

前言

作為一個新世紀的人類，沒聽過電腦網路，沒用過它，你周圍的人大概會以為你是搭乘時光隧道來的山頂洞人。也許你已經瞭解如何使用它，可能你也看過 ISO 的 OSI 七層，也瞭解互連網路的基本概念。但是你可能會好奇這最底層的技術是如何辦到的，眾多的作業系統，眾多的網路卡驅動程式，是如何呈現在你眼前的，那麼，底下的文字將會滿足你一大堆的問號。本文將先介紹市面上的作業系統的網路功能，再介紹幾個重要的網路驅動介面標準及網路卡運作原理，最後討論如何開發網路驅動程式。

網路作業系統之比較

在談驅動程式之前，讓我們先來看看市面上幾個流行的作業系統是怎麼看待網路。我們將會談到 DOS、Windows 3.x、Windows 95、Windows NT 3.x、OS/2 2.x、NetWare、Unix。

	DOS	Windows 3.x	Windows 95	Windows NT	OS/2	Netware 3.12	Unix
發表年代	1981	1990	1995	1993	1987	1993	1970
內建網路	No	No	Yes	Yes	Yes	Yes	Yes
網路架構	C/S	C/S PtP	C/S PtP	C/S PtP	C/S	C/S	C/S
網路驅動程式 介面	Many	Many	NDIS 3.0	NDIS 3.0	NDIS 2.0.1	ODI	DLPI

C/S:Client/Server PtP:Peer-to-Peer

表一：各作業系統之比較

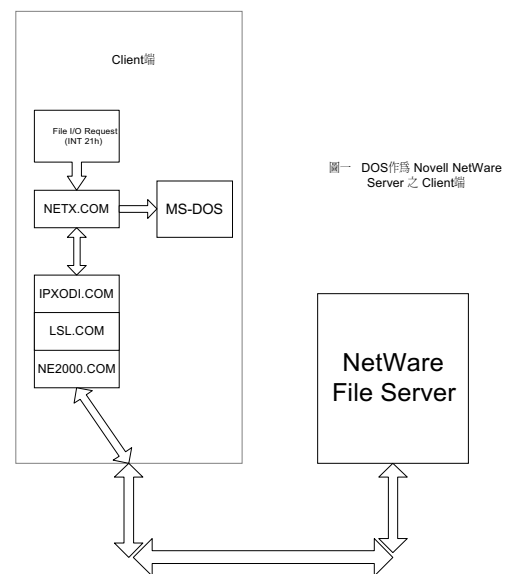
圖中的發表年代是根據其最早問世的版本，由圖中可得知除了早期 DOS/Windows 3.1 並沒有內建網路功能之外，其它的作業系統都將網路視為必備的工具，但也由於 DOS/Windows 3.x 沒有內建網路功能，其網路驅動程式介面也顯得多樣性，而有百家爭

鳴的現象。值得注意的是，雖然 Peer-to-Peer 的網路程式架構連並不是近幾年來才發展的，不過各大廠似乎都較鍾愛 Client/Server 的方式，有一點要注意的是 Windows3.x 的 peer-to-peer 功能是指 Windows for Workgroups 3.x 的功能，細節請詳閱本文。

DOS

1981 年 Microsoft 推出 DOS 以來，這套作業系統彷彿就像影子般圍繞在眾多 PC 的身旁，無論你是否喜歡它或是討厭它，相信你一定會用過它。DOS 的版本有數種，MS-DOS、PC-DOS、Novell-DOS，不過大同小異，彼此間相容性頗高，所以我們就拿大家所常見的 MS-DOS 來介紹。在網路的環境中，MS-DOS 主要是用在 Client/Server 環境下，當作 Client 端來使用，經由一些 Client 端網路程式，你就可以使用 Server 所提供的資源。

譬如，你想連上 Novell 的 NetWare server，如圖一，你必須載入以下四種 Client 端網路程式，LSL.COM（作為網路卡驅動程式以及協定堆疊之橋樑），NE2100.COM(網路卡驅動程式)，IPXODI.COM（IPX/SPX 協定堆疊），NETX.COM(網路介面，其主要功能為攔截檔案 I/O 的要求，以決定是送往 MS-DOS，存取本地的檔案，或是送往網路，存取遠端伺服器之檔案)。



好了，你現在已經知道如何連上 Server 了，讓我們來談談更下層的長相，也就是網路卡驅動程式的樣子。在 MS-DOS 下，驅動程式有兩種格式，一種是大家學習組合語言最常遇到的 TSR（Terminate and Stay Resident），通常是以.COM 的樣子出現，你可以在命令列下將其載入，如，C:\NETWOEK>ne2100。另外一種則是.SYS 的格式，這種則是原始的 MS-DOS 驅動程式模樣，你可以以”device=”的樣子寫在 CONFIG.SYS 檔裡將其載入。當載入時，當然是做些初始化的動作，它會將相關的硬體及軟體各就定位，並告訴 MS-DOS 這驅動程式放在那個中斷向量上，當你開始玩網路，傳送資料時，上層軟體以及下層硬體便是透過中斷向量來叫驅動程式做事。

Windows 3.x 系列家族

介紹完 DOS 後，我們就先來瞧瞧 Windows 3.x 系列家族與網路之間的互動，雖然說圖形介面的使用者環境並不是 Microsoft 所提出的構想，首先應用此項技術的也不是他，然而 Windows 3.x 系列家族在市場上卻是套成功的產品。我們可以將 Windows 3.x 系列家族分成兩個部份討論，一個是 Windows 3.x 以及 Windows for Workgroups 3.x。

Windows 3.x 與 MS-DOS 相同主要是在 client/server 的環境下當作 client 端來使用。其使用網路的方法與 DOS 很相近。在進入 Windows 前，將 Client 端網路程式載入，進入 Windows 後，你就可以在檔案管理員看到網路磁碟機的資訊，並使用 Server 的資源了。如果您曾用過 Windows 的話，想必會覺得使用它是沒有啥困難的，但是當你的好奇心，突然想要殺死隻貓的時候，一定會發現 Windows 的內部結構，真是不是普通的複雜，在這裡，我不會讓你搞的頭昏眼花，我把基本的架構闡述一下，至於可能會頭昏的細節，就要麻煩你自己去參考有關的書籍了。

Windows 3.1 有兩個模式，一個叫做標準模式，一個叫做增強模式。在標準模式下，MS-DOS 的應用程式，不能和 Windows 的應用程式多工，而增強模式則利用虛擬機器 (Virtual Machine) 的概念，讓使用者可以同時執行許多個 MS-DOS 應用程式，以及許多個 Windows 應用程式，這些應用程式是由一個叫做 VMM(Virtual Machine Manager)的東西所管理。由於要達成多工，同時使用系統之資源，所以就有個叫做 VxD(Vitual Device Driver)的東西，它是一個 32 位元，運作在 Intel 80386 以上的 CPU 的驅動程式，用來協調各個應用程式使用系統的資源，每個資源都有自己的 VxD。如果你的硬體只有老舊的 MS-DOS 的驅動程式，那其架構如同圖二的左半邊，存取下層的資源時，你無法達成多工，但如果你的硬體有附這個特殊的 VxD 型態的驅動程式，那麼架構就如圖二的右半邊，想當然爾，從上到下都可以多工了。

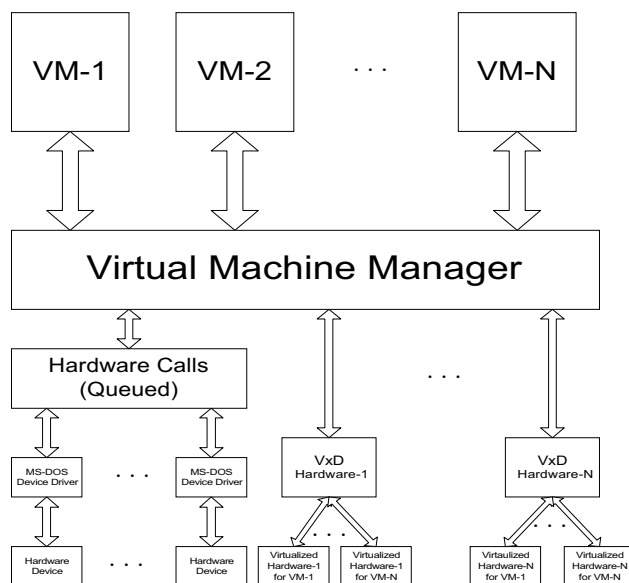
而 Windows for Workgroups 3.x(WFW 3.x)則是將 peer-to-peer 的網路環境概念引進來，每個 WFW 3.x 成員都可以將自己的檔案或是列表機分享給其他的成員。另外，WFW 亦可以使用在 Client/Server 的環境中，作為 LAN Manager，IBM LAN Server，Novell NetWare，Banyan Vines 等的 Client 端，來存取 Server 上的資源。

WFW 與 Windows 3.1 或是 Windows 3.11 有啥不同呢？很簡單，WFW 有內建網路的功能，而 Windows 3.x 沒有，就資源分享來說，WFW 要比 Windows 3.1 容易些

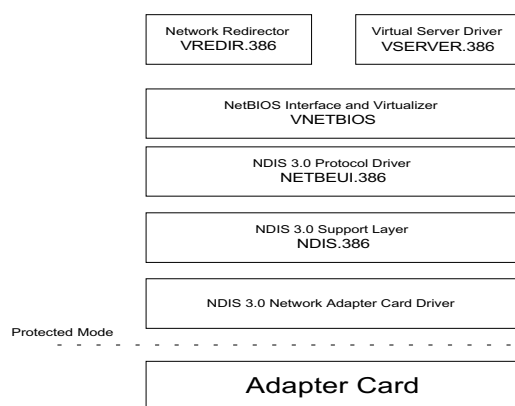
在這裡就不詳述 WFW 如何互連網路的步驟，讀者可翻閱 WFW 手冊，或是相關之書籍，應該就能上手了。讓我們來談談 WFW 3.1 與 WFW 3.11 有什麼不同，就使用者來說，可能沒有啥不同，不過對於網路卡驅動程式撰寫者來說，就有些不同了，WFW 3.1 採用 NDIS 2.1.0 的網路驅動程式介面，而 WFW 3.11 則是採用 NDIS 3.0 的網路驅動程式介面，這兩種介面的差別在於 NDIS 2.0.1 為 16 位元，運作在真實模式下，而 NDIS 3.0 則是 32 位元，運作在保護模式下，也許你還沒有啥感覺，一句話來說，就是效率的差別，32 位元的東西運作在 32 位元的 CPU 上，當然是輕鬆許多，當然為了相容性，WFW 3.11 還是讓您可以使用 NDIS 2.0.1 或是 ODI（也是一種網路驅動程式介面）所撰寫出來的驅動程式，當然老話一句，效率稍差，至於這些介面的詳細細節，容我在下一個章節再作詳細的介紹，圖三為一個 WFW

3.11 的網路架構，VREDIR.386、VSERVER.386 為網路導向器用來辨認是否存地本地資源或是網路資源用、VNETBIOS 則是協定層介面、NETBEUI.386 則是協定堆疊、NDIS.386 為協定堆疊以及網路卡驅動程式的橋樑。

Windows 95



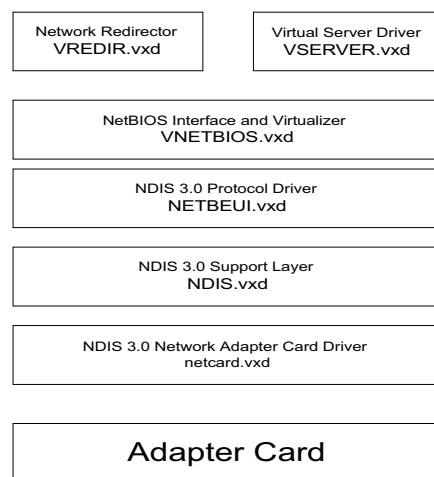
圖二 Windows 增強模式下的架構



圖三使用 NDIS 3.0 為網路驅動程式介面的 WFW 3.11 網路架構圖

前身稱做 Chicago 的 Windows 95，除了從 16 位元跳至 32 位元之外，支援 PnP，內建網路功能，新的使用者介面環境，種種嶄新的一面，的確讓人感受到 Microsoft 希望 Windows 95 能夠取代 Windows 3.x。有沒有取代，可能因人而異，在這裡比較重要的，還是要告訴你 Windows 95 的內建網路功能，到底長什麼樣子。

在網路架構方面，Windows 95 可以架在 Client/Server 的環境中，以 Client 端之角色與 Novell Netware Server，Windows NT Advanced Server，Banyan vines，SunSoft NFS，FTP NFS 等 Server 端相連接，亦可以 peer-to-peer 的方式與 Windows NT，Windows 95 等 Windows 家族系列產品相連接。本身支援數種通訊協定，較為流行的有 TCP/IP，IPX/SPX，NETBIOS 等。這些協定堆疊與前面 WFW 3.11 所述的相同都是根據 NDIS 3.0 所撰寫的。其網路架構圖可參考圖四。



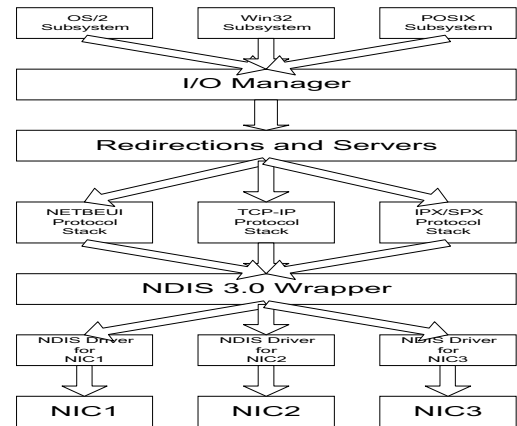
圖四 Windows 95的基本網路架構圖

Windows NT

由於擺脫 16 位元的束縛，Windows NT 比起其他 Windows 系列的作業系統顯得強悍許多。頂著跨平台的可攜性，支援多個處理器的硬體架構，再加上 C2 等級的安全性，高效率以及強健的磁碟及檔案系統，另外還可處理分散式型態的計算環境，這麼多的優點，也難怪工作站等級的 UNIX 系統也不得不注意這後起之秀。

在網路架構方面，Windows NT 可以作為 Novell NetWare Server、Microsoft LAN Manager Server、IBM LAN Server、Banyan vines、Sun NFS 等的 client 端，亦可以與其他 Windows NT 或是 Windows for Workgroups 以及 Windows 95 等 Windows 家族系列產品以 peer-to-peer 的方式互相溝通，當然最重要的，也是 Microsoft 推出 NT 的重點目標，當作 Server 端產品使用。Windows NT 它本身支援了三種最流行的協定堆疊，TCP/IP，NetBEUI，IPX/SPX，這些協定與網路卡驅動程式都是依循 NDIS 3.0 的介面標準所撰寫的。

當你載入協定驅動程式以及網路卡驅動程式時，它會告訴 Windows NT 和 NDIS 3.0 函式庫關於自己的一些進入點（entry points），以後 Windows NT 需要驅動程式傳遞訊息時，就是靠這些進入點來呼叫驅動程式已達成訊息的傳送。圖五是 Windows NT 的網路架構圖。

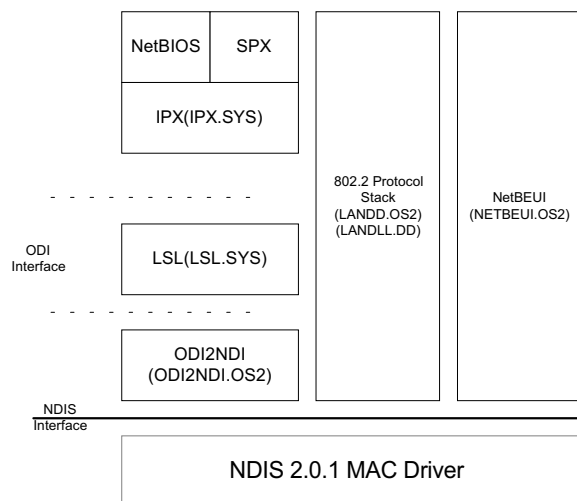


圖五 · Windows NT 網路架構圖

OS/2 2.x

OS/2 是 1987 年，IBM 以及 Microsoft 所推出的一套作業系統，其當初的設定目標是要取代 DOS 在 PC 上的地位。當然，這目標從來也沒實現過，也因為如此，再加上 Microsoft 獨自發展的 Windows 3.0 在市場上告捷，1991 年，Microsoft 宣告放棄發展 OS/2，正式與 IBM 分道揚鑣，OS/2 遂由 IBM 獨立發展。這歷史背後的複雜情節，並不能掩蓋 OS/2 設計的基本觀念以及技術層面的光芒，作為一個多工環境，Server 等級的系統確是一個不錯的選擇。它在網路架構上可以作為 Microsoft LAN Manager、IBM LAN Server、DEC Pathworks

、Novell NetWare、UNIX 等的 client 端，它亦有 Microsoft LAN Manager、IBM LAN Server、Novell NetWare 等 server 端程式可讓其當作 server 來使用。由於歷史的因素，OS/2 是以 NDIS 2.0.1 作為其網路驅動程式介面，不過 1991 年 Novell 與 IBM 的合作，它亦提供了 ODI 網路驅動程式介面的架構，能讓 Novell NetWare 在 OS/2 上執行。無論是哪一種，當你混合著使用時，譬如，上層協定堆疊是使用 ODI 介面所撰寫的，而下層的網路卡驅動程式則是根據 NDIS 2.0.1 介面所撰寫的，你可能就需要一些轉換這些標準的程式，像是 ODI2NDI.OS2 之類的。不過你不用擔心，這轉換程式是 IBM 以及 Novell 的事。圖六左半邊為 OS/2 作為 NetWare Server 的 Client 端，使用 NDIS 2.0.1 網路卡驅動程式的架構圖，右半邊則是使用其他種協定堆疊之架構圖。



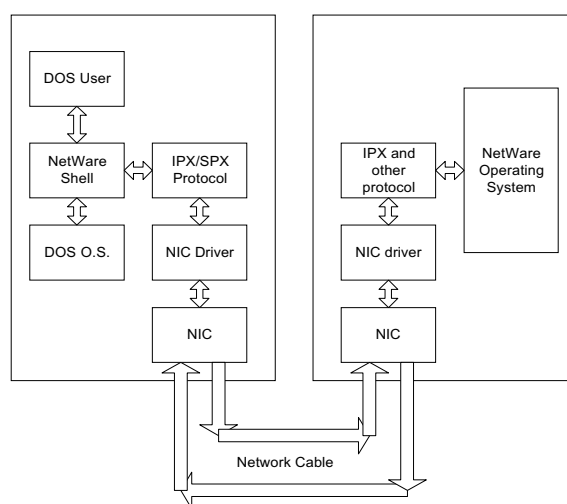
圖六 · OS/2 使用 NDIS 2.0.1 網路卡驅動程式之架構圖。

Novell NetWare

在 PC 區域網路的市場，不談 Novell NetWare，你大概會被歸類為井底之蛙了。雖然 Microsoft、IBM 虎視眈眈，想要瓜分這塊大肉餅，不過 Novell NetWare 在目前這個市場上仍居於龍頭老大的地位。Novell NetWare 是架構在 client/server 的網路環境下，你需要一台機器執行 NetWare 作業系統，然後其他的機器則執行 client 端軟體，作為連上 NetWare Server 之用。PC 市場上的作業系統大部份，如 DOS 以及 Windows 3.x，Apple Macintosh，OS/2 1.x 以及 2.x，Windows 95，Windows NT 等，都有 client 端軟體來供已存在的區域網路市場使用。

Novell NetWare 它提供了許多 Server 端的資源供 client 端使用，如檔案資料庫，列表機，數據機等。如前面所提，它是以 IPX/SPX 協定堆疊作為溝通之用，而這些堆疊以及網路卡驅動程式都是遵循 ODI 介面所發展的，透過這個介面讓網路卡驅動程式的撰寫以及協定堆疊分開來。使得程式設計師的工作輕鬆許多。圖七為 Novell NetWare 的 Client/Server 環境圖。

圖七·Novell NetWare的Client/Server 網路架構圖



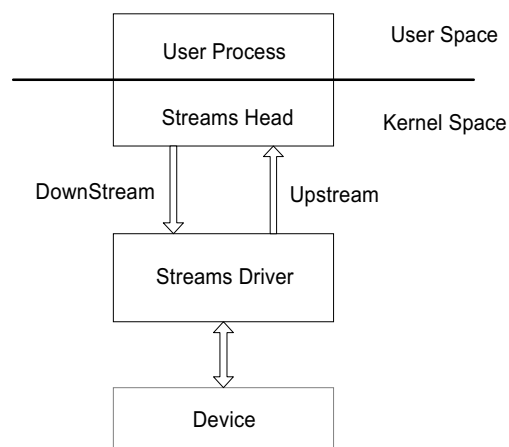
UNIX

UNIX 系統的發展可以追溯到 1970 年代，是由 AT&T Bell 實驗室的兩位工程師 Dennis Ritchie 以及 Ken Thompson 所發展出來的，伴隨其所發展出來的 C 語言，可說是 UNIX 系統下標準的程式語言。至今有許多版本的 UNIX，如 Sun Microsystems 的 SUN OS 以及 Solaris、Hewlett Packard 的 HP UX、IBM 的 IBM AIX、Digital 的 Digital Ultrix 等。不過基本的精神概念，學術殿堂的理論應用，都可以在其上找到影子，甚至更為強大。UNIX 的網路架構是以 client/server 的型態所建立的。而其上的堆疊型態，由於各方巧合的因素，首推 TCP-IP，另外還有 XNA，SNA，NetBIOS 等。

UNIX 上的 API 形式有三種：Sockets

，TLI，以及 Streams。前兩種主要用在應用程式的撰寫上面，而第三種則主要用來讓程式設計師撰寫協定堆疊之用，這種機制是建立一個全雙工的 I/O 管道(pipe)，讓使用者行程(Process)與 Streams 型態的驅動程式溝通之用，當使用者行程利用系統呼叫開啓裝置，如，open()，系統會建立一個 streams 型態的連結將使用者行程與裝置連結起來，請參看圖八。

圖八，UNIX streams架構圖

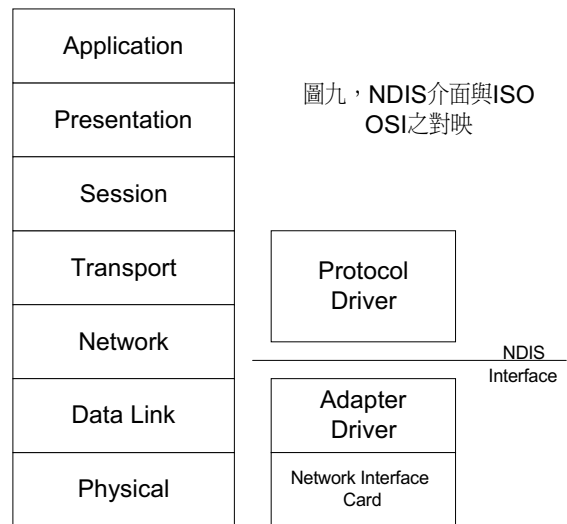


網路驅動介面標準

雖然 ISO 的 OSI Model 在 1983 年就公諸於世，不過在 1980 年代末期，問世的網路產品除了需要控制最低層的硬體-網路卡，還要有跟高層的應用程式溝通的能力。每家廠商有自己的網路協定，從網路卡驅動程式，到中間的協定堆疊，乃至於上層的應用軟體，無不唇齒相依，環環相扣。這也造就了開放架構下的 TCP-IP socket 上的應用軟體蔚為風潮，彷彿談到網路程式設計就不能不提及 TCP-IP，程式設計師只消呼叫幾個函式，一切下層的動作都不必去操心了。但是在低層方面呢？網路卡的製造商可沒那麼幸運有這麼一套規範讓大家去立正站好。為了與各個流行的溝通協定相容，常常一張卡有許多種驅動程式。使用者或許沒感覺，但是撰寫網路卡驅動程式的人員可辛苦了。

在 1987 年 FTP Software,Inc.這家公司率先定了一個 packet driver 介面，這是第一個在 DOS 下讓驅動程式撰寫員可遵循的標準，當然軟體界老大 Microsoft 也不甘落人後與 3COM 公司在 1988 年 5 月發表了 NDIS(Network Driver Interface Standard)。NDIS 定義了一個標準介面，介於 MAC 層(Medium Access Control Layer)以及上層協定堆疊之間。以 OSI Model 來說，就是在第二層以及第三層之間的一個介面，請參考圖九。

它讓溝通協定與網路卡驅動程式間之資料交換輕鬆許多。目前存在的版本有 1991 年六月發表的 NDIS 2.0.1 以及伴隨 Windows NT3.1 所出現的 NDIS 3.0。基於這兩種驅動程式有些許的差異，我們先將它分開來介紹，再做整體之比較。介紹完 NDIS，我們接著介紹 packer driver 介面，然後介紹 Novell 公司所定的 ODI 介面，以及 UNIX 系統上的 DLPI 介面。



	Packet Driver	NDIS 2.0.1	NDIS 3.0	ODI	DLPI
發表年代	1987	1991	1993	1989	1984
有關組織	FTP	3COM Microsoft	Microsoft	Novell	AT&T
定址	16	16	32	16	16-64
Software entry points	0x60~0x80	Dynamic	Dynamic	Dynamic	Dynamic

表二，各個網路驅動程式之比較

在本表二中的發表年代是根據最早問世的版本年代，而其相關的組織也是指當初制訂標準的主要組織。由圖中可看出除了 Packet Driver 之外，其它介面的 Software entry points（也就是上層軟體要利用此介面所提供之功能時，所需呼叫之處）皆為 Dynamics，也就是說這些介面並不會被固定在某個記憶體位址上，其最大的影響就是作業系統可以充分利用其擁有的資源，這也是多工的基本條件。

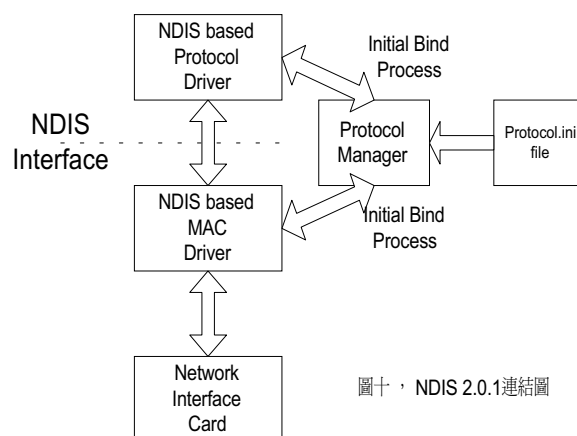
NDIS 2.0.1

遵循 NDIS 2.0.1 這套規格比較常見在 DOS，OS/2 上。會與 NDIS 2.0.1 相關的驅動程式可分成兩種，一個就是媒體存取控制(Medium Access Control, MAC)驅動程式，也就是網路卡驅動程式，另一個當然就是協定堆疊驅動程式(Protocol Driver)。（有些人把

應用程式介面，API，以下的驅動程式，也就是前面所提的兩種驅動程式，統稱為網路驅動程式，容易與網路卡驅動程式相混淆，故以下行文皆稱網路卡驅動程式為 MAC 驅動程式。)

每個驅動程式都有自己的進入點 (entry point)，也就是給別人呼叫，開始使用之處。而協定管理員則是負責協調與管理驅動程式間的連結。有了這基本的認識，你大概已經瞭解七分了，剩下的只是如何讓作業系統知悉這些驅動程式的存在，以及如何互相連結的。

通常每個系統都會要求另外有個說明檔，請參考圖十，譬如在 DOS 下或是 OS/2 要作為 Microsoft Lan Manager 的 client 端，你要有個 protocol.ini 的檔案。協定管理員就是靠這個檔案來知道如何作連結這些驅動程式。至於載入的方法，則是因各個作業系統而有所不同，以 DOS 來說，就是利用 config.sys 檔，或是從命令列中載入。而圖



圖十，NDIS 2.0.1連結圖

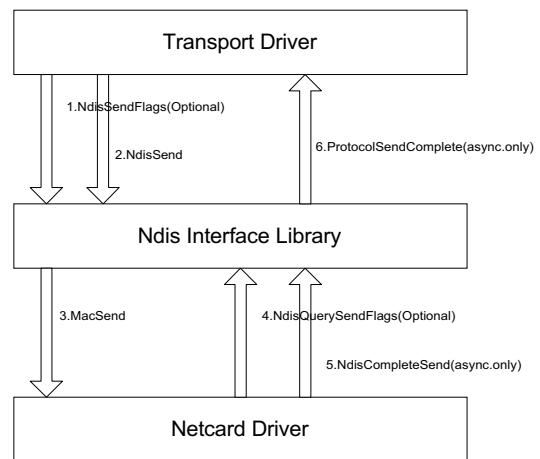
形化介面的作業系統都會有很容易上手的設定程式，你只要按照步驟一步步作就可以將這些驅動程式載入，簡單吧，七分加三分，你已經有了個骨架出來了，如何填滿要不要填滿，就看你自己要不要動手親自操作，瞧瞧了。

NDIS 3.0

NDIS 3.0 是微軟為它的 Windows NT 3.1 作業系統所發展的一套 32 位元的網路驅動程式介面規範。除了 NT 之外，WFW 3.11 以及 Windows 95 的網路架構皆是以 NDIS 3.0 為範本的。基本傳收資料的架構沒變，不過細節製作之相異性不得不讓我們把它拿出來說嘴一番。

首先主要的差異是，NDIS 3.0 的驅動程式是 32 位元以 C 來撰寫的，比起 NDIS 2.0.1 的 16 位元，組語介面的原式，顯然要容易撰寫些。當然有於是用 C 撰寫的，其移植性也提高不少，驅動程式只需更改一小部份就很容易地從 NT 移植到 WFW 3.11 等。

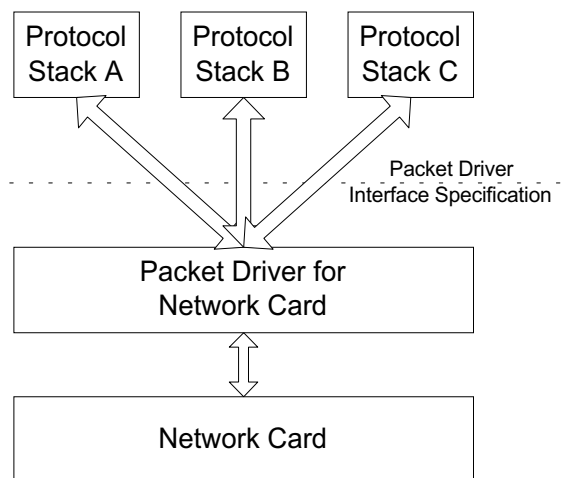
另外，NDIS 3.0 驅動程式是執行在保護模式下且被放置在延伸記憶體上，然而 NDIS 2.0 驅動程式則是在真實模式下執行，並且放置在傳統記憶體上。如同 NDIS 2.0.1 一樣，NDIS 3.0 也有類似協定管理員的程式，不過名稱爲 NDIS Interface Library Wrapper，姑且稱其爲 NDIS 介面函式庫，功能類似 NDIS 2.0.1 的協定管理員，換湯不換藥，名字變了，功能其實沒變。也許你看到這邊，沒錯你已經瞭解他們的功能是什麼，但是可能對實際封包收送的流程可能還有些狐疑，我們接下來就讓你看看實際收送的樣子是什麼模樣，請參考圖十一，這是在 NDIS 3.0 下收送封包的流程，原來不過只是呼叫幾個函式而已。



圖九，NDIS 3.0 架構下傳送資料

Packer Driver

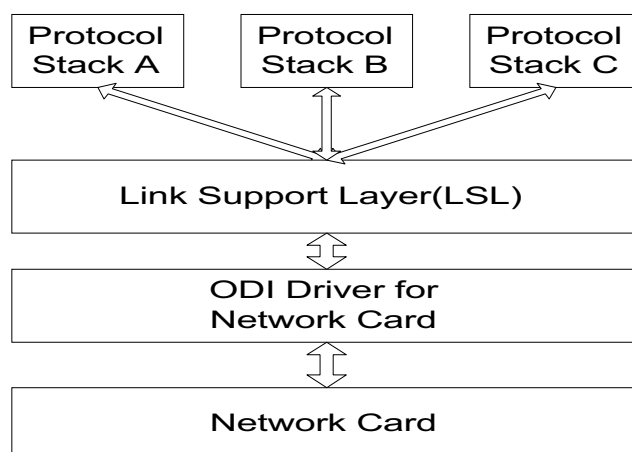
packer driver 規格 (packet driver specification) 是由 FTP Software, Inc. 制訂的，目前已發展到 version 1.11，請參考圖十二。其作用也是與 NDIS 一樣，將網路卡驅動程式的撰寫以及協定堆疊程式的撰寫分開來。這個規格通常是用在 DOS，與 TCP/IP 的環境下使用。由於他主要用在 DOS 下，所以它的連結方式就是使用一些軟體中斷來溝通，詳細的方法前面提過在此就不贅述，如果你寫過組合語言程式大概你已經知道我在說些什麼了。



圖十二，Packet Driver 架構圖

ODI

Novell 的網路架構是採用自己發展的 Open Data-Link Interface(ODI)介面。其功能與上所述的沒啥不同。其主要的組成成分為 Multiple Link Interface Driver(MLID)，Link Support Layer (LSL)，Protocol Stack。哇，挺複雜的樣子，其實說穿了沒有什麼，MLID 就是根據 ODI 介面所寫的網路卡驅動程式。而 LSL 就好像是 NDIS

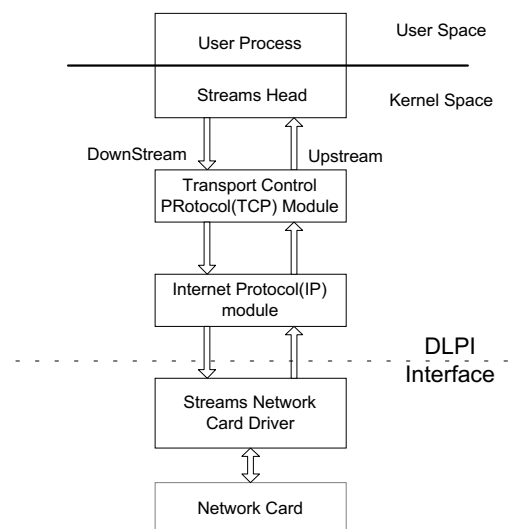


圖十三，ODI介面架構圖

2.0.1 介面中的協定管理員 (Protocol Manager) 或是 NDIS 3.0 裡頭的 NDIS 介面函式庫。至於協定堆疊，顧名思義就是 IPX/SPX，TCP/IP 之類的東東囉，請參考圖十三。

DLPI

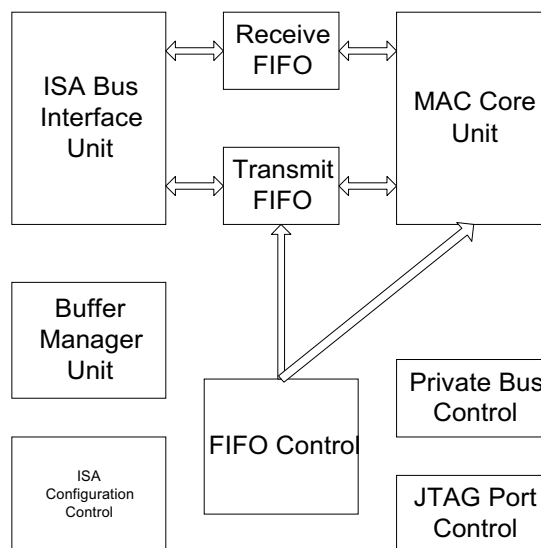
Data Link Provide Interface(DLPI)則是 UNIX 上與前所述的各個介面類似的東西，你可以參考圖十四。DLPI 只描述了這個介面應該提供些什麼東西，實際的製作則是由各家廠商的 UNIX 所內含的。各家廠商的 UNIX 提供的 DLPI 介面名稱可能或許有些不同，譬如，SCO UNIX 則是叫做 LLI (Logical Link Interface)，不過通常都是會提供一些原式(primitive)，你可以利用這些原式發展你自己的協定堆疊，發展你自己的網路卡驅動程式，然後經由編譯後再與作業系統連結起來，成爲核心的一部份，最上層的應用程式就可以使用一些像是 open()，write()，read()等的 system call 來使用你的網路卡囉。



圖十四，DLPI介面架構圖

硬體概念介紹

談了這麼多有關網路作業系統，驅動程式介面的介紹，但是你要傳送資料，仍然是要靠網路卡來完成，網路卡有些什麼東西？當然簡單想想應該就可以猜得出來，主要的成分不外乎是 RAM，作為存放 FIFO 的資料，ROM，記住一些硬體的設定，在主機板上作 DMA 的控制晶片，MAC Core Unit 處理有關各個網路型態的動作，也就是作 MAC 工作的晶片，譬如，CSMA/CD(802.3)，Token Ring，Token Bus 等。當然還有一些線路及晶片，作為管裡記憶體 buffer，Bus，Port，Error Checking 等之用，請參考圖十五。



圖十五，PCnet-ISA 組塊圖

有了這些的概念我們來談談收送資料。我們可以用系統的記憶體來映射到網路卡上的記憶體，但是這些記憶體網路卡要用，網路卡驅動程式也要用，那怎麼辦呢？所以就有個所有權的概念出來，網路卡會判斷那個記憶體 buffer 屬於它的，當有資料來時，它就利用 DMA 將資料放在這個屬於它自己的記憶體 buffer，然後將所有權換成網路卡驅動程式，接著透過中斷告訴作業系統，網路卡驅動程式收到訊息後，就去讀取資料，然後再轉換所有權，代表此記憶體 buffer 可讓網路卡再度使用。你看這所有權的概念是否就像區域網路 token ring 中的 token 呢？

如何撰寫一個網路驅動程式

終於來到你可能感覺最實際的地方了，我不會教你什麼指令如何使用，什麼函式怎麼運用，但是我會告訴你你如何踏出這第一步。首先，你要確認你的網路驅動程式是要運作在什麼作業系統環境下，可能是 DOS，可能是 95，或是 UNIX 等，接著你要決定你要使用哪一種介面，packet driver，NDIS，ODI 等。知道方向之後，當然就是去找尋

這些介面提供了什麼函式你可以運用，packet driver 你可以在 Ftp 公司的 web site 找到（<ftp://www.ftp.com/support/pub/packetd>）。如果是 NDIS 你則需要購買 Microsoft 所出的 MSDN，這裡面有一個 DDK 的東西，全名是 Device Driver Kit，看到名字你大概就猜得出來裡面有些什麼玩意了，裡面有各種硬體驅動程式的撰寫方法，函式使用方法，以及一些範例程式，當然你還必須去購買編譯器，組譯器之類的東西。而 ODI 介面你可以洽詢 Novell 公司，他們也有自己特別的組譯器，可以產生特殊的程式碼，與 Netware 作有效率的溝通，而 DLPI 呢，你則可以詢問你的作業系統經銷商，相信能得到滿意的答覆。

範例程式

不要嚇到了，這裡怎麼可能容納得下那印出來可能有上百頁的程式碼呢，只是要告訴你程式裡面不外乎包含這種東西，初始化的函式，將硬體以及一些軟體物件作初始化，收送資料的函式，I/O 控制的函式之類的，中斷來時一些處理動作之函式。就這麼簡單，你怎麼還不敢快動手呢？

參考書籍:

- 1.Dhawan,Sanjay,1995,Networking Device Drivers, New York :Van Nostrand Reinhold.NY.
- 2.Norton,Danial A, 1992, Writing Windows Device Drivers,Addison-Wesley Publishing company, Inc., MA.
- 3.Custer,Helen,1992,Inside Windows NT,Microsoft Press.
- 4.Romkey,John,1994,Packet Driver Specification,rev.1.11.FTP Software 2 High Street North Andover,MA.
- 5.Malamud,Carl,1990.“Analyzing Novell Networks” ,Van Nostrand Reinholk,New York,NY.
- 6.Triebel,Malter,The 80386DX Microprocessor,Hardware,Software,and Interface.
,ISBN 0-13-249566-X

附註：本文作者 楊東成爲交大資科系研究生，林盈達爲該系副教授 (email:ydlin@cis.nctu.edu.tw)