國 立 交 通 大 學

管理學院（資訊管理學程）碩士班

碩 士 論 文

架構高安全性資料通訊管道於企業虛擬私有網路

Secure Extranet Communication over General Enterprise VPN

研 究 生：李文正

指導教授：黃景彰 博士

中 華 民 國 九 十 四 年 七 月

架構高安全性資料通訊管道於企業虛擬私有網路

Secure Extranet Communication over General Enterprise VPN

研 究 生：李文正　　Student：Wen-Cheng Lee

指導教授：黃景彰 博士　　Advisor：Dr. Jing-Jang Hwang

國 立 交 通 大 學

管理學院（資訊管理學程）碩士班

碩 士 論 文

A Thesis

Submitted to Institute of Information Management

College of Management

National Chiao Tung University

In Partial Fulfillment of the Requirements

For the Degree of

Master of Business Administration

in

Information Management

July 2005

Hsinchu, Taiwan, the Republic of China

中 華 民 國 九 十 四 年 七 月

架構高安全性資料通訊管道於企業虛擬私有網路

研究生：李文正　　指導教授：黃景彰 博士

管理學院（資訊管理學程）碩士班

摘要

在企業傳統的資料傳遞架構中，為了安全地傳遞資料，需添購昂貴資訊安全硬體設備或是應用軟體，隨著攻擊活動日新月異的變化，企業需要付出更多的代價來升級與替換安全軟硬體設備。本論文利用開放原始碼軟體（open-source software）的組合，來達到低建制成本與低維護成本並達到加強資料傳遞時之安全性。也因為開放原始碼軟體的高公開性，讓此系統可以針對實際需求來做適當的特性調整，讓企業跨國網路對資料傳遞時安全性的保障彈性更大。

開放原始碼軟體間的搭配整合度不高，需要記憶許多不同種類的使用方式與操作步驟，非常容易出錯，造成系統潛在性不安全。因此本論文利用 Tcl/Tk 適合開發圖形介面的特性，撰寫一個友善的圖形操作介面，來協助管理人員可以輕易地透過單一圖形介面來完成系統的所有功能，避免操作的風險，同時降低維護的成本與使用的障礙。

關鍵字：開放原始碼軟體（open-source software）、圖形介面

Secure Extranet Communication over General Enterprise VPN

Student：Wen-Cheng Lee Advisor：Dr. Jing-Jang Hwang

Institue of Information Management

National Chiao Tung University

Hsinchu, Taiwan, Republic of China

Abstract

In traditional enterprise data center, a huge amount of investment on hardware and software to keep the data transfer safe and secure. Facing the evolution of network attack day by day, more and more expense are necessary and inevitable on upgrade both software and hardware. However, open source software (OSS) community keeps decent close step on security update which offers an opportunity for better security and less expense. Due to the nature of OSS, the developers are distributed overall the world and different project are built on different style .

It is very difficult to integrate various security OSS projects together and to serve in consistent direction. In this thesis, a robust environment is developed to solve the security problem on enterprise extranet data exchange. This solution provides lower expense and faster the security update on basis of OSS. It includes operation interface on OSS packages and a sophisticated graphical user interface (GUI). On this GUI, it reduces the operation barrier on system administrator and reduces the risk on operation mistakes, especially built on different OSS packages. The GUI is written on popular Tcl/Tk language for easier maintenance and low development error.

keyword：open source software (OSS),graphical user interface(GUI)

誌 謝

感謝我的指導教授黃景彰博士，在黃老師的悉心指導下，讓我能順利地完成學習的過程，也非常感謝所有的口試委員，因為口試委員們的指點，讓論文的內容能夠更完整，尤其是廖耕億博士的大力幫忙與協助。

特別要感謝的是工作上的伙伴及長官在碩士班學習期間給我支持及鼓勵，不論是前一份工作的長官朱光宇經理或是現職的長官及同事，都提供了許多的幫忙及方便，我才能順利完成課業。而在學期間，更是要感謝交大資管所裡的老師和同學們，他們的指導與鼓勵，讓我獲益良多，也因為同學們有許多人都已在工作職場中有相當的經驗，從他們的身上不僅學得到許多專業知識，更能學習到各工作領域中的一些決策及生活哲學，因此能在許多方面都有所成長。

最後要感謝我的家人，先前因為家人的健康問題而讓學習過程進度停頓，近日家人的健康已逐漸回復至保養階段，所以家人都全力支持我安心的投注在課業上，尤其是內人靖芬，下班後還要侍奉公婆及教育二名幼兒，更是辛苦，但她仍全力支持我，讓我能無後顧之憂。

目錄

表目錄

圖目錄

第1章 緒論

1.1 研究背景

隨著網際網路（Internet）的快速成長和普及化，越來越多的企業組織紛紛將原本封閉的私有網路或區域網路與網際網路相連結，一方面可以提供多種的服務和資訊給客戶，另一方面也可以透過網際網路來獲得許多的資源。近年來，許多人開始把網際網路的技術應用到組織內部來，形成了所謂的企業內部網路（Intranet），讓公司內部的員工也可以透過網路的輔助，更有效率的互相溝通並合作完成工作。發展初期的時候，企業內部網路大多侷限於同一家公司甚至是在同一棟建築物內，但是隨著企業全球化的趨勢，橫跨多地的分公司或是出差在外的員工，就可以利用網路來完成資料的通訊溝通與交換，網際網路也因此隨著商業的運用機會增加而一直不斷地增加新的使用者，而且各式各樣的電子商務、商際網路（Extranet）的解決方案也如雨後春筍般地出現，讓分散各地的組織可以透過網際網路，更快速且有效率的分享資源，形成一個邏輯上的虛擬私有網路（Virtual Private Network，VPN）。

但隨著網際網路的日漸普遍及應用程式的多元化發展，網際網路的使用者越來越多且每年都以倍數在增加中，這使得目前被廣泛使用的 IPv4 的定址位址已不敷使用；因此已有專家預估，約到西元 2010 年的時候，將會沒有 IPv4 的位址可供分配使用。面對網際網路如此驚人的成長速度，新一代的 IP 通訊協定應運而生，我們將此新一代的通訊協定稱之為 IPv6；IPv6 通訊協定改變了原本 IPv4 通訊協定中的部份欄位選項，更加入了符合現代需求的功能，同時 IP 位址的長度也從原先的 32 位元發展成 128 位元，以滿足未來網際網路更多定址位址的需求。

另一方面，網際網路上的安全問題隨著網際網路的商業應用程度地增加及近來網路犯罪的事件日益頻繁而日漸受到廣泛地重視，而 IPv6 此新一代的 IP 通訊協定能夠解決現有 IPv4 協定的若干問題，並增加網際網路的安全性；然而，目前雖然已有部分機構在建構 IPv6 的網路環境，但以現階段企業所使用的網路設備而言，仍大多數是以支援 IPv4 此通訊協定為主，但也已有不少企業所使用的網路設備，已同時具備支援 IPv4 及 IPv6 這二種通訊協定的能力，因此想利用 IPv6 與 IPv4 的相容性，建立一個在 IPv6 網路基礎上的 VPN，並針對該 VPN 所面臨的安全問題來做為主軸而加以探討，並提供一個較容易的操作介面來協助管理人員管理整體的環境。

1.2 研究動機與目的

隨著網際網路（Internet）的快速成長和盛行，漸漸地改變了人們原來的生活方式，人與人之間的溝通變得更方便及及多元，對許多人而言，網際網路已經不再是一個陌生的名詞，相當多的企業組織紛紛將原本封閉的公司內部私有網路和 Internet 相連接，透過與網際網路的連接，一方面可以提供各種服務和資訊給客戶，另一方面也可以透過 Internet 來獲得許多資源。後來許多人更開始把 Internet 的技術應用到組織內部，形成了所謂的 Intranet，讓公司內部的員工也可以透過公司內部的網路有效率的互相溝通和合作完成工作。網路的發展，初期多侷限於同一家公司或是同一棟建築物內，但是隨著企業全球化的趨勢，橫跨各地的分公司、關係企業或是出差在外的員工就可以利用 Internet 來完成資料的通訊，Internet 隨著商業的運用機會增加，各式各樣的電子商務、商際網路（Extranet）的解決方案也如雨後春筍般地出現，讓分散各地的組織可以透過 Internet 更快速而有效率的形成一個邏輯上的虛擬私有網路（Virtual Private Network ，VPN）來分享彼此的資源。而 Internet 上的安全問題隨著網際網路的商業應用增加及各種網路犯罪的案件日益頻繁而受到重視 ， 尤其是自從開放的網際網路成為商業媒介而吸引大眾的關注之後，網際網路上攻擊事件便時有所聞。其中特別以 2000 年 2 月上旬時所發生的一連串著名網站受到攻擊的事件最引人注目，攻擊駭客們選擇了著名的電子商務公司及大眾傳播媒體的相關網站做為其攻擊的目標 ， 在此波事件中，遭受攻擊的網站有 Yahoo.com、eBay.com、amazon.com、buy.com、CNN、ZDNet 等公司的網站（”FBI investigates，“2000，February 9；Cha & Schwarts，2000，February 10）。在這些事件中，專家們相信攻擊者是利用所謂阻斷式的阻斷服務攻擊（Distributed Denial-of-Service（DDoS）attacks）（Copeland，2000，February 18；Kerstetter，2000，February 21）來完成此次的攻擊行動[1]，網路上各種窺探或攻擊的工具軟體幾乎是垂手可得的，也因此，網路上的安全問題也不斷地隨著攻擊工具的推陳出新而有所不同。

Internet 所使用的 TCP/IP 協定本身有許多先天上的安全性缺點，影響大眾對於在 Internet 上進行商業應用的接受程度。早期 Internet 發展初期，使用者大多是為了學術用途居多，對於網路安全上的需求不是十分地重視，但當 Internet 運用在商業用途的時候，所需的安全考量就大大不同於運用在學術上的了，畢竟重要的商業資料若在網路上被竊取、竄改或是遭到非法入侵破壞，所付出的代價是十分難以估計的。因此在系統安全的問題上，雖然已經有所謂的 trusted operating system（如 Sun Microsystem 之 trusted Solaris）的設計，但是因為這些系統的限制太多，反而造成使用者使用上的不便利，所以在一般的應用上並不適合；然而一般使用的作業系統（commodity operating system)，雖然使用較方便且價格也較便宜，但卻隱含了一些系統規格本身設計不良或是實作上的缺點，因此如何補正這些缺陷並擴充作業系統本身對安全性的支援，便逐漸成了一個重要的課題，早在 2000 年秋天的美國國安局(National Security Agency)報告中便指出這個現象，NAS Advisory Board 的 review 以及 ISSO 也都確認有在 commercial-off-the-shelf (costs)系統上提供安全作業環境的需求。[2]

近年來網際網路的技術日益成熟，各類地應用迅速成長，人類生活對於資訊科技的依賴也愈來愈深，許多與個人隱私、公司重要資料甚至於國家安全有關的機密資料不停的在網路上傳遞著，由一個系統到另一個系統，如果這些資訊讓有心人士竊取或是不當使用，對社會國家及公司所造成的危害實在無法估量。根據美國 FBI/CSI（Federal Bureau of Investigation/Computer Security Institute）的 2004 年安全調查報告中指出，由於電腦攻擊而帶來的各種損失與以前相比有一定幅度的降低, 但令人擔憂的是由企業內部員工的疏失及攻擊所帶來的損失，已經逐漸成為最大的威脅.，報告中也指出在所有安全出現問題的機構中有 99%已經安裝了防毒軟體，更有 98%的機構安裝了防火牆軟體，所以資訊安全問題不再是單單針對外部攻擊便足夠了。另外由 CERT/CC（Computer Emergency Response Team/Coordination Center）的一份調查報告中也說明了日益嚴重的資訊安全問題，所以加強網路上的資訊安全，讓資訊能夠更安全地被傳遞，是一個很重要的課題。

因此本研究的研究動機及目的為：

1. 針對公司內部網路資訊流通，加強現有網際網路的資訊安全措施。

2. 加強網路資料傳輸的資訊安全一定都得依靠高成本的商業套裝軟體或委外發展嗎？有沒有一種可以達到簡單有彈性且能自行發展的方式。

3. 資訊安全的管理操作可否變得更容易?操作介面是否可以較命令模式更友善?

1.3 研究方法

本論文是以企業內部網路的環境為基礎，討論架設一個虛擬私有網路可能會面臨的安全問題並參考其他文獻所提出的架構，來建置整個系統的雛形。同時希望能採取結合業界上的一些軟體的方式，使其更符合實務上的需求，並對系統持續不斷地做改良。完整的步驟如下：

步驟一、收集及研讀相關論文報告。

主要的方向，以跟 IPv6、VPN、IPSec 等相關的議題來收集，並加以仔細研讀，以加強本研究不足的地方。

步驟二、雛形系統之初步分析。

首先，定義 VPN 所應具備的功能，及各部分組成的模組，並設計一套適用於企業內部網路環境的 VPN。當這些前置工作完成時，就可以先建立一個簡單的雛形。

步驟三、論文實作架構確定。

經由不斷的測試及改良，獲得最後完成的系統架構。

步驟四、系統設計、測試。

實作 VPN 系統，並作完整的測試。

步驟五、成果分析和比較。

分析成果，與預期的結果作個比較，找出相異的地方，探討其原因並加以修正。

步驟六、後續研究討及報告撰寫。

找出可供將來後續研究的方向，及目前系統不足的地方，完成報告。

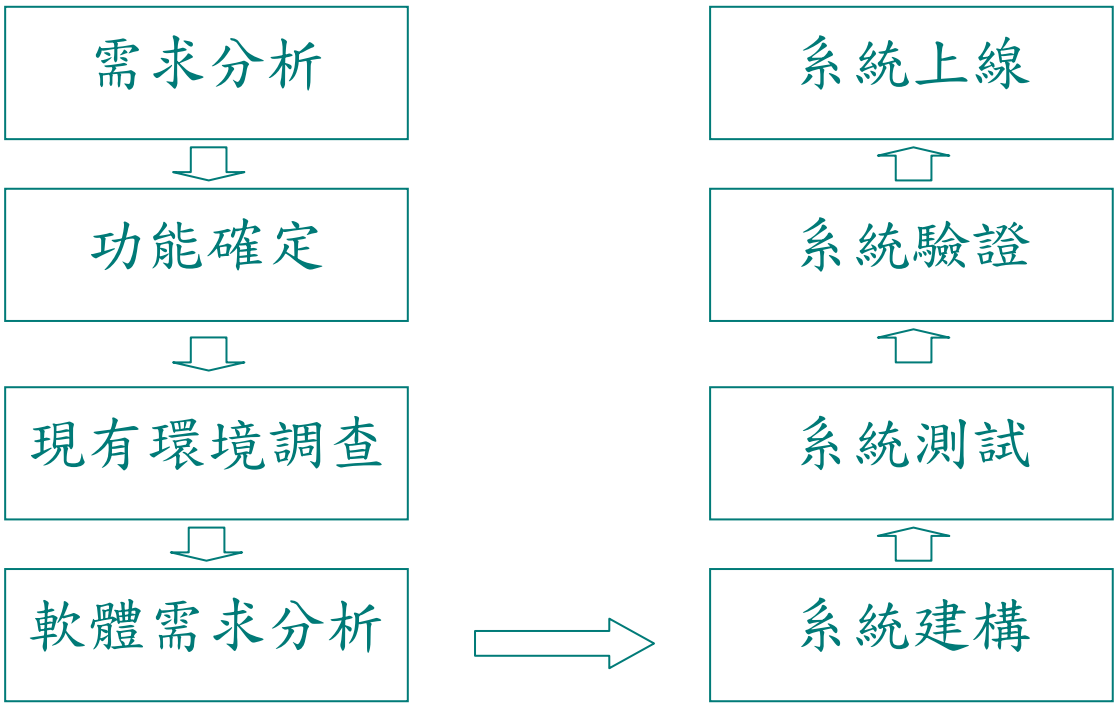

圖 1 研究方法圖示

第2章 文獻探討

2.1 新一代的定址方式 IPv6

2.1.1 IPv6 之簡介

由於近年來全球資訊網(world wide web)及網際網路(Internet)的蓬勃發展，使用網路的人口迅速的不斷增加，Internet 上的可用位址預期很快就會發生不夠的現象，因此從 1992 年底的時候，開始不斷有人提出新一代的網際網路通訊協定(IPng；Internet Protocol next generation)的建議書。在此同時，基於以往 Internet 之使用經驗得知，原本之 IP 協定有許多不足的地方，如其對及時服務、擁塞控制、自動設定及保密措施之支援等都有所不足，因此，在設計新一代的通訊協定時，除擴充可用的位址空間以解決最緊迫之位址不足的問題之外，亦對原來 IP 協定各方面的功能重新檢討，以力求改善。為反映這些需求，Internet 工程特別小組(Internet Engineering Task Force，IETF)於 1992 年 6 月對下一代網際網路通訊協定(簡稱 IPng)發起提議徵文，經過長期的討論，於 1995 年 1 月確立主要里程碑並發表 RFC (Internet Requests For Comments)1752，“The Recommendation for the IP Next Generation Protocol”，所以新一代之 IP (Internet Protocol)通訊協定(簡稱 IPv6)被提出來解決此一問題，其意義為第六版的網際網路通訊協定（IP version 6）。

IPv6 有下列的改變[3]：

1. 擴充的位址：定址空間從 32 位元調整為 128 位元，且刪除了廣播位址，增加了任一傳播（anycast）。

2. 簡化的標頭格式：固定的標頭長度及較少的欄位，所以路由的效率較佳。

3. 改善的選項、延伸支援：IPv4 將選項放在 IP 標頭的後面，而 IPv6 則是將選項放在單獨的延伸標頭，所以需要時，再處理即可。

4. 流量標籤：在 IPv4 中，所有封包都一視同仁地由傳送途中的路由器自行處理，路由器並不會對所傳送的封包做記錄，但是 IPv6 對流量的認定為一連串從起始端送到終點的封包，傳送途中的路由器會將流量做適當的記錄以便追蹤管理，即 IPv6 標頭中的流量標籤能識別封包的屬性，在同一個資料流量內的標籤是一樣的。

5. 認証與機密：IPv6 使用在 RFC1826 中所定義的 IP 認証標頭及 RFC1827 中的 IP 包裝安全資料欄位。

IPv6 提供 2 的 128 次方的定址空間，不但能暫時解決目前面臨的 IP 不足問題，它在 IP 的標頭認証、路由與安全等功能上，也有許多的改善，因此網路服務供應商也就能夠提供具 QoS 功能的網路，同時 IPv6 使用 IPSec 的安全傳輸方式，對於近來逐漸受重視的網路安全問題，提供了十分正面的幫助。

2.1.2 IPv6 基本標頭（header）與擴展標頭（extension headers）

IPv6 此一通訊協定並不是一個全新的網路通訊協定，它是從原有的 IPv4 通訊協定演進而來，它將 IPv4 中不產生作用的功能除去，保留了有用或預期要使用的部份，並且針對 IPv4 所缺乏的部份加以改善，譬如在區域使用的單點傳播中，提供 IPv4 所缺乏的隨插即用 （plug & play） 的能力，所以當使用者移動主機後，不需要重新註冊位址。在 IPv6 網路中的每一個封包其 header 總長度為 40 位元組（bytes），如圖 2 所示，header 中各欄位功能列示如表 1。

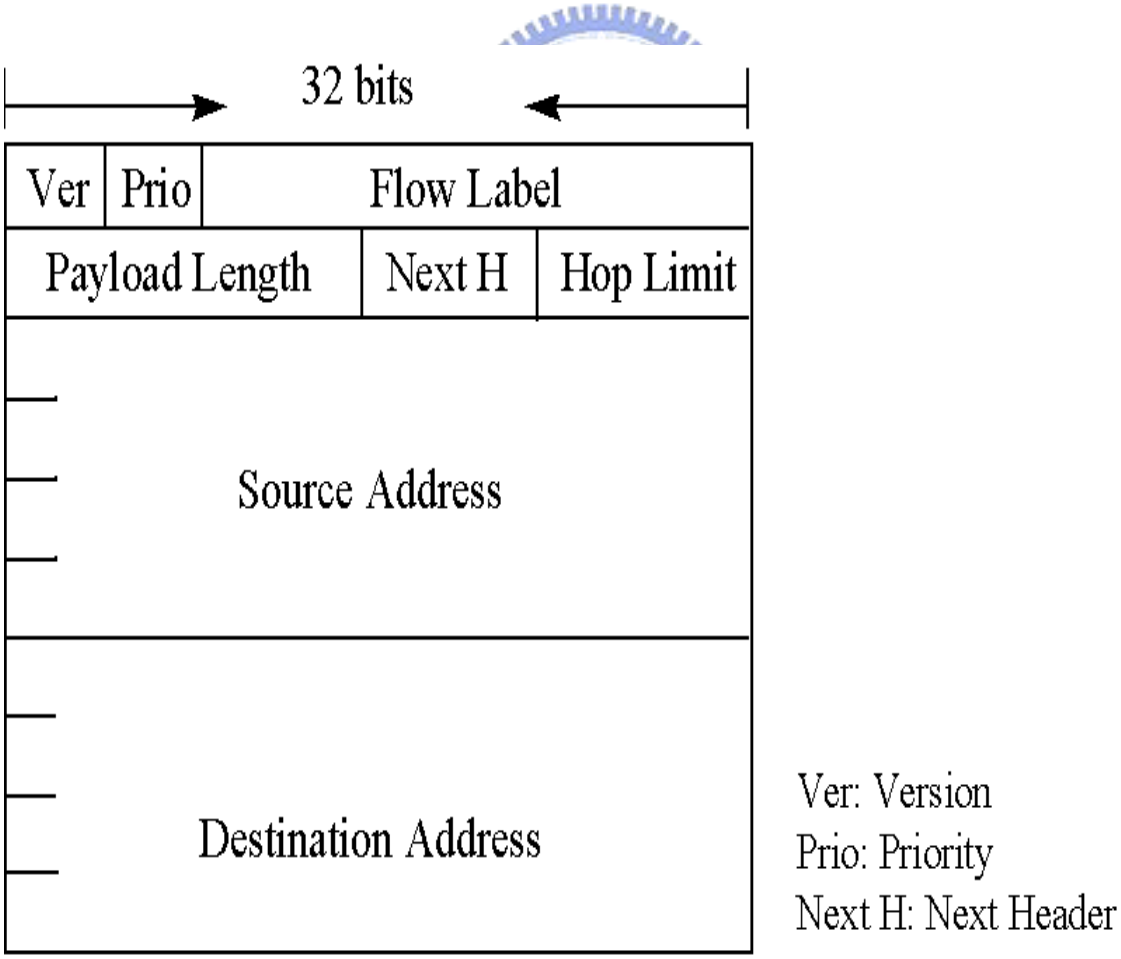

圖 2　IPv6 基本標頭格式 （資料來源：[3]）

表 1 IPv6 基本標頭各欄位說明

欄位名稱	位元數	欄位功能
Version	4	IP 的版本號碼，其值為 6
Priority	4	傳送者標示其所要傳送的封包之優先權值，此值是該傳送者所要傳送資料之間相對的優先順序
Flow Label	24	傳送者標示其所要傳輸之封包，在經過 IPv6 路由器時，所需的特殊處理方式。例如即時服務（real time service）等
Payload Length	16	值為正整數，代表連接在 IPv6 標頭之後的資料長度，不含基本標頭 40 位元組
Next Header	8	表示緊接在 IPv6 標頭之後的標頭形態
Hop Limit	8	值為正整數，所傳送的封包每經過一個 Hop。時，其值減 1，當其值為 0 時，則此封包即被丟棄。
Source Address	128	封包之起始傳送者的介面位址。
Destination Address	128	封包接受者的介面位址。

資料來源：[3]

在 IPv6 通訊協定中，其基本標頭的長度和欄位格式是固定的，這與 IPv4 是不一樣，因為 IPv4 標頭中包含了選擇性的功能（Options），所以長度會因為所採用的選擇不同而有所差異，而 IPv6 將選擇性的功能獨立出來，放在擴展標頭內，而與基本標頭分開，這種設計的目的乃是為了能提高傳輸封包的處理效能。擴展標頭在傳輸封包中是放在基本標頭和傳輸控制協定（TCP）標頭之間，其所包含的選擇性的功能如表 2 所示。

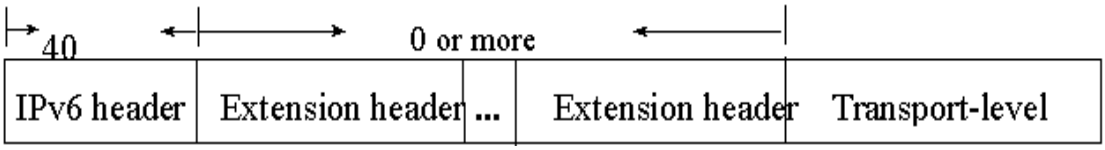

圖 3 extension headers 位置圖 （資料來源：[3]）

表 2 IPv6 選擇性功能表

Options	功　能
Routing	傳輸起源點指定路由路徑
Fragmentation	封包的分割及組合辨識
Authentication	完整性（Integrity）和認証
Security Encapsulation	隱密性（Confidentiality）
Hop-by-Hop option	指定每個經過的 Hop 都需要處理的 option
Destination Options	僅需要目的地節點檢視的資訊

資料來源：[3]

2.1.3 IPv4 與 IPv6 封包格式之比較

IPv4 的標頭格式如圖 4

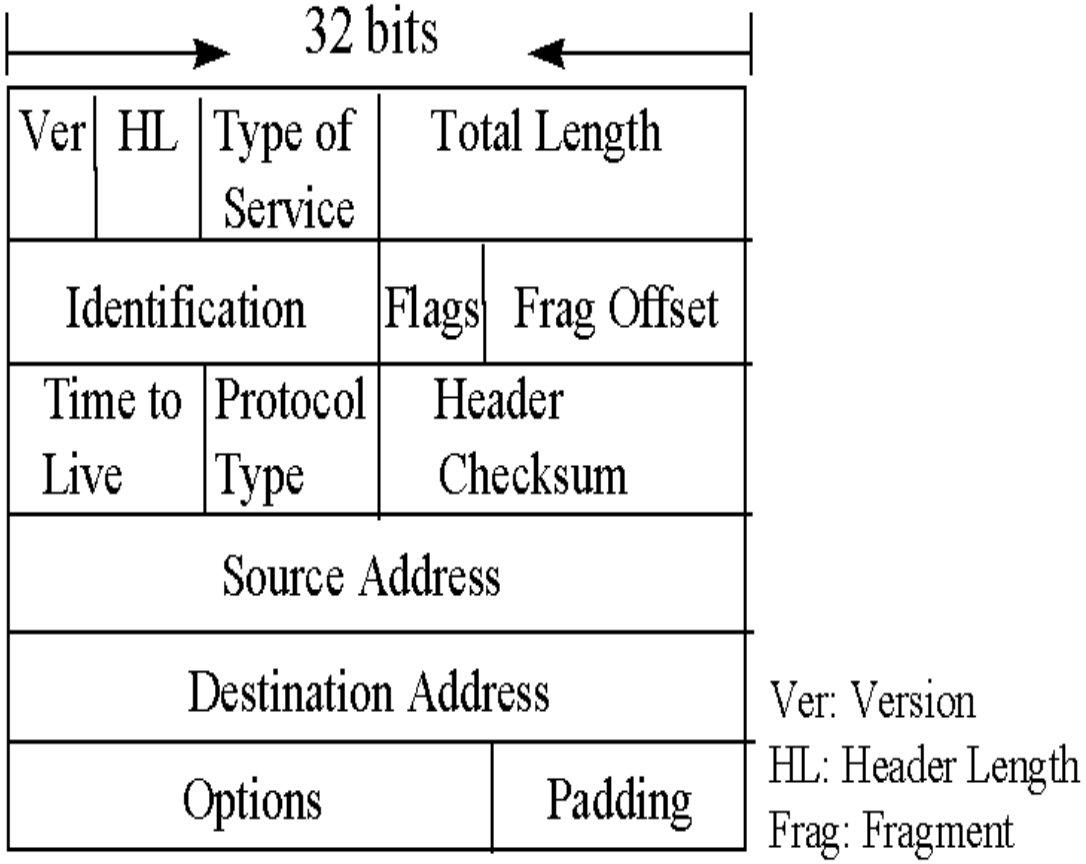

圖 4　IPv4 的標頭格式（資料來源：[3]）

而 IPv6 的標頭格式如前面所介紹，首先，IPv6 取消以下 6 個在 IPv4 之欄位：

表 3　IPv6 取消 IPv4 部份的欄位

項次	內　　容
1	IP 標頭長度(Header Length)：由於 IPv6 係採固定標頭長度，故不再需要。
2	服務型式(Service Type)：此欄位由其它機制取代。
3	識別(Identification)、旗號(Flags)和區段移補 (Fragment offset)：由於 IPv6 只支援端點對端點 (end-to-end) 分割，故不再需要這些欄 位。
4	標頭檢查和(Header Checksum)：靠著媒介存取 (media access) 控制程序中的檢查和，不再需要在 每一切換上檢查及更新檢查和，主要好處是減少標頭處理的負擔。

資料來源：[3]

而 IPv6 與 IPv4 之間 options 的相異之處，整理如下：

表 4　IPv6 與 IPv4 options 的相異之處

		IPv4	IPv6
特性比較	長度限制	40 bytes	None
	存在位置	基本標頭的一部份	Extension Header
	執行處理	只要 Option 存在，經過的路由器都必須處理	經過的路由器不須處理，只要由目的地主機處理即可
功能比較	Source Routing	Yes	Yes
	Security	compartment，user group	Authentication、Security Encapsulation
	Fragmentation	None	Yes
	Record Route	Yes	None
	Stream ID	Yes	None
	Internet Time stamp	Yes	None
	Hop-by-Hop Option	None	Yes
	Destination Option	None	Yes

資料來源：[3]

2.1.4 IPv6 實驗網路（6Bone）

6-Bone 是為了在 Internet 上推廣 IPv6 的一個全球性 IPv6 測試平臺。它於 1997 年 6、7 月間開始運作，其相關活動皆屬 IETF 下 ngtrans 工作小組的一部份。6-Bone 的主幹是由許多相互連接的網路服務提供者(ISP)及用戶網路所組成。事實上，它是一個以架構在原 IPv4 網路上，使 IPv6 封包透過通道(tunnel)轉運的虛擬網路。目前，6-Bone 的目標在於獲得一些使用經驗，以便對 IPv6 的傳送提供早期的政策及程序。未來，當陸續有網路服務提供者及用戶網 路提供 Internet 上 IPv6 傳輸後，它將以一種透通的方式逐漸被取代。
6Bone 虛擬網路的組成份子大致分為四種角色[5]：

1. 網際網路服務提供者（ISP）：法國 G6、英國 UUNET 等。
2. 網路軟硬體企業：Cisco、Bay Networks 等。
3. 學術研究單位：美國 UCLA、日本大學、丹麥大學等。
4. 官方或政府：美國 NASA、歐洲 RIPE 等。

在 2001 年時，臨近我們的日本已經有三家網絡網路服務提供者正式提供 IPv6 的網路服務供商業使用，但是台灣目前只有中華電信擁有 IPv6 的實驗網路供測試用途，而未正式提供該項服務。

2.2 虛擬私有網路(VPN)

2.2.1 何謂虛擬私有網路(VPN)

「虛擬私有網路」的關鍵字在「虛擬」二字的意義，所指的便是一個企業透過公眾的廣域網路系統，讓分散在各地的子公司或是合作廠商的私有區域網路彼此通訊的網路架構。雖然實際網路的連結方式和傳統定義的私有網路連結方式不同，但在邏輯上其作用和所達成目的相同，故稱為虛擬私有網路（Virtual Private Network，簡稱 VPN）。也因為其所利用的媒介為公眾的廣域網路系統，所以在 VPN 上所流傳的商業資料就應該注意不被竊取或竄改。因為網際網路（Internet）在全球的盛行，許多企業為了享受網際網路的優點和便利而紛紛和網際網路連結，內部也開始將原本用於網際網路的技術來建構網際網路。一般傳統的私有網路大多是透過數據專線、ISDN、或是電話撥接的方式與各子公司連線或是使用者直接連線後達到資料互相傳輸的目的。但是如此的花費卻相當的驚人，特別是數據專線，除了成本較高的缺點外，一旦建立好之後，也不容易遷移和擴充。如果使用 VPN 的話，同樣可以達到相同的目的，而且尚有下列的優缺點：

優點：

1. 以前與遠方的網路連線方法可能是採租用專線或是以訊框傳遞（Frame Relay）網路來達成，不論是鋪設專線或是以傳輸量來計費的 Frame Relay 方式，其費用都十分地高，若是透過網際網路來達成溝通的目的，則其花費之成本會大幅地縮減。除了初期的建構所需成本較低外，每次通訊傳輸資料的費用也由長途的費率降為區域的費率。
2. 擴充性提高了，因為透過網際網路的關係，增減所欲傳輸的對象都十分方便，因為由硬體設備的增減轉變為軟體的設定。
3. 因為網際網路的普及，使得可傳輸合作的對象可以遍佈全球，也因此增加了企業的競爭力，與以往所不同的是，虛擬私有網路是經由網際網路為媒介來傳輸資料的，而網際網路是一個低成本、低收費，而且是一個開放性的環境，適合全球化的通訊。

缺點：

1. 不能保證傳輸資料時所需的傳輸品質，因為 VPN 是架在網際網路之上，所以傳輸資料均會不可避免的經過網際網路，因此若是在網路使用量大的尖峰時段傳輸資料的話，往往會因頻寬不足而變得非常緩慢。
2. 因為網際網路所用的 TCP/IP 協定的安全方面的問題，使得資料的傳輸安全性受到質疑。

VPN 服務的方式一般可分為二種，一為 Client to Site （如圖 5），另一種為 Site to Site （如圖 6）

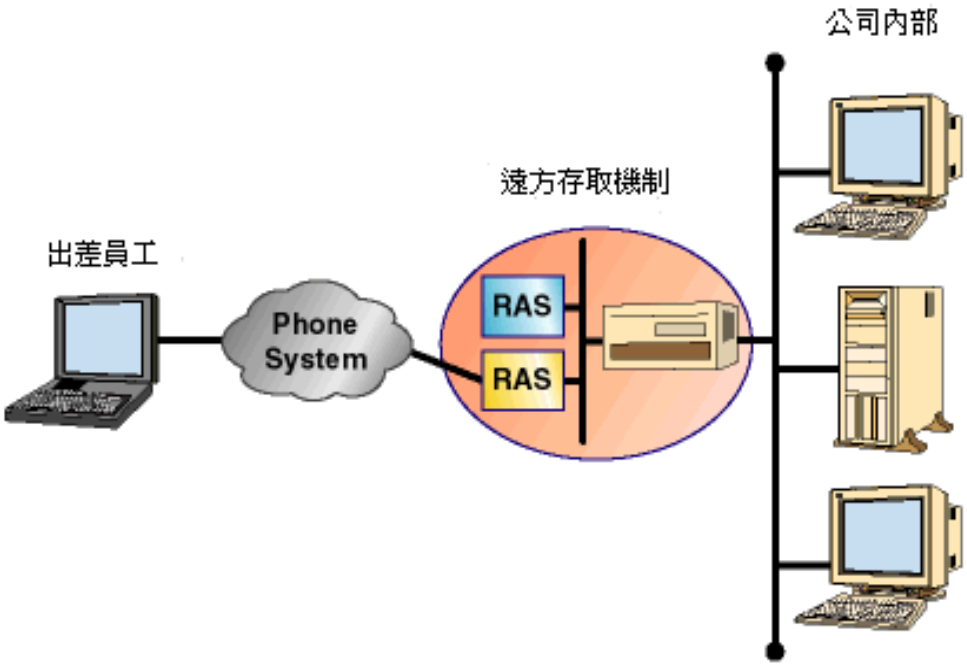

圖 5 Client to Site 的 VPN 模式 (資料來源 [6])

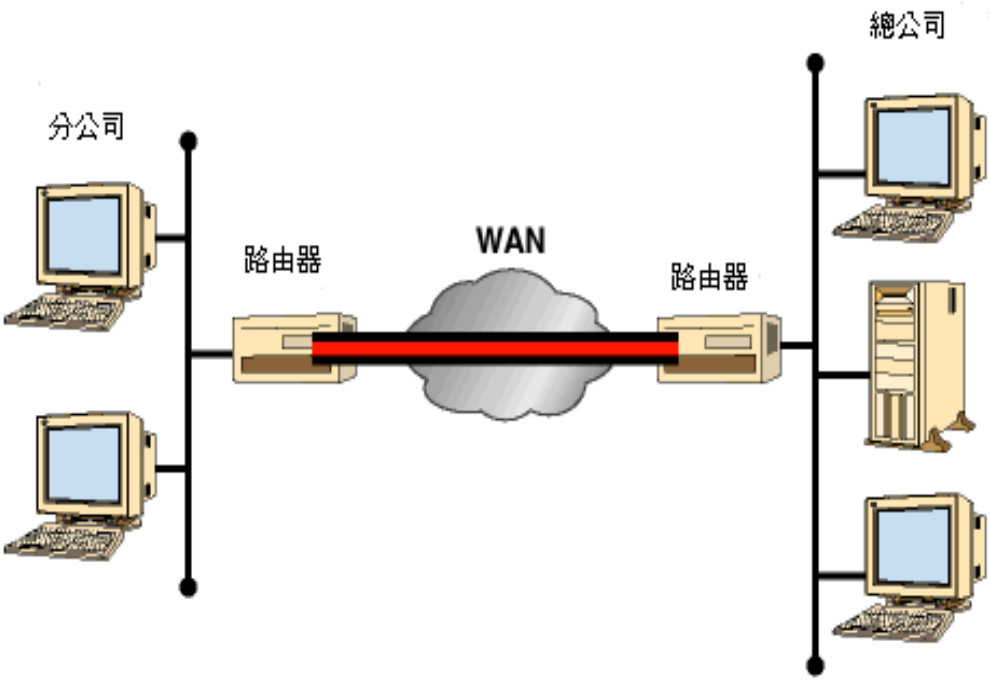

圖 6 Site to Site 的 VPN 模式 (資料來源 [6])

2.2.2 虛擬私有網路的保護對象

一般人在談到網路上所需要保護的對象，第一個聯想到的通常是保護系統內的檔案資料：公司的企劃書、財務報表、薪資及稅單或是研發的成果。這些檔案資料固然十分的重要，但是網路所帶來的便利及好處並不僅僅如此而已，網路所帶來的好處更可能包括了提昇公司的企業形象或商譽等無形的資產，若是不好好地小心保護網路內的重要資料，當網路上機密資料遭別人所盜取而利用時，所損失的可能不只是檔案資料這種有形的資產，也可能使得公司的形象受損，影響商譽，甚至影響了公司的獲利盈收。但我們也不能因噎廢食，因為網路所帶來的好處，可以讓公司的員工在家中就可以繼續上班時未完成的工作，調閱公司內部的資料，而不必受限於一定要到辦公室才能工作，所以若是好好利用網際網路這種管道來進行通訊的話，可以提昇員工的產能，但要如何在網際網路這麼不保險的環境中，利用虛擬私有網路來降低傳輸時的風險，是值得思考的問題。

2.2.3 虛擬私有網路解決網際網路安全問題的技術

虛擬私有網路在保護網際網路上傳遞資料的技術方面，最常見的方式有防火牆（firewall）、身份驗證（Authentication）、編碼加解密（Encryption & Decrypion）、通道封裝（Tunneling）等四種方式，透過這四種方式，便可以建溝一個具安全性的資料傳遞管道。

2.2.3.1 防火牆（firewall）

在概念上，Internet 的防火牆與實體建築的防火牆功能是一樣的，建築物的防火牆可以在發生火災的災難時，有效地隔離火勢的散佈，減少可能的損失，而 Internet 的防火牆也是同樣的道理，它防止網際網路上的一些危險進入到企業內部網路來造成災害。防火牆是介於內部網路與網絲網路之間的系統，主要的功能為阻擋外界直接看到企業的內部網路，限制能提供的服務；同時也管制企業內部能看到外界的服務，它會檢驗封包的 IP 位址或是要求連線的 port 來決定是否提供該項服務。

可以想像將內部網路與網際網路之間的出入口加以管控的話，所有傳入或傳出的資料封包都一定要經過這個管制點，則只要管理好這個管制點的安全性，便等於管好了整個內部網路與網際網路之間溝通的安全問題，因此將管制點簡化到最少，這顯然比管埋許多的出入口要來得容易許多，而且以目前的軟、硬體而言，架構防火牆的安全方式會比其他的來得容易且省錢。也因為架構防火牆的管制點是網路傳輸的要點，所有的網路傳輸都會經過這個地方，所以若是要對網路活動加以記錄的話，也會遠比架構防火牆前來得有效率。但是有了防火牆並非就代表網路的絕對安全，實際上，防火牆有一些弱點，例如多數的防火牆不能夠預防病毒的破壞、防火牆是針對內部網路與網際網路間的傳輸加以管控，所以對於內部網路的使用者而言，若其畜意破壞也是不能有效地管理的、若是內部網路與網際網路間的傳輸有防火牆未管制到的出入口時，該出入口便會成為內部網路安全性上的一個大漏洞、防火牆大多是透過事先定義好的政策來管制網路資料的傳輸，所以一旦事先定義的政策不夠嚴謹時，仍會造成網路資料傳遞時安全性的一個問題。

就防火牆的技術性而言，其實都十分地雷同，大多都以減少內部網路及網際網路管制口數量為主，達到類似集中化管理的方式。而一般防火牆常見的的架構有下列四種：

- 封包過濾路由器：路由器並不在乎封包的內容，它只管封包的來源及目的地是否符合規則或是事先定義好的路徑表（Routing Table），若是符合的話，則讓該封包通過，反之，則將封包加以阻擋，不予通過。如（圖 7）

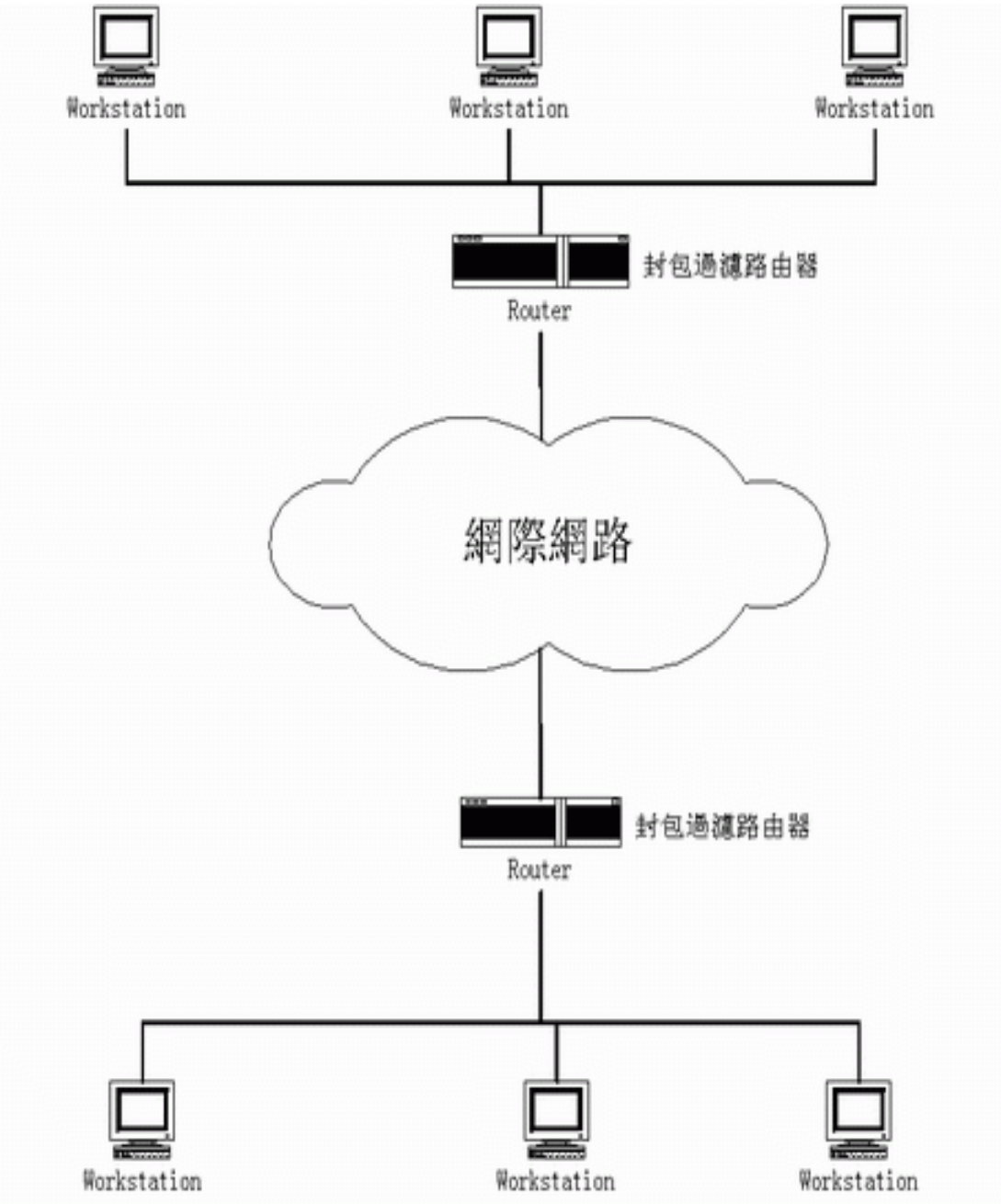

圖 7 防火牆常見的架構之封包過濾路由器 (資料來源：[7])

- 防禦主機（Bastion Host）：利用路由器的封包過濾的功能外，再加上防禦主機的輔助來負責網路的安全防護。（如圖 8）

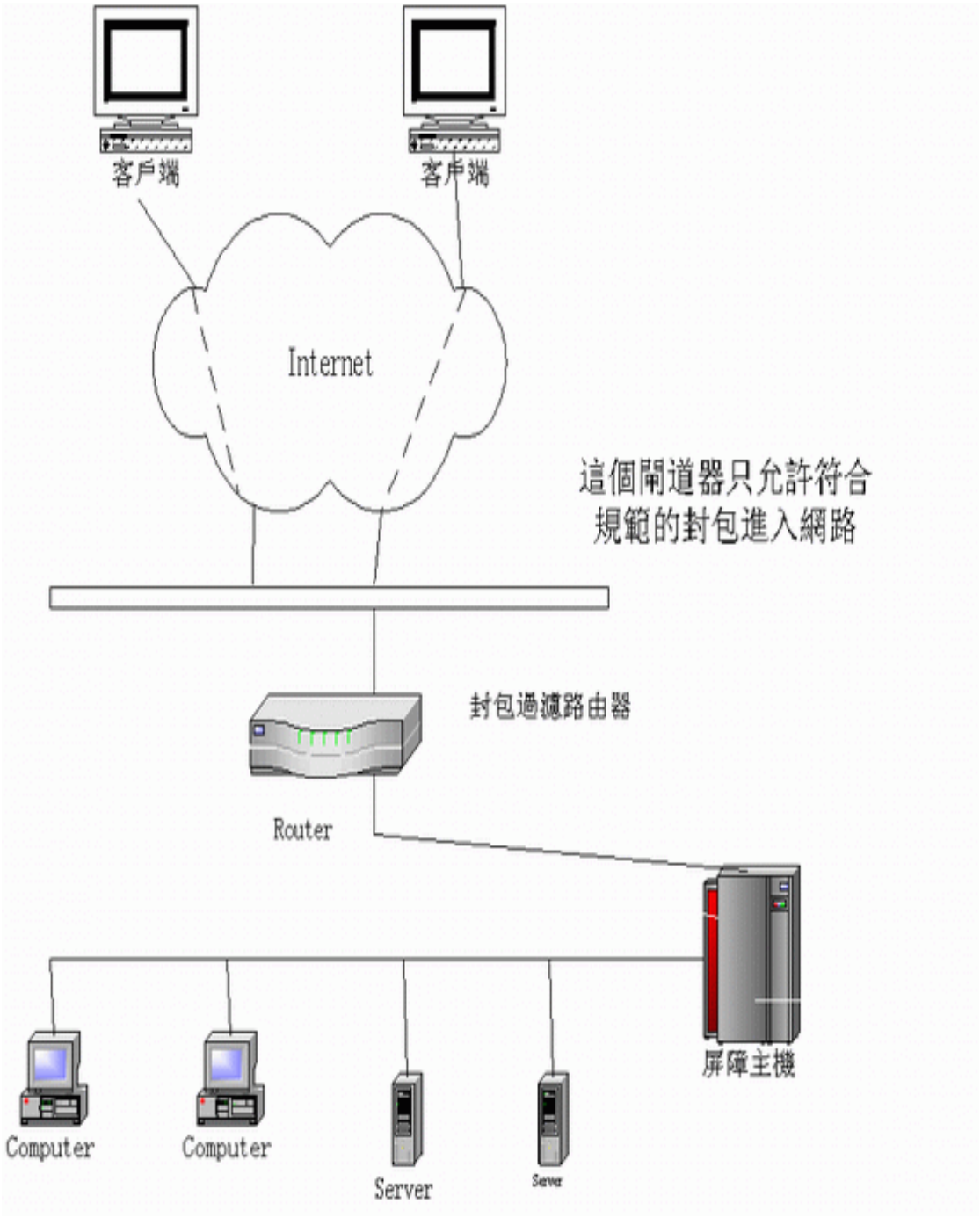

圖 8 防火牆常見的架構之防禦主機 (資料來源：[7])

- 鄰界網路（perimeter network）：鄰界網路又稱為 DMZ（DeMilitarized Zone）非戰區，在內部網路與外部網路交界地帶中間所開設出來的一個緩衝區，至少要由二部以上的路由器所構成，一部當作對外網路的閘道，另一部則作為通往內部網路的閘道，在這二部路由器中間，除了防禦主機或是其它的路由器外，不應該設置其它的機器。（如圖 9）

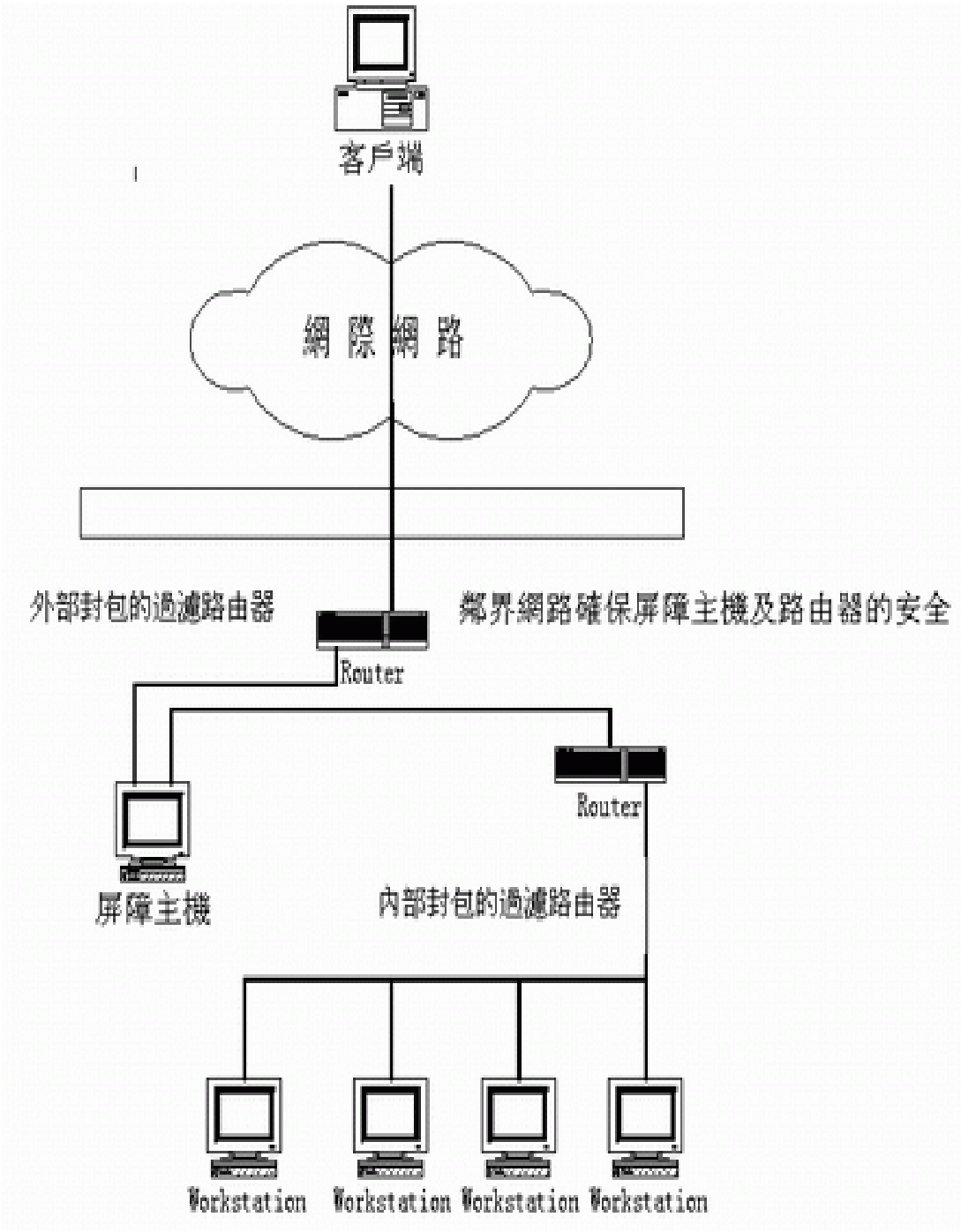

圖 9 防火牆常見的架構之鄰界網路 (資料來源：[7])

- Proxy 伺服器：Proxy 運作的方式類似防禦主機，Proxy 伺服器執行一個特殊的程式後，讓外界以為它是內部的一台主機。(如圖 10)

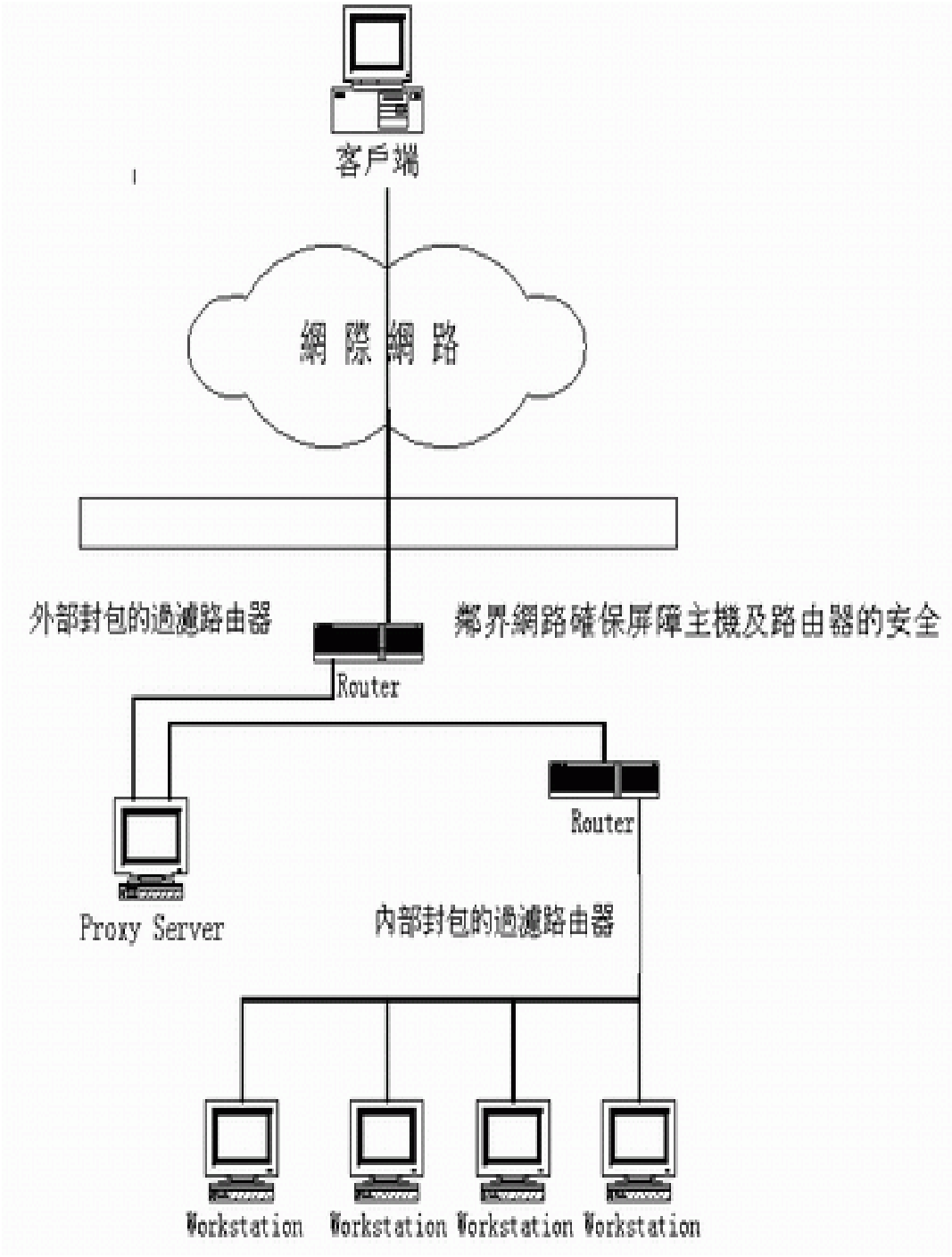

圖 10 防火牆常見的架構之 Proxy 伺服器 (資料來源：[7])

2.2.3.2 身份驗證（Authentication）

有人想透過網路與你連線，你無法看到他，也無法聽到他，如何確定跟你連線的人，真的就是你所想的人，而不是另一個人，。因此如何正確地辨識合法的使用者與設備，使真正需要連線的人或設備能夠互相溝通，形成一個虛擬私有網路並讓駭客不易侵入，這就是身份驗證的技術，最常使用的技術有使用者的名稱、密碼或是智慧卡等。

2.2.3.3 編碼加解密（Encryption & Decryption）

編碼加解密的技術已有一段時間，需要秘密通訊時，都會利用到它。而虛擬私有網路是建構在網際網路上，為了私有的資料在傳送的過程中，不會被不相關的人所竊取或修改，所有在網路上傳遞的封包都需要經過加密的過程，當該封包到達目的地之後，再將其解密。在傳遞的過程中，即使封包被人所截取，也只能看到一些無意義的亂碼，而無法得知封包所傳送的的真正內容。

編碼加解密的技術可分為二大類：對稱式密碼學（Symmetric cryptography），又稱為秘密金鑰密碼（Secret-key cryptography）及非對稱式密碼學（Asymmetric cryptography），又稱為公開金鑰密碼學。對稱式密碼學的加解密使用同一把金鑰，如 DES 即是如此。而非對稱式密碼學的加解密動作，則是使用不同的金鑰，其中以 RSA 的方式最常被使用。因此如果駭客想要知道網路上封包的內容，則必須先解開金鑰（Encryption Key）。但是金鑰的長度不同，破解所需的時間就會有所不同，譬如使用 SSL 的技術，秘密金鑰的長度為 40 bits，可能不到一秒就可能被破解；但是 56 bits 的 DES 技術，可能就要數年才能破解，相對地來說，安全性便比較高，但目前 SSL 的加密技術已經從先前的 40bits 之外，已加強到 128bits，其安全的強度為 40bits 的 288 倍，至於 DES 方面，目前則多以 Triple-DES 的技術來取代 DES，其實際有效金鑰長度為 168 bits，同樣地，其安全性又再提昇了許多。

2.2.3.4 通道封裝（Tunneling）

通道封裝的技術是為了將資料在網際網路上傳輸所發展出來的一種資料封裝方式（Encapsulation），也就是說，在網際網路上建立一個特殊的通道，讓私密的資料在此一通道內傳遞。目前在此一方面所常使用的通訊協定有 IPSec（IP Security）、PPTP（Point to Point Tunneling Protocol）及 L2TP（Layer2 Tunneling Protocol）及 SSL 等四種。

IPSec 是一種第三層的通道封裝技術，專門針對 IP 的環境所設計，不但符合現有的 IPv4 環境，也同時可以符合未來 IPv6 的網路需求；而 PPTP 與 L2TP 則都是第二層的通道封裝技術，PPTP 是由 Microsoft 及 Ascend 所提出來，而 L2TP 則是由 Cisco 所提的 L2F（Layer2 Forwarding）所演變而來。三者之間最大的不同為 IPSec 適合多點傳輸的功能，所以可以同時使用 Internet 與虛擬私有網路，而 PPTP 或 L2TP 只能執行點對點的虛擬私有網路功能，不能同時使用 Internet 。

而近年來，由於企業對於行動工作的資訊存取需求不斷地增加，所以遠端存取的 VPN（Remote Access VPN）逐漸在市場上受到重視，而 SSL VPN 便因為設定及安裝十分地簡單，且只需透過幾乎是隨處可得的能支援 SSL 之瀏覽器就可以建立起一個加密通道，尤其是在 clinet-to-site 的部份，SSL VPN 更是有其優勢；SSL VPN 除了安裝、設定十分容易方便上的優勢之外，還具有內部資訊安全控管、遠端存取權限控管等功能，而且其總擁有成本低。而傳統的 VPN 設定都必須加裝客戶端軟體，整體的設定也較為繁鎖，因此增加了系統的複雜度，同時也降低了使用者在使用時的友善程度，尤其傳統的 VPN 無法像 SSL VPN 可以在任何有支援 SSL 之瀏覽器的地方便可以設定、使用，這使得 SSL VPN 的後勢

發展被看好，甚至有專家及業者認為 SSL VPN 可望取代部份傳統的 IP VPN 市場。

SSL VPN 當然也有其缺點。首先，由於它是在應用程式層上運作，而不像 IPSec VPN 是在網路層上加密運作，所以一般而言，效能無法與 IPSec VPN 相提並論，其次，由於 SSL VPN 只適用於 HTTP 的通訊協定，即使愈來愈多的操作都可以在 WEB 平台上正確地運作，但在未全面應用程式都可以在 WEB 介面上運作前，一些非 Web-based 的應用程式，便必須經過額外的測試與設定，才能夠順利在 SSL VPN 上存取，不像 IPSec VPN 的可用性及透通性高，只要上網並設定完成後，所有 IP-based 的應用程式便都可以直接使用、存取無誤。另外 SSL VPN 使用者可以存取的應用程式也必須由網路管理者事先定義好，這也會增加管理者的部份負擔，而 IPSec VPN 則不必再由管理者針對應用程式再去設定任何東西。

表 5 IPSec VPN 與 SSL VPN 的優缺點比較

	優點	缺點
IPSec VPN	1. 廣泛服務的支援，可用性高 2. 可與其它產品做完整整合，進而確保最高之安全性 3. 效能佳	1. 須在用戶端的電腦上安裝軟體 2. 設定較複雜 3. 建立的地點較受限
SSL VPN	1. 市面上多數的作業系統皆提供且支援 SSL VPN 模式之 Web browser，所以幾乎到處都能設定 2. 透過 NAT 或是代理伺服器（Proxy device）運作時，對遠端使用者幾乎沒有影響 3. 不需安裝軟體於使用者電腦上 4. 總擁有成本低	1. 有大量連線時，因整體加解密運算而可能會大大降低了效能 2. 無法應用於 Site-to-Site 之 VPN 架構 3. 僅適用於 HTTP 通訊協定 4. 端點安全性

2.3IPSec 的運作機制

IPSec 是一個由 IETF 所提出並負責維護的通用架構。它為 IP 提供了許多安全措施的

服務，包括了目前最常用的 IPv4 及逐漸推廣的 IPv6。IPSec 定義了一個高階的元件導向架構，而不是詳細的去規範加密的演算法或是如何交換金鑰的方法。在概念上，IPSec 是設計用來保護網路本身的，所以對於架構在網路上的應用是不會受到影響的，因此應用程式應該可以不必修改便可以正常運作。由於 TCP/IP 通訊協定的運用相當地廣泛，所以若昇級 IPSec 來保護網路的安全性，目前所有的網路運作程式一樣能夠一如往常地傳遞資訊，所以若昇級到 IPSec 的話，能夠兼顧現存的系統。

由 IETF 所公告的 IPSec 相關文件主要討論了加密演算法、驗證演算法和金鑰管理三個議題，這些元件對於定義系統的安全架構有所助益，同時 IPSec 也希望能夠加入新的演算法到這個通訊協定組內，而不會對現有的架構產生太大的變化。使用符合 IPSec 的產品或是服務，主要是能夠得到增加安全性及這些產品間的互相合作的特性。

IPSec 使用認証標頭（Authentication Header；AH）和 ESP（Encapsulating Security Payload）兩個通訊協定來提供傳輸的安全，關於此二個通訊協定都分別在 RFC（如 RFC 1825、RFC 1826、RFC 1827 等） 中有詳細的描述。

認証標頭提供資料來源的確認以及可自由選擇的 Anti-replay 服務，所以其用途如下：

1. 提供 IP 裝載資料內容的確認
2. 提供 IP 裝載資料的認証
3. 提供 IP 裝載資料的唯一性
4. 經由序列號碼防止入侵

ESP 提供完整性、有限的傳輸流量限制及可自由選擇的 Anti-replay 服務，所以其功能為：

1. 經由加密的裝載資料完整性
2. 經由公眾金鑰加密的資料原始認証
3. 提供 Anti-replay 的功能
4. 經由安全閘道器作交通流量的限制。

IPSec 系統在傳遞資料封包的時候，必須透過事先協調好的安全參數來指明所採用的加密或者身份驗證的演算法、金鑰或其它的參數，所有的安全相關參數都會存放在所謂的 SA（Security Association）的資料結構中，因此，要達到什麼安全的程度，端賴二方所共享的 SA 內容而定。

SA 的主要工作包括：

1. 資料加解密
2. IPSec 金鑰的交換（IKE / ISAKMP）

3. IPSec 金鑰的管理（SAD）

4. 負責 IPSec 系統 Security Policy （SPD）

5. 所需要的密碼演算法、數學運算式及常用的功能函式包裝成函式庫，以加強軟體的再使用性。

IPSec 支援二種加密的模式:「傳輸模式」(transport mode)(圖 11)和「通道模式」(tunnel mode）(圖 12)。傳輸模式只對封包的裝載資料（payload）加密，而通道模式則對標頭與實際資料加以加密的動作。一般而這，通道模式是比較安全的一種方式，因為傳輸模式的標頭並未加以加密，所以可以從其標頭得知許多資料，因此較不安全。

整個 IPSec 系統大致可分為兩大主體：資料封包轉換與金鑰管理（如圖 13）[8]。

資料封包轉換：受保護的資料封包透過 AH 和 ESP 兩種轉換來達到傳送過程安全保密的目的。這二種方式均須透過事先協調好的安全參數來指明所採用的加密或身分驗證的演算法、金鑰與其他參數，這些參數都存放於一個稱為 Security Association（SA）的資料結構中。

金鑰管理：如何與其他主機協調決定 SA 及如何管理本身使用的 SA。

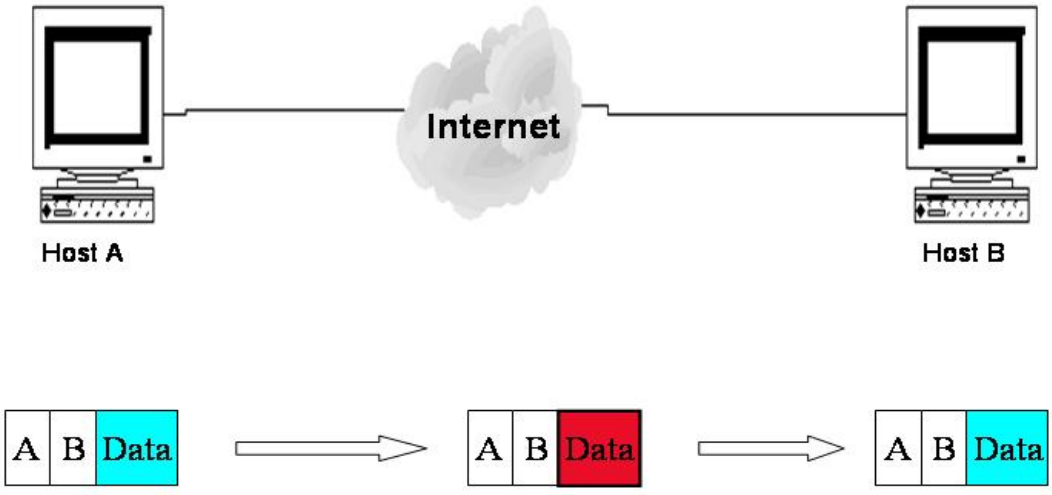

圖 11 IPSec 之 Transport Mode

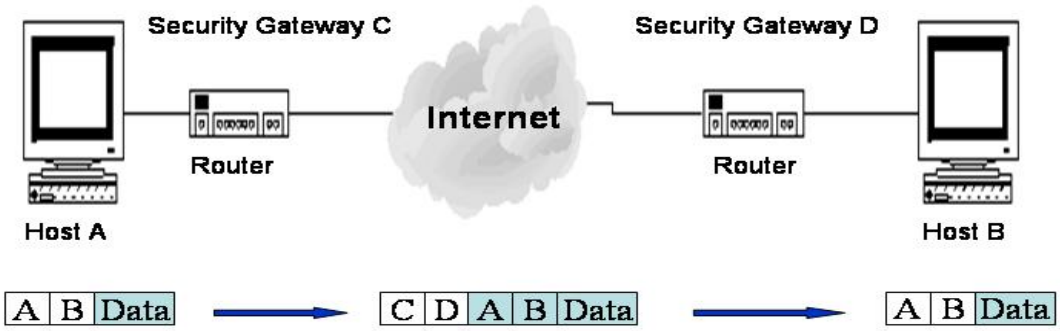

圖 12 IPSec 之 Tunnel Mode

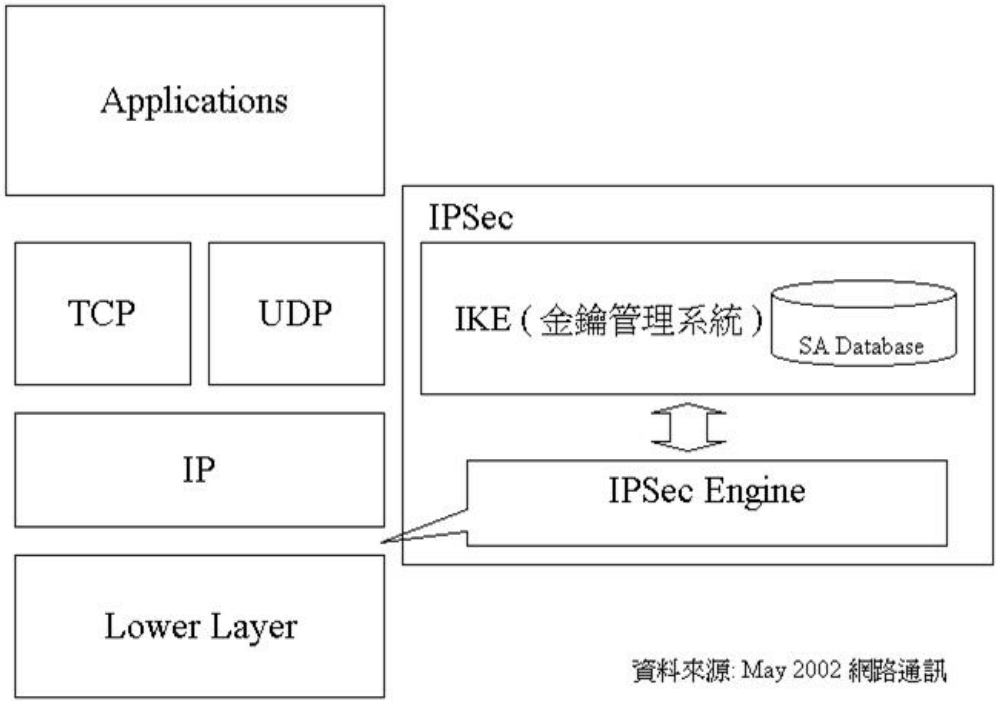

圖 13 IPSec 之 功能架構

2.4 Sun Microsystem 之作業系統概述[9]，[10]

2.4.1 Solaris 作業系統簡介

Solaris 作業系統是昇陽電腦（Sun MicroSystem）所推出的作業系統名稱，其提供了一個完整且具實用性、擴充性、安全性及容易管理的作業環境，昇陽電腦更將 Solaris 作業系統定位於是個最值得信賴並可以滿足市場上商業需求的作業系統，不論各種商業類型及企業規模的企業都能適用，透過昇陽電腦所提出的“Free Binary License”計劃，昇陽電腦計劃從 Solaris8 這個版本的作業系統開始，免除了 Solaris 終端使用者的授權費用，且其適用範圍包括了原始碼及可執行碼的存取部份，讓使用者可以在 8 個中央處理器（CPU）以下的系統中使用 Solaris 作業系統時，可以跟以往版本的使用經驗一樣不受限制，也不必再付任何授權使用的費用，而且使用者可以從網際網路上自由地下載 Solaris 的程式碼，如果使用者需要一份作業系統的安裝媒體或是 Solaris 作業系統的操作手冊紙本時，也只要繳納媒體製作的成本費用及運送過程中所產生的運費即可，因此企業可以不必花費許多的金錢在購買企業使用的作業系統上。

再者，在今日的 UNIX 作業環境中，Solaris 作業系統是市場的領導品牌之一；因為在昇陽電腦中存在一個「網路即電腦 (The Network Is The Computer)」的觀念已超過 15 年以上，所以 Solaris 作業系統的原始設計就是以網際網路上的應用為主，而 TCP/IP 這個網際網路最重要的通訊協定，更在 Solaris 作業系統的網路運算功能中有超過 15 年以上的使用經驗，經過這麼長的時間考驗後，Solaris 作業系統擁有一個小而穩定的核心系統、作業系統程式模組化、系統管理容易及擁有可擴充元件和定義清楚的介面等設計優勢，因此 Solaris 作業系統成為建構企業關鍵應用軟體的一個最佳應用環境的解決方案之一，而且 Solaris 作業系統更能夠協助使用者充分利用「網路效應」的驚人力量，創造出企業的競爭優勢。而 Solaris 8 作業系統以後的版本，其功能不斷地加強，也比以往的版本功能更加完整，並且透過昇陽電腦的測試計劃，可以確保大部份現有的應用軟體仍然可以繼續使用，而不必大量地修改原始碼程式並重新編譯後才能執行，節省了企業許多應用程式開發的時間。

當企業開始在際網路上發展業務的時候，可以發現早已有許多的企業已經在網際網路上發展各式各樣的業務了，這個現象，昇陽電腦認為代表的意義為在網際網路上存在許多各式各樣的資訊設備必須一同作業來完成企業的相關業務，而這些資訊設備要一同作業的話，便必須能彼此互相溝通協調，才能完成這些業務上的需求，所以不論是從 PDA、個人電腦到工作站或大型電腦主機等設備，都有可能會需要彼此互相合作來完成一項業務。昇陽電腦更對市場承諾昇陽電腦一定會支援開放產業標準及一般業界中大眾所熟知的通用標準，因此在 Solaris 作業系統中，昇陽電腦提昇了一些更高層次的共通功能，透過這些新功能的輔助，可以協助使用者在網際網路上與眾多的使用者及不同的資訊設備相互溝通。同時，Solaris 作業系統是一個具備許多功能的使用環境，例如 Solaris 作業系統便提供支援許多種不同語言的功能，包括文字表述複雜的阿拉伯語、希伯來語等，只要透過使用 Solaris

作業系統，企業只需要安裝一次作業系統，便可以在伺服器主機或個人電腦上擁有支援多種語言的環境，而且可以在任何時候，為了因應實際應用的需求而很容易的增加或移除其中某一種語言的環境，Solaris 作業系統可支援 37 種以上的不同語言及 123 個地區。同時，Solaris 作業系統也提供與其他作業系統一同作業的彈性空間，例如 Solaris 環境中，可以透過 PC NetLink 軟體的協助來提供 Windows 環境的網路服務功能，包括整合目錄、授權及檔案、列印服務等；而目前在 Intel 平台上很流行的 Linux 作業系統，也可以在不須修改程式碼的情況下，使用 Linux 這個免費的作業系統和 Solaris 作業系統做溝通，這項功能，大大地提昇了 Solaris 作業系統與其他不同環境系統的合作關係。

此外，Solaris 作業環境從一開始就將網路功能包括在作業系統之中，而且不斷地更新並加強，以滿足不同時期中客戶需求的做法，使得 Solaris 作業系統得以成為企業應用系統中最主要的作業系統之一，而且在 Solaris 作業系統環境中已經被使用證明過的許多功能之中，有許多的功能是可以讓現有的資料中心或企業業務的運作更加地靈活彈性，同時，也能增加一些新的網際網路方面的應用方式，為企業的網路環境帶來更多的效益，所以昇陽電腦希望成為企業工作上的共同夥伴，這也就是為什麼昇陽電腦認為 Solaris 作業環境不僅是最佳的 UNIX 平台，更是網際網路時代最佳的作業系統，也是企業的最佳夥伴。

在網際網路上發展業務，其業務的運作與系統運作時間、員工、客戶及各工作夥伴甚至各供應商之間的關係十分地密切，而且目前這種關係已從早期的區域性逐漸透過網際網路而擴展為全球性連結，在現在的市場環境中，業務相關系統的停機時間不僅會影響企業整體的運作，連帶也會造成整個供應鏈活動的影響，尤其是提供線上即時服務這一類型的企業應用，其影響更是明顯，這意謂著市場對於零當機時間的系統需求，不再侷限於傳統的金融機構或電信公司而已，在網際網路環境中的每一個單位，都可能會有零當機時間的需求存在，無論是客戶端、供應商、合作夥伴或是公司內部的使用者，都會希望能夠每天 24 小時之間都能得到需要的服務而不被中斷，所以穩定的作業環境也是 Solaris 標榜的特色之一。

Solaris 作業系統以產品、架構、服務及通訊這四個基礎為支柱，提供系統整體的基礎服務（圖 14）。其中產品強調 Solaris 作業系統所提供的功能是企業整個作業環境中的一環；而架構則是描述 Solaris 如何建構及如何與其他網路架構之間互相配合來建構使用者所需的應用環境；在服務方面，Solaris 作業系統則包括了優良的客戶支援、專家服務、顧問諮詢服務及其他可以提高企業生產力的服務，以提昇企業整體的競爭力；而通訊則是強調客戶及軟體開發者最感興趣的地方。透過這四項功能，所以 Solaris 作業環境的產品便擁有使用性、擴充性、多方性及可靠性等特性。

圖 14 Solaris 作業環境中的四大基礎 （資料來源[9]）

昇陽電腦擁有 15 年以上網路運算相關協定的支援經驗，而 Solaris 作業系統也提供一個純熟的網路運算架構，透過支援目前最新的網路通訊協定及標準，Solaris 作業系統成為市場上具備靈活、可信賴、最新網路運算應用軟體的理想開發平台。

由於昇陽電腦在各種不同標準上的影響力及對開放原始碼標準的鼓吹與支持，Solaris 8 作業系統開發出許多新的網路運算功能，例如服務位址通訊協定（Service Location Protocol，簡稱 SLP）這個標準，它是一個新的網際網路工程作業標準（Internet Engineering Task Force；IETF），它讓管理者可以建構一個以服務為主的網路環境，不需要針對因為系統結構或者是使用者端的一些改變而必須投入更多的管理心力。此外，Solaris 作業系統也是市場中第一個支援新一代定址方式 IPv6 這個新的 IETF 標準的重要商用軟體，它可以讓網際網路的效能與作用發揮得更完全並可以適用於新一代的網路環境。

Solaris 作業系統同時也提供頻寬管理及網路多重路徑（MultiPath）等服務來加強資料中心能提供更好品質服務的能力，這些功能都是許多資料中心在提供服務時，常常會被要求的網路運算功能，透過這些功能能夠加強、滿足企業環境的各項需求。在網際網路上，TCP/IP 是最常被採用的通訊協定之一，而實際上，大多數的的企業商業應用系統，也是依循這個通訊協定來撰寫及開發。在 Solaris8 的作業系統中，昇陽電腦發現也解決了一些 TCP/IP 方面的問題，例如利用一個新的 TCP 登錄機制，讓每次通訊連結之後會自動的終止；而 Solaris 作業系統更利用 LDAP 這個協定原有的優勢，來整合使用者端與應用軟體，以彌補現有 NIS 及 NIS+服務的不足，這項功能，讓現今的 Solaris 認証機制更加地完善及彈性。

Solaris 作業環境中最新及被使用驗證過的特性有下列幾點：

- 動態設定（Dynamic Reconfiguration）功能：經過改良後，更能支援多通道及負載平衡等功能，例如主機透過網路多重路徑的功能，能將網絡的負載分散到其他的網路介面。

- Live Upgrade：由於 Solaris 作業系統的模組化，所以使用者可以線上安裝

一些需要的升級套件，並透過一些簡單重新啟動的動作，即可完成升級的步驟。

- Real Time：Solaris 加強了作業系統本身的功能，所以能提供更短的回應時間及應付不斷增加的網路活動。
- Failed Device Lockout：當系統重新啟動時，會自動將損壞或是有問題的設備從線上排除，避免因為該故障的設備而造成系統不斷地重新啟動。
- IPv6：從 Solaris8 以後版本的作業系統都支援這個下一代的網際網路通訊協定，對於 IPv6 對於位址數量的限制較小，可以發展許多網路應用，而且 Solaris 作業環境可以啟動 IPv6 enable 模式，即使是在 IPv4 的網路環境中，一樣可以完成所需的網路功能，不需改變設定。
- IPSec：在 Solaris8 以後的作業系統版本中都加入了此項 IP 安全功能，可加強網路安全、防止侵入、資料的機密性並可建置虛擬私有網路。
- 雙堆疊的通訊協定（Dual-Stack Protocol）：Solaris 作業系統支援 Dual Stack Protocol，如圖 15；Dual-stack socket interface 如果允許 IPv6 的通訊協定時，會優先使用 IPv6 的通訊協定，否則會自動改成 IPv4 的通訊協定，如圖 16 所示。

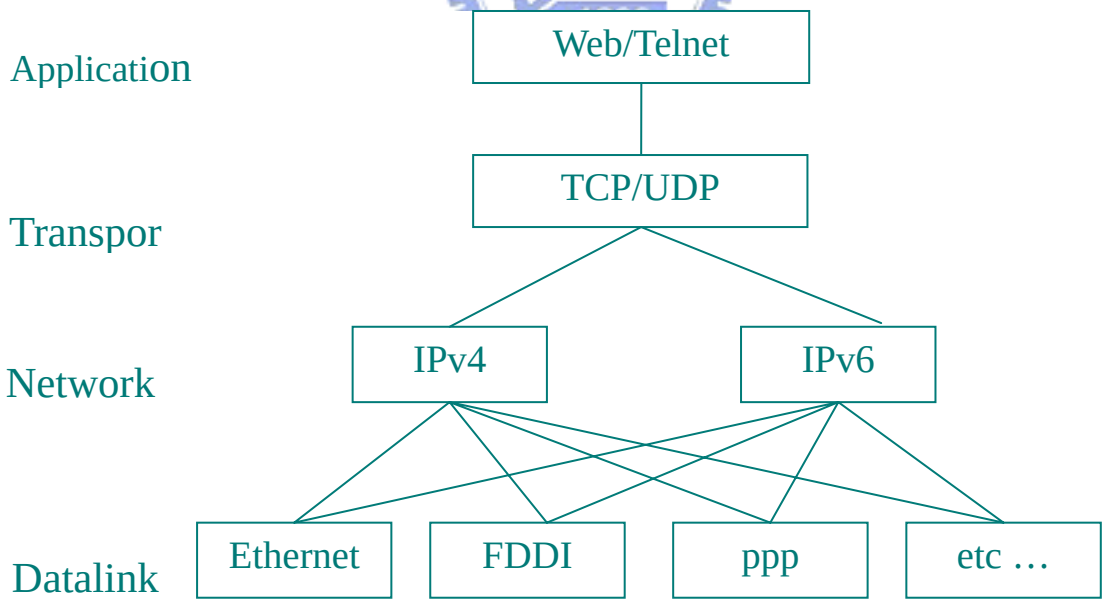

圖 15 Solaris Dual Stack Protocol （資料來源：[9]）

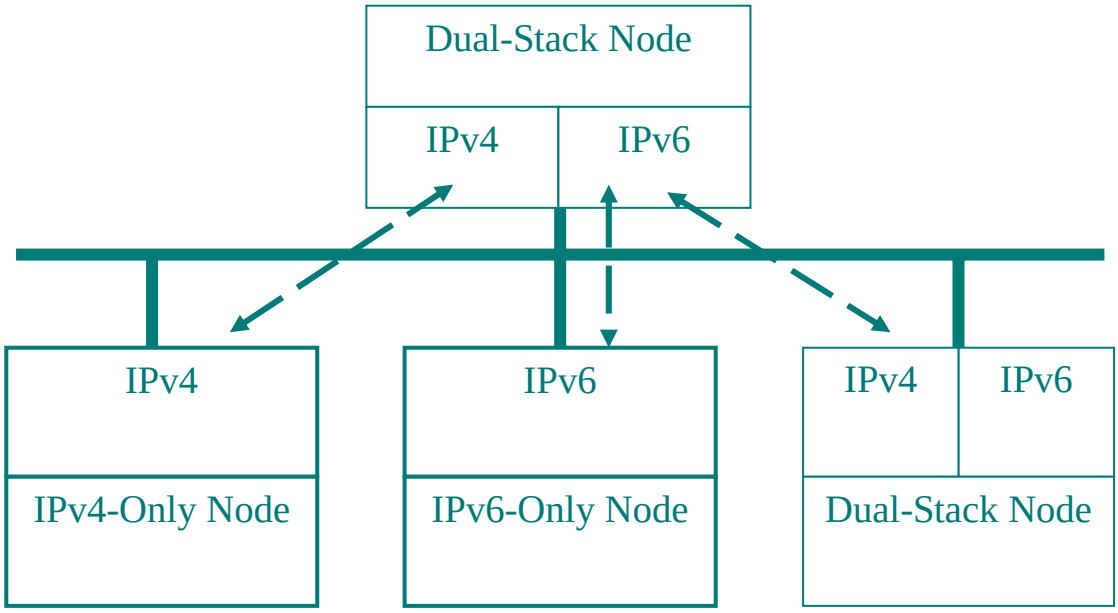

圖 16 Solaris Dual Stack socket interface 通訊協定選擇之順序

(資料來源：[9])

- Mobile IP：Solaris 作業系統開始可以利用 IP 位址來管理一些無線設備，讓這些高移動性的設備，可以透過固定的 IP 位置來進行資料存取的作業，而不必管這項設備是從哪裡連上網際網路的。這個功能允許使用者可以在任何地方從任何可以網路存取的設備作業。

- Java Virtual Machine（JVM）擴充性的強化：加強了 JVM 的功能，當系統的 CPU 增加時，效能也會隨之強化。

- Removable Media：加入目前市場上一些新的儲存媒介的驅動程式，如 Jazz、Zip、DVD 等，以允許使用者在選擇儲存媒介時，更有彈性。

- Solaris Web Start：將作業系統的安裝過程從早期的文字命令模式改為更容易操作的網頁安裝模式，簡化了安裝及配置的過程。

- Web-base Enterprise Management（WBEM）：將許多系統相關的管理工作改為以 Web 為基礎的系統管理介面，讓系統管理者可以更輕易地完成系統管理工作，而這些系統管理工具不一定是由昇陽電腦所提供，也可能是一些市面上已存在的系統管理工具，只要利用昇陽電腦所提供的 Sun WBEM SDK 軟體作為管理應用軟體的開發工具，讓 Solaris 應用軟體的管理，可以像在其他平台上一樣的便利。這是昇陽電腦所提供以 Java 技術為基礎的工具及服務。

- Role-base Access Control（RBAC）：支援 RBAC 此更前端的安全功能，以角色為基礎的控制機制，讓系統管理者可以在執行各項系統管理工作時，更有效的管理系統所需的安全功能。

2.4.2 Solaris 的網路設定

以下就 Sun Solaris 的作業系統環境下，針對如何設定並定義 IPv6 的網路介面、建構一個資料傳遞通道，作一個簡單的說明。

首先，在 Solaris8 的作業系統環境下，對 IPv6 與 IPv4 新增的特性比較如下：

1. 擴大了路由（routing）及定址的能力：IPv6 增加了網路定址位址的大小，從原先的 32 位元擴大到 128 位元，因此可以提供更大量的定址空間，並且在 IPv6 中定義了新的 anycast 位址，anycast 的功能為其定義了一組網路的點，當有任何封包是要傳送給該 anycast 位址的時候，它便會傳送給該組網路點中的某一個點，而 anycast 位址運用在 IPv6 的路由上時，則可以允許這一組網路位址根據其網路的流量情形來決定其路由的路徑。

2. 封包標頭（Header）格式的定義：原有的 IPv4 的部分標頭欄位已經在 IPv6 中被改為選項或是已被移除，如此便可以讓 IPv6 的標頭在封包控制及保持頻寬的成本上，不會因為可定址空間增加而造成成本也大幅增加的情形。雖然 IPv6 的位址是 IPv4 的位址四倍長，但是在標頭的大小上卻只有 IPv4 的二倍大，沒有因此便成為 IPv4 標頭的四倍大小。

3. 增加選項的支援：IPv6 標頭中的選項可以允許比較有效率的傳送封包、較少必要的長度限制及保留將來可以更有彈性的新增一些選項欄位。

4. 服務品質管理：IPv6 增加了在封包上加上標籤的方式，因此可以針對特殊的網路服務需求或是特別的網路流量來做一些區隔管理，因此更能快速的針對不同的網路狀況及各種需求來加以適當地分隔，而讓網路的品質更能充分的發揮。

5. 較佳的安全性：IPv6 中定義了一些認証及資料完整性的檢查，這樣便更能確保網路上傳遞的資料之正確性與資料安全。

Solaris 作業系統的網路功能設定方面，在每一次的系統開機過程當中，作業系統都會自動參考位於/etc 目錄下的網路相關設定檔的定義來自動完成相關的網路功能的設定，一般而言，網路功能的相關定檔有 hostname.<interface>、hosts、netmask、defaultrouter 及 nodename 等檔案。其中 hostname.<interface>檔案為相關的網路介面定義檔案，在每一次系統開機的過程中，會自動參考其內容，並針對該內容啟動適當的網路介面，其中的 <interface>會根據所使用的網路介面不同而有所差異，當該主機有二個以上的網路介面時，便必須在/etc 目錄下產生二個該類型的檔案，若是在開機的過程中，作業系統在/etc

目錄下並未看到 hostname.<interface>類型的檔案時，則該網路介面便不會被啟動，因此便不能透過此介面來進行網路的功能，此時便必須由系統管理人員以手動的方式將網路介面設定啟動。以目前較常見的網路介面名稱有：

hostname.le0

hostname.eri0

hostname.ip.tun0　（此介面通常為設定 tunnel 時使用）

hostname**6**.eri0　（此設定檔為 IPv6 的網路設定檔）

hostname6.ip.tun0　（此介面為設定 IPv6 的 tunnel 時使用）

而每一個網路介面都必須設定一個 IP 位址，如此才能與其它的機器在網路上彼此溝通，而在 Solaris 作業系統中，IP 位址定義的相關檔案為/etc 目錄下的 hosts 檔，這個檔案中的內容會定義機器名稱（Hostname）及 IP 位址的對應關係，所以必須在此檔案中定義機器本身所擁有的 IP 位址，不論是 IPv4 或 IPv6 的系統，均可在此檔案中定義，除此之外，若要得到網路上其他主機的機器名稱與 IP 位址的對應關係時，也可以定義在這個檔案中，或是設定 NIS、DNS 等名稱服務（Name Service）的機制即可。/etc/hosts 的內容範例如下：

#內容中，在#後方的任何文字會視為註解說明。

```
# Internet host table
#IP Address    主機名稱      主機的別名
127.0.0.1       localhost                #系統 loopback 位址
192.9.200.1          vpn1      loghost   #設定主機的 IP Address 及主機名稱
192.9.200.2          vpn2
…
```

而在 Solaris 8 及以後版本的作業系統中，若是啟動 IPv6 功能的話，則作業系統會參考/etc/inet/ipnodes 這個檔案的內容來對應 IPv6 的搜尋，一般而言，通常會在 Solaris 的環境內設定/etc/nsswitch.conf 的檔案內容，來決定作業系統在位址搜尋時所使用名稱服務的優先順序關係，舉例而言，我們會在 /etc/nsswitch.conf 設定下列的定義：

hosts：files　nis

ipnodes：files

此範例的意義為當作業系統要搜尋 IPv6 的位址時，會僅以/etc/ipnodes 檔案內的定義優先參考，當要搜尋 IPv4 的位址定義時，則會參考 /etc/ipnodes 及 /etc/hosts 的檔案設定，搜尋不到需要的位址對應關係時，則會繼續從 NIS 的資料庫中去找尋所需要的資料。關於 Solaris 環境中的名稱服務，可參考圖 17 及圖 18。

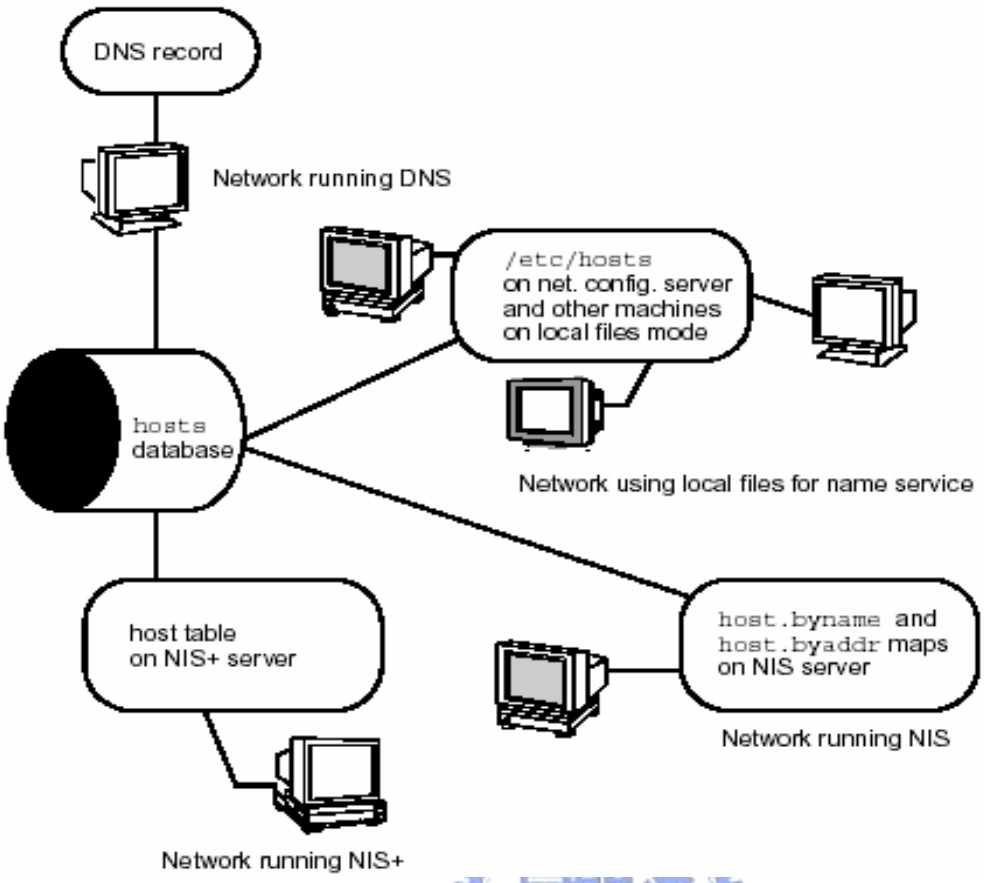

圖 17 名稱服務所使用的主機資料庫 (資料來源[9])

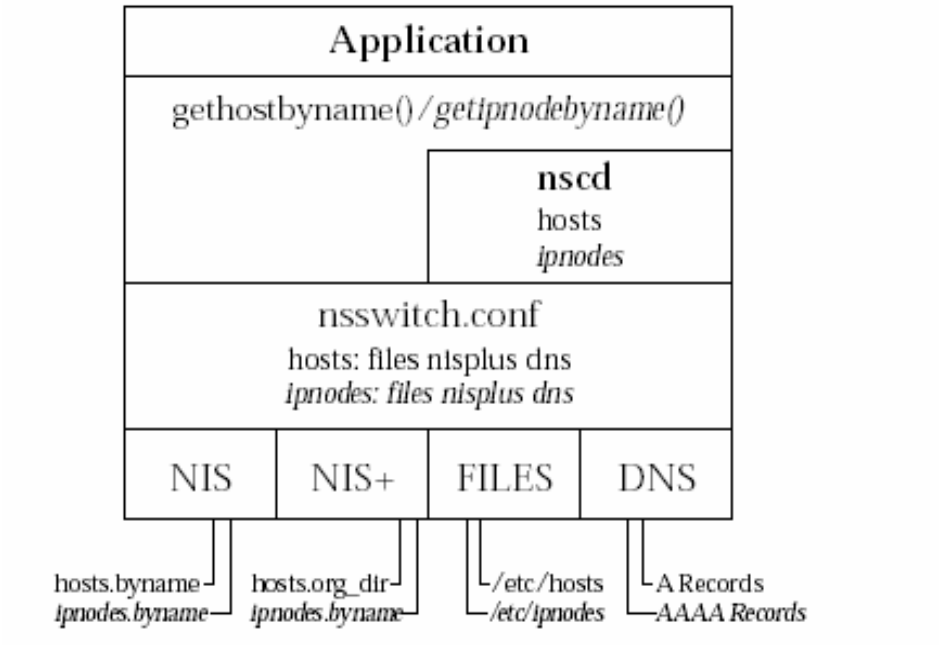

圖 18 nsswitch.conf 與名稱服務之間的關係 (資料來源[9])

所以，在 Solaris 的作業系統中，將 IPv6 啟動的步驟為：

1. 以超級使用者（super user）的身份登入想啟動 IPv6 的主機。

2. 執行下列指令（command）。

   touch /etc/hostname6.*interfacename*

   其中 interfacename 為網路介面名稱，需視實際情形而定，如 le0、le1。

3. 將機器重新啟動（reboot）。

4. 再以超級使用者登入，執行下列指令以確定 IPv6 的網路介面正常啟動。

   ifconfig –a

5. 將 IPv6 的位址設定到對應的網路介面或是適當的名稱服務資料庫中。

6. 若此網路介面要以手動的方式設定時，則可以單純地使用 ifconfig 指令便能完成所有的網路相關設定，但這種設定方式的缺點為，在每次主機重新開機之後，這些以 ifconfig 指令設定的網路相關設定便會消失，不會再產生作用，所以便必須再以 ifconfig 指令設定一次方能讓網路功能正常運作。

當網路介面的設定完成之後，這台主機便可以開始與網路上的其他主機進行溝通，但若要建立起一個資料溝通的專屬通道時，則必須在這通道二端的主機都完成在網路上的通道相關設定，才能順利建構起一個資料溝通的專屬網路通道；專屬網路通道的設定方法如同前面設定網路介面時所述的步驟一般，但通道二端的主機必須將彼此的 IP 位址都定義到該通道網路介面中，且目的地位址與來源位址成對組合才行。
設定專屬網路通道範例如下：

```
[root@vpn1]ifconfig hme0 plumb        #先設定主機 vpn1 的網路介面
[root@vpn1]ifconfig hme0 192.9.200.1 up
[root@vpn1]ifconfig hme0 inet6 plumb up   #設定主機 vpn1 的 IPv6 網路介面
[root@vpn1]ifconfig hme0:1 inet6 plumb
[root@vpn1]ifconfig hme0:1 inet6 4ffe:400:350:2db3:a00:20ff:fee6:3195/64
[root@vpn1]ifconfig hme0:1 inet6 up
[root@vpn1]ifconfig ip.tun0 inet6 plumb tsrc 192.9.200.1 tdst 192.9.200.2 up
    [root@vpn1]ifconfig ip.tun0 inet6 addif 4ffe:400:350:2db3:a00:20ff:fee6:3195
    3ffe:400:350:2db3:a00:20ff:fee6:3196 up
[root@vpn2]ifconfig hme0 plumb        #再設定主機 vpn2 的網路介面
[root@vpn2]ifconfig hme0 192.9.200.2 up
[root@vpn2]ifconfig hme0 inet6 plumb up   #設定主機 vpn2 的 IPv6 網路介面
[root@vpn2]ifconfig hme0:1 inet6 plumb
[root@vpn2]ifconfig hme0:1 inet6 3ffe:400:350:2db3:a00:20ff:fee6:3195/64
[root@vpn2]ifconfig hme0:1 inet6 up
[root@vpn2]ifconfig ip.tun0 inet6 plumb tsrc 192.9.200.2 tdst 192.168.0.1 up
    [root@vpn2]ifconfig ip.tun0 inet6 addif 3ffe:400:350:2db3:a00:20ff:fee6:3196
```

4ffe:400:350:2db3:a00:20ff:fee6:3195 up

[root@vpn1]ifconfig –a　　#以 ifconfig 指令確認所有網路介面已正確建立

lo0:flags=1000849 mtu 8232 index 1

inet 127.0.0.1 netmask ff000000

hme0: flags=1000843 mtu 1500 index 9

inet 192.9.200.1 netmask ffffff00 broadcast 192.9.200.255 ether 8:0:20:86:af:11

hme0:1:flags=2000841<UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 9

inet6 4ffe:400:350:2db3:a00:20ff:fee6:3195/64

ip.tun0: flags=2200851 mtu 1480 index 11

inet tunnel src 192.9.200.1 tunnel dst 192.9.200.2

inet6 fe80::c0a8:2/10 --> fe80::c0a8:1

ip.tun0:1:flags=2202851 mtu 1480 index 11

inet6 4ffe:400:350:2db3:a00:20ff:fee6:3195/128 3ffe:400:350:2db3:a00:20ff:fee6:3196

如此便可以啟動在 Solaris 環境中的網路功能，而且可以在網路中建構起一個安全的資料通道。

2.4.3 如何在 Solaris 環境中設定 IPSec

但是這個安全的資料通道是如何做到的呢？因為在昇陽電腦作業系統 Solaris8 或以後的版本中，均已將 IPSec 的安全機制納入到作業系統中，所以同樣可以很輕易地透過一些設定檔的設定之後，作業系統便能提供 IPSec 機制的安全性，而不需要再花費許多額外的成本。

在 Solaris 作業系統中設定 IPSec 的方法說明如下：

在 Solaris 作業系統中，共提供了二個工具程式來輔助使用者完成 IPSec 的相關設定，這二個工具程式分別為/usr/sbin/ipsecconf 及/usr/sbin/ipseckey。其中 ipsecconf 程式協助使用者設定主機的 IPSec 規則（policy），不論是流入（inbound）或流出（outbound）的網路流量（traffic），都會根據所定義的規則去決定所需採取的行動，而 ipseckey 程式則是協助使用者管理整合在 IPSec 中的 SA，同時 ipseckey 程式也允許系統管理者 root 以手動的方式來啟動 SA 的管理。當主機在開機的時候，會先自動檢查/etc/inet/ipsecinit.conf 的檔案是否存在，若存在的話，便會自動依據設定檔中的設定來啟動 IPSec 的規則，其檔案的內容為 ipsecconf 相關的規則定義，每一個規則都需定義在個別的一行，其規則定義的格式如下：

{pattern} action {properties}

當網路封包符合 pattern 欄位的定義時，便會再看 action 欄位定義的動作以決定要採取什麼動作來處理這個封包，而 properties 則是定義要動作時的一些特性，下列為這三個欄位定義值的參考表。

表 6　/etc/inet/ipsecinit.conf 設定檔欄位參考表

設定值	定義
Pattern	
Saddr	來源主機的 IP。
daddr	目的地主機的 IP。
smask	來源的網路遮罩。
dmask	目的地的網路遮罩。
Sport	來源埠。
Dport	目的埠。
Ulp	通訊協定，如 TCP、UDP 或其它的 layer4 的通訊協定。
Action	
apply	當 pattern 符合時，便根據 properties 欄位定義其動作。
permit	若符合 pattern 及 properties 二個欄位的定義時才允許後續動作。
bypass	當符合 pattern 時便不再檢查其他的規則。
properties	
auth_algs	AH 所使用的演算法，如 MD5 或 SHA1
encr_algs	ESP 所使用的加密演算法。
encr_auth_algs	AH 及 ESP 所用的加密演算法。
Dir	定義該規則是針對 inbound 或是 outbound。
Sa	定義 SA 的特性，可能為 unique 或是 shared 之一。

ipsecinit.conf 範例如下：

在主機 vpn1 的/etc/inet/ipsecinit.conf 的設定如下：

```
{saddr vpn1 daddr vpn2}     apply     {auth_algs MD5 sa shared}
{saddr vpn2 daddr vpn1}     permit    {auth_algs MD5}
```

第一行的規則定義了從 vpn1 到 vpn2 的所有網路封包都會加上 AH 標頭，並且使用 MD5 為認証的演算法，同時使用已存在的 SA 資料。而第二行則定義了從 vpn2 到 vpn1 的所有封包如果有加上 AH 標頭且使用 MD5 為其認証演算法時，則允許這個封包通過。同樣地，在 vpn2 的 /etc/inet/ipsecinit.conf 也必須有相對應的設定，如此在 vpn1 與 vpn2 這二台主機

間的所有互相溝通的封包便都會受到 IPSec 安全機制的保護。

當 IPSec 的規則定義完成之後，便需要設定一個 SA，在 solaris 的環境中，這項功能可以透過/usr/sbin/ipseckey 這個程式的協助來完成，其格式如下：

[add|update|delete|get|flush|dump|save] [ah|esp] spi [spi#] dst [dst addr] [options]

ipseckey 程式是在命令模式下一一輸入 SA 的相關資訊，而 SA 所使用的金鑰便會在此時由使用者設定完成。若在機器重新開機之後，仍要維持相同的 SA 設定時，則可以在設定 SA 相關資訊完成之後，便將相關設定存放在一個設定檔案中，在每次需要使用到 SA 的資訊時，便以 “ipseckey –f 設定檔”的方式來將 SA 相關設定讀入系統中來完成設定。ipseckey 的範例如下：

```
[root@vpn1]ipseckey                    #此時會進入 ipseckey 的命令模式
ipseckey>add ah spi 5457 src 192.9.200.1 dst 192.9.200.2 authalg MD5
authkey AF123BCDE89C53482AA4221CF89E343D  #key 可以由使用者自行決定
ipseckey>add ah spi 6543 src 192.9.200.2 dst 192.9.200.1 authalg MD5
authkey F123BCE6583132CF68DACB9FC8339D4B
ipseckey>dump              #顯示所有的 SA 相關資訊
Base message (version 2) type DUMP, SA type AH.
Message length 136 bytes, seq=3, pid=27140.
SA: SADB_ASSOC spi=0x198f, replay=0, state=MATURE
SA: Authentication algorithm = MD5 #使用 MD5 為認証演算法
SA: flags=0x0 < >
SRC: Source address (proto=0/)
SRC: AF_INET:    port = 0, 192.9.200.1 (vpn1).
DST: Destination address (proto=0/)
DST: AF_INET:    port = 0, 192.9.200.2 (vpn2).
AKY: Authentication key.
AKY: f123bce6583132cf68dacb9fc8339d4b/128
  LT: Lifetime information
CLT: 0 bytes protected, 0 allocations used.
.
.
.
Dump succeeded for SA type 0.
ipseckey>save ah /etc/inet/key        #將 SA 的相關資訊存在/etc/inet/key 檔案中
ipseckey>^D
[root@vpn1]cat /etc/inet/key
#This key file was generated by the ipseckey(1m) command's 'save' feature.
#begin assoc
```

```
add ah spi 0x198f encr_alg 0 auth_alg md5 replay 0 src vpn1 dst vpn2    authkey
f123bce6583132cf68dacb9fc8339d4b/128
#end assoc
#begin assoc
add ah spi 0x1551 encr_alg 0 auth_alg md5 replay 0 src vpn2 dst vpn1    authkey
af123bcde89c53482aa4221cf89e343d/128
#end assoc
```

2.5 Open Source

2.5.1 Open Source 的定義

在近年發展迅速的電腦世界中，發生了許多對網際網路的發展有重大影響的事件，這些事件都很特殊且令人津津樂道，其中較為人所熟悉的幾件事如：

1. 世界上最重要的計算機標準組織（IETF，網際網路標準制定委員會）於 1989 年成立，IETF 是一個開放性的國際社團，由參加它通信論壇 (mailing list) 的人們所共同組成，任何人只要願意付合理費用，即可成為 IETF 的一員，他們一年會舉行三次的會議，協調網際網路的作業、管理和演進，並且解決中小型範圍的通訊協定和架構問題，任何人在會議中都具有發言權，藉由會議討論去瞭解意見一致的程度，但是決議需獲得大多數與會人士的同意，但也不需要全部人士都同意方可。

2. 1991 年一位芬蘭的大學生（Linus Torvalds）寫了一個作業系統的核心，並且將此核心系統公開發表，在很短的時間內，Linux 作業系統的使用者不斷地增加，目前已超過七百萬人以上，且使用者數量仍不斷地在成長。Linux 作業系統更成為目前成長最快的伺服器作業系統之一，IBM、Sun 的伺服器都已宣佈支援 Linux 作業系統。

3. 1991 年 CERN 位於瑞士日內瓦的高能物理實驗室的程式師們 Tim Berners-Lee 等人發展出一種可以散佈資訊的超文字主從式(client-server)系統，稱為全球資訊網路 (World Wide Web)，此系統可以產生多媒體整合的電子文件於網際網路上，而 Web 大量地被使用的結果，造成了網際網路上的活動量激增，使用者只要透過 Web 瀏覽器，便可以很輕易地找到網際網路上的許多豐富資訊，使得網際網路成長得更茁壯。

4. 1994 年夏天在學校宿舍裡的兩位大學生（Yahoo 的楊致遠和 David Filo）透過使用可自由取得的軟體建立一個大學生的網站資料手冊，並成功地建立了一個新的商務模式(Business Model)，也建立了一家價值數百萬美元以上的公司。

這些以往我們未曾想到卻正在網際網路中發展的事都十分地重要，因為他們都大大改變了人們使用網際網路的習慣並促成網際網路的發展。如果在數年前，你問一些大公司的資訊部門主管們對於所謂自由軟體 (free software) 的看法時，他們很可能會告訴你，他們沒有使用所謂自由軟體這類的軟體，原因是因為這些所謂的自由軟體沒有特定的人或廠商可以支援，所以軟體本身不夠強健，若軟體出現問題錯誤而不穩定時，不能達到一般商業軟體的整體品質。但是這個觀點在這幾年之間正在逐漸改變中，自由軟體逐漸被許多大公司在找尋企業解決方案時納入參考考量之一。而小公司基於成本的考慮，使用自由軟體的情形更是較大公司頻繁。由於使用人數的增加，許多實用的軟體工具又不斷地被公開，無形中，資訊從業人員對自由軟體的觀點已經從以往的不穩定逐漸有所不同了。

1998 年一月 Netscape 受到 Eric Raymond "奠基" 論文—“教堂與市集”的內容影響，決定釋放其最著名的產品 "通訊家族" (Communicator) 之原始程式碼出來，讓人們能夠自由地取得這項產品原始程式碼，這個舉動大大地震撼了整個產業界。即使後來電腦市場上，仍然有需收費的官方版產品販售，但任何想要製造或販賣同類產品的人員，都可以很自由地利用這個開放原始程式碼作為發展基礎去發展個自的產品，創造自己的利基。在 Eric Raymond "奠基" 論文內容中指出，Linux 作業系統在技術上和市場上成功的關鍵是因為其以自由及可散佈的原始碼為基礎的發展方法。因此 Raymond 認為並宣揚開放原始碼軟體不只比非開放原始碼軟體更為強健，而且能有較好的支援及創新，因為整個軟體的使用者都是此軟體的協同發展者及驗證者，此一現象在使用者人數愈多的軟體中愈是明顯。

1998 年四月的時候，O'Reilly 公司舉辦了一場會議，會中邀集了許多自由軟體發展者中的領導人，一起討論 Raymond 論文中的內容，並且希望努力提高自由軟體在電腦領域中的氣勢。此次會議的關鍵性結果是這些自由軟體發展領導者們同意採用 Raymond 論文中所使用的名詞 -- 開放原始碼軟體 (open-source software)，來釐清和 "自由軟體" 相關的包袱；因為即使英文的 free 有兩個意義，人們已經習慣把自由軟體當成非商業性軟體，而不是非獨佔性軟體，所以傾向於免費的觀念，但自由軟體創始人 Richard Stallman 認為，”free software”中的 free 如同"free speech" (言論自由)中的 free 一般，是自由的意思，而非是"free beer" (免費啤酒)中的免費意思。在會議之後不久，Corel、Informix 和 Oracle 等公司便相繼宣告他們已經或願意在不久的將來要把他們的產品移植到 Linux 的作業平台上，此舉無疑大大提昇了開放原始碼軟體的聲勢，因為 Linux 作業系統正是開放原始碼軟體中為人所熟知的軟體之一。1998 年七月的時候， IBM 公司此國際大廠更宣告加入 Apache 小組，讓 Apache 的 Web 伺服器成為 IBM WebSphere 這項產品的核心，不僅如此，IBM 公司更投入一個由程式工程師所組成的小組，協助將 Apache 與 NT 的作業平台整合。1998 年九月的時候，Intel、Netscape 和投資公司 Greylock 和 Benchmark Partners 更投資 Linux 的發行商 Red Hat 軟體公司，此時，人們才開始意識到，早已有許多開放原始碼軟體已默默構成網際網路的使用基礎，如網域名稱服務 (DNS) 中使用的 BIND (the Berkeley Internet

Name Daemon)、傳送 e-mail 所使用的 sendmail 及 Perl、Tcl 和 Python 這些程式語言更已被 Yahoo!和 Amazon.com 這些知名網站大量採用來發展維護其網站，除此之外，這些程式語言更被納入許多作業系統中，成為其作業系統的一部份，如 Sun 及 Red Hat Linux 作業系統等。

忽然之間，開放原始碼成了電腦領域中一個被廣泛討論的話題，但開放原始碼到底是什麼呢？又如何能從開放原始碼中找到能獲利的商機呢？

根據 Open Source Initiative (OSI)對開放原始碼軟體的定義，開放原始碼軟體基本上必須符合：

1. 可以自由地再散佈

不論是把軟體送人或是將軟體放在一個完整的系統來銷售，都不會要求你付任何授權金，但並非是完全免費的，開放原始碼軟體可以收取複製的成本費用（如製作軟體光碟的費用）、散佈的成本費用或服務的費用。

2. 開放與散佈程式原始碼的自由

這個軟體必須有原始碼，且必須允許可以散佈原始碼。即使有些產品不直接提供原始碼，也必須要提供一個公開的來源，例如可以自由地從網際網路上直接免費下載或附贈免費的光碟、磁片等方式。這個原始碼必須是程式設計師可以修改的，不可以故意混亂原始碼或是提供一個必須用處理器先編譯的中繼程式。

3. 允許程式被修改以及產生衍生作品

這個授權必需允許修改程式及產生其衍生的產品，而且必須允許衍生之產品可以用與原來的軟體的授權同樣的條件被散播。

4. 需保留原著作原始碼的一致性(Integrity)

如果原始碼在建立時可以允許散佈修改檔案，這個授權可以禁止原始碼在修改狀態下不允許被傳遞。這個授權需清楚的允許散佈衍生的軟體，但可能需要衍生的產品跟原來的軟體有不一樣的名稱或版本，。

5. 授權條款對任何個人或團體都一視同仁，不得有歧視及差別待遇

6. 授權條款不得對特定領域或活動的應用有差別限制

這個授權不能限制軟體僅能使用在某一特定領域。舉例而言， 這個授權不能限制這個程式僅能被用在一個特定的公司或是特定的研究之中。

7. 授權條款的內容自動適用於衍生的作品

這個授權提供的適用性應該是針對所有利用這個軟體及由這個軟體而衍生而成的產品。

8. 授權條款不得專屬於特定產品之下

這個授權提供的權益不能針對某特定產品。如果這個程式是由某個軟體中摘錄並以這

個計劃的名義來散佈時，則全部擁有這個授權書的人都擁有與原始散播者相同的權益。

9. 散佈管道必須保持技術中立性，不得限制特定方式或平台才能取得.，且不得與特定技術相結合。

10. 授權條款不得對隨同散佈的其它軟體做出限制

這個授權書不能夠限制其他同樣以這個授權書散佈的軟體。舉例而言， 這個授權書不能堅持其他用同樣方式傳佈的軟體都要是開放性原始碼的型態。

為了協調程式發展者，開放原始碼的發展方式顯得特別重要，許多這類的軟體在發展之初並不是為了要有商業行為的販售目的，而只是單純地為了要解決各自所遇到的問題，有時候，這些解決方案只對特定的情形有所幫助，但常常一間公司也只需要這些可以運作的軟體便可以解決其所遇到的問題，因此許多開放原始碼專案便以此方式開始，而當程式發展者發現還有其他人也面臨相同的問題並正在想辦法解決時，這個程式就會開始流傳到各地。Raymond 便認為「最好的程式通常來自於作者要解決自己本身所遇到的問題，然而這通常也是許多人的面臨的問題，所以這個程式便開始散佈」。網際網路增加了程式發展者間彼此合作的機會，於是就出現了多方共同進行程式發展的這種新發展模式，程式發展者之間因此可以合作得更緊密，也因為開放於網際網路上而分享到更多。且如同社會學中的“Delphi”效應一般，程式擁有越多的使用者，便越容易發現程式的錯誤，因為會有越多考驗程式的機會，因此 Eric Raymond 形容這個情形為「把全世界都當成你的智庫」，而 Linus 也正因為將使用者視為程式的協同發展者，所以讓 Linux 發展地如此成功。

開放原始碼運動中最有意義的一面是讓程式發展跨越單一組織的限制，我們稱之為"開放原始碼"(opensourcing)，它的含意為"原始程式碼流出" (outsourcing)。開放原始碼能讓小型軟體發展團隊壯大，就最簡單的情形來說，散佈在全球各地的電腦工作者與程式撰寫高手組成的社群，彼此以獨立的方式透過網路合作完成軟體，而軟體測試方面更是由龐大不特定的志願試用者持續檢視軟體實用性與是否有任何的漏洞存在，並隨時隨地回報所發現的軟體漏洞。經由如此嚴密反覆的找漏洞與隨時修正的品質改良過程，開放原始碼軟體在穩定性方面的提升速度十分地快捷，與商業化私有軟體採用有限並特定的時間與人力審視軟體的品管方式迥然不同。

Raymond 認為開放原始碼專案在起始的時候，必須由起始者提出基本的想法或是未來的發展方向，接著再由所有遇到相同問題的人員共同參與，將其發展完成。因為大部份的公司在公司運作的過程當中，會不斷地遇到許多的問題，因此會有發展軟體來解決問題的需求，但大部份並非想改行賣軟體，所以開放原始碼的發展方法是可行的。

傳統發展軟體的方式所需的成本很高，而要維護既有軟體的成本，一樣也很高，但若是此軟體不能解決所遇到的問題時，就這個問題而言，此軟體就毫無用處，開放原始碼則是聚集全球遇到相同問題的人，因此可以透過全球參與的人員來分散成本到所有參與的組織中並且增加發展軟體所需的資源，這也讓軟體的發展者和協助除錯的使用者的人數增加，讓程式多了許多的考驗機會，可以很快地找到程式中的問題，此外，開放原始碼發展的過程

中也培育了能維護這個軟體的社群。

開放原始碼運動中最常被提出的範例之一就是網際網路標準制定委員會(Internet Engineering Task Force 簡寫成 IETF)的運作方式，IETF 所定出的標準規格，促成了網際網路的誕生，而 IETF 制定這些網際網路標準的過程中，多是藉由通信論譠(mailing list)的彼此討論，每個工作群組真正面對面的開會討論，只有在每年只舉行三次的會議中才有機會，任何對制定網際網路規格有興趣的人都可以參加 IETF 的通信論譠及討論會，而且每個人都有發言權，但是如果你沒有充分準備或團體的支持，那麼你的聲音將會很容易被忽略或被其他聲音壓過。長期參與 IETF 的哈佛大學人士 Scott Bradner 便指出："IETF 中沒有所謂的投票過程，只有彼此試著去瞭解意見一致的程度，每個參與的人可以透過舉手、低哼示意或其他主席可以認可的方式來表達想法，但是最終決議的結果需獲得大多數人的同意，方算成立。若最終結果為 51 比 49，便不算有了最終結果，而要再次討論，但最終結果也不需要全部的人都同意才算數，但必需是 80 比 20 或 90 比 10 這種絕對多數的結果，才算有了最終結果。"所以大致上來說，IETF 的運作模式就如 Dave Clark 所描述的"沒有王者也沒有尊者，只有大略的意見一致和可執行的程式。"

當然，除此之外，就像其他許多成功的開放原始碼專案一樣，IETF 有一個基本的未來想法，這些未來的想法大部份都會在網際網路結構公佈欄(Internet Architecture Board，簡稱 IAB)中有詳細的描述。而 IETF 是一個由下而上的網際網路標準制定團體，在每個 IETF 的工作團隊裡，都有一名分組總監來帶領各團員，這些分組總監稱之為 Area Directors（簡稱 AD），他們並非由任何中央機關、政府或團體所指派，這些 AD 是在尋找最佳解決方案的過程中受同儕推派而產生的，專門負責技術審核與管理，而這些 AD 都屬於 Internet Engineer Steering Board (簡稱 IESG)或 IAB 的會員。IAB 是一個負責監督審核 TCP/IP 以及公佈各類標準的機構，而 IESG 與 IAB 這兩個機構都是經由 ISOC 認可的機構，IESG 通常都會由一名總指導來帶領會員，而這名總指導有時也會管理 IAB 內部的一些事。IETF 的內部存在許多政治相關的問題，這都需要靠各個工作團隊的主席竭盡所能，來維持標準製定的過程能有所進展。有時候，分組總監釐定出問題之後，便會徵求志願解決問題的人，但更常見的情形是一群對某個問題有興趣的人共同組成了一個工作團隊，並提出其解決方案，然後爭取其他人的認同。IETF 的成員大多依靠他們技術上的長處，來評估這些互別苗頭的解決方案，評估的重點在於這些解決方案是否簡單(simplicity)與諧和(interoperability)。

開放原始碼運動的提暢者 Eric Raymond 更把一份被稱為不名譽的萬聖節文件(Halloween Document)之微軟內部的備忘錄內容整理之後，反應到開放原始碼運動中，在這份備忘錄中，將開放原始碼軟體簡稱為 OSS，並認為 IETF 和 Linux 一樣都是微軟公司需要消除的威脅之一。為什麼微軟公司會如此看待開放原始碼軟體呢？因為在 IETF 各個不同的工作團隊中，結合了許多專業人士的想法，這讓 IETF 的工作團隊能夠很快速地建構出一個結構模型(architectural model)，並將之整合到 OSS 的專案計畫中。而且 OSS 的許多專案計畫中，因為有眾多實用的工具程式和簡潔的協定組合，所以能在伺服器的應用上，佔有一席之地，這種情形大大地威脅了微軟的軟體王國，所以微軟想藉由擴充 OSS 所使用的協定或發展新的協定來阻擋 OSS 專案計畫進入市場的入口。

在中古世紀的時候，由於缺乏健全的資訊傳播管道與方法，所有文件都僅能依靠手寫

這種昂貴的代價才能複製，所以當時的文件必需具有即時的價值才會以手寫的方式進行複製，如貿易紀錄、銀行交易記錄等，因此連當時的外交書信內容都簡潔到僅能傳送出即時的價值而已；而至於煉金術士、僧侶、科學家和哲學家的手稿，因為其內容就當時而言，大多無即時的價值，所以通常只有非常低的優先權會進行手寫複製，因此這些手稿的資訊便傳播得十分緩慢且傳遞範圍十分地有限。但是中國印刷術的發明改變了這個情形，印刷術將資訊傳播的門檻降到很低，以往獨力研究創新的學者們，開始可以和各地其他的學者們成立一個共同的研究團體，當然這個研究團體的建立必須有賴於公開地分享彼此的資訊，這種研究團體的誕生可說是因為學術自由的觀念，後來便把這類團體的研究過程稱之為科學方法。如果沒有形成研究團體的需求及學術自由的觀念，就不會有這些事發生，藉由彼此分享資訊的行為，使得科學團體能夠一同研究相同的問題，這種情形已經持續了有好幾個世紀。想像一下，如果科學家保留其研究的結果而不公開，甚至將結果商品化，所有要使用的科學方法都要再額外付費，那現今的世界將會是何種情形。當然這個想法目前聽起來已不可能出現，但是獨門秘方的確會使得科學的發展窒礙難行，而且科學也沒有演變成這種模式，因為科學的確不能這樣演進。

而網際網路就如同電腦數位時代的印刷術，網際網路再一次地大幅降低進入資訊傳播的門檻，原始程式碼不必再如最初的 UNIX 發展時代一樣，必須將要執行的程式先用紙帶複製後才能散佈，也不需要再用軟碟片、光碟片、行動碟或是其他形式的儲存媒體複製後才能散佈，只需透過 FTP、Web 的伺服器或是目前所流行的 P2P（Peer to Peer）軟體，便可以成為一個方便且便宜的軟體散佈平台。

雖然開放原始碼運動帶來了許多的希望，但不能忘記這幾個世紀以來，科學的遺產是開放原始碼發展模式的根基。在現今，電腦科學發展和電腦工業彼此間的關係尚未穩定的情形下，往往有為了短期內的經濟效益而迫使電腦軟體公司發展新的專利產品的情形產生，當越來越多的電腦科學研究工作是由電腦工業界所發起而不是學術界時，電腦工業界便必須透過公開分享創意的方式來培育學術界，也就是用開放原始碼發展的模式，然而，電腦工業廠商的這種作法並不是真的為了要嘉惠大眾或是其他更偉大的目的，這種作法只是為了自己的商業利益這個最現實的理由。

對開放原始碼領域的頂尖軟體發展者而言，金錢上的實際回饋幾乎不是他們最主要關心的事情，他們認為他們參與的是一場名垂不朽的發展過程，從歷史結果上也顯示出科學上的成就遠比金錢上的成功持續更久的時間，人們可能會記得過去一些有名的企業家，如：卡耐基、洛克斐勒、和福特，但我們卻一定會記得過去一些影響深遠的科學家或發明家，如：愛因斯坦、愛迪生、牛頓等，當目前這個年代的歷史被寫下之後，後人在電腦工業的歷史中，也許除了記得比爾蓋茲這位成功的軟體業鉅子之外，其他人的名字可能就記不得多少了，但人們卻很有可能仍然記得 Linus Torvalds 發展 Linux 作業系統的事蹟。而除了聲譽的成就之外，更重要的是，工業需要來自科學的創新研究，不斷地刺激市場成長或開發出新的市場，開放原始碼軟體便能以科學的速度和集合眾人的創造力來發展軟體並對軟體除錯，所以電腦工業中，也需要來自於開放原始碼發展的新一代的點子，開放原始碼不僅為電腦科學帶來進步，也帶動電腦工業不斷地向前邁進。

2.5.2 開放原始碼授權書之簡介

開放原始碼的授權書在最佳的情況下，能夠培養軟體發展者和軟體使用者間共享、同榮的合作關係。然而，開放原始碼的授權書有非常多種，在 2005 年四月為止，自由軟體基金會（Free Software Foundation，簡稱 FSF）所公佈的授權書已有六十種以上之多，而這些授權書目前也有許多已經建立起成功的模式，如 BSD 式的授權書、GNU 通用公眾授權書（GPL）或 LGPL、Mozilla 公眾授權書、Apple Public Source License (APSL)與 Q Public License (QPL)等，茲將三個最主要的授權書介紹如下：

BSD 式的授權書 (BSD-style License)

BSD 式的授權書這個名稱源自於 Berkeley Software Distribution UNIX，其為最早期發展出來的開放源始碼授權書，規範也最為寬鬆，BSD 式的授權書是由加州大學柏克萊分校所發展出來，其前身為一個 UNIX 版本的研發計畫。基本上，BSD 授權書的內容條款十分地彈性，只要符合授權條款的條件，便允許使用者自由進行使用、複製、修改、散佈或銷售的動作，所以軟體的開發人員可以在自行開發的產品中包含 BSD 軟體的元件，而該產品可以透過一般的商業管道來進行銷售，而不必如 GPL 的授權方式嚴格限制必須將所販售的產品，包括衍生的產品，都必須採用同樣的 GPL 開放原始碼機制公開。BSD 授權的特點是內容文字十分地精簡，且對於使用者的規範而言，主要也僅針對原始碼與可執行碼散佈時應載明的事項及特定組織名稱是否具背書(Endorsement)效力等項目進行規範。除 BSD UNIX 本身採取 BSD 形式授權之外，亦有許多軟體的授權條款採取 BSD 形式的授權方式，例如常見的 X-11、Apache software license、Python Copyright 等都是。BSD 式的授權書可以授權給衍生的私有軟體(Proprietary software)，就如同傳統的商業軟體一般不提供原始程式碼，至於衍生的私有軟體中的改變是否要回饋給原來的公用版本，廠商可以自行決定，並無嚴格規範。

然而在開放原始碼的社群中，有部份人士對於 third parties(非主流廠商)的產品利用公用的軟體來獲取私人利益，卻對原來的公用版本沒有任何回饋貢獻的現象感到憤恨不平，此種現象在經濟學中稱為 "搭便車"問題。僅管授權書中的條款沒有任何的規定要求，但這種授權方式仍引來許多志願者彼此合作去發展公用軟體，許多網際網路上的重要軟體都是因此而產生，且都採用 BSD 式的授權方式，如 BIND、Apache 和 sendmail 等都是耳熟能詳採 BSD 式授權方式的軟體。

GNU 通用公眾授權書 (GNU General Public License -- GPL)

GNU 的通用公眾授權書為源於 1983 年美國自由軟體基金會(Free Software Foundation)的 GNU 計畫，由 Richard Stallman 所草擬。當時，原本的目的是為了規範在自由的基礎下，GNU 計畫下創作軟體的分享。但漸漸隨著許多非 GNU 計畫中的軟體也引用了 GPL 的授權方式，造成 GPL 成為目前開放原始碼軟體中最常見的授權方式之一。根據 GPL 授權條款的規定，其條款明確地指出授權條款保障的是所有使用者可以自由複製、散佈與修改的權利，其它活動則不在條款授權的涵蓋範圍之內。此種授權方式實現了 GNU 專案"反版權"(Copyleft)的觀念。他們認為"版權"(copyright)阻斷了複製和衍生工作的可能性，而"反版權

"卻允許無限制的複製、修改。然而，"反版權"的使用者仍必須負擔一些責任，且散佈衍生的原始碼時，必須是免費且只能採用"反版權" 式的授權方式。

GNU 專案的重點是"free software”，其中 free 這個字在英文中有免費及自由二種意義，在這裡的定義為自由而不是免費。你可以出售自由軟體，但同時必須附上或確定能讓使用者可以得到軟體原始碼。GPL 的授權條款定義十分地嚴格，以致於很難和採取其他授權方式的軟體並存，例如你加強了一個採用 GPL 授權方式的軟體，那麼你所加強的軟體衍生部份也同樣必須採用 GPL 的授權方式，別無選擇。

然而，後來 GPL 的授權方式有了另一種比較寬鬆的授權型式，專供函式庫採用，稱之為 LGPL，其主要是針對在 GPL 的架構下，所有衍生成果均必須涵蓋於 GPL 的授權範圍之下，且因 GPL 無法與私有軟體進行整合，相對地，也減少了市場上軟體開發者將 GPL 授權的開放原始碼軟體加入其應用的機會。特別是一些程式庫(Library)方面的開發，如果仍堅持必須以 GPL 的方式進行授權的話，則不符合應用於市場上一般軟體開發過程的需求，因為軟體開發者開發出來的程式庫成果必須且僅能以 GPL 的授權形式公開，如此便不易創造出自身的利基及獲利。換句話說，假設這些程式庫的功能於私有軟體領域中也存在的話，因為功能相近卻可以不必公開原始碼的情況下，一般而言，很多的程式開發人員可能便會捨棄使用開放程式碼的程式庫而選擇私有軟體領域的程式庫。所以，在 GPL 這樣嚴格的條款限制下，開放程式碼的程式庫顯然不具有發展的機會，同時這也與原本開發程式庫的意義相背離了，基於此種需求，FSF 提出了 LGPL（GNU Lesser General Public License）這種新的授權方式。LGPL 仍是基於 GPL 的精神所發展出來授權方式，所以可以與 GPL 相容，也可以使得開放原始碼軟體有機會與 GPL 或非 GPL 授權的軟體結合。

GPL 的授權方式因為嚴格規範需將原始碼開放，所以擅長於防止獨家的壟斷行為，在 UNIX 市場發展過程中，扮演著相當重要的角色，在 2002 年四月份的 Freshmeat.net 報告中也指出，在其調查的 25,286 個開放原始碼軟體中，有 71.85%的軟體是採用 GPL 的授權方式，而同年的 Sourceforge.net 網站的報告中也有類似的結果，在網站中的 23,651 個軟體中有 73%的軟體採用 GPL 的授權方式。也因此，GPL 是開放原始碼軟體中最常被使用的授權方式，採用 GPL 授權方式的成功範例包括大眾所熟知的 Linux 核心程式(kernel)、GNU C 語言編譯器和 Samba 檔案伺服器等。

Mozilla 公眾授權書 (Mozilla Public License -- MozPL or MPL)

MPL 是網景公司(Netscape Communication)通訊家族開放原始碼版本 5 的一部份，由於各種授權條款之基本精神以及規範的方式有所差異，如 BSD 與 MPL 的授權方式強調研發成果的自由性，而 GPL 授權方式強調其對於開放原始碼的開放性，Apple Public Source License (APSL)與 Q Public License (QPL)則包含了衍生作品與原創作者間授權或相關權利的授與規範，在這樣的情況下，MPL 公眾授權書試圖找出 BSD 式的授權和 GPL 授權之間的均衡點，也因此，引申出許多開放原始碼的運作授權模式，以及衍生出許多種商業化的可能性。在 MPL 之下，允許個人自行發展衍生軟體，但是若 MPL 所涵蓋的原始碼部份有修改的話，則必須在網際網路上公開，讓其他人可自由取得，而 MPL 比 GPL 寬鬆的地方在於如果額外增加的原始碼 (不是修改過的原始碼)構成一個比較大型的專案時，那麼額外增加的部份可以採用不同的授權方式或也可以不公開。

即 MPL 條款將原始程式碼與可執行程式碼的授權方式分開；根據 MPL 的授權條款，即使經過多次版本的修改後，原始程式碼仍須保持 MPL 開放原始碼的模式，但可執行程式碼版本則可以交由軟體開發者自由選擇要以何種開放原始碼或是改採專屬軟體的形式散佈，也就是說，MPL 允許可執行程式碼修改版本的部份可以與一般私有軟體一樣，相同地禁止軟體使用者任意複製、散佈或修改，但因為 MPL 對於原始碼必須公開的規定，所以也保留了開放原始碼的特性。除了對於原始碼與可執行程式碼的分開授權之外，MPL 另一與其它開放原始碼授權機制不同的地方為 MPL 允許原始碼可以採取「多重授權」(Multiple-licensed Code)的模式。所謂的多重授權的意思為不限定原始碼只能單一性地選擇採用 MPL 的授權條款，根據 MPL 的授權條款內容規範，原創作者可以依照其意願來指定原始碼中某一部分原始碼採取 MPL 授權方式，而其它部分則採用另一種授權方式，而此處的"其它授權方式"並不侷限於任何一種授權方式，甚至可以是私有軟體的授權方式或其他開放原始碼的授權方式。

以上便為目前電腦軟體領域中最常見的三種開放原始碼授權書的介紹，將其授權條款內容的差異比較如表 7，但由於 LGPL 的授權條款乃是根據 GPL 授權條款來修改的，所以 LGPL 與 GPL 的授權條款已有所差別，因此將其視為一種授權方式來比較。

表 7 主要的開放原始碼軟體授權方式的差異比較

授權方式 / 特性	BSD	GPL	LGPL	MPL
允許使用、複製、修改及散佈軟體	是	是	是	是
允許開發私有軟體	是		是	是
有義務公開修改後的原始碼		是	是	是
有義務公開衍生產品的原始碼		是		由軟體開發者自行決定
允許多重授權方式				是

而開放原始碼的授權方式，正如同開放原始碼的經濟模式一樣，到目前為止都還在演進中，且不斷有新的授權條款被擬定，所以軟體開發者在開發軟體時，若是利用已存在的開放原始碼軟體加以修改的話，則授權方式需看所使用的開放原始碼軟體當初的授權方式而定，但若是全新的軟體開發的話，則可以好好地考慮該軟體所採取的授權條款。以下為考慮採用的授權條款時可以參考的事項：

1. 此軟體開發的目的是否為一般個人使用或單純以研究發展為目的

2. 所開發的軟體程式是否為以研究發展為目的的函式庫

3. 此軟體程式是否需要享有最大的商業空間

4. 此軟體是否需兼顧自由軟體之開放性與商業運用空間

2.6Tcl/Tk 語言

2.6.1Tcl/Tk 是什麼?

tcl 是一個高階的電腦語言，而 tk 是一個使用者界面發展工具。Tcl/Tk 是一個跨平臺 (cross-platform)、可擴充 (extensible) 的高階 scripting 語言，語法介於 shell script 與 C 語言之間，把這兩者結合起來，就形成了強有力的 GUI 發展套件，可用以發展 GUI 應用程式，其語法對熟悉 UNIX Shell Script 的人員而言，十分容易理解及撰寫。電腦環境中還有那些常見且具有相近優點的語言可以考慮呢？例如 Java、 Perl/Tk、 Python/Tk、 Guile/Tk 等語言其實也具有類似的優點，但因作者此次選擇以 tcl/tk 為系統開發的語言，所以對其它語言便不多加介紹。

Tcl/Tk 的 parser 只對命令列作很簡單的代換， 然後就把代換完的結果丟給該命令 (第一個字串) 處理； 每個命令看到的參數，則一律是一個個的字串。 例如若是程式設計師下： set msg "hello， world!" 則 parser 處理完後成為 set msg hello， world! 三個字串， 並把控制權交給 set 這個命令。

在 Tcl/Tk 的程式裡面， 所有命令看到的參數都是字串。 這在 tcl/tk 中很重要，也是 Tcl 與一些高階程式語言有所不同的地方。

Tk 程式包含兩部分 ：

1. initialization script ：建立起程式運作的基本環境， 例如建立視窗等等。 所有 initialization script 處理完之後就進入 event loop， 等待視窗使用者或系統對視窗上的元件下命令， 有事才動作。 (Perl: 呼叫 MainLoop 以進入 event loop; Wish: 不必呼叫任何函數。)

2. event handlers : 指定視窗上各個元件在接收到各種事件 (event) 時， 應該作何種反應。如程式中只有 initialization script，沒有 event handler，則這個程式要用 ^c 或 kill process 的手動方式將其終止。

2.6.2 Tcl/Tk 的一些基本名詞

每一個視窗或視窗元件 (例如一個 button， 一條 scrollbar 等等) 叫做一個 widget 。而 widgets 依其種類分成不同的 class， 通常第一個字母大寫時即代表一整個 class， 例如 Button， Scrollbar， ... 及範例中的 Label。

一個 widget 的大部分外觀與特性可以透過 configuration option 來設定， 例如字形，外緣寬度 ... 及"前景顏色" fg， "背景顏色" bg 等等

2.6.3Tcl/Tk 的優缺點

Tcl/Tk 的優點如下 ：

1. Free 。其軟體取得十分容易且不必產生費用。
2. 基本功能簡單，敘述與英文句子較接近，不須具備 object-oriented 或 threads 或 Microsoft Fundation Classes 的基礎也可學習，所以易懂、易上手，但是也因為敘述易懂，所以程式部份可能會顯得較冗長。
3. 沒有複雜的資料結構。
4. Tk 提供的功能幾乎已成為所有此類 (跨平臺， 可擴充) 語言的標準 GUI 元件。Tk 語言不僅跨平臺， Tk 的觀念與術語甚至跨程式語言， 是值得學習的工具 (Tool Kit)。
5. 維護程式時較容易。因為所有的程式碼都攤在陽光下， 所以維護比較容易。 (可用 grep、 sed 等工具協助搜尋並修改程式片段。) 此外， 它提供了較多的彈性，可以做一些 visual programming 不容易做的事情。

Tcl/Tk 的缺點如下 ：

1. 也由於基本功能簡單，敘述與英文句子較接近，所以程式部份可能會顯得較冗長。
2. 沒有複雜的資料結構 ， 故要有 HASH 等功能時不易表達。
3. 可以提供物件導向的功能， 但受限於語言基本架構， 很難把所有重要物件導向功能完整表達出來。
4. 因為所有的程式碼都攤在陽光下， 程式碼的安全性便不佳 ， 現可用 Tcl compiler 將 Tcl 程式碼編碼以保護 Tcl 程式碼。

2.6.4 Tcl/Tk 解譯器的取得

Tcl/Tk 軟體的取得方式常見的方式有 ：

1. 安裝 Linux 或是 FreeBSD 作業系統時，可以直接選取安裝即可。
2. 若是Sun Solaris 作業系統的話，則可以在Sunfreeware (http://sunfreeware.sun.com) 的網站上 download到所需的tcl 及 tk的軟體版本。
3. 在 MS Windows 下的版本， 可以參考 「Cygwin: 微軟視窗底下的自由軟體環境」， 把 gcc/g++ 和 Tcl/Tk 安裝起來 。
4. 其他作業系統的使用者可以到 Tcl/Tk 的網頁（http://resource.tcl.tk/resource/software/ports/） 下載可執行檔及豐富的文件。
5. 安裝完畢後，可以找到一個叫做 widget 的檔案 ， 執行該檔案。 這個範例程式展示 Tcl/Tk 的 GUI 基本功能， 同時可以讓你看看要達到這些功能的 Tcl/Tk 程式要如何撰寫。

2.6.5 Tcl/Tk 擴充套件 (extension) 的觀念

1. Tcl 程式庫是所有 Tcl/Tk 應用軟體最基本的部分， 外面包著一層殼 shell (類似UNIX 系統中的csh、tcsh) ，讓程式設計師下命令。 包著 Tcl 程式庫的 shell 叫做 tclsh。
2. 每個擴充程式庫 *extension* 提供不同的額外功能 ， 例如：
   (1). Tk 程式庫提供 GUI 元件； Tcl+Tk 外面包的 shell 叫做 wish (windowing shell)。
   (2). Tix 程式庫提供更高階的 GUI 元件； Tcl+Tk+Tix 外面包的 shell 叫做 tixwish。
3. 你可以在不同的 extension shell 底下用 info commands 指令看看它們各提供多少命令。
4. 用 Tk 建立 GUI 固然比用 C/C++ 方便， 但它所提供的元件還是稍嫌低階。 而「Tix 擴充套件」則提供許多更高階的常用元件 (widgets)， 例如 File Dialog Box， Tabbed Notebook 等等， 讓程式設計師可以用更短的程式碼建構出複雜的 GUI。

2.7 IP Filter 的功能

2.7.1IP Filter 簡介

IP Filter 是由Darren Reed 在許多人的幫忙下所撰寫完成的軟體，它可以安裝於Linux、FreeBSD、Sun Solaris等許多不同的作業系統平台上，而且目前已經是FreeBSD 2.2版及NetBSD 1.2 版作業系統的一部分了，此軟體提供了防火牆（FireWall）及網路位址轉換（NAT，Network Address Translation）的功能，透過使用者自訂的一些規則（rule）方法，來對網路介面上所傳遞的封包進行監控管理，而且該軟體現在已經發展成可以動態裝載的模組化程式，並且可以與若干作業系統的核心程式（kernel）直接進行溝通，意即當作業系統已經啟動IP Filter 的功能時，即使軟體的規則改變或是加入新的規則設定，也都可以在不需要重新開機的情形之下，讓改變過或是全新產生的規則能夠立即產生作用。該軟體及相關資訊可從http://coombs.anu.edu.au/~avalon/ 網址取得。

2.7.2IP Filter 軟體的規則（Rule）制定

IP Filter 的規則讀取順序採由上而下逐條讀取的原則，每當 IP Filter 讀完一條規則並完成判斷之後，無論結果是允許通過或是否定通過該封包，均會在該封包加上一個標籤（Flag）然後繼續讀取下一條規則，直到所有的規則都已經讀取一遍之後，再以最後通過的規則為主來決定該封包要如何處置。如下例：

```
block in all
pass in all
```

第一條規則所代表的意義為將所有經過該網路介面的封包攔阻，不讓其通過該網路介面；而第二條規則的意義則是完全不同，其意義為所有經過該網路介面的封包都予以通過，不會加以攔阻，所以此二條規則是互相抵觸的，但是，因為 IP Filter 的讀取規則的特性，其會讀完所有的規則並以最後符合的規則為準來處置封包，所以經過該網路介面的封包會被最後符合的規則，即第二條規則決定其處理的方式，所以所有的封包都得以通過該網路介面。

如果不想等到所有的規則都讀取完之後，才真正地決定其所適用的規則，則 IP Filter 提供了 quick 該關鍵字來解決此一問題，當 IP Filter 在讀取規則時，只要該封包符合該規則的定義，而且此規則之中有定義 quick 該關鍵字，則 IP Filter 即不會再繼續往下一個規則進行，而會直接執行此規則定義的動作，如下例：

```
block in quick all
pass in all
```

所以 IP Filter 讀取規則的時候，第一條規則被讀取之後判斷符合條件，因為該規則有

定義 quick 的關鍵字，所以便不會再繼續讀取下一條規則，而會直接地執行該規則所定義的動作，所以結果便是所有的封包都無法通過該網路介面。因此當所定義之規則眾多的時候，如何訂定規則的優先順序，便足以影響整體所需讀取及判斷的規則數目，也因此會影響系統整體效能的表現。

第3章系統設計

3.1為什麼選擇以 Solaris 作業系統建構 IPv6 的環境

本個案研究對象為國內某高科技產業的研發單位，由於企業不斷地成長，所以除了國內原有的辦公室據點之外，也開始增加了數個海外據點辦公室及投資海外的新公司，平時這些辦公室彼此之間如各自獨立的公司一般各自運作，有任何資料需要交換時，便會透過電子郵件或是其他常見的方式傳遞資料，但因為會有各辦公據點共同合作的專案產生，所以便會有別於一般資料傳遞的特殊需求產生，此時所要傳遞的不再是一般單純的資料，而是合作專案中的研發資料，這些資料對企業而言是十分地重要，所以在傳遞過程中的安全性便需要更加注意。然而，並不是非常頻繁或經常性地有傳遞這些重要資料的需求，所以若是為了這個需求而去租用一條專線的話，並不符合經濟效益，因此想找出一種可以彈性建構出較安全的網路環境來傳遞這些資料，所以虛擬私有網路便是一個值得考慮的解決方案。

但是，目前企業所建構的虛擬私有網路乃是針對二個辦公據點間的網路環境，所以這二個辦公據點間的資料傳遞便都會在虛擬私有網路的安全機制下受到保護，這對防範企業外部的非法入侵行為有很好的效果，但是對於企業內部的非法入侵行為，便無法達到相同的保護。根據近來的一些統計數字顯示，企業內部的非法入侵行為已經較企業外部非法入侵行為的次數為多，所以在傳遞重要的資料時，已經不僅僅要防範企業外部的人士，同時也要考慮防範企業內部的人員，只讓必要的相關人員知道資料傳遞的相關資訊會是必須的措施。

企業在專案進行中需使用許多不同的電子輔助設計軟體來協助完成工作，而這些電子輔助設計軟體先前多只支援 Sun Solaris、HP UNIX 及 IBM AIX 等工作站主機大廠的作業系統，在近年來才正式支援 Linux 作業系統，所以在研究個案的環境中，只有 Sun Solaris 及 Linux 二種作業系統的環境存在，而電子輔助設計軟體所支援的 Solaris 作業系統的版本通常並不會以目前市場上最新版本為主，相反地，各電子輔助設計軟體的廠商都會以軟體開發時，最穩定的版本為正式支援的作業系統版本，所以網路環境中的 Sun Solaris 作業系統均統一安裝目前多數電子輔助設計軟體廠商正式支援的 Solaris8 版本，此即本研究採用 Solaris8 而不用更新的 Solaris10 為建構環境的作業系統之主因。

在網路環境方面，目前 IPv4 定址空間的數量限制，沒有辦法達到使用一組 IP 位址便可以不論在那裡，使用者都不需要大幅修改網路設定，便可以在網路上進行溝通的點對點需求，然而 IPv6 有機會達成這個需求，而且在 IPv4 的網路環境中，也不易針對網路傳輸品質進行管控，而這個需求在 IPv6 的網路環境中可以較輕易地達成，加上 IPv6 在新選項中的一些設定，可以增加整體網路環境的安全與效能，再者，個案中的企業目前所使用的網路設備也都支援 IPv6，所以本研究採用 IPv6 的網路環境來建構所需的環境。

因此，本研究基於下列考量而採用 Solaris 作業系統建構 IPv6 的環境：

- 企業各辦公據點之間的機密資料傳遞需求並不頻繁，且資料量通常都不多，基於成本考量下，選擇不使用商業軟體而採用開放原始碼軟體的做法，而且開放原始碼軟體其軟體品質在網際網路中眾多使用者的測試下，不見得會比商業軟體的品質差，反而可以保留企業自己開發量身訂作所需軟體的空間。

- Solaris8 為最早支援 IPv6 的作業系統之一，所以在 Solaris 作業系統中設定 IPv6 的網路環境十分地容易。

- Solaris 作業系統的環境及可支援 IPv6 的網路設備在個案企業的工作環境中已經存在，所以不必再多花額外的成本來添購其他的設備。

- Solaris 作業系統內建 IPsec 機制，可以協助使用者輕易地建立起一個通道來加強資料傳遞時的安全性。

3.2為什麼選擇 IP Filter 及 Tcl/Tk 來建構系統

在最初研究該主題時，尚未有接觸過 IP Filter 該軟體的相關經驗，只是想要在軟體廠商控制了大部份的軟體市場生態，而消費者自主權實在是很小的情形之下，能夠找出一種使用者自己能夠充分掌握的空間，不必所有需求皆要靠軟體廠商發展的高成本且功能強大的套裝軟體方能達成，也由於近幾年來，使用者的意識逐漸覺醒，所謂的開放原始碼（OpenSource）的觀念也正不斷地為使用者所接受，所以想要在此一潮流之中，尋找一個空間，讓近年來愈來愈受重視的資訊安全問題，不再是被動地受限於軟體廠商所開發出來的商業軟體，而能夠真正地讓使用者決定自己的需求，進而產生適合的軟體。另一個原因則是因為軟體廠商所開發出來的套裝軟體，往往十分的龐大且功能眾多，但是使用者所需要的功能，往往只有其中的一小部分，無形中所花費的成本有部分浪費未用，十分的可惜。然而要開發出一個全新的軟體是非常花費時間、物力及人力的事情，因此，不禁回想起在早期電腦世界剛剛蓬勃發展的時候，大多數的軟體設計者十分樂意地與他人分享自己的程式及相關經驗，但是隨著網路的發達所帶來的龐大商機，吸引了許多的電腦人員加入發展電腦軟體的行列，但也逐漸將網路上分享的軟體，一步一步地轉變成昂貴的收費軟體。但是仍然有一群人依然為此觀念在努力著，所以發展出開放原始碼的方式，開放原始碼的方式不僅一方面能讓許多十分方便的軟體工具，能夠在網際網路上繼續地傳播，另一方面，也能讓軟體作者的著作權，不會因為昂貴的收費而造成盜版猖獗，著作權不被尊重。
在資料搜集的過程當中，偶然地發現網路上有此軟體，該軟體已經具備本論文的基本功能，而且又是開放原始碼的授權方式，符合本研究想要發展不受軟體廠商限制，但是又能符合系統安全的要求，而且該軟體的來源碼（Source Code）也可以在網路上取得，只要對其來

源碼加以修改，便能夠符合作者的安全需求，而且該軟體已能夠支援 IPv6 的定址方式，表示該軟體能夠順應網路的未來潮流，所以便決定利用該軟體來完成此論文系統的主體。

至於選擇 Tcl/Tk 為本研究中的程式語言主因如下：

1. Tcl/Tk 可支援大多數平台的程式語言。方便日後移植到其他作業平台。

2. Tcl/Tk 的語法簡單易懂，維護程式較容易。

3. Tcl/Tk 在撰寫使用者介面(User Interface)方面比其他程式語言容易。

基於上述理由，故本研究之系統便以 Tcl/Tk 所撰寫的使用者介面系統將所有的元件組合，加強資料在網路上傳遞時的安全性。

3.3 虛擬私有網路(VPN)的設計

本論文在建立虛擬私有網路的時候，採取利用 Soalris 8 可以設定建立通道（tunneling）的新特性，在虛擬私有網路的二端設定並建立起虛擬通道。

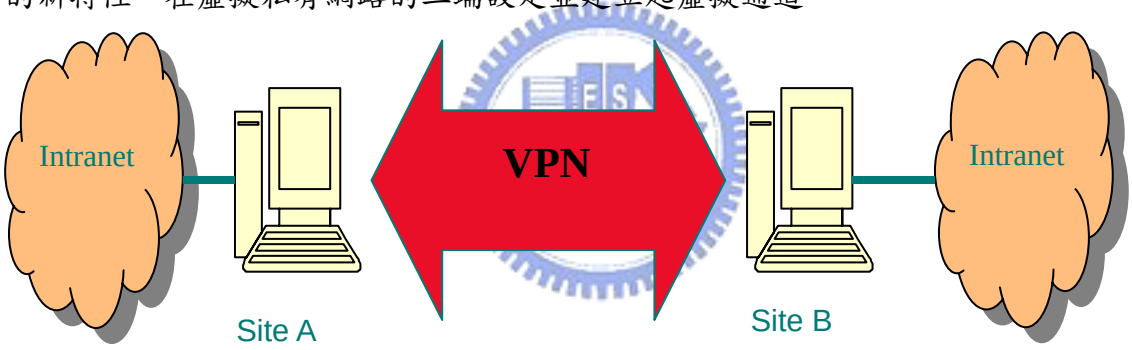

圖 19 系統建立通道模式前

圖 20 系統建立通道模式後

Solaris 8 要達到建立虛擬通道的目的，乃是透過下列模組的功能來完成：

strmod/tun

strmod/atun

為了要將在傳遞溝通的過程中相關性降至最小，在傳遞的二個節點中間所需經過的路由器，並不需要一定能支援 IPv6 的協定。這種機制稱為通道模式(tunneling)。基本上，IPv6 的封包是放在 IPv4 的網路封包內，而此 IPv4 的封包是真正透過 IPv4 路由器傳送的封包格式。 圖 21 解釋此通道模式的機制：

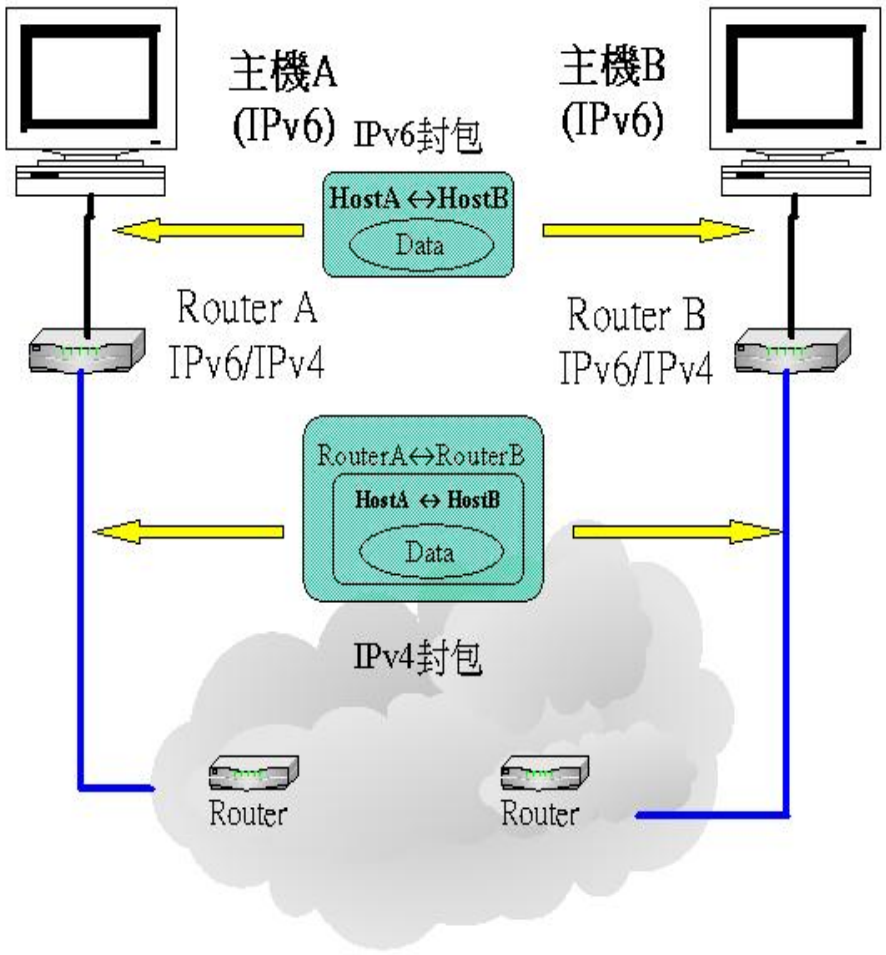

圖 21 通道模式的機制

在傳遞溝通的過程中，通道模式的型式可分為在二台機器間定義通道模式 (圖 11)及在二個路由器中定義通道（圖 12）。目前在網際網路上所定義的通道模式大多是為了其它的用途，例如MBONE (IPv4 multicast backbone)。而此類的通道模式在可預見的未來，將會愈來愈常被使用。

3.4 友善的圖形操作介面

本研究是利用 Open Source 軟體中的 IP Filter 及 Sun 的作業系統 Solaris 的系統指令來達成本研究加強資料傳輸時的安全性目的 ， 在許多的系統操作經驗中顯示，透過友善的圖形介面操作會遠比命令模式下操作指令更能讓使用者容易接受，尤其對不熟悉系統指令的管理人員而言，更是如此，光是要記憶許多的指令及路徑就可能讓管理人員傷透腦筋，因此本研究一開始便朝發展出一個簡易的圖形介面為方向，方便協助系統管理人員能夠在很短的時間內能夠利用該系統來加強網路上的資訊安全。

當使用者以系統管理人員(root)的身份登入的時候，只需一個指令即可開始使用該系統的功能 ，完成 IP Filter 及網路參數的設定。圖 22 為進入系統後所見到的啟始介面，此一介面也是設定 VPN 網路相關參數所見到的畫面。

此一介面可設定主機的網路參數，如 IP Address、是否支援 IPv6 等。若要建立一個 VPN 通道的話，則只需點選 Tunnel Setup 的按鈕選項，則系統即會開啟通道設定的新視窗（如圖 23），將設定一一填入後，只要執行 submit 選項後，所設定的網路參數會立即生效，因此系統管理人員不必記憶設定系統網路參數及通道設定時所需的指令及每個指令的語法格式及選項，系統管理人員只需記得每一欄位所代表的意義，一一填入即可完成設定，而個個選項的名稱也都儘量以一般常見的名詞命名，所以要了解其意義並不困難。

VPN 設定畫面中各欄位的說明如下：

Network Interface Name：網路介面的名稱，此為 Sun Solaris 所能辨認的網路介面的敘述，如 hme0 或 eri0；如該主機有二個網路介面的話，則需填寫下面的 Network Interface Name 欄位。

Supports：確認該網路介面支援 IPv4/IPv6 的程度

IP Address：填入網路介面所需的 IP Address

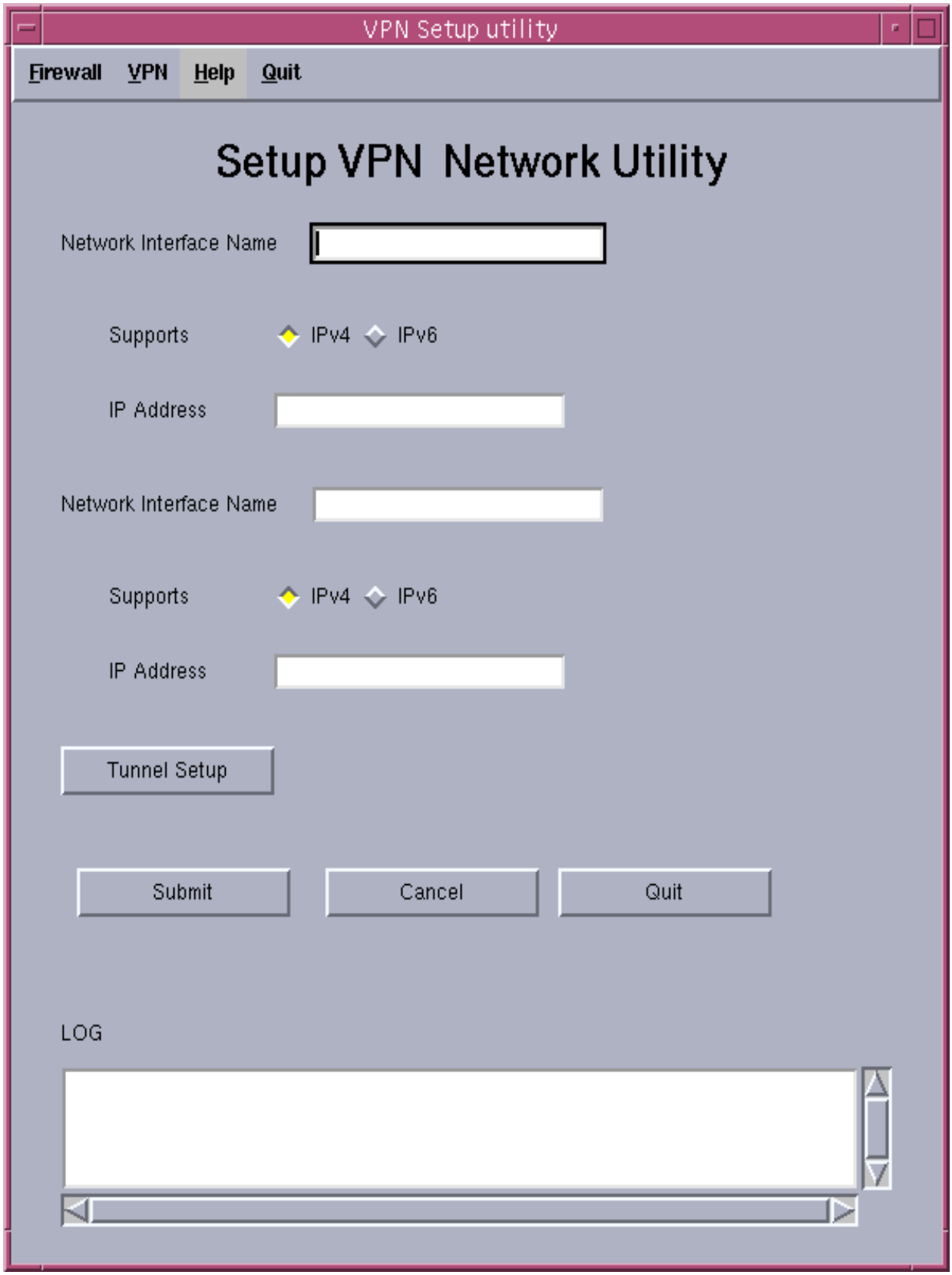

圖 22 VPN 網路設定的圖形介面

通道設定畫面中各個欄位的說明如下：

Tunnel No.：針對每一個 VPN 連線都必需確認其編號，而該編號可以由管理人員自行設定。

Source：來源主機的 IP Address，分為 IPv4 及 IPv6 二種。

Target：目的地主機的 IP Address，分為 IPv4 及 IPv6 二種。

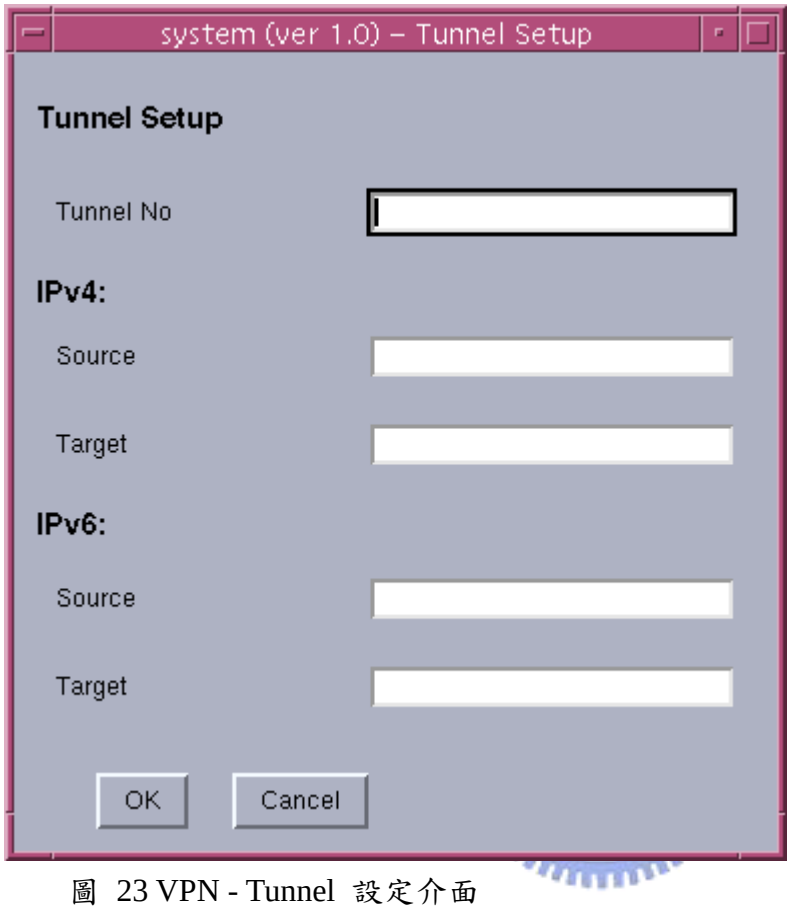

圖 23 VPN - Tunnel 設定介面

若是要設定 IP Filter 的 firewall 功能時，則系統管理人員只要以滑鼠點選上方的 Firewall 選項即可看到如圖 24 的畫面。此一畫面為進入 firewall 設定後所看到的預設畫面，畫面中會將目前系統中所執行的規則全部列出，而每一行在"#"符號後方的所有內容均視為說明，不會交由 IP Filter 執行或判斷。系統管理人員可以在畫面中立即編修所要增刪的規則內容，確定內容編輯完成後，務必要以滑鼠點選 Save 按鈕選項將所編輯過的規則記錄下來，如此離開本系統之後才不致於將剛才所做的改變遺失，而每次點選 Save 功能時，系統會自動將原先記錄檔的檔名後方加入時間註腳，以便日後要重回特定時間的規則時，可以輕易地回復到當時的情形。

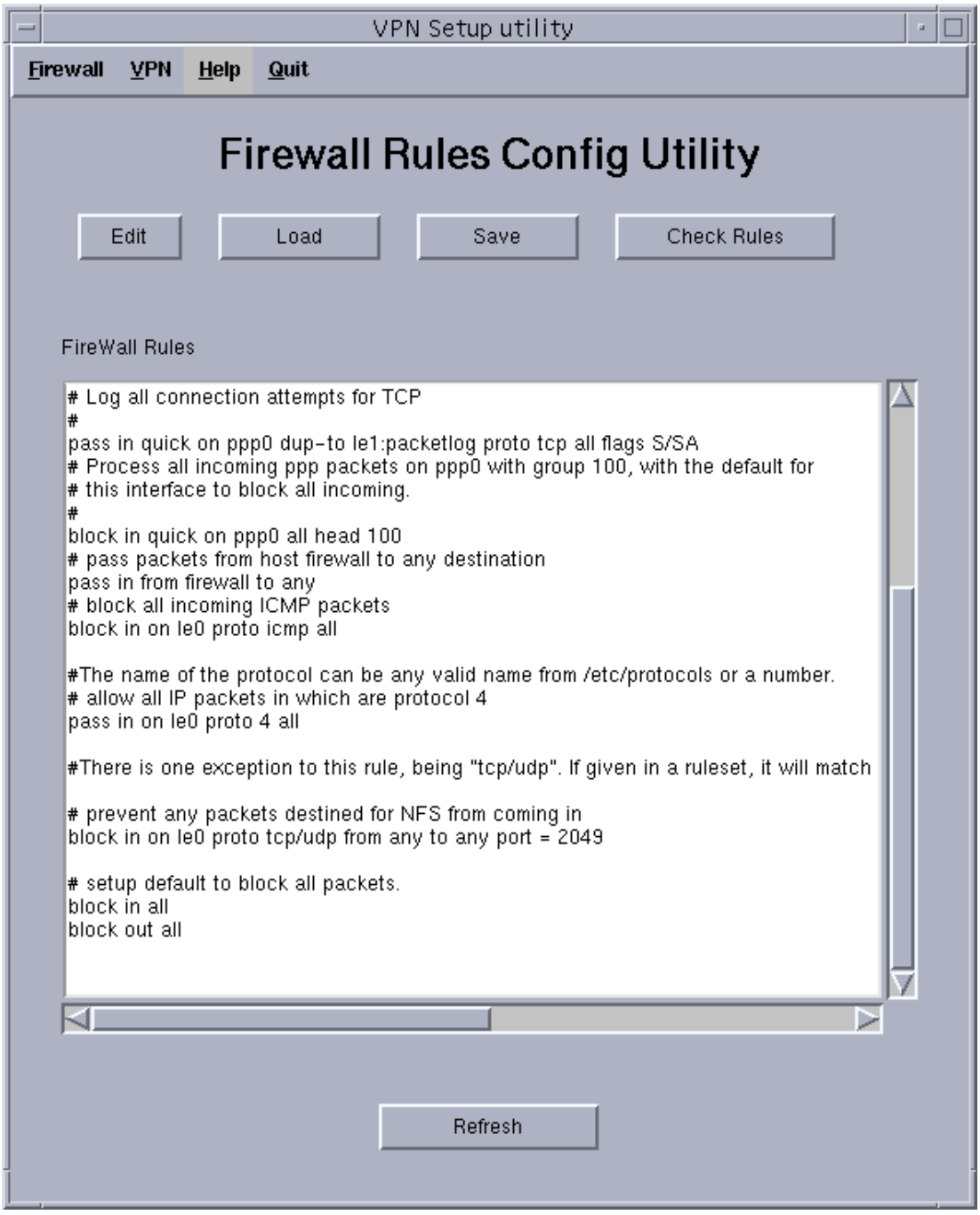

圖 24 Firewall - 設定的啟始畫面

而 load 按鈕選項則是專門為了將 firewall 規則回復到某一特定時間時所用。 使用者可以透過選單的方式點選所需的設定檔案，每一個設定檔案的最後一個欄位即為該記錄檔 save 時的時間點，若沒有時間點的檔案則為目前執行的設定檔。舉例來說，若最後一個欄位的數字為 200504160324 則代表此檔案的儲存時間為 2005 年 04 月 16 日 03 點 24 分。其畫面如圖 25 所示。

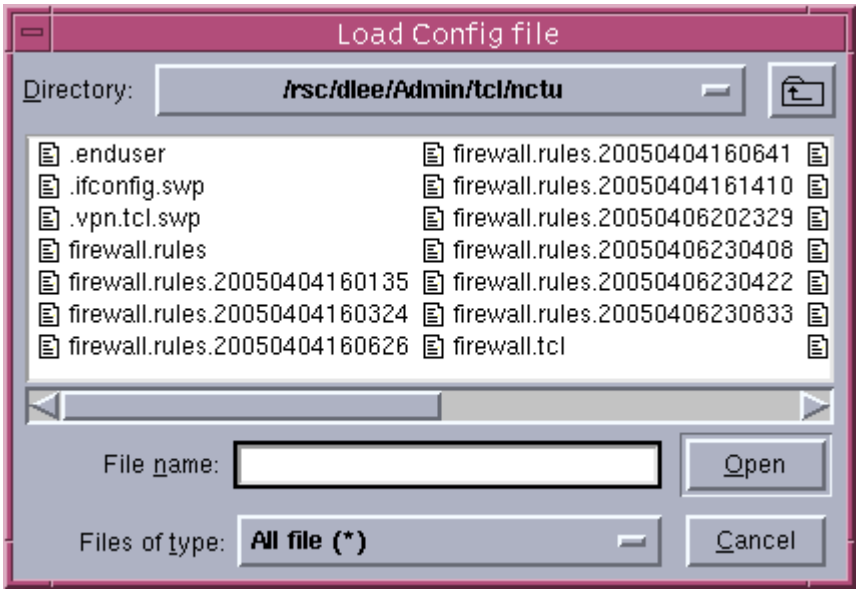

圖 25 Firewall - Load config file 畫面

當你編輯 Firewall 設定檔時，若要確認所設定的規則是否有誤時，則可以利用“Check Rules”的選項先 check 是否有不符合 IP Filter 的設定。若有問題的話，則會在畫面中顯示錯誤，否則便會將所有檢查過的規則一一列出，而設定檔中的說明部份則不會列出。其畫面如圖 26 所示。

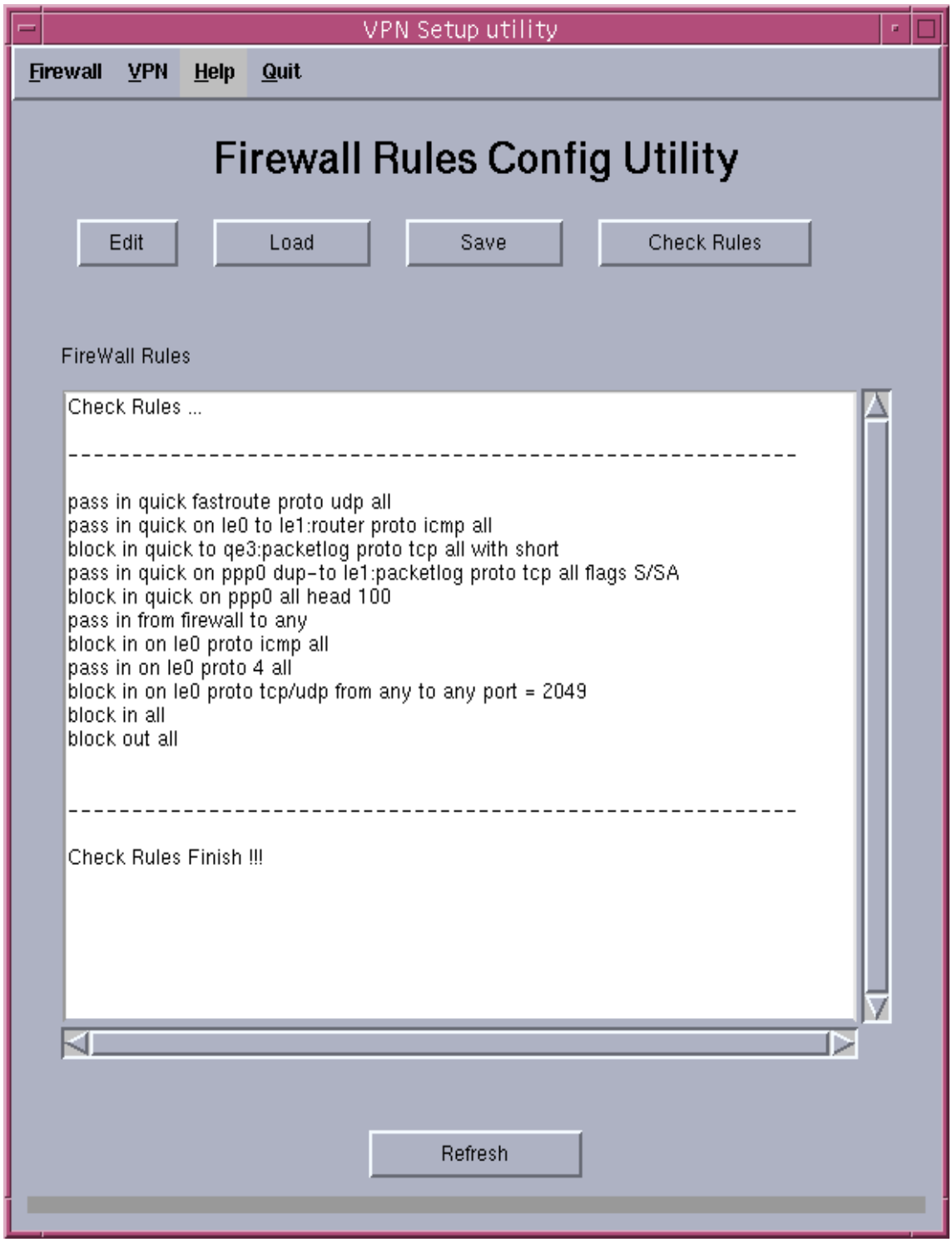

圖 26 Firewall – 檢查 firewall rules 介面

而 Help 選項則是將此系統各個欄位簡略地提示使用者知道，因其內容已於前面介紹過了，在此不再重複。而 Quit 選項則是讓使用者點選離開系統，但是為了避免使用者是誤選到該功能，所以會出現新的視窗再次與使用者確認動作，如圖 27，若確認後，方會真正離開系統。

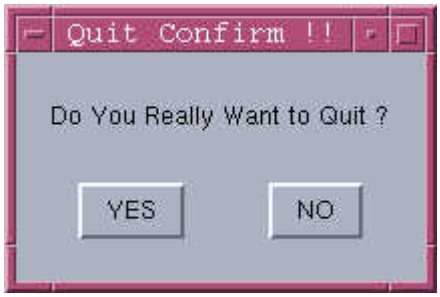

圖 27 系統 Quit 確認視窗

第4章測試驗證

4.1 驗證環境的建構

為了驗證本論文中所提之方法確實可以增加資料傳遞時的安全性，故擬建構一個模擬之測試環境，並利用封包解讀之工具的輔助，來了解設定 VPN 通道前後對於所傳遞之資料封包所造成的差異性。

硬體需求：

1. 二台測試主機，以便安裝測試軟體，進行測試。

2. 本文以 Solaris 為測試的作業系統，故機器的硬體以 Sun Workstation 為測試的機種。

3. 一台監控的主機，以監控網路上所傳遞的封包內容。本論文擬利用 Solaris 的內建工具 snoop 為分析資料封包的工具軟體，故監控機器仍以 Sun Workstation 為監控的機種。

4. 一台網路設備，以便模擬建構出一個簡單的網路環境，本論文以集線器（Hub）為測試設備。

軟體需求：

1. 作業系統，本論文以昇陽電腦之 Solaris 8 為測試機器上所安裝的作業系統，因為昇陽電腦從 Solaris 8 之後的作業系統可以支援 IPv6 的網路功能，符合本論文的需求。

2. 編譯器（Compiler），本論文同樣以網路上免費的 GNU C 編譯程式（簡稱 GCC）為本論文所使用的編譯器，以符合本論文希望使用開放原始碼軟體來完成達到增加系統安全的目的。

3. 監控軟體，市場上有許多的軟體可以達到此功能，但是同樣希望能以開放原始碼軟體為主要考量，昇陽電腦的 Solaris 作業系統中所提供的 snoop 程式便可以用來進行網路封包的監控與分析，所以便不另外安裝其他軟體。

所以測試環境的示意圖如下（圖 28）：

通道二端的二台主機分別為主機 A 及主機 B，其設定如下：

VPN 主機 A（vpn1）：

IPv4 的 IP Address （192.168. 0.2）

IPv6 的 IP Address （3ffe:400:350:2db3:a00:20ff:fee6:3196/64）

VPN 主機 B：

IPv4 的 IP Address （192.168. 0.1）

IPv6 的 IP Address （4ffe:400:350:2db3:a00:20ff:fee6:3195/64）

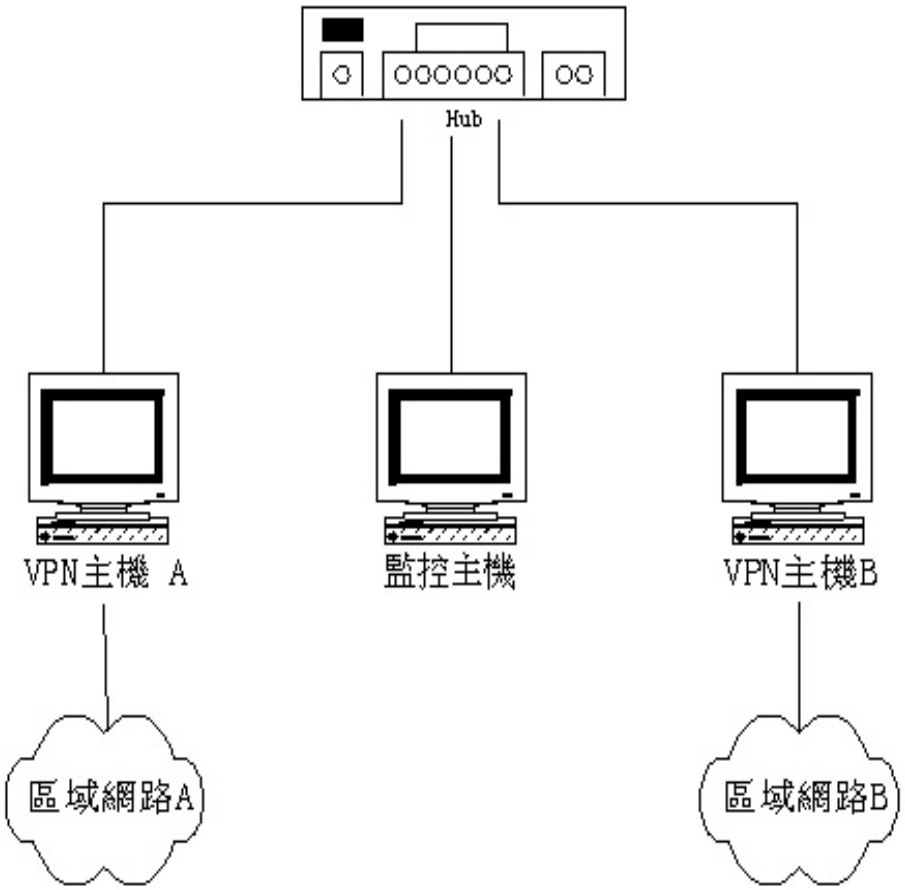

圖 28 測試環境示意圖

將 vpn1 及 vpn2 二台主機的網路介面透過 UI 的操作介面設定完成後，一併將 vpn1 及 vpn2 之間的通道也設定完成，並透過事先設定好的 IPSec 機制，將在 vpn1 及 vpn2 二端的封包都經過 IPSec 的安全機制。

4.2 IPSec 機制的確認

要在Solaris環境中確認是否有支援ESP的加密演算法十分地容易，只要以root的身份執行”ndd /dev/ipsecesp ipsecesp_status”指令即可，若得到的status為 1 的話，代表沒有支援ESP，若status為 3 的話，則表示該系統已可支援ESP。Solaris作業系統在安裝時，並不會主動安裝SUNWcry 及SUNWcryrx這二個軟體元件，所以必須要另外安裝後重新開機即可，這二個軟體可以從http://wwws.sun.com/software/solaris/encryption/download.html這個網址下載後安裝即可。

[root@vpn1]ndd /dev/ipsecesp ipsecesp_status
ESP status

Authentication algorithms = 2
Encryption Algorithms = 1 <------- 沒有支援 ESP
Packets passing authentication = 0
Packets failing authentication = 0
Packets apparently decrypting badly = 0
Packets failing replay checks = 0
Packets failing early replay checks = 0
Failed inbound SA lookups = 0
Inbound PF_KEY messages = 0
Inbound ESP packets = 0
Outbound ESP requests = 0
PF_KEY ACQUIRE messages = 0
Expired associations (# of bytes) = 0
Discarded inbound packets = 0
Discarded outbound packets = 0
[root@vpn1]
[root@vpn2 /]
ndd /dev/ipsecesp ipsecesp_status
ESP status

Authentication algorithms = 2
Encryption Algorithms = 3 <------- 有支援 ESP
Packets passing authentication = 186
Packets failing authentication = 0
Packets apparently decrypting badly = 0
Packets failing replay checks = 0
Packets failing early replay checks = 0
Failed inbound SA lookups = 0
Inbound PF_KEY messages = 15
Inbound ESP packets = 186
Outbound ESP requests = 151
PF_KEY ACQUIRE messages = 2
Expired associations (# of bytes) = 0
Discarded inbound packets = 0
Discarded outbound packets = 0

4.3網路監控工具的應用

本論文所使用的網路監控工具為使用 Sun Solaris 作業系統本身所附的工具程式 snoop 為主，並搭配 sniffer 的軟體來輔助，以截取機器在網路上所傳遞的所有封包內容，以驗證論文中的想法。

作業系統 Solaris 所提供的 snoop 工具程式可以將所有網路上的封包截取下來分析，snoop 的語法如下：

1. **snoop <host1> <host2>**

   截取 host1 與 host2 之間傳遞的封包。

2. **snoop -o filename <machine1> <machine2>**

   截取 host1 與 host2 之間的封包並將結果存在 filename 的檔案中。

3. **snoop -i filename**

   看先前截取並存放在 filename 中的封包內容。

本論文的測試方法便是利用 snoop 此作業系統程式，將所有與通道二端的主機相關的封包截取下來先存放在一暫存檔後，再同樣以 snoop 程式來一一檢視所截取的封包內容。

4.4 實際封包(packet)內容的驗證

將系統設定成之後 ， 實施傳遞的封包內容 dump 出來的結果如下 ：

```
ETHER:   ----- Ether Header -----
ETHER:
ETHER:   Packet 1 arrived at 10:46:54.90
ETHER:   Packet size = 138 bytes
ETHER:   Destination = 8:0:20:ee:c4:e3， Sun
ETHER:   Source      = 8:0:20:86:af:11， Sun
ETHER:   Ethertype = 0800 (IP)
ETHER:
IP:      ----- IP Header -----
IP:
IP:      Version = 4
IP:      Header length = 20 bytes
IP:      Type of service = 0x00
IP:              xxx. .... = 0 (precedence)
```

IP: ...0 .... = normal delay
IP: .... 0... = normal throughput
IP: .... .0.. = normal reliability
IP: Total length = 124 bytes
IP: Identification = 56442
IP: Flags = 0x4
IP: .1.. .... = do not fragment
IP: ..0. .... = last fragment
IP: Fragment offset = 0 bytes
IP: Time to live = 60 seconds/hops
IP: Protocol = 41 (IPv6) <-------此封包中使用到 IPv6 的通訊協定
IP: Header checksum = e08a
IP: Source address = 192.168.0.2， 192.168.0.2
IP: Destination address = 192.168.0.1， 192.168.0.1
IP: No options
IP:
IPv6: ----- IPv6 Header -----
IPv6:
IPv6: Version = 6 <-------此封包中使用到 IPv6 的通訊協定
IPv6: Traffic Class = 0
IPv6: Flow label = 0x0
IPv6: Payload length = 64
IPv6: Next Header = 58 (ICMPv6)
IPv6: Hop Limit = 60
IPv6: Source address = 4ffe:400:350:2db3:a00:20ff:fee6:3196
IPv6: Destination address = 3ffe:400:350:2db3:a00:20ff:fee6:3195
IPv6:
ICMPv6: ----- ICMPv6 Header -----
ICMPv6:
ICMPv6: Type = 128 (Echo request)
ICMPv6: Code = 0 (ID: 460 Sequence number: 0)
ICMPv6: Checksum = 5cd

ETHER: ----- Ether Header -----
ETHER:
ETHER: Packet 2 arrived at 10:46:54.90
ETHER: Packet size = 138 bytes
ETHER: Destination = 8:0:20:86:af:11， Sun

```
ETHER:  Source          = 8:0:20:ee:c4:e3， Sun
ETHER:  Ethertype = 0800 (IP)
ETHER:
IP:     ----- IP Header -----
IP:
IP:     Version = 4
IP:     Header length = 20 bytes
IP:     Type of service = 0x00
IP:             xxx. .... = 0 (precedence)
IP:             ...0 .... = normal delay
IP:             .... 0... = normal throughput
IP:             .... .0.. = normal reliability
IP:     Total length = 124 bytes
IP:     Identification = 42774
IP:     Flags = 0x4
IP:             .1.. .... = do not fragment
IP:             ..0. .... = last fragment
IP:     Fragment offset = 0 bytes
IP:     Time to live = 60 seconds/hops
IP:     Protocol = 41 (IPv6)
IP:     Header checksum = 15ef
IP:     Source address = 192.168.0.1， 192.168.0.1
IP:     Destination address = 192.168.0.2， 192.168.0.2
IP:     No options
IP:
IPv6:     ----- IPv6 Header -----
IPv6:
IPv6:     Version = 6
IPv6:     Traffic Class = 0
IPv6:     Flow label = 0x0
IPv6:     Payload length = 64
IPv6:     Next Header = 58 (ICMPv6)
IPv6:     Hop Limit = 255
IPv6:     Source address = 3ffe:400:350:2db3:a00:20ff:fee6:3195
IPv6:     Destination address = 4ffe:400:350:2db3:a00:20ff:fee6:3196
IPv6:
ICMPv6:   ----- ICMPv6 Header -----
ICMPv6:
```

ICMPv6: Type = 129 (Echo reply)
ICMPv6: Code = 0 (ID: 460 Sequence number: 0)
ICMPv6: Checksum = 4cd

第5章結論

5.1 研究成果

隨著電腦硬體設備的不斷進步以及網路的興起，資訊系統的面貌也變得越來越不同，所能達到的功能也越來越複雜，也因此，資料傳送的安全性也越來越被人們所重視，但是利用網路來傳送資料，雖然可以節省許多的成本，但也因此產生了許多安全性方面的考量。而隨著網路世界的意氣風發，原有的定址方法已逐漸顯現出其不足的現象，故本論文利用網路上的 OpenSource 的軟體，配合 Solaris 作業系統提供新一代的定址方式來架構私人的虛擬私有網路，可以解決現有定址方法的位址不足之窘境，並可以獲得 IPv6 的優點：
本研究具體的成果有：

1. 利用 IPv6 新一代的定址方式來彌補 IPv4 定址方式的若干缺點。

2. 利用 Solaris 的 IPv4/IPv6 的共存模式，可以在不必更換現有多數的 IPv4 網路設備的情形下架設私人的虛擬私有網路，意即不必有額外的成本增加。

3. 建立私人的虛擬私有網路，並將資料其透過 tunneling 的方式傳送，使得所傳送的網路封包不易為人所竊取或是修改，可以達到資料保全之目的。

4. VPN 和以其為基礎所發展的系統，皆已經成為許多業界所追求的解決方案。這樣的話，對於與業界合作時，可以很容易地與實務結合在一起。

5. VPN 的技術，已有許多良好的支援，因此在實作上並不會十分困難，而系統以 UNIX 為基礎，更能利用 UNIX 的優勢發展這樣的技術，尤其現今多數的大型主機多是 UNIX 的作業環境，更使得此系統很容易移植到大型主機上運作。

6. 本論文所採用的軟體多為網路上的 OpenSource 軟體，取得容易且不需要增加成本。

7. 透過本論文友善且簡單的介面設定私人的虛擬私有網路，可以讓企業設定私人的虛擬私有網路時，更有彈性，也因此更增加其安全性。

從作業的資訊安全角度來看，網路的安全性一直為許多人所懷疑，隨著電腦科技的不斷精進，系統遭非法入侵的手法也不斷翻新。企業如何保障自己的網路資訊安全並利用 Internet 此便利且便宜的媒介來完成各分公司或合作夥伴之間的機密資料交換，是一個滿有趣的課題。如果經過省慎的規劃，相信對企業的發展將有很大的幫助。

5.2 未來研究方向

本論文研究尚有許多可以較深入的地方：

1. 本研究目前僅只限於 Sun workstation 的 Solaris 作業系統，但是論文中所使用的許多元素皆可適用於其它的作業平台，例如近年來持續發燒的 Linux 平台，所以未來可繼續更廣泛地研究此安全的企業內部網路，使其能適用於其他的作業平台。

2. 本研究中的安全性考量尚有可加強的地方，例如資料封包在網路傳遞時，仍有可能會為人所攔截竊取，因此可以在資料封包在網路傳遞時，利用網路封包中的一些欄位，加強整體的安全性[11]。

3. 本研究中所使用的系統可以再加強為將所有通道中傳遞之資料封包所使用來溝通的 port 都導入到非一般眾所周知的 port，如此更可以增加資料傳遞時的安全性。

4. 本研究之系統目前啟動方式必須二個端點事先達成設定共識 ， 並確定連線設定的許多細節 ， 但這些動作可逐步以自動化的方式來達到更安全的目的。

參考文獻

1 黃景彰,「網上駭客的攻擊與反制」,資訊安全通訊第七卷第一期,p65 – p.77,2000。

2 馮立琪、黃盈源,「軟體包裹技術在系統安全上應用之探討」,資訊安全通訊第七卷第二期,p12 – p.21,March 2001。

3 PETE LOSHIN著,1999,袁文宗、侯鏞譯,IPv6 Clearly Explained網際網路的未來,旗標,2000。

4 R.Hinden,"IP Next Generation Overview" , Communications of the ACM,Vol.39,No.6,June 1996。

5 趙德卉,「IPv6 現況調查及轉換過程中 DNS 問題之研究」,國立交通大學資訊管理研究所碩士論文,1997。

6 http://www.nts.com/library/index.html。

7 Charlie Scott、Paul Wolfe & Mike Erwin,Virtual Private Networks,O'Reilly,1999。

8 柯岳明,「VPN的安檢員-IPSec & IKE」,網路通訊,p72-p78,May 2002 。

9 http://tw.sun.com/press/news/Solaris_8Operating.htm。

10 http://tw.sun.com/press/news/2000-index.html。

11 高笙庭,「利用 IPSEC 達成封包傳輸隱匿之目的」,台大資工所碩士論文, 2000。

13 BUCK GRAHAM,TCP/IP ADDRESSING : DESIGNING AND OPTIMIZING YOUR IP ADDRESSING SCHEME 2nd Edition,Academic Press,2001。

14 Pete Loshin,TCP/IP Clearly Explained 3rd edition,Morgan Kaufmann,1999。

15 陳柏飛,「以防火牆為基礎之虛擬私有網路的身份驗證系統設計與實作」,國立交通大學資訊管理研究所碩士論文,1998。

16 Peter H. Salus,Big Book of IPv6 Addressing RFCs, Morgan Kaufmann,2000。

17 Clark、Chapin、Cerf、Braden＆Hobby,"Towards the Future Internet Architecture",RFC1287,December 1991。

18 Dixon,"Comparison of Proposals for Next Version of IP",RFC1454,May 1993。

19 Hinden R.＆S. Deering,"IP Version 6 Addressing Architecture",RFC2373",July 1998。

20 Hinden R.、M. O'Dell、S. Deering,"An IPv6 Aggregatable Global Unicast Address Format",RFC2374",July 1998。

21 R.Gilligan、E.Nordmark,"Transition Mechanism for IPv6 Hosts and Routers",RFC1933,April 1996。

22 萩野純一郎博士著,2002,賴虹燕譯,IPv6 網路程式設計,博碩文化,2004。

23 張晃崚,"SSL VPN的選擇與建置",http://www.ringline.com.tw/epaper/forum93102.htm。

24 樂家福,"漫談IPSec（The Internet Protocol Security Standard）及SSL（Secure Sockets Layer） VPN(上)",http://dbmaker.syscom.com.tw/mag/86/coverstory_02.htm。

25 樂家福,"漫談IPSec（The Internet Protocol Security Standard）及SSL（Secure Sockets Layer） VPN(下)",http://dbmaker.syscom.com.tw/mag/87/seeing_01.htm。