

第十章 Internet 系統模型

10-1 Internet 系統簡介

由前面幾章的介紹，我們大略可以了解整個 Internet 網路原理，以及所需的基本通訊協定。自本章開始我們將介紹一連串有關 Internet 網路的應用環境，以及相關的通訊協定。在整個 Internet 網路環境之下，也許包含許多不同型態的網路，這些網路所使用的連結技術皆由傳輸層以下的通訊協定所達成，目前我們所探討的問題是網路連結之後，如何利用網路達成所需的服務，這也就是『**應用層**』(**Application Layer**) 所提供的功能。

基本上，Internet 網路在傳輸層以後並沒有提供標準通訊協定（如，交談層或表現層）來銜接各種應用軟體，每一應用軟體都依照自己需要發展各自的協定，因此就有各種不同的『**應用系統**』(**Application System**) 產生，譬如，檔案伺服系統 (FTP)、郵件系統 (SMTP) 等等，它們都有各自的通訊協定，但它們之間也有許多共通點，也就是本章探討的重點，讓我們對 Internet 應用系統有一個概括的認識。

10-2 Internet 系統模型

由網路服務的觀點來看，Internet 應用系統是建立在兩個（或兩個以上）通訊端點上，只要連結上這個通訊端點，且兩者之間給予適當的協議，雙方便能依照服務內容來運作，它的系統模型就如圖 10-1 所示。圖 10-1 (a) 為網路系統模型，經過下層通訊軟體連接之後，所建構的應用模型如圖 10-1 (b) 所示，因此我們便可依照圖 10-1 (b) 來探討應用系統的運作模式。在應用系統模型之中，為了讓雙方都能正常運作，必需包含下列功能：

- ※ 通訊端點定址
- ※ 多工服務端點
- ※ 通訊語法

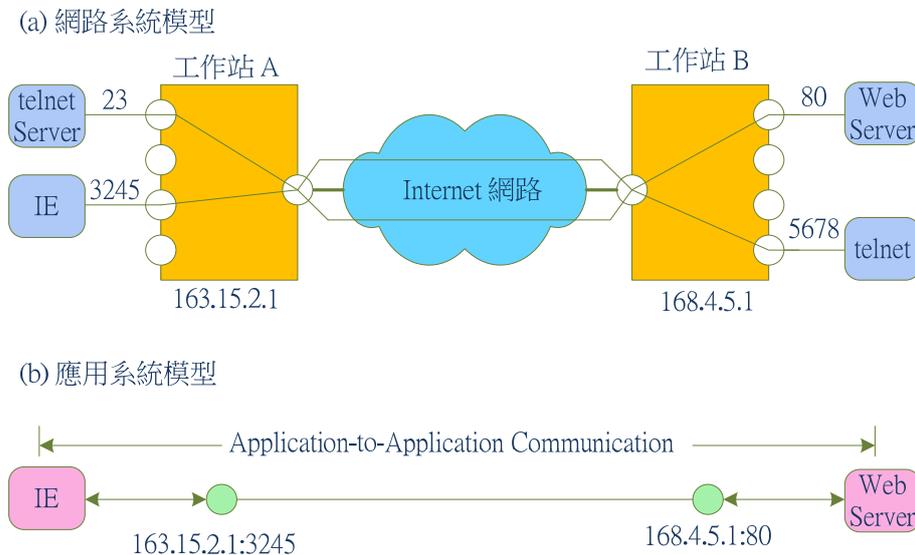


圖 10-1 Internet 應用系統模型

以下針對這三個功能來介紹。

10-2-1 通訊端點定址 – Domain Name

無論 Internet 或 Intranet 網路所訴求的重點是透過遠距離之間的傳輸，而其中也許會經由多種網路型態連結，連線雙方也許是不同網路架構，因此連線雙方必須有一個**傳輸獨立 (Transport Independent)** 的介面來互相通訊。依照此標準介面便可通行無阻的互相通訊，而不用考慮所經過的網路型態，此介面是由 TCP/UDP 層所提供，通訊雙方的連線方式如圖 10-1 (b) 所示。

我們將各種特殊的服務程式連結到傳輸埠口，任一埠口提供某一特殊服務，該埠口就稱之為『**服務埠口**』，在 Internet 網路上我們只要連結到該埠口，就可以得到某一特殊的服務。基本上，我們將服務埠口定名為『**IP 位址：傳輸埠口**』(如 163.15.2.1:80)，IP 位址為主機電腦在 Internet 網路上的唯一位址名稱 (假設是唯一的)，而傳輸埠口為該主機上 TCP/UDP 的連接埠口。此定名方式就如同電話號碼 (IP 號碼) 裡再連接分機號碼 (TCP/UDP 埠口) 一樣，在電話系統上，如能知道電話號碼和其分機號碼，在世界任何一個角落裡，都能撥接到其所需服務的地點，Internet 網路也如同此道理。

想要在任何一部主機電腦的傳輸埠口上連接某一個應用程式 (如 Web Server) 時，都而要公告週知，讓網路上所有使用者知道，的確不是一件容易的事，但最起碼我們可以將一些較常用的應用程式固定在某一埠口上，使用者只要知道該主機電腦的 IP 位址，便可以連結到一般常用的應用程式，譬如，將 Web Server 固定在埠口 80、Telnet Server 在埠口 23、ftp Server 在埠口 21 上。

一般使用者就不需要特別去記憶，哪些應用程式是連接到第幾埠口，這些埠口就稱之為『著名埠口』(Well-known Port)，一般常用的著名埠口如表 7-1 所示 (或 Linux 系統的 /etc/services 檔案)。

雖然有了著名埠口可幫助使用者免去記憶傳輸埠口的困擾，但使用者還是需要知道某一主機的 IP 位址，才能連結到該服務，可是要記憶某一主機的 IP 號碼也是非常的困難，因此 Internet Society 提出『網域名稱』(Domain Name) 的架構。簡單的說，網域名稱是利用有結構性的命名方式來取代 IP 位址，讓使用者記憶有意義的名稱總是比沒有意思的數字來得容易。譬如：

linux-1.cu.edu.tw ↔ 162.15.2.62

linux-2.cu.edu.tw ↔ 163.15.2.30

當使用者以網域名稱來連接伺服器時，是由誰來負責將網域名稱轉換為 IP 位址呢？在 Internet 網路上有一套完整的『網域名稱系統』(Domain Name System, DNS) 來負責。基本上 DNS 也是屬於分散式處理環境，任一區域的 DNS 只負責該區域內名稱的查詢動作，如果超越管轄地區，便轉送到相關區域的 DNS 系統查詢。本書第十五章有針對 DNS 系統詳加介紹，這裡僅用圖 10-2 來說明 DNS 的查詢動作。圖 10-2 中，工作站 A 執行 telnet linux-1.cu.edu.tw 命令後，假設工作站內未紀錄 linux-1 主機的 IP 位址，便發出訊號向管轄內的 DNS 伺服器查詢。伺服器回應 linux-1 的 IP 位址後，工作站 A 才確實執行 telnet 163.15.2.62 命令，連結到 163.15.2.62，又當主機執行 telnet 命令，Telnet 伺服器的著名埠口是 TCP 23，因此，所指定到的通訊端點是 163.15.2.62:23。

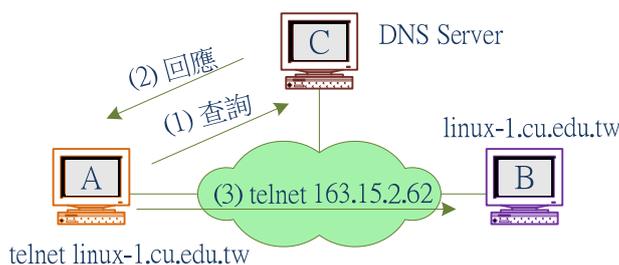


圖 10-2 網域名稱查詢動作

DNS 的命名方式是用樹狀結構，由根目錄 (root) 『 . 』 往下延伸，第二層表示組織單位 (或區域)，第三層以下再由各組織單位往下延伸，如圖 10-3 所示。早期 DNS 系統是由美國發展出來，因此，第二層的組織單位是以美國地區為主，各組織代碼的功能如表 10-1 所示。各國的國碼 (Country Code) 也架設在第二層上，譬如，『 tw 』 代表台灣、『 jp 』 代表日本、『 ca 』 代表加拿大等等。各國的組織單位也是由各自國家代碼以下 (第三層) 來延伸。任何一個代碼表示一個網域，

譬如台灣教育部的網域是『edu.tw』，它的書寫方法是倒過來的方式，由最下層往上層來書寫，其中包含的網域是『edu』、『tw』、『.』，但一般書寫都將根的網域(『.』)簡略掉，而書寫成『edu.tw』(一般電腦都會自動補上『.』)。

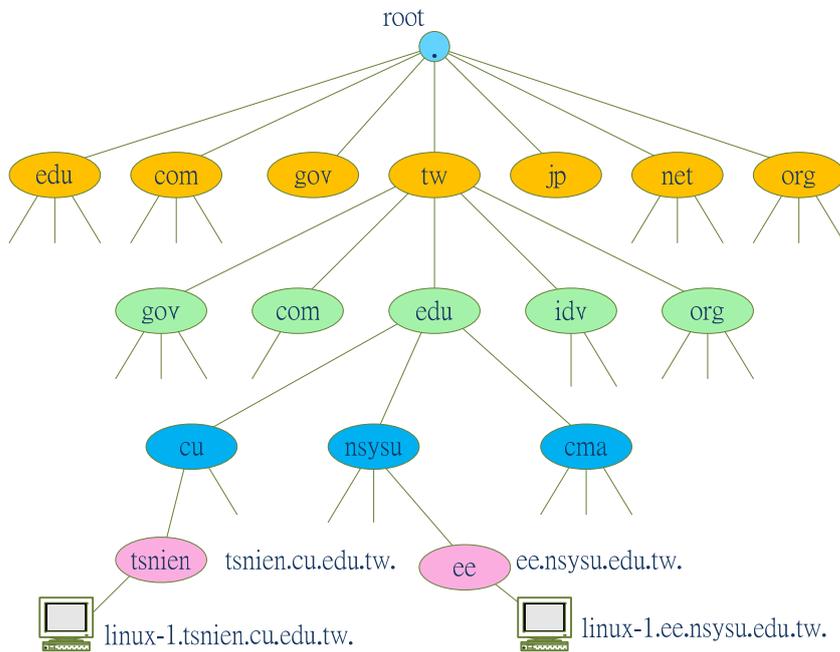


圖 10-3 樹狀的 DNS 系統

既然 DNS 是屬於分散式系統，因此在某一網域下是否再發展出下一層的次網域，是由該網域的 DNS 系統來負責，這種管理工作將會在第十五章介紹。如圖 10-3，我們在 edu.tw 網域下再設定次網域為 cu.edu.tw、nsysu.cu.edu.tw，又在這兩個網域下的次網域上都有一個名為 linux-1 的主機，但這兩個主機的網域名稱分別為：linux-1.tsnien.cu.edu.tw 和 linux-1.ee.nsysu.edu.tw。

表 10-1 各區域之網域名稱 (部分)

區域名稱	代表組織單位
com	商業組織 (Commercial Organizations)
edu	教育單位 (Education Institutions)
gov	政府機關 (Government Institutions)
mil	軍方單位 (Military Group)
net	網路支援中心 (Network Support Centers)
org	非營利事業組織 (Organizations)
au (aus)	Australia (澳大利亞)

ca (can)	加拿大 (Canada)
cn (chn)	中國 (China)
fr (fra)	法國 (France)
hk (hkg)	香港 (Hong Kong)
it (ita)	義大利 (Italy)
jp (jpn)	日本 (Japan)
kr (kor)	南韓 (Korea)
tw (twn)	台灣 (Taiwan)

10-2-2 多工服務端點 – Multiplex

依照我們所瞭解，網路應用服務是架設 (或銜接) 在主機電腦的某一傳輸埠口上，當這個傳輸埠口被某一客戶連接時，其它客戶端是否可同時要求連線呢？一般網路伺服器都必須允許同時連接多個客戶的請求，否則便失去伺服器的功能，但應如何來達到多工連線？其實它的動作非常簡單，當伺服器接受到連線請求後，系統便再產生一個次處理程式 (Sub-process) 利用 fork() 系統呼叫，而將連線訊息轉移給次處理程式，並由次處理程式負責雙方的通訊，原來主程式又回到等待連線的狀態下，如此便可以達到多工的功能，我們用圖 10-3 來說明傳輸埠口的多工處理。譬如，網路上有一個 Telnet 伺服器位於 168.13.4.5:23，當 Uase_A (163.15.2.30:1456) 要求連線，伺服器接受連線後，便執行 fork() 功能呼叫產生 P1 處理程式，也將連線訊息轉移到 P1 上，便啟動 P1 和 User_A 之間通訊，它們之間的通訊連線是 168.13.4.5:23 對 163.15.2.30:1456；同樣的，User_B 要求連線後，它們之間連線是 168.13.4.5:23 對 163.15.2.62:2378，雖然所有連線還是經過同一傳輸埠口，但是每個連線的號碼都不會一樣，因此還是可以區分出來的。(有關多工連線的實現，請參考第十章說明)

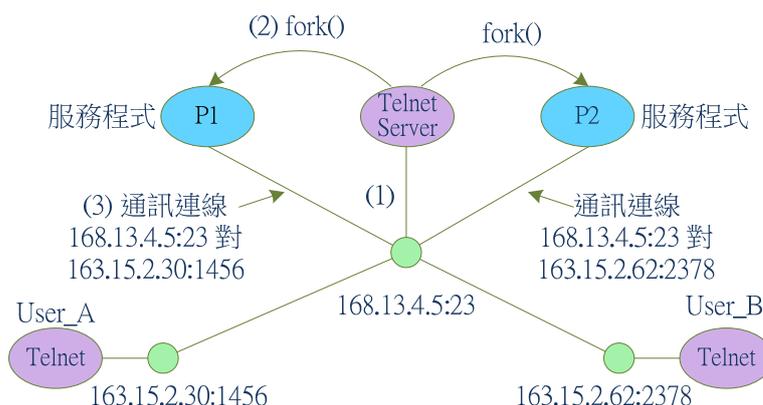


圖 10-4 通訊端點之多工連線

10-2-3 通訊語法 – NVT ASCII

Internet 網路為了要包容各種異質性電腦間互相通訊的需求，通訊語法採用最通用『網路虛擬終端機』(Network Virtual Terminal, NVT) 語法，將兩個通訊端點上的通訊行為，模仿成像終端機對主機之間的通訊一樣，這樣比較容易避免掉不同電腦之間語意相容性的問題。早期 Internet 尚未普遍時，NVT 是由 RFC 854 定義在 Telnet 通訊協定上使用，它的做法是無論客戶端的終端機是何種型態，都將其映射到 NVT 模式，同樣的，也將伺服器的終端機映射到 NVT 模式，不論雙方電腦系統的終端機是何種型態，雙方都可以依照 NVT 模式來通訊，後來發展的應用系統大多也採用這種模式。基本上，NVT 是一個有鍵盤與印表機的字元裝置，使用者由鍵盤上鍵入字元，並將該字元以 ASCII 碼傳送給伺服器，也有可能伺服器以 ASCII 碼傳送印表機，印表機再印出該字元的圖樣(如鍵盤上 "A" 的圖樣)，因此它們之間是以最簡單的 ASCII 碼來傳遞，因此稱之為『NVT ASCII』。但在 Internet 上傳輸會有稍微改變，這些變化也只侷限於控制字元部份而已，基本上還是以 NVT ASCII 模式傳輸。

NVT ASCII 是採用 7 位元的英文字母表示法，每一個 7 位元字元都以一個 8 位元(一個位元組)來表示，而將其最高位元設定為 0。如此，有關應用程式的控制字元就可設計在 128 數字(0x7F)以後的字碼，也演變成不同應用系統的通訊協定。但各種控制字元的編碼也必須避開有些 8 位元的編碼字型，譬如，中文字型的圖形文字編碼等等。因此，各種應用系統的通訊協定大多製作在 0x7F 以後的控制碼，其中較接近於最後(0xFF)部份，較不易和其它編碼衝突。另外有關 NVT 的換行控制碼，則一律以 <CR>(Carriage Return, 0x0D)和 <LF>(Line Feed, 0x0A)兩個控制碼組合而成，可寫成 \r\n 來表示。而原來單純的 CR (Carriage Return)，則以 <CR>和 <NUL>(NULL, 0x00)組合而成，也可以寫成 \r\0 來表示。

我們用 Telnet 協定來說明 NVT ASCII 的動作程序，Telnet 是以 IAC(Interpret As Command)(字碼為 0xFF)來表示控制命令的開始和結束，而以 SE(Sub-option End)表示命令選項的結束，而其中包含的命令則用 NVT ASCII 表示。我們用圖 10-4 來說明 Telnet 如何利用 NTV ASCII 來協調雙方終端機型態，首先，客戶端送出 <IAC, WILL, 24> 給伺服器，如將其編碼成 NTV ASCII 為 <0xFF, 0xFB, 24>，其中 24 表示協調終端機的選項，而 0xFB 是 WILL 的字碼(請參考第十三章 Telnet 介紹)。伺服器收到後回應 <IAC, DO, 24> 表示同意協商終端機型態給客戶端，並通知客戶端送終端機型態 <IAC, SB, 24, 1, IAC, SE>，客戶端再送出終端機型態給伺服器 <IAC, SB,

24, 0, 'V', 'T', '2', '2', '0', IAC, SE>，其中 VT220 也是 NTV ASCII 編碼表示為 0x56、0x54、0x32、0x32、0x30。

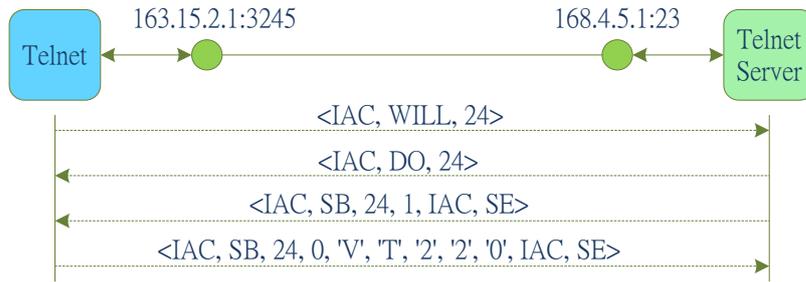


圖 10-5 NTV ASCII 控制程序範例

既然，我們將應用系統中兩端通訊端點都模擬成『網路虛擬終端機』(NVT)，這表示各種應用系統的通訊端點都可以用 Telnet 命令連結到它，並可用鍵盤輸入 (ASCII) 各種命令，以下我們用 Telnet 連結到 163.15.2.30 主機內 FTP 伺服器 (Port 21) 的通訊端點，並用鍵盤輸入各種命令，來觀察它的反應：(其中有底線部份表示使用者輸入控制字元)

```
[tsnien@linux-1 tsnien]$ telnet 163.15.2.30 21
Trying 163.15.2.30...
Connected to 163.15.2.30.
Escape character is '^]'.
220 linux-2.cu.edu.tw FTP server (Version wu-2.6.1-18) ready.
user tsnien
331 Password required for tsnien.
pass good478
230 User tsnien logged in.
help
214-The following commands are recognized.
USER      PORT      STOR      RNT0      NLST      MKD       CDUP
PASS      PASV     APPE     ABOR     SITE     XMKD     XCUP
TYPE     DELE     SYST     RMD      STOU
STRU     ALLO     CWD      STAT     XRMD     SIZE
MODE     REST     XCWD     HELP     PWD      MDTM
```

```
QUIT    RETR    RNFR    LIST    NOOP    XPWD
214 Direct comments to root@localhost.
pasv
227 Entering Passive Mode (163,15,2,30,223,54)
port 163,15,2,62,4,4
200 PORT command successful.
```

由以上的介紹，我們可以瞭解 Internet 應用系統發展的三大基礎：通訊端點的定址法、通訊埠口的多工連線、以及 NVT ASCII 的控制字元。雖然各種應用系統都有各自的通訊協定，也大多是依照這三個基礎來發展，這對我們學習 Internet 網路應用有很大的幫助。

10-3 主從式架構

一般 Internet 應用系統都是將某一服務，建構在傳輸埠口上讓一般使用者使用，該服務程式就稱之為『伺服器』(**Server**)，而使用該服務者也稱之為『客戶』(**Client**)。客戶端和伺服器端之間必須透過某一傳輸通訊協定，來互相交換訊息，如，HTTP 或 FTP 通訊協定，對整個應用系統的架構就稱之為『主從式架構』(**Client/Server Architecture**)。至於客戶端必須具有下列特性：

- 客戶端由伺服器端取得所欲完成的訊息，執行還是在本身工作站上。也就是說，伺服器只存放執行程式，而客戶端是將執行程式下載到本地電腦上執行，才能得到所要的結果。
- 程式是否下載執行，完全由客戶端使用者決定，並且依次只能執行一個交談訊息 (**Session**)。
- 由客戶端主動向伺服器端要求連線。
- 客戶端可以同時連接多個伺服器，但同時只能和一個伺服器交談。
- 只要合乎雙方通訊協定，不需要有特殊硬體設備或作業系統。

相同的，伺服器端必須具備有下列性：

- 能提供某一特殊服務 (如，列印功能)，並能同時接受處理多個客戶端的要求連線。
- 該服務必須能自動執行，無論客戶端何時請求，都能即時提供應有的服務。
- 一般執行於多重處理 (**Multi-tasking**) 以上的主機電腦。

- 隨時等待客戶端的連線要求。
- 能接受各種客戶端連線要求，但只能提供某一特殊服務。

基本上，一般所稱的伺服器為軟體設備，但有些硬體製造商宣稱其主機電腦為伺服器，這是表示該主機具有足夠的硬體能力來安裝伺服器軟體，譬如，它的 CPU 執行速度夠快、或記憶體容量夠高、或甚至主機電腦的穩定度足以承擔伺服器的工作，因而宣稱該主機為伺服器。但話又說回來，伺服器端之工作站必須隨時等待客戶端的要求，又可能同時接受多個客戶服務，因此，伺服器主機電腦的工作負荷也相對非常的重，確實需要特殊功能的主機來服務。

客戶端和伺服器端之間的通訊行為，大多採用『要求/回應』(Request-and-Response) 的作業模式，也就是說，客戶端向伺服器端提出某種『要求』(Request) 服務，伺服器再依照客戶端的要求『回應』(Response) 所需的訊息。基本上，伺服器並不主動發送任何訊息給客戶端。例如，客戶端向檔案伺服器要求傳送檔案，伺服器接收到檔案名稱，再檢視各種權限制後，再將檔案傳送給客戶端，完成所需的服務。

10-3-1 單一主機多重服務

既然伺服器是以應用程式為主，而並非以主機電腦為觀點，因此，在同一主機上可安裝多個伺服器軟體，提供多個伺服器。例如，在一部主機電腦上可安裝 HTTP Server、DNS Server、FTP Server 之伺服器軟體，該主機便能提供 Web 伺服器、名稱伺服器和檔案傳輸伺服器之功能，如圖 10-5 所示。一般較小的企業，客戶端的要求不是很多的話，這種安裝較節省費用，但如果客戶端要求量過多時，整個傳輸效率會降得很厲害，基本上建議將較常用的伺服器安裝於特殊的主機電腦上效率較高。

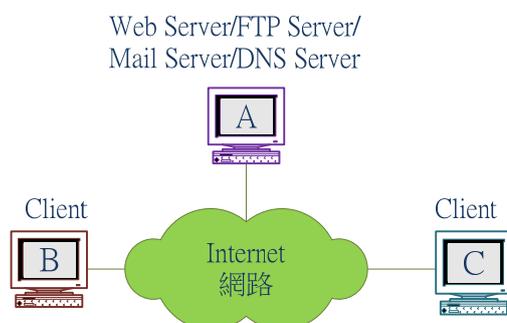


圖 10-6 單一主機多重服務

除了同一主機可安裝多個不同的伺服器外，在一個主機上(只有一片網路卡) 也可以設定多個虛擬主機 (Virtual Host)，每一個虛擬主機有一獨立的 IP 位址，而每一個 IP 位址同樣可以連接多個伺服器。如圖 10-6 所示，主機 A 原有的 IP 位址是 163.18.3.45 (網路卡的 IP)，但可在主機上設定 163.18.3.46、163.18.3.47、163.18.3.48 等三個虛擬主機，每一主機的都可連接獨立的伺服器，譬如， www.ee.cu.edu.tw、www.cs.cu.edu.tw、www.msg.cu.edu.tw，而三個 Web Server 都連接在各自 IP 的埠口 80 上。

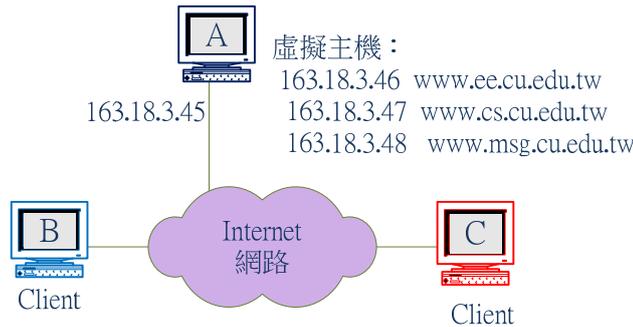


圖 10-7 虛擬主機的多重服務

10-3-2 單一主機單一服務

我們將某一特殊的應用系統安裝在一台主機上，讓這台主機專門處理這項服務，不論安全性的保護、系統備份、或異動登錄 (Log) 都可針對整部主機來管理，因此，對於這項服務所可能發生的安全漏洞也較小。再說，我們亦可使用高效能的主機來安裝較常使用的服務 (如，Web Server、Mail Server)，也能將服務效率提到最高。所以，單一服務單一主機的架構，是目前 Internet 網路最普遍的應用環境。它的架構如圖 10-7 所示，某一學校有 Web Server(www.cu.edu.tw)、Mail Server (cc.cu.edu.tw)、FTP Server (ftp.cu.edu.tw)，都將它們安裝在不同主機上，以提高傳輸效率。

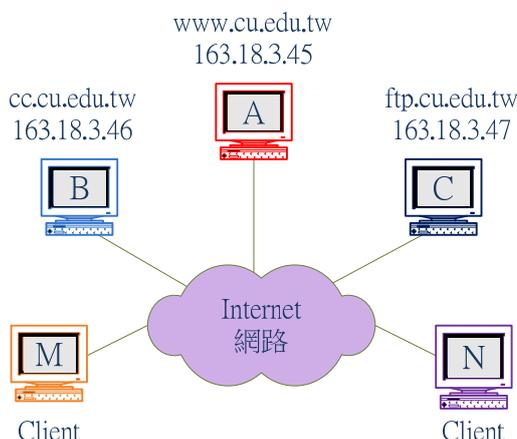


圖 10-8 單一服務單一主機架構

10-3-3 單一服務多重主機

如果單一主機電腦所提供服務的處理速度不能滿足於提供客戶端的要求，也就是說，客戶端的『要求』數量過多，主機上的處理速度不足於應付，而造成處理速度過慢。或者地理環境的因素，如將某一伺服器（Web Server）安裝於固定地點（例如台北），任何地區（如高雄）需要該服務都必須到該地點存取資料，如此可能會浪費不少頻寬。如有前面兩種因素存在，我們可考慮將同一伺服器內之資料，複製儲存於多個伺服器，而將其分散多個地區，不但可以分散主機電腦的處理負荷，也能提昇各地區的處理效率。如圖 10-8 所示，我們將同一 Web Server 複製並分散在多個地區，以提高伺服器的執行效率。單一服務多重主機的做法非常簡單，目前連結網路大多使用網域名稱（Domain Name），只要將各地區的網域伺服器加入不同的轉譯地址，便可將同一的伺服器指定到不同的網路位址上。如圖 10-8 中，Client_D 在網路區域 M，當它存取 Web Server A 時，經過 DNS 查詢位址為 164.5.3.78；相同的，Client_E 存取同一 Web Server_A 卻被指定到 185.3.56.2 的主機上，如此便可分散伺服器的傳輸量，一般國際性網站都用此方法。

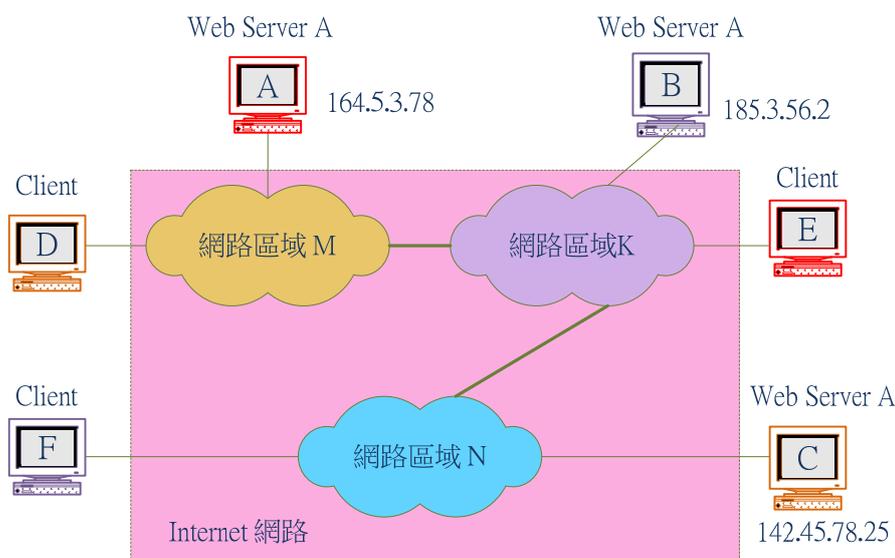


圖 10-9 單一服務多個伺服器

10-4 動態伺服器功能

雖然主從式架構是以『要求/回應』（Request-and-Response）的作業方式，但任一程式（Client 或 Server）都可以再向其它伺服器提出要求，而成為另一伺服器的客戶。因此，當客戶端向某一伺服器提出要求，而伺服器端不足於提供它所要的需求，也許該伺服器會再向其它伺服器請求服務，此功能稱之為『動態伺服器』（Dynamic Server），其架構如圖 10-9 所示。在圖 10-9 中，客戶端向

Server K 查詢某一資料，Server K 無法提供此資料 (或伺服器內定設計)，便透過網路向 Server M 查詢，得到回應後，再轉送回應給客戶端。

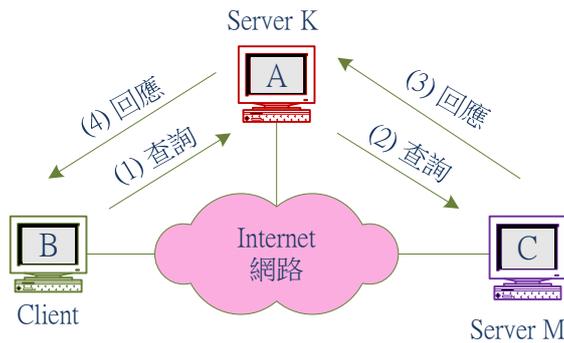


圖 10-10 動態伺服器架構

目前在 Internet 網路上，使用動態伺服器架構的情況非常普遍。例如，客戶端向某一 DNS 伺服器查詢網路位址，而該網域並非該伺服器所管轄，則該伺服器會將訊息轉送給其它伺服器，由另一伺服器負責查詢。又如 Web Server 也具有動態伺服的功能，當客戶端傳送查詢命令給伺服器時，伺服器可透過標準介面 (如 Common Gateway Interface, CGI) 向其它伺服器 (如 SQL Server) 查詢所需的訊息，與被伺服器連結之目的伺服器的所在位址無關。也就是說，目的伺服器也許與原伺服器在同一部主機內、或同一區域網路內、或在 Internet 網路上的任何地區。

10-5 連接與非連接伺服器架構

應用系統是建立在傳輸層上，基本上傳輸層提供兩種連接方式：連接導向 (Connection-Oriented) (TCP) 與非連接 (Connectionless) (UDP) 方式。連接導向方式表示雙方通訊之前必須建立通訊通道，再依照通訊通道來互相傳送訊息，又稱之為『**可靠性**』 (**Reliable**) 傳輸。至於非連接方式，表示通訊雙方並未建立通訊通道，傳送端直接將訊息發送到網路上，目的端並不保證可以接收得到，因此稱之為『**不可靠**』 (**Unreliable**) 傳輸。但連接導向通訊前必須建立連線，也許會浪費許多負荷 (Overhead)，反而非連接方式的效率較高。

應用系統應該採用何種傳輸模式較為適合？這是值得系統工作人員思考的問題。我們將一些考慮的重點歸類如下：

(A) 通訊範圍

依照 TCP/IP 通訊協定，網際層 (IP) 採用非連接方式，表示網際層並不負責偵測封包是否安全到達的功能，因此對於較遠的通訊距離，封包也許會經過多個路由器的轉送，才能到達目的地，因此，傳輸層就必須採用連接導向的連線方式 (TCP)。但反觀過來，如果網際層的封包轉送在一定區域內 (如一般區域網路)，不會經過太多的路由器，則採用非連接方式 (UDP) 效率反而會較高。例如，網路檔案系統 (Network File System, NFS) 所傳送的訊息大多限制於某一區域網路之內，因此採用非連接 (UDP) 以提高效率；而 Web 伺服器大多應用較遠地區之間的傳輸，因此，採用連接導向 (TCP) 方式。

(B) 封包大小

當傳送端發送某一訊息給接收端，如果訊息較大時，傳輸層會將該訊息分割成若干個封包，再依次經由網際層傳送到對方。但因網際層採用電報傳輸 (Datagram) 方式，當封包到達目的時，封包次序可能和原來不同，如何去偵測哪一個封包遺失，或回復原來的次序，這就必須透過可靠的傳輸層來負責，因此，對於傳輸訊息較大時，採用連接導向方式 (TCP) 較為適合。但就傳送訊息較短 (也許傳輸距離也遠) 的服務，本身訊息沒有分割的可能，則可考慮使用非連接方式，其效率較高。例如網域名稱伺服器 (DNS)，客戶端和伺服器端之間的傳送訊息很短，在一個 IP 封包內就足以承載，因此，採用非連接方式 (UDP) 效率反而較高。

有些伺服器為了提供不同環境之應用，提供有 TCP 和 UDP 的選擇連接方式，也就是說，在同一個傳輸埠口提供兩種連接方式，客戶端可依照自己的需求 (傳輸距離或訊息大小)，來決定連接方式，譬如，Time Server 是 37/tcp 與 37/udp，nameserver 是 53/tcp 與 53/udp 上。

習題

1. 請簡述 Internet 應用系統模型中通訊端點的定址方法。
2. 何謂『網域名稱系統』(Domain Name System, DNS) ?
3. 請說明一般 Internet 應用系統如何達到多工 (Multiplex) 的功能 ?
4. 何謂『NVT ASCII』 ?
5. 請簡述一般 Internet 應用系統，如何利用 NVT ASCII 來達到雙方的通訊 ?
6. 請簡述『主從式架構』(Client/Server Architecture) 的特性。
7. 何謂『單一服務多重主機』 ? 在何種情況之下需要這種架構 ? 應如何來實現它 ?
8. 何謂『動態伺服器』(Dynamic Server) ? 並請舉例說明之。
9. 一般應用系統選用 TCP 或 UDP 傳輸服務的依據為何 ? 並請舉例說明之。