

在 Edge Device 上實作 IP over ATM 通訊協定 The design and implementation of IP over ATM on Edge Device

賴威光
Wei Kuang Lai

國立中山大學資訊工程研究所
Institute of Computer and Information engineering
Nation Sun Yat-Sen University
Kaohsiung, Taiwan, R.O.C

蘇讚元
Tzann-yuan Su

國立中山大學資訊工程研究所
Institute of Computer and Information Engineering
Nation Sun Yat-Sen University
Kaohsiung, Taiwan, R.O.C

摘要

在本篇論文中，我們在區域 ATM 網路(Local ATM Network)和傳統的 Ethernet 網路間的連接設備(Edge Device)上實作 IP over ATM 通訊協定，提出 EDIM(Edge Device Interface Model)的方法，來解決封包在異質網路間的轉送問題，並探討位址解析的問題，提出 Proxy ARP 的方法，使得在傳統 Ethernet 網路的上的應用程式能不需要修改就能享受到 ATM 所帶來快速及高頻寬的好處。

Abstract

In this paper, we propose an Edge Device Interface Model(EDIM) to implement IP over ATM on an edge device. Programs developed for TCP/IP can communicate between Ethernet LANs and ATM LANs without any modification and enjoy the high bandwidth provided by ATM networks.

1. 序論

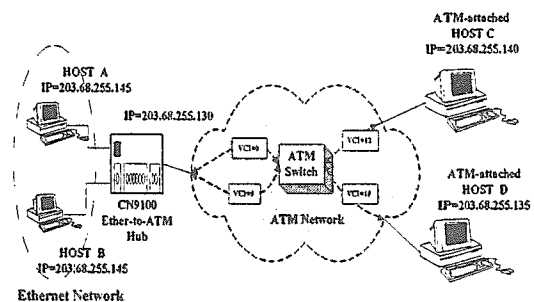
由於區域網路的快速成長，以及多媒體應用的導入，使得傳統分享式媒體(shared media)的網路環境面臨效能(performance)和管理(management)上的限制與瓶頸，但是近年來高速網路的蓬勃發展，尤其以非同步傳輸模式 ATM(Asynchronous Transfer Mode)

在本篇論文中，我們是採用 IP over ATM 的方式來達到傳統網路與 ATM 網路的互連，現在有關於架構在 ATM 網路上的 IP 通訊協定的研究，已經有 IETF 組織所訂定的『RFC 1577 Classical IP over ATM』來規劃標準的方法[4]、RFC 1483 Multiprotocol Encapsulation over ATM』來規

定 IP packet 和 ATM cell 的封裝方法[6]、『RFC1626 Default IP MTU for use over 『ATM AAL5』定義 IP datagram 在 ATM 傳送的最大傳輸單元(Maximum Transfer Unit; MTU)[5]、『RFC 1755 ATM Signaling support for IP over ATM』和『ATM Forum UNI 3.1 specification』規定 IP over ATM 如何連接設定[2][3]。我們將在本篇論文中探討在 Edge Device 上實作 IP over ATM，並設計出 EDIM(Edge Device Interface Model) 來解決位址解析(address resolution)與封包封裝(encapsulation)的問題。

2. 網路架構

我們以 CNet CN9100 Ethernet-to-ATM Hub 做為開發 IP over ATM 協定功能的平臺，此平臺採用 Intel 960 晶片，我們採用支援 Intel 960 晶片的 USNET Multitask 即時作業系統。目前在 CN9100 Hub 上的 ATM 網路連接部份只有支援 PVC 的連接方式，因此在連接設定上只考慮 PVC 方式。實作 IP over ATM 的網路架構如圖(2.1)所示，ATM-attached Host 都裝置具有 Classical IP over ATM 功能的 ATM 網路卡，整個網路架構成一個 Logical IP subnet (LIS)，所有的機器必須具備相同的 IP network/subnet 位址。

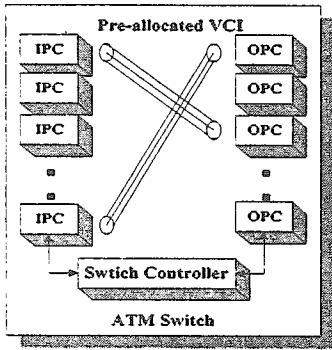


圖(2.1)實作 IP over ATM 網路架構圖

This research is supported in part by NSC contract number NSC85-2221-E-110-005.

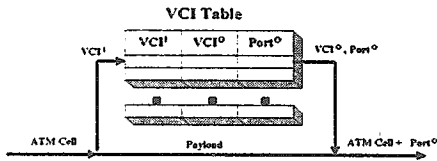
ATM switch 的內部結構如圖(2.2)，所有

的輸入和輸出埠都具有控制器稱為輸入埠控制器 IPC(Input Port Controller)及輸出埠控制器 OPC(Output Port Controller)，而每個 switch 內也有一個控制器用來配置 VCs 及管理各個埠控制器內的 VCI 表格。每一條 VC 是由(VCI¹，VCI⁰)來識別，IPC 的主要元件就是 VCI 表格



圖(2.2) ATM Switch architecture

如圖(2.3)所示，此表格的主要功能即是將輸的 VCI¹轉換成輸出的 VCI⁰及輸出埠(Output Port)。在 CN9100 Hub 的 ATM 部分，我們是使用 AAL5 adaptation Layer，



圖(2.3) VCI table

3. 系統架構

通訊協定(protocol)是網路通訊的基礎，應用程式透過層層負責不同工作任務的通訊協定的處理才能在網路上來傳遞，基本上通訊協定是以軟體來實現的，USNET Multitask RTOS 的 TCP/IP 通訊協定的運作方式是，它訂定每層通訊協定所需要的功能函式，透過設計一個 Protocol Table 來存放每個功能函式的指標，

因此系統宣告一個陣列指標 PTable *protoc[3]，即可以用 protoc[0] 存取到 CN9100(Driver) 的所有函式功能，如透過 protoc[0]->screen() 就可以呼叫到 CN9100 的 screen 函式，相同的 protoc[1] 存取到 Ethernet Layer 的所有函式功能，protoc[2] 目前在系統保留沒有使用。在功能函式中的 screen() 和 writE() 是最重要的二種函式呼叫，screen() 是用來接收來自下層的封包，即當封包在目前的層次處理完後，可透過呼叫上層的 screen(mess) 函式來把 mess 訊息送給上層通訊協定，writE() 是用來傳送封包給下層通訊協定來處理，即當封包在目前的層次處理完後，可透過呼叫下層的 writE(mess) 函式來把 mess 訊息送給下層通訊協定。

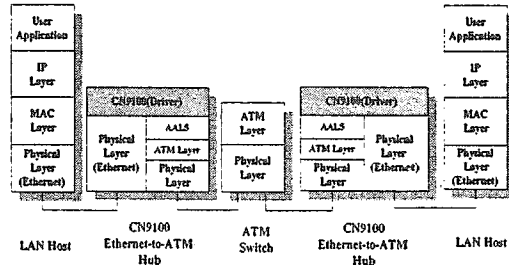
另外對於封包格式的处理，CN9100 Hub 系統藉著宣告一個 MESS header 結構，來處理每一個進來的封包，即當封包從底層(CN9100 Driver)接收進來時就會附加 MESS header 在封包的前面，然後跟隨封包到每一層通訊協定，如此方便在每一層去取得關於此封包的一些重要的資訊，MESS header 的結構如圖(3.1)所示。

Unsigned long "timeStamp" /* Time Stamp */	Unsigned long "target" /* P address */	Unsigned short "len" /* message length */	Unsigned char "proto" /* protocol number */	char "offset" /* offset to current level */	Unsigned char "cmd" /* network configuration code */	char "conn" /* connection number */	short "id" /* message ID */	short "mess" /* mess pointer */
--	--	---	---	---	--	---	-----------------------------------	---------------------------------------

圖(3.1) MESS header 的結構

在 MESS 結構中的 offset 為偏移值，也就是其值都會指到目前要處理封包的開始位置，例如當封包到達 IP 層時，其 offset 必須是等於 Length(MESS header) + Length(Ethernet header) 這樣指標就可以很容易存取到 IP datagram 開始位址的地方。

CN9100 Ethernet-to-ATM Hub 具備有使傳統 Ethernet 網路連上 ATM 網路的功能，不過它只單純將 Ethernet 的封包直接轉成 ATM cell 再送給 ATM 網路，本身不具備 TCP/IP 通訊協定的能力，其原始通訊協定堆疊(Protocol Stack)架構如圖(3.2)所示。



圖(3.2) CN9100 Hub Protocol stack diagram

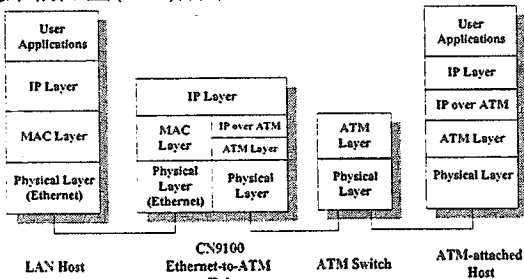
基本上 CN9100 Hub 的系統是由一個最重要的 BridgeTask 工作在執行，主要負責二個事項：第一，將從 Ethernet 埠及 ATM 埠接收的資料封包存放到緩衝區(buffer)中，即放到系統的 arrive queue 中；第二，從緩衝區(系統的 depart queue)取出封包依目的位址資訊來送到 Ethernet 埠或 ATM 埠，底下為其程式碼。

```

Void BridgeTask()
{
for( ; ; ) {
    int_off(); /* 關掉 interrupt()功能 */
    DEC_Receive(); /* call bridge_forward()接收封包*/
    SARA_Receive(); /* call bridge_forward()接收封包 */
    SARA_Free_TX_Desc();
    DEC_Free_TX_Desc();
    int_on(); /* 恢復 interrupt()功能 */
}
}
    
```

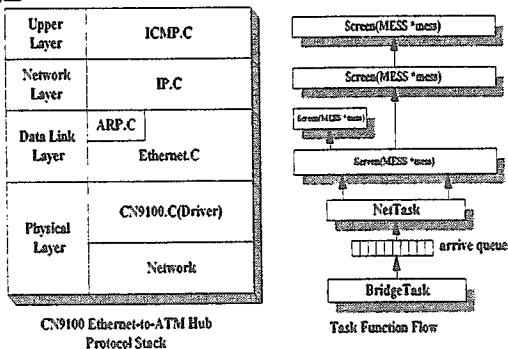
DEC_Receive()會呼叫 bridge_forward()函式來接收 Ethernet 網路上的封包，同樣的 SARA_Receive()也會呼叫"bridge_forward()"函式來接收 ATM 網路上的封包，因此我們設計增加一個 NetTask 工作，這個工作是會固定循環從緩衝區(即系統的 arrive queue)取出封包送給上層(Ethernet Layer)的通訊協定處理，也就是會呼叫 Ethernet 層專屬的 screen()函式。

接著我們要將 TCP/IP 通訊協定的功能加入到 CN9100 Hub 中，使之具備有網路層的功能，然後在此平臺上來發展 IP over ATM 協定軟體。我們的做法在使傳統的 Ethernet 網路上執行 TCP/IP 通訊協定的機器，能夠透過 CN9100 Hub 來與具有 Classical IP Over ATM 協定功能的 ATM-attached 主機互相通訊，即我們要在 CN9100 Hub 中實作 Classical IP over ATM 協定的軟體，其通訊協定架構如圖(3.3)所示。



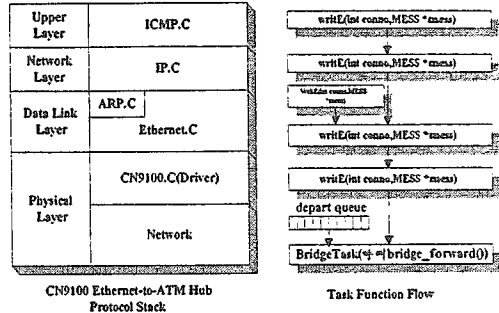
圖(3.3) CN9100 Hub Protocol stack diagram with IP over ATM

我們移植了 USNET 的 Ethernet Layer(Ethernet.c)、ARP Layer (ARP.C)、IP Layer(IP.C)及 ICMP Layer(ICMP.C)到 CN9100 Hub 系統上，整個通訊協定移植完後，系統在底層 Physical Layer 以 NetTask 固定去檢查 arrive 佇列是否有封包，若有封包則將封包往上傳送到 Ethernet 層去處理，往上傳送則是透過呼叫 Ethernet 層的 screen() 函式來處理，同樣的當 Upper Layer 收到封包處理完畢後，則是又透過每層的 writE() 函式來將封包層層往下送到網路去，圖(3.4)(3.5)所示為移植到 CN9100 Hub 上的 TCP/IP 通訊協的 Protocol stack 資料接收/傳輸流程。



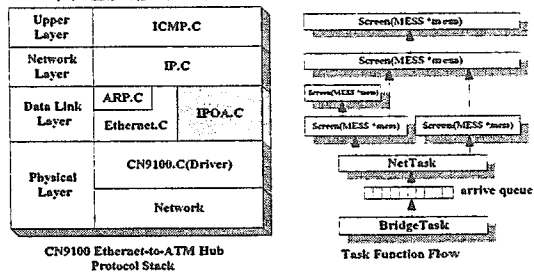
圖(3.4) 移值 IP 後之接收封包的流程

因此我們設計的 IPOA Layer 就是要放在 Data Link 層的位置，即在 Ethernet 層的相同位置，用來負責處理 ATM cell 和 IP datagram 的轉換。

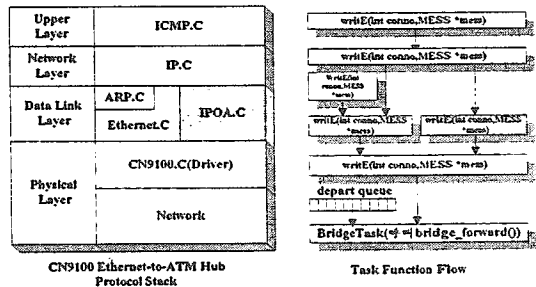


圖(3.5) 移值 IP 後之傳送封包的流程

，我們以圖(3.6)(3.7)來概觀整個 Protocol stack 傳送及接收的流程。



圖(3.6) 具 IPOA 功能之接收資料的流程



圖(3.7) 具 IPOA 功能之傳送資料的流程

在上圖中我們將設計 IPOA 層的專屬 screen() 及 writE()函式，即在 Protocol Table 中陣列指標中多增加了 IPOA 的選項如下所示：

```
宣告 PTable *protoc[4]
protoc[3]->screen()
/*可以存取到 IPOA 層的 screen 函式*/
protoc[3]->writE()
/*可以存取到 IPOA 層的 writE 函式*/
```

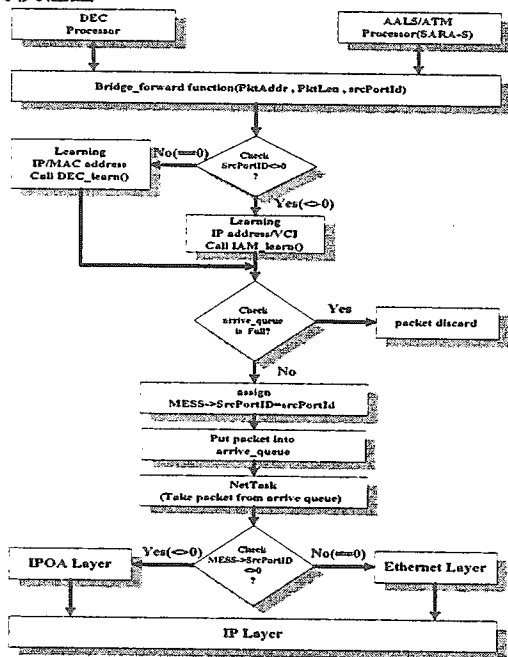
4. EDIM(Edge Device Interface Model)的方法

在 IPOA 層完成後，接著必須要考慮 IPOA 層和其上層(IP Layer)與下層(CN9100 driver)間的

關係，即必須以一個系統概觀的方式來描述封包在整個協定堆疊傳遞的流程，底下將描述我們的 Edge Device Interface Model (EDIM) 的方法，EDIM 將分成封包接收及封包傳送來敘述。

4.1 EDIM 封包接收的設計與實現

底下將描述 CN9100 的 bridge_forward() 函式與 IPOA 層之間的關係，如圖(4.1)為封包接收的模組圖。



圖(4.1) 系統接收封包的模組圖

CN9100 Hub的封包接收是以 bridge_forward() 來負責接收與傳送封包，而在此函式內部最重要的工作即是位址學習，其規格如下有三個傳入參數：

```
bridge_forward(PktAddr, PktLen, srcPortId)
unsigned char *PktAddr;
unsigned short PktLen;
unsigned short srcPortId;
```

PktAddr 是一個指標指到封包在記憶體的啓始位址，PktLen 是記錄整個封包的長度，srcPortId 則是埠號(port number)。

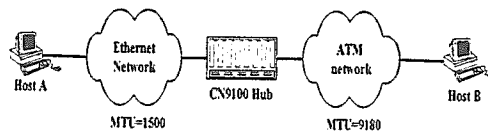
當我們收到來自網路的封包時，封包進入 CN9100 Hub 後，系統會在此封包前面附加一個 MESS 標頭，如此可以隨時在各層次取得與封包相關的重要資訊，所以我們透過在 MESS 標頭的 SrcPortID 埠識碼欄位，來記錄封包是從 Ethernet 網路(當 SrcPortID 等於零)或是從 ATM 網路來的(當 SrcPortID 不等於零)，當封包是來自 Ethernet 網路則必須將其 MAC 位址與埠號碼(port number)學習起來，反之從 ATM 網路來的封包則將其 IP 位址與 VCI 學習起來，接著檢查到達佇列(arrive queue)是否滿了，若沒有滿則將 srcPortId 參數

指定給 MESS 標頭的 SrcPortID，然後封包會存放到佇列中，否則封包將會被丟棄，接著 NetTask 會從到達佇列(arrive queue)中取出封包，再透過在封包前面的 MESS 標頭的 SrcPortID 欄位來判斷封包要往上送給 Ethernet Layer 或是送給 IPOA Layer，即當 SrcPortID 為零時，表示封包一定是從 Ethernet 埠進來的，因此其封包格式為 Ethernet header+IP datagram，所以此種格式的封包就一定要送到 Ethernet 層來解讀，相反的當 SrcPortID 不等於零，表示此封包一定是從 ATM 埠進來的，因此其封包格式為 LLC/SNAP header+IP datagram，所以這種封包就要送到 IPOA 層來解讀。

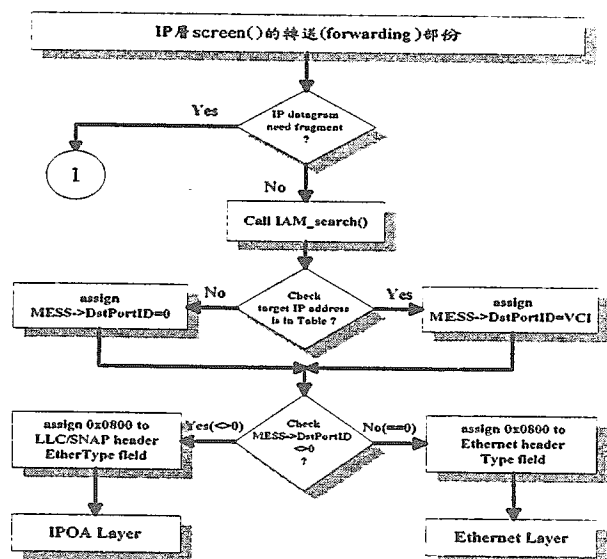
接著 IPOA 層處理封包完畢後，封包可以不必做任何變換就將 IP datagram 傳給 IP 層來處理，在 IP 層的 screen() 函式的運作下，IP datagram 除非是送給自己的封包，才會又往上層來傳送，否則都將執行轉送的動作，也就是 IP datagram 將開始要轉送到下層通訊協定來處理，這部份將是下一節所要敘述的。

4.2 EDIM 封包傳送的設計與實現

當 IP Layer 的轉送功能要將 IP datagram 往下送時，要先考慮 MTU(Maximum Transfer Unit)的問題，即 Ethernet 網路的 MTU=1500 bytes，而 IP over ATM 的 MTU=9180bytes，如圖(4.2)所示。



圖(4.2) Ethernet 網路和 ATM 網路的 MTU 問題

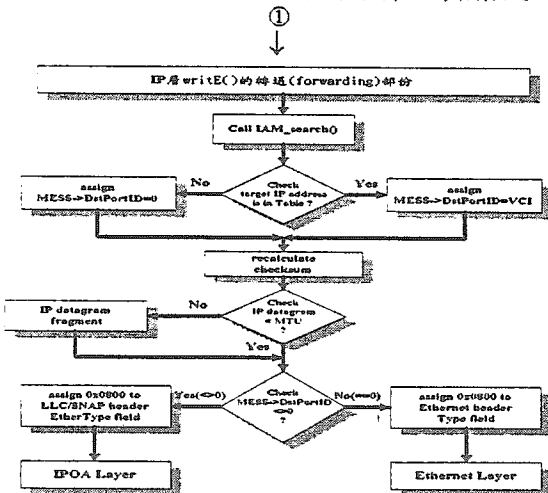


圖(4.3) 系統傳送封包的模組圖(一)

所以很明顯在封包的轉送過程中，從 ATM-attached Host B 來的封包有可能會大於 1500bytes，所以就必須要執行切割 (fragmentation) 的動作，而從 Ethernet 網路來的封包，由於其 MTU=1500bytes 的限制，遠遠小於 ATM 網路的 MTU=9180 的限制，所以不需要執行切割的動作，如圖(4.3)所示為 IP 層 screen() 函式的轉送 forwarding) 部份與下層的封包傳送模組圖，當 IP datagram 需要分割則是直接呼叫 IP 層的 write() 函式來作處理，接著若不需要切割，則必須要分辨 IP datagram 是往下送給 Ethernet Layer 或是 IPOA Layer。

我們利用在 bridge_forward() 函式中，曾經執行過 IAM_learn() 函式所建立的 IP 位址和 VPI/VCI 的對照表，藉著呼叫 IAM_search() 函式來查詢此表格得知是否有 IP address 和 VPI (out) / VCI (out) 符合的，若在表格中有找到相對應的 entry，則將取得的 VCI 存放到 MESS 標頭的 DstPortID 欄位，反之將 MESS 標頭的 DstPortID 欄位存入零，接著藉由判斷 DstPortID 來決定 IP datagram 要往下送給 Ethernet 層或是 IPOA 層，當 DstPortID 不等於零時，我們可以確定封包是要送給 IPOA Layer，反之則是送給 Ethernet Layer，然後我們必須指定 IPOA 的 LLC/SNAP 標頭的 EtherType 欄位，即指出是 IP 通訊協定(其值為 0x0800)，而在送到 Ethernet 層之前也必須指定 Ethernet 標頭的 Type 欄位，指出是 IP 通訊協定(其值為 0x0800)。

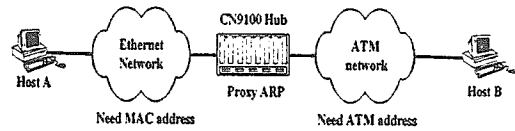
為 IP 層 write() 函式的轉送(forwarding) 部份與下層的封包傳送模組圖，這部份的 IP datagram 處理是來自圖(4.3)的 call IP Layer write() 模組，即是需要將 IP datagram 切割(fragment) 的部份，當然切割後的每個 IP datagram 還是必須透過 IAM_Search() 來得知 DstPortID 的值，然後才能判定要送往 IPOA 層還是 Ethernet 層，其運作流程是和圖(4.3)相似的。



圖(4.4) 系統傳送封包的模組圖(二)

5. Proxy ARP

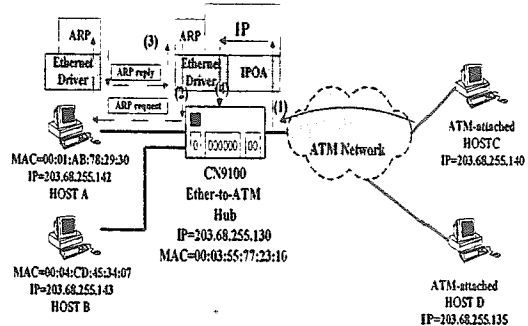
這部份的功能在整個系統中是一個重要的關鍵，因為在傳統的 Ethernet 網路中，主機之間的通訊都是藉 MAC 位址來識別，也就是 Ethernet 網路界面卡上的實體位址，由於 CN9100 Hub 系統必須處理 Ethernet 網路與 ATM 網路的封包，來自 ATM 網路的封包不具備有目的 MAC 位址的資訊，來自 Ethernet 網路的封包則沒有 ATM 位址的資訊，因此 CN9100 Hub 必須扮演一個通訊代理者的角色(如圖 5.1)所示，才能使封包的轉送能夠正確的運作，接著將分為二種情況來描述解決的方法。



圖(5.1) CN9100 Hub 扮演 Proxy ARP 的角色

5.1 從 ATM 網路到 Ethernet 網路的訊息封包

如圖(5.2)所示，當 ATM 網路上的 HOST C 欲傳送資料給 HOST A 時(1)，當 CN9100 Hub 系統的 Ethernet 層要往下層送出封包時，發現沒有目的地主機 HOST A 的 MAC 位址，所以在轉送的過程中就一定要有一個機制來取得目的地的 MAC 位址，即 CN9100 hub 要扮演一個 Proxy ARP 的角色，於是它委託 ARP 層發出 ARP request 到網路上(2)，詢問誰有 HOST A 的 MAC 位址，當 HOST A 的 ARP 層收到 ARP request 後並確認是自己的 IP 位址，接著就會發出 ARP reply。當 CN9100 hub 收到此回應訊息後(3)，才能順利的將封包轉送到 Ethernet 網路上的主機 HOSTA(4)。

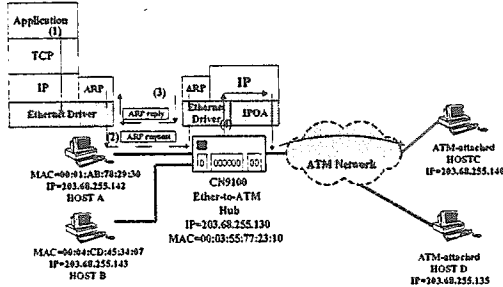


圖(5.2) Proxy ARP 的運作方式(ATM to Ethernet)

5.2 從 Ethernet 網路到 ATM 網路的訊息封包

同樣的如圖(5.3)所示，當 Ethernet 網路上的 HOST A 欲傳送資料給到 ATM 網路的 HOST C 時(1)，由於 Ethernet 網路的先天特性，只能辨

識 MAC 位址，所以 HOST A 會委託 ARP 層發出一個 ARP request(2)到網路上來查詢誰有 HOST C 的 MAC 位址，但 ATM 網路上主機是不具有 MAC 位址，所以 CN9100 hub 就必須扮演一個 Proxy ARP 的角色，要將 ARP request 封包攔截並代替 ATM 網路上的主機 HOST C 回答 ARP reply，且需回答 CN9100 hub 自己的 MAC 位址，當 CN9100 hub 的 ARP 層收到 ARP request 後，接著就會發出 ARP reply，當 HOST A 收到此回應訊息後(3)，才能順利的將封包送給 CN9100 hub，然後 CN9100 hub 也才會正確轉送封包到 ATM 網路上的主機 HOST C(4)。



圖(5.3) Proxy ARP的運作方式(Ethernet to ATM)

6. 結論

非同步傳輸模式(ATM)網路擺脫以訊框為基礎的分享式區域網路及多協定企業網路環境，改採以細胞(cell)為基礎的交換式網路環境，並利用終端對終端(end-to-end)連接，整合區域網路及廣域網路。Classical IP over ATM 整合了傳統 TCP/IP 網路和 ATM 網路，使得以訊框為基礎的封包得以在高速的 ATM 網路來傳送，加快了傳輸的速度，並提供了優越的頻寬能力，降低了傳統網路管理的複雜度。

本篇論文主要是針對在不同網路之連接設備上，跟據 RFC1577、RFC 1483、RFC1626 等標準文件，設計並實現 IP over ATM 通訊協定，雖然我們實作的 IP over ATM 軟體由於硬體上的限制無法實現更多的功能，但目前 ATM 的相關產品還是相當昂貴，使用者要享受到 ATM 網路的許多好處更是要花費龐大，而在 CN9100 hub 上開發 IP over ATM 軟體，使得傳統 Ethernet 網路上的機器不需要每台購置昂貴的 ATM 網路界面卡，只要透過這樣一台中介設備就能連上 ATM 網路，將是使用者升級到 ATM 網路的很好選擇。

7. 參考文獻

[1] Postel, J. B., ed. "Internet Protocol DARPA Internet Program Protocol Specification", RFC 791, September 1981

[2] The ATM Forum, "ATM User-Network Interface(UNI)Specification", <ftp://ftp.atmforum.com/pub/UNI/ver3.1>, Prentice Hall, 1994

[3] Cole, R. G.; Shur D. H., "ATM Signaling Support for IP over ATM", IETF Network Working Group, RFC 1755, February 1995

[4] Laubach M., "Classical IP and ARP over ATM", IETF Network Working Group, RFC1577, January 1994.

[5] Atkinson R., "Default IP MTU for use over ATM AAL5", IETF Network Working Group RFC 1626, May 1994.

[6] Heinanen J., "Multiprotocol Encapsulation over ATM AAL5", IETF Network Working Group, RFC 1483, July 1993.

[7] Plummer, D. C., "An Ethernet Address Resolution Protocol", RFC 826, November 1982.

[8] Bradley, T.; Brown, C., "Inverse Address Resolution Protocol", IETF Network Working Group, RFC 1293, January 1992.

[9] Debanjan Saha; Dipak Ghosal; H. Jonathan, "A design for Implementation of the Internet Protocol in a Local ATM Network", IEEE Magazine, 1994

[10] Cole, R. G.; Shur, D. H.; Villamizar, C., "IP over ATM: A Framework Document", February 23, 1996

[11] Borden, M.; Crawley, E.; Davie, B.; Batsell, S., "Integration of Real-time Services in an IP-ATM Network Architecture", IETF Network Working Group, RFC 1821, August 1995

[12] Stevens, W. R., "TCP/IP Illustrated Volume I-The Protocols"

[13] Wright, G. R.; Stevens, W. R., "TCP/IP Illustrated Volume 2- The Implementation"

[14] Alles, A., "Internetworking with ATM", <http://cell-relay.indiana.edu/cell-relay/docs/cisco.html>, Cisco System, May 1995

[15] ATM Forum "LAN Emulation over ATM specification - version 1", ATM Forum specification, February 1995.