行政院國家科學委員會補助專題研究計畫執行進度報告

※※※※※※※※※※※※※※※※※※※※※※※※※
※　　　　　　　　　　　　　　　　　　　　　　　※
※下一代虛擬私有網路核心技術之研究(1/2)—子計畫三※
※　以無線網路提供下一代 VPN 服務關鍵技術之研究 ※
※　　　　　　　　　　　　　　　　　　　　　　　※
※※※※※※※※※※※※※※※※※※※※※※※※※

計畫類別：□個別型計畫　　■整合型計畫
計畫編號：NSC　90—2213—E—002—076
執行期間：90 年 8 月 1 日至 92 年 7 月 31 日

計畫主持人：國立台灣大學電信工程學研究所 蔡志宏教授

本執行進度報告包括以下應繳交之附件：
■九十一年度預核清單影本一份
□變更經費對照表一份(欲變更經費者才檢附)
□欲參加九十一年度傑出研究獎遴選者應備資料

執行單位：國立台灣大學電信工程學研究所

中　華　民　國　91　年　5　月　24　日

行政院國家科學委員會專題研究計畫執行進度報告

下一代虛擬私有網路核心技術之研究(1/2)—子計畫三：以無線網路提供下一代 VPN 服務關鍵技術之研究

A Study on The Key Technologies for Providing Next Generation VPN Service in Wireless Networks

計畫編號：NSC 90-2213-E-002-076

執行期限：90 年 8 月 1 日至 92 年 7 月 31 日

計畫主持人：國立台灣大學電信工程學研究所 蔡志宏教授

一、中英文摘要

本計畫擬於兩年時間，研究開發以無線寬頻網路提供下一代虛擬私有網路之關鍵技術，主要研發之關鍵技術包括兩大部分：

1) 建置無線 VPN 於固定無線接取鏈路或無線區域網路時，用以支援多個對服務品質敏感之即時多媒體交通流及一般 TCP/IP 交通流之封包分類器及排程器。

2) 無線接取節點以及無線終端設備之自動設定機制，IP VPN 閘道器功能，以及 VPN 通道協定。

這些關鍵技術之目標在於使同一個無線接取網路中同時支援多個VPN服務成為可能，而且各 VPN 可以擁有不同服務品質、計價及安全需求，並有不同之使用群。此外，無線終端設備之自動 VPN 設定功能將使無線VPN之使用者可以自由無縫地連接所擬連接之 VPN。

本計畫第一年已逐步完成上述系統所需關鍵技術之測試。

關鍵詞：虛擬私有網路、無線、服務品質

Abstract

This project aims to develop key technology within 2 years, for providing the next generation Virtual Private Network (VPN) services in the broadband wireless network environments. The targeted technologies include 2 major parts: 1) packet scheduler for supporting multiple QoS - sensitive real-time multimedia streams and regular TCP/IP streams over fixed wireless links or LANs, and 2) auto-configuration mechanism, IP VPN gateway functions and VPN tunneling protocol for the wireless access node and the wireless terminals. The goals of these key technologies are to provide multiple VPN services simultaneously in the same wireless access network, while the supported VPNs are with different QoS, security requirement and are for different user groups. In addition, the wireless VPN terminals shall be provided with auto-configuration mechanism, such that the wireless VPN users can access the desired VPN(s) and applications in a seamless fashion.

This year, we have completed the testing and integrating several key technology component for the prototype.

Keywords: VPN, Wireless, QoS

二、計畫緣由與目的

近年來，由於 TCP/IP 協定和區域網路在企業環境使用上的快速成長，以及符合成本效益的 IP 服務已經引起對以 IP based VPN 的強烈需求。雖然在一開始的時候，IP VPN 只是用來取代在私有廣域網路上專線的使用，而現在，由於 IP 技術的發展，

不僅可以找到多種實現 IP based VPN 的方法，而且 VPN gateway 也提供了更強大的功能，而其中的一些 tunneling protocol，像是 L2TP，IP in IP，MPLS 等等，都扮演著很重要的角色，同時 VPN 的特性已經跟以往大不相同。根據 RFC 2764，VPN 有以下幾種: leased line VPN， Virtual Private Routed Network， Virtual Private Dial Network。由於使用者有各種不同的需求，這將會導致 VPN 市場的快速發展。然而目前一些 VPN 的解決方案缺少了 QoS 保障，還有動態調整 VPN 設定的彈性。因此我們需要下一代的 VPN，而且在工業界和學術界有許多研究人員往這方面努力。

另一個在網路技術方面的重要發展趨勢是平衡有線與無線寬頻存取網路的成長。我們特別關心的是無線寬頻存取的進展。雖然頻譜有限，但由於有以下的優點: 允許使用者能夠輕易的設定終端設備(像筆記型電腦，PDA)而不用實體的線路，對 mobility 的些許或全力支援，或僅是想要更高的聯網速度，這些好處都促使無線接取市場的快速成長，而在區域網路中有 802.11a，802.11b,在短距離有藍芽和 Home RF 的技術發展。

然而，在電信大廠對於 3G 行動服務的引進以及未來 4G 無線科技的研究，我們相信目前頻寬(2Mbps~11Mbps)的限制可以被輕易的突破。最後在有線網路上所提供的服務幾乎都可以透過無線寬頻介面來享用，即使像影音串流這種需要 QoS 保障與大頻寬的服務，都可以在這種環境實現。

目前有明確的證據顯示 VPN 和無線網路是未來的兩大關鍵技術，而它們有很大的發展空間，以及市場與使用的廣大需求。因此我們可以推斷，遲早,在相同的無線寬頻網路上提供具彈性與多重 VPN 服務，不只可行，而且 VPN 與無線產品也強烈需要這種功能。

然而，在無線寬頻環境上要提供無線 VPN 服務將面臨兩大挑戰。第一，在大部分寬頻無線標準解決方案，特別是在無線區域網路，對於空中介面，MAC，及封包排程機制缺少處理 QoS 的能力。第二，若要允許多重 VPN 共存在相同的無線寬頻接取網路，並且要能夠維持使用者加入或離開任何 VPN 的彈性，那麼在網路端與使用端就需要一個具有強固性和能夠自動調整組態的協定與機制。不論無線寬頻標準或實體層技術的進展如何，我們相信在 VPN 與無線領域皆需要更深入的研究。

在本計畫我們著重的技術包含兩個部分。第一，可在固定式無線網路或區域網路支援多個對 QoS 敏感的多媒體即時串流及一般 TCP/IP 交通流的封包排程器及空中存取介面。第二,對於在無線存取節點及無線終端使用者之間的自動調整組態機制，IP VPN gateway 功能及 VPN tunneling 協定。這些技術所要達成的目標是在相同的無線接取網路上同時的提供多重 VPN 服務，但是不同的 VPN 服務所需的 QoS 都相異，而且對不同的使用族群也需不同的安全等級。此外，無線 VPN 終端設備必須要能夠提供自動組態設定機制使得無線 VPN 使用者能以無縫式的接取意欲的 VPN 與應用。

三、研究方法、結果與討論

3.1 VPN 的架設與選定

3.1-1.使用 FreeS/WAN 實作 VPN

本計畫初步以 FreeS/WAN 軟體實作出一組 VPN 閘道器，使位於不同所在地的子網路能夠相通。實驗硬體設備如表一：

表一 VPN 閘道器硬體設備

主機代號	B	F
CPU	P3 733	P3 933
記憶體	128M SDRAM	256M SDRAM
主機板	ASUS CUBX	ASUS TUSL2-C
硬碟機	Seagate ST330631A	Western Digital WD300BB-00AUA1
網路卡	3Com 905C×2 3Com 905B	Intel Pro/100 S Desktop×2

實驗環境中作業系統為 Linux 2.4.17，軟體為 FreeS/WAN 1.91。

3.1-2.關於 FreeS/WAN

FreeS/WAN 是一個在 Linux 上實作 IPSEC 的軟體，採用 3DES 加密及 MD5 數位簽章認證。IPSEC 為 IETF 所提出的應用於 IP 網路的安全架構，由多個協定組成，主要提供在 IP 層的認證及加密功能。

IPSEC 主要有三個部分：1)ESP(Encapsulating Security Payload)提供資料加密及認證。2)AH(Authentication Header)提供封包認證。3)IKE(Internet Key Exchange)替 AH 及 ESP 交換所需參數。

而 FreeS/WAN 中分別有程式實作上述協定，包括 KLIPS(**K**ern**el IP S**ecurity)，經修改 Linux 核心，可達到實作 AH、ESP 協定和封包處理的目的。在 Pluto(IKE daemon)部份，我們實作 IKE，完成系統建立連線。

3.1-3.實驗設定與結果

將兩台 VPN 閘道器設定為 ESP 及 AH 的 tunnel mode，表示由子網路電腦所送出的封包將會在閘道器被包上一層新的 IP header，到了另一端的閘道器時整個封包都會被解密及認證，如果通過封包的認證，就把解密後的封包依 destination IP 位址查 routing table 加以傳送。

我們將兩端(R248 及 R554)的子網路皆設為 private IP segment 並將 FreeS/WAN 設定及完成 public key 交換之後，進行兩個子網路的對測，發現所有網路應用(WWW、FTP、telnet 等)皆可正常運作，對於兩端子網路中的使用者而言並不需要進行額外的軟體安裝與設定，達成架設 VPN 的目標。

3.2 Linux 路由器的效能驗證

3.2-1.驗證目標

在本研究中，我們一共設計了三個實驗。實驗一是針對 Linux 路由器的 IP 封包轉送能力做量測，藉此估計以 PC 為架構的路由器效能上限。實驗二是針對目前市面上常用的商用網路協定分析模擬軟體 Chariot，我們將 Chariot 所量到的數據與實際從網路上所量得的結果加以比較，而這項結果就可以在日後使用 Chariot 時作為誤差校正的用途。實驗三則是使用 Chariot 來驗證 Linux 路由器啟動 CBQ(Class-Based Queuing)時的效能。

3.2-2.實驗方法、結果與討論

軟體包括 Chariot v4.2，SmartWindow v6.53.18，以及 SmartApplication v2.32。電腦硬體設備如表二：

表二 實驗用電腦的設備

主機代號	A	E
筆記型電腦型號	Acer TravelMate 613TXV	IBM ThinkPad A22e

主機代號	C	D
CPU	P3 1000×2	P3 450
RAM	256M SDRAM	128M SDRAM
主機板	ASUS CUV4X-D	ASUS P2B-F
硬碟機	Seagate ST330620A	IBM DTLA-307030
網路卡	Intel Pro/100 S Server×1 Intel Pro/100 S Desktop×2	3Com 905C×2
作業系統	MS Windows 2000 Professional	

實驗一是直接使用 SmartBits 200 產生 UDP 交通流灌入 Linux 路由器(主機 B)，封包經過路由器會轉送到 SmartBits 的另一個介面。藉由量測 SmartBits 送出與收到的封包數量差，可以得到 Linux 路由器的封包遺失率。另一方面，可以灌大量封包到路由器，使它達到穩態後再量測此時某個封包的單向延遲時間；這裡封包單向延遲時間指的是封包在路由器中 store and forward 所消耗的總時間，並沒有計入封包本身的傳輸時間。

SmartBits 在實驗中送出的交通流是 CBR，我們調整交通流的兩個參數，封包大小與乙太網路的線路使用率。封包大小(frame size)的範圍是 64~1518bytes，線路使用率指的是除了 MAC 層的 frame 以外，再加上 PHY 層的 overhead (含 preamble 與 minimum inter-frame gap，共 20bytes)，全部佔用實體線路的比例。線路使用率與上

層處理時的資料率並不相同，因此我們必須先做轉換再解讀量測結果才有意義。

量測的結果列在表三。由結果很明顯可看出在封包很小的情況下，Linux 路由器很快就抓襟見肘，大量遺失封包，但是對中等大小(或以上)的封包，就能維持超過九成的線路使用率。

表三 實驗一的結果

封包大小 (bytes)	線路使用率	封包產生速率(fps)	封包遺失率	封包通過速率(fps)
64	40%	59524	0.00%	59524
	50%	74405	13.22%	64565
	60%	89286	52.53%	42380
	70%	104167	72.00%	29162
	80%	119048	76.92%	27482
	90%	134409	87.03%	17432
	100%	148810	84.10%	23655
128	70%	59242	0.00%	59242
	80%	67568	12.14%	59366
	90%	75988	32.17%	51541
	100%	84459	54.15%	38723
256	90%	40783	0.00%	40783
	100%	45290	0.34%	45137
512	90%	21151	0.00%	21151
	100%	23496	0.14%	23463
1024	90%	10776	0.00%	10776
	100%	11973	0.01%	11972
1518	100%	8127	0.00%	8127

封包大小 (bytes)	線路使用率	封包產生速率 (fps)	封包單向延遲 (μs)
64	30%	44643	17.1
	40%	59524	30.6
128	50%	42230	18.3
	60%	50710	30.7
	70%	59242	29.4
256	80%	36232	19.1
	90%	40783	21.6
	100%	45290	67.5
512	80%	18797	22.4
	90%	21151	22.4
	100%	23496	42.6
1024	90%	10776	28.8
	100%	11973	43.4
1518	90%	7314	38.2
	100%	8127	54.1

在實驗一中，我們觀察到封包的單向延遲時間大約在 17~38 微秒的範圍內，由此可以推斷 Linux 路由器處理每個封包要花二、三十微秒的時間，大封包需要做的 I/O 量較大，因此處理時間會較長。這些處理時間事實上也是為效能的瓶頸。

以 20 微秒的處理時間為例，同樣長度的時間在 100Mbps 的乙太網路上可傳送 250bytes 的資料；換句話說，如果交通流的封包大小超過 250bytes，即使下一個封包是 back-to-back 緊跟著正在處理的封包後面傳過來，在它完全送過來之前，就能處理掉現在這個封包。反之，若是封包來得比處理的速度還快，那很快就會有封包遺失。

我們粗略的以 200~300bytes 做為大封包與小封包的分界點，只有在小封包不斷湧進時，才會看到可觀的封包遺失率。這種情況在以 CPU 為核心的 routing 架構較易發生。

實驗二的架構如圖五所示，在圖的下半部(含主機 A、C、D 及 switch)主要是用來傳送 Chariot 的設定訊息以及回傳的報告訊息，而上半部(含主機 C、D 及 hub)則是用來傳送 Chariot 所產生出的模擬資料交通流，如此設計的目的是要隔離真正要測試的模擬交通流與 Chariot 本身所產生的其他交通流。而 hub 與 SmartBits 200 是用來在不影響測試資料流的傳送下，量出實際在網路中資料流動的參數。然後我們以 SmartBits 測得的數值當作基準，拿來與 Chariot 報告的數值加以比較，求得 Chariot 結果與實際交通流之間的誤差。

我們使用 Chariot 中內建的 IPTVv.scr 檔來模擬產生一個 UDP 交通流，由主機 C 的 x.y.1.2 經由 hub 送到主機 D 的 x.y.1.1。

有關 Chariot 中的參數設定：

- file_size = 146000bytes
- send_buffer_size = 1460bytes

並根據我們的目的，對 send_data_rate 設計了下列三種情況：

- send_data_rate = UNLIMITED
- send_data_rate = 7Mbps
- send_data_rate = 8Mbps

其中 file_size 參數主要是隨著所使用

的 send_data_rate 來做動態的調整，以便使得 Chariot 取樣到的點數夠多、取樣頻率也能適中，而 send_buffer_size 參數是用來控制送出封包的大小。

實驗二所送出的 layer 2 封包格式如圖六 packet style 1 所示。但是 Chariot 所設定之參數及回傳的報告數據皆以 layer 7 所見為準，所以我們在與 SmartBits 比較時都會先轉換成在 layer 2 所見的數值。實驗二的結果在表四。就 Chariot 所顯示的結果(需轉換到 layer 2 來看)與用 SmartBits 量測到的實際結果來說，誤差值約在±1%。

表四 實驗二結果

項次	(a)	(b)	(c)
Chariot 設定的交通流速度 (Mbps)	∞	8	7
Chariot 顯示的 throughput (Mbps)	9.573	7.867	6.826
Chariot 的結果轉換至 layer 2 (Mbps)	9.934	8.189	7.087
SmartBits 200 量得的 throughput (Mbps)	9.857	8.221	7.102

在實驗二中，情況(a)在 send_data_rate 是 UNLIMITED 的情況下會得到約 10Mbps 的結果，主要是由於 hub 所造成的，因為實驗所使用的 hub 只支援 10Mbps 的速度。

造成約 1%的誤差的主要來源是因為 Chariot 每次所回傳量測值的精確度只能到 1ms，而 SmartBits 則可以達到 0.1μs。

故以後在使用 Chariot 當作量測工具時，把所得的量測數值轉換到 layer 2 後，再考慮約 1%的誤差，即為真實網路中的交通流參數。

在實驗三中，我們使用的電腦與實驗二類似，主要的不同點是在於使用 Linux 路由器(主機 B)，並且開啟它的 CBQ 功能。

我們使用 Chariot 中內建的 IPTVv.scr 檔來模擬產生兩個 UDP 交通流，這兩個交通流皆由主機 C 的 x.y.3.1 流到主機 D 的 x.y.1.1；再使用內建的 Throughput.scr 檔來模擬產生一個 TCP 交通流，而它是由主機 C 的 x.y.4.1 流到主機 D 的 x.y.1.1。

CBQ(Class-Based Queuing)是目前常見的排程器之一。它的作法是將交通分成數種不同的 class，每個 class 有自己的 queue 與設定的頻寬。一般在頻寬充裕的情況下每個 class 可以按照比例分配可用的頻寬，一旦發現有 class 因為頻寬不足而無法完全使用設給它的頻寬時，則會開始限制某些超過設定頻寬的 class，讓它不再超過。CBQ 特別的地方在於 class 的分類是有階層性的，當 parent class 有多餘的頻寬時，能讓它的 children 優先享用，而不會均分給所有的 class。不過在實驗三裡我們並不打算測試頻寬分享的功能，而是著重在 CBQ 估算各 class 使用的頻寬與限制頻寬的能力上。

由於實驗三是要驗證 Linux 路由器的 CBQ 性能，所以 Chariot 在前一個實驗中所設的參數只有 send_data_rate 比較重要，我們必須把 send_data_rate 調到比 CBQ 設定的 allocated bandwidth 大，才能判斷出 CBQ 的效能如何。此外，我們也探討在有小封包出現時，對 CBQ 的效能有何影響。實驗的結果見表五。

表五 實驗三的結果

Allocated bandwidth (Mbps)			Chariot 量得的 throughput (Mbps)		
UDP1	UDP2	TCP	UDP1	UDP2	TCP
0.1	0.1	1	0.103	0.103	1.041
0.1	0.1	1	0.073	0.103	1.041
0.5	0.5	5	0.545	0.545	5.413
0.5	0.5	5	0.384	0.544	5.439
1	1	10	1.183	1.182	11.571
1	1	10	0.830	1.176	11.526
2	2	20	2.480	2.485	21.153
2	2	20	1.282	2.492	22.047
3	3	30	3.857	3.864	30.652
3	3	30	1.622	3.812	30.602
4	4	40	4.548	4.554	48.374
4	4	40	1.808	4.404	46.733

Packet Style	UDP1	UDP2	TCP
	1	1	3
	2	1	3

在實驗三中，我們可以看到 Linux 的 CBQ 效能，在處理小封包交通流時的表現遠遜於處理大封包交通流，這可能是因為

Linux 路由器的 CPU 在某段時間內相對於以大封包傳送的交通流接收到太多的中斷訊號，而來不及處理所造成，所以導致小封包的遺失率很高，因而使得 throughput 下降。

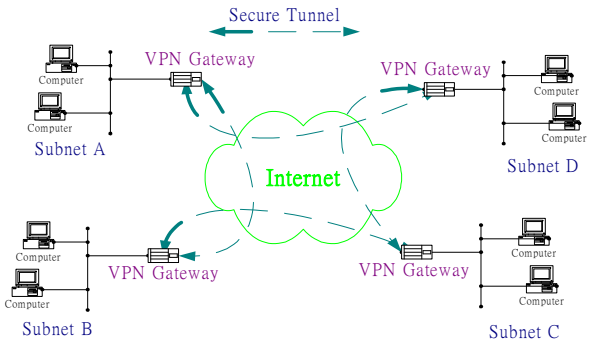
而且就所得的數據來看，最壞的情況可以跟 allocated bandwidth 有 28%的誤差，最好的情況也有 3%的誤差，所以 CBQ 的效果似乎不是那麼地準確，只能提供一個大概的機制來限定頻寬而已。

四、參考文獻

1. B. Gleeson, A. Lin, J. Heinanen, G. Armitage, A. Malis, “A Framework for IP Based Virtual Private Networks,” RFC 2764, February 2000.
2. M. Suzuki, J. Sumimoto, “A Framework for Network-based VPNs,” draft-suzuki-nbvpn-framework-02.txt , November 2000.
3. T. Sloane, R. Bach, R. Bubenik, A. Young, J.J. Yu, “Network based IP VPN Architecture using Virtual Routers,” Network Working Group draft-ouldbrahim-vpn-vr-02.txt, November 2000.
4. R. Droms, “Dynamic Host Configuration Protocol,” RFC 2131, March 1997.
5. R. Braden, D. Clark and S. Shenker, “Integrated Services in the Internet Architecture: an Overview,” RFC 1633, July 1994.
6. T. Li, Y. Rekhter, “A Provider Architecture for Differentiated Services and Traffic Engineering (PASTE),” RFC 2430, October 1998.
7. D. Black, “Differentiated Services and Tunnels,” RFC 2983, October 2000.
8. IEEE 802.11: IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, June 26, 1997.
9. D. Stiliadis and A. Varma, “Rate-proportional Servers: a Design Methodology for Fair Queueing Algorithms,” *IEEE/ACM Trans. Networking,* vol. 6, no. 2, pp. 164 –174, Apr. 1998.
10. O. Yaron and M. Sidi, “Generalized Processor Sharing Networks with Exponentially Bounded Burstiness Arrivals,” *Proc. of IEEE INFOCOM '94,* vol. 2, pp. 628 –634, 1994.
11. A.K. Parekh, R.G. Gallager, “A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: the Single-node Case,” *IEEE/ACM Transactions on Networking,* vol. 1, no. 3, pp. 344 –357, Jun. 1993.
12. F-M. Tsou, H-B. Chiou and Z. Tsai, “A Delay/Jitter Guaranteed Traffic Scheduler with Fair Bandwidth-Sharing for Internet Multimedia Services,” submitted to IEEE Trans. Multimedia.
13. S. Lu, V. Bharghavan, R. Srikant, “Fair Scheduling in Wireless Packet Networks,” *IEEE/ACM Transactions on Networking,* vol. 7, no.4, pp. 473 –489, Aug. 1999.
14. J. Gomez, A.T. Campbell, “A Channel Predictor for Wireless Packet Networks,” *2000 IEEE International Conference on Multimedia and Expo, 2000. ICME 2000*. vol. 3 , 2000 , pp. 1269 –1272.
15. A. Zahedi and K. Pahlavan, “Capacity of a Wireless LAN with Voice and Data Services,” *IEEE Trans. Commun.*, vol. 48, no. 7, pp.1160-1170, Jul. 2000. S. Lu, V. Bharghavan, and R. Srikant,

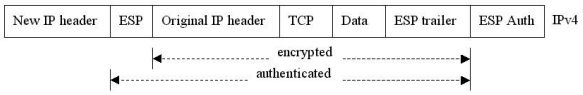
"Fair Scheduling in Wireless Packet Networks," Proceedings of ACM SIGCOMM '97, pp. 63-74, 1997.

16. S. Floyd, and V. Jacobson, "Link-Sharing and Resource Management Models for Packet Networks, IEEE/ACM Trans. Networking, vol. 3, no. 4, pp. 365-386, Aug. 1995.
17. P. Bhagwat et al., "Enhancing Throughput over Wireless LANs using Channel State Dependent packet Scheduling," Proceedings IEEE INFOCOM '96, vol. 3, pp. 1133-1140, Mar. 1996.
18. "Linux Advanced Routing and Traffic Control" (http://lartc.org/)
19. ''FreeS/WAN''(http://www.freeswan.org/)

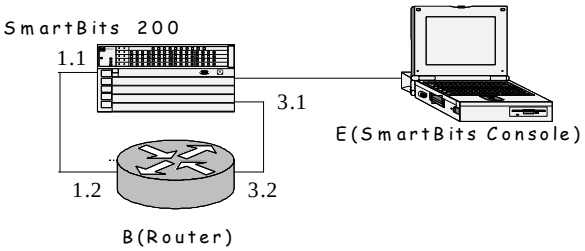


圖一 虛擬私有網路示意圖

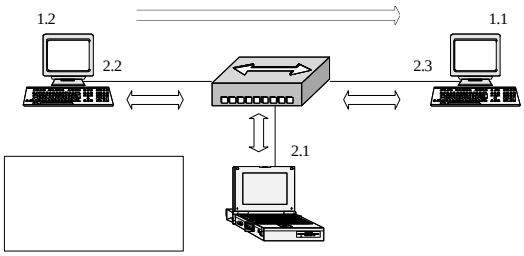
圖二 VPN 實驗架構圖



圖三 封裝後的封包格式



圖四 實驗一架構圖



圖六 實驗二、三的封包格式

圖七 實驗三架構圖