在 Linux 微型路由器上的 VPN 設計

VPN Design on Linux Micro Router

何振毅　　楊仕任　　王欣平

大葉大學資訊工程研究所　大葉大學資訊工程研究所　大葉大學資訊工程研究所

r9006002@mail.dyu.edu.tw r9006017@mail.dyu.edu.tw　swang@mail.dyu.edu.tw

摘要
ABSTRACT

本文提出一個虛擬私有網路[1]架構建置在 Linux 微型路由器(Linux Router Project; LRP)[5]上的設計。這個設計使用嵌入式系統工具和函式庫，將虛擬私有網路協議(IP Security Protocol(IPsec) 、 Crypto IP Encapsulation (CIPE))技術實作整合進 Linux 微型路由器上。透過軟體模組化的實現，這個設計具備比硬體實作更佳的彈性和延展性，同時具低成本、維護容易、擴充性佳的特性。同時本文透過網路封包截取、流量分析工具評估此架構的可行性、效能、和與防火牆整合時的影響。另外，本文也針對不同 VPN 協議在這個設計上的傳輸效能作比較。

關鍵字：虛擬私有網路、微型路由器、安全性機制。

Keywords：Virtual Private Network (VPN)、Linux Router Project (LRP)、security policy

一、簡介

隨著家庭網路和企業網路快速的發展，此時網路服務品質和網路安全性機制在家庭網路和企業網路之間就扮演著相當重要的角色。而虛擬私有網路的出現提供比傳統認證簽章方式更高的安全性和透通性。虛擬私有網路透過公眾網路模擬出私有網路，是一整合安全性機制的網路服務架構。本文提出一個虛擬私有網路架構建置在 Linux 微型路由器(Linux Router Project; LRP)[5]上的設計。本文的架構不同於 ISP 所提供 VPN 服務的昂貴和維護性不佳的缺點，它是一個便宜、容易維護的安全性網路通道架構。以下第二節會介紹本設計的架構。接下來第三節是整個設計實作過程與效能測試。緊接著第四節是測試與分析。最後第五節結論和未來工作計劃。

二、設計架構

微型路由器是一個以網路功能為導向的微型 Linux 系統，整個系統程式相當的小，足以放入 1.44mb 的軟式磁碟片。其基本硬體資源需求可以是一個小型嵌入式系統或是一個典型 PC、加上二張網路卡和一個軟式磁碟機。裝設微型路由器時，直接由微型路由器磁片開機，它可以不影響原本電腦系統前提下獨立工作。目前有許多微型路由器的設計架構，例如 gnatbox、NetBSD/i386 firewall project、floppyfw、及 LRP 等，其中大多數的架構都只提供 Linux 核心本來就有的功能，簡單的說就是只針對特定需求功能做編譯來取得一個特定功能且容量極小的核心當作整個架構的中心，而所提供的功能包括 dhcp、nat firewall、pptp 等各式服務。

虛擬私有網路核心技術在 Tunneling[2]，目前的虛擬私有網路 Tunneling 技術所使用的協定主要有幾種如表 1 所示。

表 1 VPN Tunneling 種類表

Tunneling type	OSI Layer	說明
Point to Point Tunnel Protocol(PPTP)	2	由 3Com 和微軟共同發展的遂道技術
Layer 2 Tunneling Protocol(L2TP)	2	將 PPTP 和 Layer 2 Forwarding 合併
IP Security Protocol(IPsec)	3	由 IETF 所制定的標準安全通道技術
Crypto IP Encapsulation(CIPE)	3	一個開放性原使碼計劃

其中 IPsec 和 CIPE 的技術都建構在 Layer 3，這使得使用者可以使用 Internet 的多點傳輸功能(包括 Internet、Intranet、Extranet、Remote Access 等)。相對的，PPTP 及 L2TP 的技術建構在 Layer 2，因此它們只能執行點對點傳輸功能，使用時受到相當的拘限。但是以效能來說，架構在 Layer 2 上的 PPTP 及 L2TP 會比在 Layer 3 上處理的較快。 IPsec[3]是個標準，具有高支持度；CIPE 則是一個實作 VPN Tunnel 的開放性原始碼計劃[6]，但是兩者除了使用不同加密演算法以外，以及採用不同 Tunnel 傳輸模式(TCP/UDP)，基本上具有相似的 VPN 架構。目前虛擬私有網路的路由器多半皆為特定嵌入式系統開發的軟體，而現有以微型路由器為基礎的 LRP 也具有 IPsec-VPN 的功能[2]，但採用比較舊的規格且僅限於初步支援，另外核心部份也是採用比較舊的 2.2.X 版本，對於擴充性上會有一定的影響。而本文所實作的架構和 LRP 所實作的 VPN 架構比較，不但是擴充性更佳且整體的功能也更為完整。特別是為達到多點傳輸功能，所以本設計架構同時加入 IPsec 和 CIPE 來架構 VPN。而這個架構本文就稱為微型虛擬私有網路路由器(VPN-µRouter) 。同時會在第四節部份分別測試 LRP 的 VPN 微型路由器和本文的 VPN-µRouter，針對傳輸效能上的比較做討論。

三、設計實作

如前面提到，本文將會使用兩種 VPN Tunnel 技術來實現本文的 VPN-µRouter。IPsec 部分是以最新虛擬私有網路技術套件 FreeS/Wan[7]包裝在本文開發的 VPN-µRouter 中。此 VPN-µRouter 可保障 net-to-net 的安全服務品質，提供符合最新 IETF 制定的 IPsec 標準。讓使用者透過專屬的私人通道與遠方的友人或公司進行私密的通訊，其實作架構簡圖如下圖 1,2 所示:

至於 CIPE 部份，則採用 CIPE 計劃所提供實作 CIPE 技術的 VPN 套件來實作 VPN-µRouter。實作架構和 IPsec 部份幾乎一樣，而封裝如圖 3

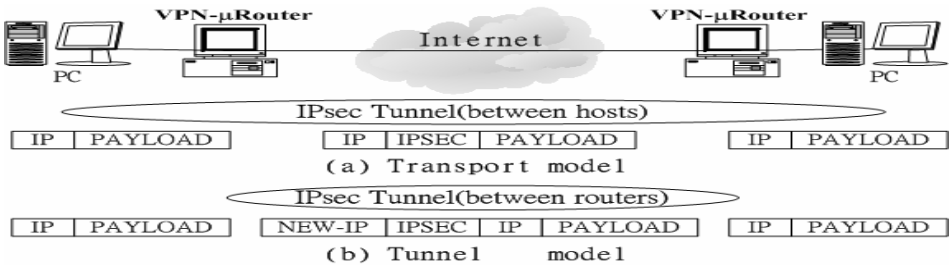

圖 1. 不同傳輸模式封裝架構簡圖

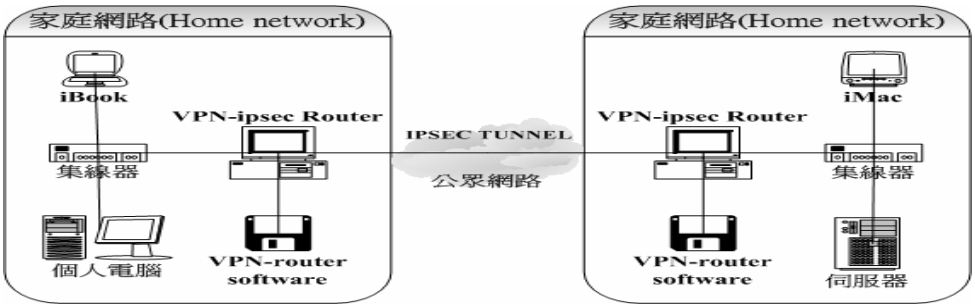

圖 2. 實作架構簡圖

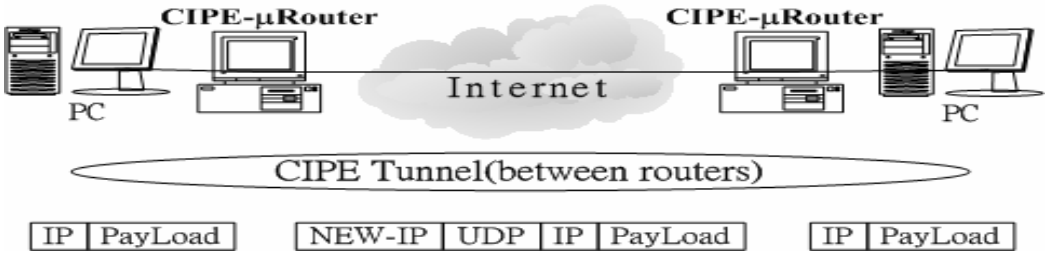

圖 3. CIPE 封裝示意圖

四、測試與分析

4.1VPN-µRouter 建置

本文架構基礎植基於 Linux 系統，因此我們一樣需要在微型路由器系統下實作虛擬的 Linux 目錄環境，然後將預先準備的程式、核心、函式庫等做成 LRP 套件寫入到 1.44mb 的軟式磁碟片中並針對硬體和網路環境進行設定。首先需要有 boot loader 程式負責啟動核心和載入 root filesystem，此處我們使用 Syslinux。Syslinux 是一個在 MS-DOS/WINDOWS FAT filesystem 下載入 Linux 操作系統的程式，它可以在微型 Linux 下用於載入分散式套件或 raw image。首先我們將一軟式磁碟片做高度度格式化並將 syslinux 載入軟式磁碟片中。接著用文字編輯器產生一個名為 syslinux.cfg 的系統開機設定檔案並編輯其相關參數設定及開機程序。

接下來我們需要建置 Linux Kernel 和 filesystem。首先取得一個新的核心原始碼並進行編譯，因受微型路由器空間限制故將原始核心碼中不必要的功能、模組都移除，將核心精簡成最佳化(建議約為 550k~750k)，在這裡我們並不對編譯核心做太多的介紹。其次是製作虛擬私有網路部份，這裡我們所採用的是 FreeS/Wan[7]，FreeS/Wan 模組實作時可以選擇編成模組或直接編入核心。產生了支持虛擬私有網路部份的核心後，尚需要一些工具程式(例如 BusyBox[8]、uClibc[9])。為減少磁碟的使用空間，我們選擇 uClibc 函式庫來編譯這些工具程式。接著我們用 uClibc 編譯一個靜態連結的 BusyBox，將 BusyBox 編譯成靜態連結的原因就是不希望過份依賴 uClibc 而增加磁碟的使用空間。BusyBox 它包含了七十多種 Linux 標準的工具程式，它所需要的磁碟空間僅僅只有幾百 Kbytes。

在經由 syslinux 啟動後，先前製作的 Linux Kernel 將會被載入記憶體中。Linux Kernel 一旦載入完畢之後，便開始初始化系統所有硬體設備，接著掛載 root filesystem 和其他自訂的 LRP 模組。此時 Linux Kernel 必須知道從哪裡可以找到 root filesystem (一般系統磁碟片的 root filesystem 會被製作成以 ramdisk 執行方式

的影像檔)。

在開始建造 root filesystem 之前必須成為 super uesr 也就是 root，其次為 root filesystem 建一個目錄叫做 floppy-linux，然後進入 floppy-linux 內為 root filesystem 建立一些相關目錄如下：

bin, etc, sbin, usr, lib, dev, var, tmp, root, proc, mnt 。各目錄功能在檔案架構系統文件中有詳述〔http://www.pathname.com/fhs/〕。最後將root filesystem 以ram disk的方式來實現，並且加入一些會用到的相關模組如 shorewall、iptable等完成一個具VPN功能的微型路由器。

本文的 CIPE 套件是使用者程式和核心無關，因此可以更方便的將它做成 LRP 模組，並將其掛載入 VPN-µRouter 中。而且這裡實作過程和 IPsec 部份的差別是在於不需要把 VPN 模組加入核心，其餘部份幾乎一樣。

到現在為止，算是把兩種不同 VPN Tunnel 技術的 VPN-µRouter 完成。剩下的就是如何讓它啟動和相關模擬測試比較。首先開始的是 IPsec 的部份，將先前制作的 VPN-µRouter 磁片放入磁碟機中並開機。此時若一切無誤將會進入文字介面模式，系統中若有裝設防火牆或其它網路服務(如:NAT)，在此均要為其做相關設定，一個 IPsec.conf 基本設定如表 2:

此時若有使用防火牆時需要注意設定 iptable，必須將 IP 協定埠 50、51(因 ESP、AH 使用到這個兩個埠)允許通過防火牆否則 IPsec 將無法正常工作。

在同樣的環境下一樣在兩邊各建立一 VPN-µRouter，並載入 CIPE-LRP 模組，設定兩邊 CIPE 參數如表 3:

表 2 IPsec.conf 基本設定

```
conn net-to-net
  left = 163.23.24.205                #本端路由器 IP 位址
  leftsubnet = 192.168.1.0/24         #本端使用者 IP 位址
  leftnexthop = %defaultroute         #本端對外路由器設定
  leftfirewall = yes                  #本端是否有防火牆設定
  right = 163.23.24.99                #遠端路由器 IP 位址
  rightsubnet = 192.168.2.0/24        #遠端使用者 IP 位址
  rightnexthop= %defaultroute         #遠端對外路由器設定
  leftfirewall = yes                  #遠端是否有防火牆設定
```

表 3 使用防火牆的 IPsec.conf 基本設定

Router_A	Router_B
#指定使用的 Device Name	#指定使用的 Device Name
DEVICE=cipcb0	DEVICE=cipcb0
#是否自動啟動	#是否自動啟動
ONBOOT=yes	ONBOOT=yes
#使用者是否可以定訂	#使用者是否可以定訂

USERCTL=no	USERCTL=no
#本地端使用的 UDP port	#本地端使用的 UDP port
MYPORT=6060	MYPORT=6060
#遠端 IP 和 UDP port	#遠端 IP 和 UDP port
PEER=163.23.24.99:6060	PEER=163.23.24.205:6060
#遠端的 virtaul ip	#遠端的 virtaul ip
PTPADDR=10.0.0.1	PTPADDR=10.0.0.2
#本地端的 virtual ip	#本地端的 virtual ip
IPADDR=10.0.0.2	IPADDR=10.0.0.1

4.2 VPN-μRouter 測試結果與分析

完成所有相關設定後就可以開始測試 VPN-μRouter。此處測試重點是 VPN Tunnel 技術實作在 LRP 的可行性和兩種不同 VPN Tunnel 架構和 LRP 的 VPNTunnel 架構效能比較，而 VPN 所使用的任何加解密技術因受篇幅所限將不再此處討論。[4]

當完成架置兩個 VPN-μRouter 並設定相關設定後，使用 Linux 下常用的偵測及抓取封包的軟體 ethereal[10]來測試本文的 VPN-μRouter 是否正常工作。首先使用 ping 工具配合 ethereal 針對 IPsec 虛擬接口裝置(即封包未經 IPsec 處理)和對外實體接口(即原本對外網卡裝置 eth0；即已經過 IPsec 處理之封包) 偵測及抓取封包，結果如圖 4、5 所示。

No.	Time	Source	Destination	Protocol	Info
1	0.000000	163.23.24.205	163.23.24.99	ICMP	Echo (ping) request
2	0.000439	163.23.24.99	163.23.24.205	ICMP	Echo (ping) reply
3	1.000152	163.23.24.205	163.23.24.99	ICMP	Echo (ping) request
4	1.000604	163.23.24.99	163.23.24.205	ICMP	Echo (ping) reply
5	2.010159	163.23.24.205	163.23.24.99	ICMP	Echo (ping) request
6	2.010658	163.23.24.99	163.23.24.205	ICMP	Echo (ping) reply
7	3.010885	163.23.24.205	163.23.24.99	ICMP	Echo (ping) request
8	3.011471	163.23.24.99	163.23.24.205	ICMP	Echo (ping) reply
9	4.010153	163.23.24.205	163.23.24.99	ICMP	Echo (ping) request
10	4.010590	163.23.24.99	163.23.24.205	ICMP	Echo (ping) reply
11	5.020167	163.23.24.205	163.23.24.99	ICMP	Echo (ping) request
12	5.020665	163.23.24.99	163.23.24.205	ICMP	Echo (ping) reply

圖 4. ethereal 偵測 IPsec 虛擬接口圖

No.	Time	Source	Destination	Protocol	Info
3	0.739798	163.23.24.205	163.23.24.99	ESP	ESP (SPI=0x09103e20)
4	0.740248	163.23.24.99	163.23.24.205	ESP	ESP (SPI=0xf0107be6)
6	1.739798	163.23.24.205	163.23.24.99	ESP	ESP (SPI=0x09103e20)
7	1.740322	163.23.24.99	163.23.24.205	ESP	ESP (SPI=0xf0107be6)
9	2.739791	163.23.24.205	163.23.24.99	ESP	ESP (SPI=0x09103e20)
10	2.740242	163.23.24.99	163.23.24.205	ESP	ESP (SPI=0xf0107be6)
17	3.739846	163.23.24.205	163.23.24.99	ESP	ESP (SPI=0x09103e20)
18	3.740251	163.23.24.99	163.23.24.205	ESP	ESP (SPI=0xf0107be6)
22	4.739805	163.23.24.205	163.23.24.99	ESP	ESP (SPI=0x09103e20)
23	4.740195	163.23.24.99	163.23.24.205	ESP	ESP (SPI=0xf0107be6)
30	5.749810	163.23.24.205	163.23.24.99	ESP	ESP (SPI=0x09103e20)

圖 5. ethereal 偵測實體對外網卡裝置 eth0 圖

從圖 4 中可以發現在經過原先 IP 層處理過後被 IPsec 處理模組取下時是普通的 ICMP 協定封包，再經過 IPsec 處理模組針對封包進行加密認證後實際送到 eth0 網路裝置時，如圖 5 所示封包已被保護並經由虛擬安全通道傳送到遠端的 VPN-μRouter 經由反向處理後再發送給遠端的目的端，經過這部份的測試證實 VPN-μRouter 在實行 IPsec 處理模組上已無問題，然而一般傳送封包大體區分為兩大類，一類是連結導向(connection-oriented)另一類是非連結導向(connectionless-oriented)，ICMP 是屬於非連結導向，即未和遠端進行連結即進行封包傳送，而傳送資料時不會局限於 MTU 限制，因此對整體封包輸出效能的影響較小，本文使用 ESP 協議對封包進行加密和認證，不論是封包在 IP 層停留時間，或是對原封包大小的改變都有其變動，對於使用連接導向的封包類型影響較大，也因此本文才會實作分別以 TCP 和 UDP 隧道傳輸方式的 VPN 機制在本文架構上比較兩者在路由器上額外處理所花費的時間和整體效能。所以在下面會針對這部份做詳細比較。

接下來我們利用簡單的 socket-ftp 程式搭配 ethereal 來測試本文的 VPN-μRouter (採用 ESP 協議搭配隧道模式)和一般路由器透過 TCP/UDP 協定傳送資料時間比較如下表 4 如示。

表 4 四種情況下在路由器時傳輸效能比較

VPN 協議	IP 封包個數	平均總共花費時間	平均傳輸總訊框數	平均傳輸率
IPsec	1594	88.126s	107379	1.21Mbps
CIPE	1594	79.126s	104321	1.31Mbps
LRP-VPN	1594	87.911s	107285	1.22Mbps
No-VPN	1594	48.697s	71092	2.05Mbps

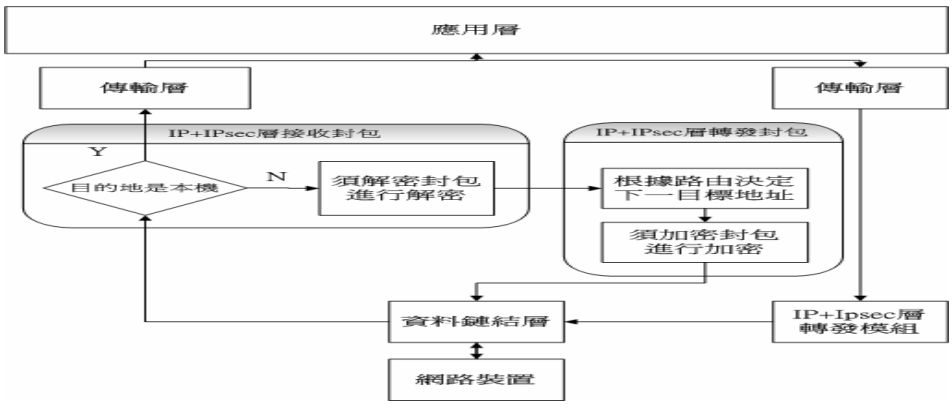

圖 6. VPN-μRouter 網路結構簡圖

在表 4 中很明顯看出未使用 VPN Tunnel 的路由器傳輸速度比使用 VPN Tunnel 的路由器快將近一倍。那是因為路由器在處理加密封裝動作上花掉不少時間，以 IPsec 部份為例，如圖 6。

這個機制將使得封包在 IP 層須經過兩階段處理，並經由一 2192bits RSA 加密，故增加了傳輸所需時間。而為何 LRP 的 VPN 架構在傳輸效能上也優於本文架構是因為它的加解密步驟比較簡單且演算法複雜度比較低和使用的加解密 KEY 比較短，所以傳輸效會比較好，但是也明顯可以判斷出它的安全性會比較低，這也是本文提出一個新的架構的原因。

下面針對兩種 VPN Tunnel 機制在傳輸效能上做更多的比較，同樣以前面使用的 socket-ftp 測試如圖 7。

可以從表 5 清楚的看到同樣大小的資料在經過不同 VPN 機制路由器所需花掉的時間是以使用 UDP 傳送的 CIPE 機制路由器比較快，頻寬佔有率也較高，但是在封包丟失率上明顯使用具有 flow control 機制的 TCP 傳送的 IPsec 比較低，特別是網路擁塞情況越嚴重時兩者丟失率差更大。

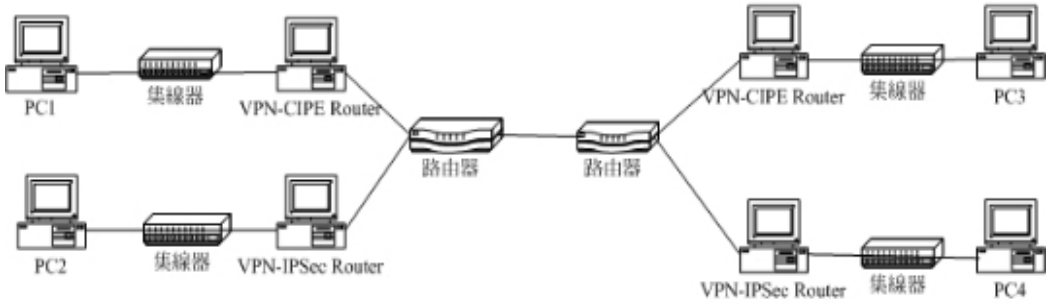

圖 7. 兩種 VPN Tunnel 實作測試環境

表 5 兩種不同傳輸模式的 VPN Tunnel 效能比較

VPN 協議	花費時間	比率	丟失率	平均延遲
IPsec	102.301 (s)	42%	7%	14ns
CIPE	62.014 (s)	58%	23%	8ns

表 6 IPsec 搭配 CIPE 和單獨 CIPE 的多點傳輸效能比較

VPN 協議	平均傳輸總訊框數	平均傳輸率	Latency
CIPE -IPsec	45890	94kbps	34ns
CIPE	33800	122kbps	11ns

本文另外一個重點即是提出現有方案中沒有支援多點傳輸的功能，不過一般多點傳輸只用於 UDP 上，如果使用在利用 TCP 協定傳輸的 IPsec 似乎比較困難，因此本文藉由把 IPsec 傳送時再套上CIPE才傳送出去和只有單CIPE來做多點傳輸效能的比較。從表 6 可得知加上兩次 VPN 封裝的 CIPE –IPsec 組合因為處理解封裝時多花了一些時間，每次傳送的資料量變少了，傳送的訊框數增加了，因此傳輸效能比單獨使用 CIPE 來得差，但是能否藉由兩次的 VPN 封裝來提高安全性是另一個有趣的問題，不過本文並不在此討論其安全方面的問題。

五、結論

本文設計並實作了一個稱之為 VPN-µRouter 具備低成本、維護容易和操作簡單的虛擬私有網路微型路由器架構。同時在本文中，確定此架構在真實環境下的可行性，可以從圖 4、5 中確定原始封包透過 VPN Tunnel 機制加密和封裝，以及驗證可以正常傳送到出去；另外在兩種具備較高靈活性的 VPN 機制的傳輸效能比較上，看到了以 UDP 傳輸模式的 CIPE 傳輸量會比以 TCP 傳輸的 IPsec 來得多，同時 CIPE 使用的加密演算法只使用 128Bit 長度的 Key，大概是 IPsec 的一半，在處理時就會稍微快一些。但是考量到 UDP 先天上的缺點，包括：不可靠性、沒有流量控制機制等。如圖表 5 的丟失率來看，雖然 IPsec 傳輸較慢，但丟失率明顯低於 CIPE。因此，如何同時維繫兩種機制的優點將會是未來的工作之一。另外在多點傳輸的方面，由於沒有實際的其他方案來做比較，因此這裡只能說本文架構能夠確實達到多點傳輸的功能，暫時無法提供不同方案的效能比較。

最後，在考量此架構未來的發展性，未來的工作包括：(1).把本文架構放入更大容量的可移動式裝置上，如：LS-120、隨身碟等, (2). 嘗試改用更快速的加密演算法並且以更低階的組合語言碼來實作程式碼，藉此降低封包在路由器處理的時間, (3).制定 QoS 策略在本文架構上，能夠在實作安全機制的同時，也能做到頻寬控管的能力, (4).為了能夠讓本文架構達到在各種硬體上執行的目的，讓 VPN-µRouter 具有跨平台能力也將是未來一大挑戰。

六、參考文獻

[1] Venkateswaran,R.,“Virtual private networks”,IEEE Potentials,Volume 20, Issue 1,11-15,Feb/Mar(2001)

[2] Zhao Aqun and Yuan Yuan and Ji Yi and Gu Guanqun,“ Research on tunneling techniques in virtual private networks, " Communication Technology Proceedings, 2000. WCC - ICCT 2000. International Conference on , Volume 1 , 691 –697,2000.

[3] Keromytis,A.D.and Ioannidis,j.and Smith,J.M, “ Implementing Ipsec, " Global Telecommunications Conference, 1997. GLOBECOM '97., IEEE , Volume: 3 ,1948 –1952,1997.

[4] Pena,C.J.C., “ Performance evaluation of software virtual private networks (VPN), " Local Computer Networks, 2000. LCN 2000. Proceedings. 25th Annual IEEE Conference on ,522 –523,2000.

[5] LRP , http://www.linuxrouter.org/

[6] CIPE ,http://sites.inka.de/sites/bigred/devel/cipe.html

[7] FreeS/WAN , http://www.freeswan.org/

[8] BusyBox , http://www.busybox.net/

[9] uClibc , http://www.uclibc.org/

[10] Ethereal , http://www.ethereal.com/