鳥哥的 Linux 私房菜

為取得較佳瀏覽結果，請愛用 firefox 瀏覽本網頁

第二章、基礎網路概念

切換解析度為 800x600

最近更新日期：2011/07/15

你的伺服器是放在網路網路上面來提供服務的，所以，如果沒有網路或者是網路不通，那麼你的伺服器當然是英雄無用武之地啦！ 此外，伺服器上面的網路服務都是用來達成某項網際網路的通訊協定，以提供相對應的服務而已。所以囉，你當然得要知道這個最基礎的網路概念，否則，當伺服器的服務出現問題時，你該如何解決啊？您說對吧！ 這部份最重要的是 TCP/IP 與 OSI 七層協定的相關概念了，這部份難的很～難的很～ 在這一章中，鳥哥以較為口語的方式來介紹這些基礎網路架構，希望能帶給朋友們快速瞭解網路是啥。 當然，想要更瞭解網路相關功能的話，文末的參考資料可以參考看看喔！ ^_^

2.1 網路是個什麼玩意兒

全世界的人種有很多，人類使用的語言種類也多的很。那如果你想要跟外國人溝通時，除了比手劃腳之外，你要如何跟對方講話？ 大概只有兩種方式囉，一種是強迫他學中文，一種則是我們學他的語言，這樣才能溝通啊。在目前世界上的強勢語言還是屬於英語系國家， 所以囉，不管是啥人種，只要學好英文，那麼大家都講英文，彼此就能夠溝通了。希望不久的未來，咱們的中文能夠成為強勢語言啊！

這個觀念延伸到網路上面也是行的通的，全世界的作業系統多的很，不是只有 Windows/Linux 而已，還有蘋果電腦自己的作業系統， Unix like 的作業系統也非常多！那麼多的作業系統 (人種) 要如何進行網路溝通 (語言) 呢？那就得要制訂共同遵守的標準才行了。這個標準是由國際組織規範的，你的系統裡面只要提供可以加入該標準的程式碼， 那你就能夠透過這個標準與其他系統進行溝通！所以囉，網路是跨平台的，並不是只有 Linux 才這麼做！因此，這部份的資料你學完後，是可以應用在所有作業系統上面的！觀念都相同啊！

另外，這一個章節旨在引導網路新鮮人快速進入網路的世界，所以鳥哥寫的比較淺顯一些些，基本上， 還有一堆網路硬體與通訊協定並沒有被包含在這篇短文裡頭。如果你的求知慾已經高過本章節， 那麼請自行到書局尋找適合你自己的書籍來閱讀！當然，你也可以在網際網路上面找到你所需要的資料。 在本章最後的參考資料可以瞧一瞧吶！

2.1.1 什麼是網路

我們都知道，網路就是幾部電腦主機或者是網路印表機之類的周邊設備， 透過網路線或者是無線網路的技術，將這些主機與設備連接起來， 使得資料可以透過網路媒體(網路線以及其他網路卡等硬體)來傳輸的一種方式。 請你想像一下，如果你家裡面只有電腦、印表機、傳真機等機器，卻沒有網路連接這些硬體， 那麼使用上會不會很麻煩？如果將這個場景移到需要工作的辦公室時， 電腦的資料無法使用網路連接到印表機來列印，那是否很傷腦筋呢？ 對吧！光用想的就覺得很麻煩吧！不幸的是，這些麻煩事在 1970 年代以前，確實是存在的啊！

- 各自為政的『網路硬體與軟體』技術發展： **Ethernet & Token-Ring**

  在 1970 年代前後，為了解決這個煩人的資料傳輸問題，各主要資訊相關的公司都在研究各自的網路連接技術， 以使自家的產品可以在辦公室的環境底下組織起來。其中比較有名的就是全錄公司的 Ethernet 技術， 以及 IBM 研發的 Token-Ring 技術了。但是這些技術有個很大的問題，那就是它們彼此不認識對方的網路技術！ 也就是說，萬一你的辦公室購買了整合 Ethernet 技術的電腦主機，但是其他的電腦卻是使用 IBM 的機器時， 想要在這兩者之間進行資料的溝通，在早期來說那是不可能

的。

- 以『軟體』技術將硬體整合： **ARPANET & TCP/IP**

  為了解決上述的網路硬體整合功能，所以在 1960 年代末期美國國防部就開始研究一個可以在這些不同的網路硬體上面運作的軟體技術， 使得不同公司的電腦或資料可以透過這個軟體來達成資料溝通。這個研究由美國國防部尖端研究企畫署 (Defense Advanced Research Project Agency, DARPA) 負責，他們將該網路系統稱為 ARPANET， 這個咚咚就是目前熟知的 TCP/IP 技術的雛形了！在 1975 年左右， ARPANET 已可以在常見的 Ethernet 與 Token-Ring 等硬體平台底下互通資料了。DARPA 在 1980 年正式推出 TCP/IP 技術後， 由於想要推展此項技術，因此與柏克萊 (Berkeley) 大學合作，將 TCP/IP 植入著名的 BSD Unix 系統內， 由於大學乃是未來人才資料庫的培養處，所以， TCP/IP 這項技術便吸引越來越多使用者的投入， 而這種連接網路的技術也被稱之為 Internet (註1)。

- 沒有任何王法的網際網路： **Internet**

  現在我們知道 Internet 就是使用 TCP/IP 的網路連接技術所串聯起來的一個網路世界， 而這個 Internet 在 1980 年代之後由於對 email 的需求以及瀏覽器圖形介面的興起，因此快速的蔓延在電腦世界中。但是， Internet 有沒有人在管理啊？很不巧的是， Internet 是一個管理相當鬆散的所在。 只要你能夠使用任何支援 TCP/IP 技術的硬體與作業系統，並且實際連接上網路後， 你就進入 Internet 的世界了。在該世界當中，沒有任何王法的保護，你的實際資料如果接上 Internet ， 在任何時刻都需要自己保護自己，免得中了『流彈』而受傷啊！

  為甚麼說 Internet 沒有王法呢？這是因為 Internet 僅是提供一個網路的連接介面， 所以你只要連接上 Internet 後，全世界都可以任你遨遊，不過也因為如此，『跨海』而來的攻擊就成了簡單的事件， 簡單說，台灣的法律僅適用台灣地區對吧？但是電腦怪客 (cracker) 可以在國外透過 Internet 對你的主機進行攻擊， 我們的法律可管不到國外地區啊！雖然可以透過很多國際管道來尋求協助，不過，還是很難協助你緝拿兇手的啊。 因此囉，在你的主機要連上 Internet 之前，請先詢問自己，真的有需要連上 Internet 嗎？^_^

- 軟硬體標準制定的成功帶來的影響： **IEEE** 標準規範

  現在我們常常聽到『你要上網啊！那你要去買網路卡喔！還得要連接到 Internet 才行啊！』 這個網路卡就是市面上隨處可見的一個介面卡而已，至於 Internet 則是去向 Hinet/Seed net 或 其他網路服務提供公司 (Internet Service Provider, ISP) 申請的帳號密碼。 問題是，是否就只有透過網路卡與 Internet 才能上網啊？呵呵！當然不是！其他不同的網路硬體與軟體可多著那！ 不過，最成功的卻是乙太網路 (Ethernet) 與 Internet ，這是為甚麼呢？這兩者的技術比較好嗎？ 當然不是！這是因為這兩者都被『標準』所支援的緣故(註2)。

  乙太網路最初是由全錄公司 (Xerox PARC) 所建構出來的，而後透過 DEC, Intel 與 Xerox 合作將乙太網路標準化。 再經由 IEEE (Institute of Electrical and Electronic Engineers 註3) 這個國際著名的專業組織利用一個 802 的專案制定出標準，之後有 19 家公司宣佈支援 IEEE 所發布的 802.3 標準， 並且到了 1989 年國際標準組織 ISO (International Organization for Standard) 將乙太網路編入 IS88023 標準，呵呵！這表示乙太網路已經是一項公認的標準介面了，如此一來，大家都可以依據這個標準來設定與開發自己的硬體， 只要硬體符合這個標準，理論上，他就能夠加入乙太網路的世界，所以，購買乙太網路時， 僅需要查看這個乙太網路卡支援哪些標準就能夠知道這個硬體的功能有哪些， 而不必

知道這個乙太網路卡是由哪家公司所製造的吶。

Tips:

標準真的是個很重要的東西，真要感謝這些維護標準的專業組織。當有公司想要開發新的硬體時， 它可以參考標準組織所發布與維護的文件資料，透過這些文件資料後，該公司就知道要製作的硬體需要符合哪些標準， 同時也知道如何設計這些硬體，讓它可以『相容』於目前的機器，讓使用者不會無所適從啊。 包括軟體也有標準，早期 Linux 在開發時就是透過了解 POSIX 這個標準來設計核心的， 也使得 Linux 上面可以執行大多數的標準介面軟體呢！你說，標準是否真的很重要啊！

除了硬體之外，TCP/IP 這個 Internet 的通訊協定也是有標準的，這些標準大部分都以 RFC (Request For Comments, 註4) 的形式發佈標準文件。 透過這些文件的輔助，任何人只要會寫程式語言的話，就有可能發展出自己的 TCP/IP 軟體， 並且連接上 Internet 。早期的 Linux 為了要連接上 Internet ， Linux 團隊就自己撰寫出 TCP/IP 的程式碼， 透過的就是這些基礎文件的標準依據啊！舉例來說 RFC 1122 (註5) 這個建議文件就指出一些可以連線到 Internet 的主機應該要注意的相關協定與基本需求，讓想要撰寫連線程式的設計師可以有一個指引的標準方向。

2.1.2 電腦網路組成元件

接下來，讓我們來談談那麼組成電腦網路的元件有哪些呢？這些元件的定義為何啊？我們得要先知道有哪些硬體嘛！ 接下來才好理解啊。在這裡，我們以底下這張連線示意圖來解釋好了：

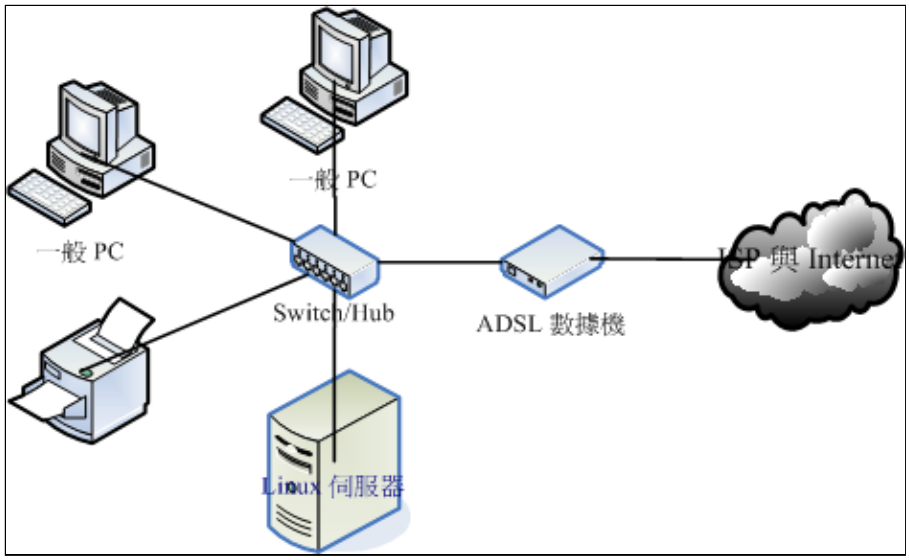

圖 2.1-1、電腦網路連線示意圖

在上圖中，我們主要需要注意到的硬體有哪些呢？大致有底下這些啦：

- 節點 (node)：節點主要是具有網路位址 (IP) 的設備之稱， 因此上面圖示中的一般PC、Linux伺服器、ADSL數據機與網路印表機等，個別都可以稱為一個 node ！ 那中間那個集線器 (hub) 是不是節點呢？因為他不具有 IP ，因此 hub 不是節點。

- 伺服器主機 (server)：就網路連線的方向來說，提供資料以『回應』給用戶的主機， 都可以被稱為是一部伺服器。舉例來說，Yahoo 是個 WWW 伺服器，崑山的 FTP (http://ftp.ksu.edu.tw/) 是個檔案伺服器等等。

- 工作站 (workstation) 或用戶端 (client)：任何可以在電腦網路輸入的設備都可以是工作站， 若以連線發起的方向來說，主動發起連線去『要求』資料的，就可以稱為是用戶端 (client)。舉例來說，一般 PC 打開瀏覽器對 Yahoo 要求新聞資料，那一般 PC 就是用戶端。

- 網路卡 (Network Interface Card, NIC)：內建或者是外插在主機上面的一個設備， 主要提供網路連線的卡片，目前大都使用具有 RJ-45 接頭的乙太網路卡。一般 node 上都具有一個以上的網路卡， 以達成網路連線的功能。

- 網路介面：利用軟體設計出來的網路介面，主要在提供網路位址 (IP) 的任務。 一張網卡至少可以搭配一個以上的網路介面；而每部主機內部其實也都擁有一個內部的網路介面，那就是 loopback (lo) 這個迴圈測試介面！

- 網路形態或拓樸 (topology)：各個節點在網路上面的連結方式，一般講的是物理連接方式。 舉例來說，上圖中顯示的是一種被稱為星形連線 (star) 的方式，主要是透過一個中間連接設備，以放射狀的方式連接各個節點的一種形態，這就是一種拓樸。

- 網關 (route) 或通訊閘 (gateway)：具有兩個以上的網路介面， 可以連接兩個以上不同的網段的設備，例如 IP 分享器就是一個常見的網關設備。那上面的 ADSL 數據機算不算網關呢？ 其實不太能算，因為數據機通常視為一個在主機內的網卡設備，我們可以在一般 PC 上面透過撥號軟體， 將數據機模擬成為一張實體網卡 (ppp) ，因此他不太能算是網關設備啦！

網路設備其實非常多也非常複雜，不過如果以小型企業角度來看，我們能夠瞭解上述圖示內各設備的角色，那應該也足夠囉！ 接下來，讓我們繼續來討論一下網路範圍的大小吧！

2.1.3 電腦網路區域範圍

由於各個節點的距離不同，連線的線材與方式也有所差異，由於線材的差異也導致網路速度的不同，讓網路的應用方向也不一樣。 根據這些差異，早期我們習慣將網路的大小範圍定義如下：(註6)

- 區域網路 **(Local Area Network, LAN)**：
  節點之間的傳輸距離較近，例如一棟大樓內，或一個學校的校區內。可以使用較為昂貴的連線材料， 例如光纖或是高品質網路線 (CAT 6) 等。網路速度較快，連線品質較佳且可靠，因此可應用於科學運算的叢集式系統、 分散式系統、雲端負荷分擔系統等。

- 廣域網路 **(Wide Area Network, WAN)**：
  傳輸距離較遠，例如城市與城市之間的距離，因此使用的連線媒體需要較為便宜的設備，例如經常使用的電話線就是一例。 由於線材品質較差，因此網路速度較慢且可靠性較低一些，網路應用方面大多為類似 email, FTP, WWW 瀏覽等功能。

除了這兩個之外，還有所謂的都會網路 (Metropolitan Area Network, MAN)，不過近來比較少提及，因此你只要知道有 LAN 及 WAN 即可。這兩個名詞在很多地方你都可以看的到喔！改天你回家看看你家的 ADSL 數據機或 IP 分享器後面的插孔看看，你就能夠看到有 WAN 與 LAN 的插孔，現在你就知道為啥有這兩個燈號與插孔了吧。

一般來說，LAN 指的是區域範圍較小的環境，例如一棟大樓或一間學校，所以在我們生活周遭有著許許多多的 LAN 存在。 那這些 LAN 彼此串接在一起，全部的 LAN 串在一塊就是一個大型的 WAN 囉！簡單的說，就是這樣分。

不過，現在的環境跟以前不一樣了，舉例來說，前幾天剛剛宣布 (2011/07)，光纖的速度已經可以到達 100Mbps/10Mbps 的下載/上傳頻寬了！再舉例來說，台灣的學術網路通通是串在一塊的，鳥哥在台南崑山連線到高雄義守大學下載 CentOS 映像檔時，你猜下載的速度有多快？每秒鐘可高達 100Mbps 左右！這已經是一個內部區網的速度了！所以，用以前的觀點來看， 其實對目前的網路環境有點不符現象了。因此，目前你可以使用『速度』作為一個網路區域範圍的評量。 或許現在我們可以說，整個台灣的學術網路 (TANET, 註7) 可以視為是一個區域網路呢！

2.1.4 電腦網路通訊協定： OSI 七層協定

談完了網路需要制訂的標準、網路連線的元件以及網路的範圍之後，接下來就是要講到，那麼各個節點之間是如何溝通訊息的呢？ 其實就是透過標準的通訊協定啦！但是，整個網路連接的過程相當複雜，包括硬體、軟體資料封包與應用程式的互相連結等等， 如果想要寫一支將聯網全部功能都串連在一塊的程式，那麼當某個小環節出現問題時，整隻程式都需要改寫啊！真麻煩！

那怎辦？沒關係，我們可以將整個網路連接過程分成數個階層 (layer)，每個階層都有特別的獨立的功能， 而且每個階層的程式碼可以獨立撰寫，因為每個階層之間的功能並不會互相干擾的。 如此一來，當某個小環節出現問題時，只要將該層級的程式碼重新撰寫即可。所以程式撰寫也容易，整個網路概念也就更清晰！ 那就是目前你常聽到的 OSI 七層協定 (Open System Interconnection) 的概念囉！

如果以圖示來說，那麼這七個階層的相關性有點像底下這樣：

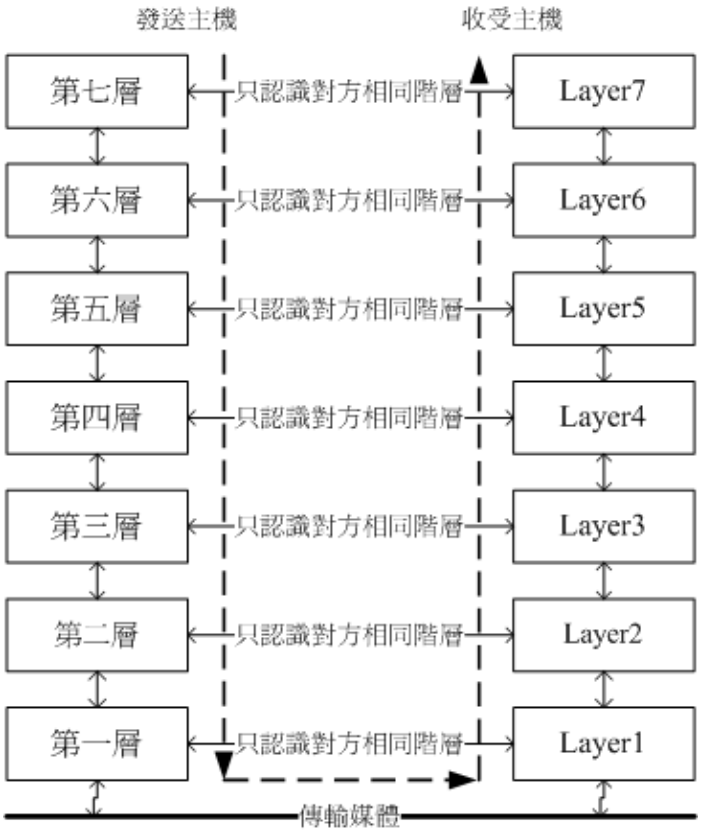

圖 2.1-2、OSI 七層協定各階層的相關性

依據定義來說，越接近硬體的階層為底層 (layer 1)，越接近應用程式的則是高層 (layer 7)。 不論是接收端還是發送端，每個一階層只認識對方的同一階層資料。 而整個傳送的過程就好像人們在玩整人遊戲一般，我們透過應用程式將資料放入第七層的包裹，再將第七層的包裹放到第六層的包裹內， 依序一直放到第一層的最大的包裹內，然後傳送出去給接收端。接收端的主機就得由第一個包裹開始，依序

將每個包裹拆開， 然後一個一個交給對應負責的階層來視察！這就是整人遊戲...喔！是 OSI 七層協定在階層定義方面需要注意的特色。

既然說是包裹，那我們都知道，包裹表面都會有個重要的資訊，這些資訊包括有來自哪裡、要去哪裡、接收者是誰等等， 而包裹裡面才是真正的資料。同樣的，在七層協定中，每層都會有自己獨特的表頭資料 (header)，告知對方這裡面的資訊是什麼， 而真正的資料就附在後頭囉！我們可以使用如下的圖示來表示這七層每一層的名字，以及資料是如何放置到每一層的包裹內：

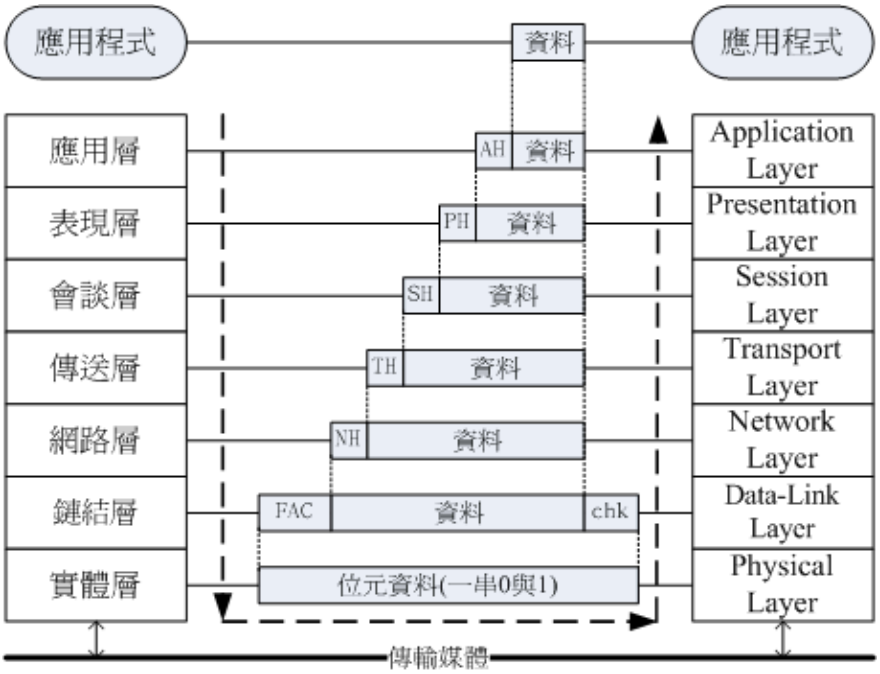

圖 2.1-3、OSI 七層協定資料的傳遞方式

上圖中仔細看每個資料包的部分，上層的包裹是放入下層的資料中，而資料前面則是這個資料的表頭。其中比較特殊的是第二層， 因為第二層 (資料鏈結層) 主要是位於軟體封包 (packet) 以及硬體訊框 (frame) 中間的一個階層， 他必須要將軟體包裝的包裹放入到硬體能夠處理的包裹中，因此這個階層又分為兩個子層在處理相對應的資料。 因為比較特殊，所以您瞧瞧，第二層的資料格式比較不一樣喔，尾端還出現一個檢查碼哩～

每一個階層所負責的任務是什麼呢？簡單的說，每一層負責的任務如下：(註6, 註8, 註9)

分層	負責內容
Layer 1 實體層 Physical Layer	由於網路媒體只能傳送 0 與 1 這種位元串，因此實體層必須定義所使用的媒體設備之電壓與訊號等， 同時還必須瞭解資料訊框轉成位元串的編碼方式，最後連接實體媒體並傳送/接收位元串。
Layer 2 資料鏈結層 Data-Link Layer	這一層是比較特殊的一個階層，因為底下是實體的定義，而上層則是軟體封裝的定義。因此第二層又分兩個子層在進行資料的轉換動作。 在偏硬體媒體部分，主要負責的是 MAC (Media Access Control) ，我們稱這個資料包裹為 MAC 訊框 (frame)， MAC 是網路媒體所能處理的主要資料包裹，這也是最終被實體層編碼成位元串的資料。MAC 必須要經由通訊協定來取得媒體的使用權， 目前最常使用的則是 IEEE 802.3 的乙太網路協定。詳細的 MAC 與乙太網路請參考下節說明。 至於偏向軟體的部分則是由邏輯連結層 (logical link control, LLC) 所控制，主要在多工處理來自上層的封包資料 (packet) 並轉成 MAC 的格式， 負責的工作包括訊息交換、流量控制、失誤問題的處理等等。

Layer 3 網路層 Network Layer	這一層是我們最感興趣的囉，因為我們提及的 IP (Internet Protocol) 就是在這一層定義的。 同時也定義出電腦之間的連線建立、終止與維持等，資料封包的傳輸路徑選擇等等，因此這個層級當中最重要的除了 IP 之外，就是封包能否到達目的地的路由 (route) 概念了！
Layer 4 傳送層 Transport Layer	這一個分層定義了發送端與接收端的連線技術(如 TCP, UDP 技術)， 同時包括該技術的封包格式，資料封包的傳送、流程的控制、傳輸過程的偵測檢查與復原重新傳送等等， 以確保各個資料封包可以正確無誤的到達目的端。
Layer 5 會談層 Session Layer	在這個層級當中主要定義了兩個位址之間的連線通道之連接與掛斷，此外，亦可建立應用程式之對談、 提供其他加強型服務如網路管理、簽到簽退、對談之控制等等。如果說傳送層是在判斷資料封包是否可以正確的到達目標， 那麼會談層則是在確定網路服務建立連線的確認。
Layer 6 表現層 Presentation Layer	我們在應用程式上面所製作出來的資料格式不一定符合網路傳輸的標準編碼格式的！ 所以，在這個層級當中，主要的動作就是：將來自本地端應用程式的資料格式轉換(或者是重新編碼)成為網路的標準格式， 然後再交給底下傳送層等的協定來進行處理。所以，在這個層級上面主要定義的是網路服務(或程式)之間的資料格式的轉換， 包括資料的加解密也是在這個分層上面處理。
Layer 7 應用層 Application Layer	應用層本身並不屬於應用程式所有，而是在定義應用程式如何進入此層的溝通介面，以將資料接收或傳送給應用程式，最終展示給使用者。

事實上， OSI 七層協定只是一個參考的模型 (model)，目前的網路社會並沒有什麼很知名的作業系統在使用 OSI 七層協定的聯網程式碼。那...講這麼多幹嘛？這是因為 OSI 所定義出來的七層協定在解釋網路傳輸的情況來說， 可以解釋的非常棒，因此大家都拿 OSI 七層協定來做為網路的教學與概念的理解。至於實際的聯網程式碼，那就交給 TCP/IP 這個玩意兒吧！

2.1.5 電腦網路通訊協定： TCP/IP

雖然 OSI 七層協定的架構非常嚴謹，是學習網路的好材料。但是也就是因為太過嚴謹了，因此程式撰寫相當不容易， 所以造成它在發展上面些許的困擾。而由 ARPANET 發展而來的 TCP/IP 又如何呢？其實 TCP/IP 也是使用 OSI 七層協定的觀念， 所以同樣具有分層的架構，只是將它簡化為四層，在結構上面比較沒有這麼嚴謹，程式撰寫會比較容易些。後來在 1990 年代由於 email, WWW 的流行，造成 TCP/IP 這個標準為大家所接受，這也造就目前我們的網路社會囉！

既然 TCP/IP 是由 OSI 七層協定簡化而來，那麼這兩者之間有沒有什麼相關性呢？它們的相關性可以圖示如下， 同時這裡也列出目前在這架構底下常見的通訊協定、封包格式與相關標準：

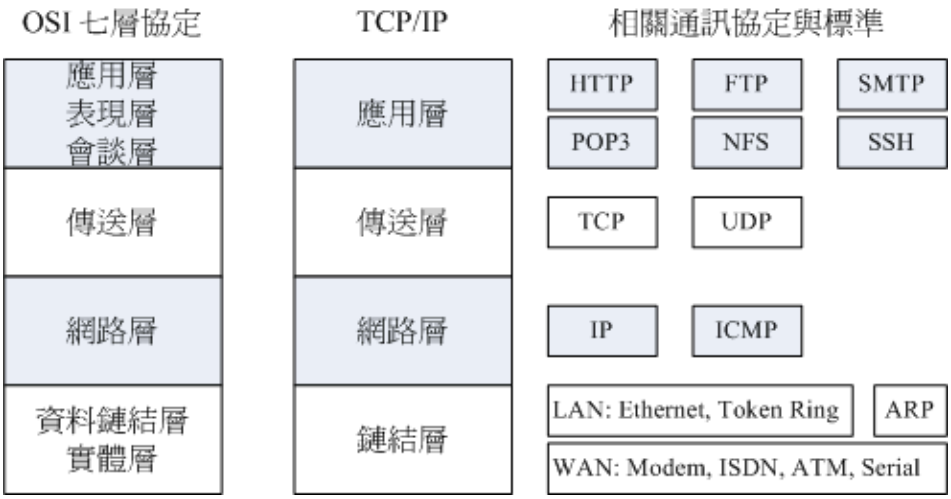

圖 2.1-4、OSI 與 TCP/IP 協定之相關性

從上圖中，我們可以發現 TCP/IP 將應用、表現、會談三層整合成一個應用層，在應用層上面可以實作的程式協定有 HTTP, SMTP, DNS 等等。 傳送層則沒有變，不過依據傳送的可靠性又將封包格式分為連接導向的 TCP 及非連接導向的 UDP 封包格式。網路層也沒有變，主要內容是提供了 IP 封包，並可選擇最佳路由來到達目標 IP 位址。資料鏈結層與實體層則整合成為一個鏈結層，包括定義硬體訊號、 訊框轉位元串的編碼等等，因此主要與硬體 (不論是區網還是廣域網路) 有關。

那 TCP/IP 是如何運作的呢？我們就拿妳常常連上的 Yahoo 入口網站來做個說明好了，整個連線的狀態可以這樣看：

0. 應用程式階段：妳打開瀏覽器，在瀏覽器上面輸入網址列，按下 [Enter]。此時網址列與相關資料會被瀏覽器包成一個資料， 並向下傳給 TCP/IP 的應用層；
1. 應用層：由應用層提供的 HTTP 通訊協定，將來自瀏覽器的資料包起來，並給予一個應用層表頭，再向傳送層丟去；
2. 傳送層：由於 HTTP 為可靠連線，因此將該資料丟入 TCP 封包內，並給予一個 TCP 封包的表頭，向網路層丟去；
3. 網路層：將 TCP 包裹包進 IP 封包內，再給予一個 IP 表頭 (主要就是來源與目標的 IP 囉)，向鏈結層丟去；
4. 鏈結層：如果使用乙太網路時，此時 IP 會依據 CSMA/CD 的標準，包裹到 MAC 訊框中，並給予 MAC 表頭，再轉成位元串後， 利用傳輸媒體傳送到遠端主機上。

等到 Yahoo 收到你的包裹後，在依據相反方向拆解開來，然後交給對應的層級進行分析，最後就讓 Yahoo 的 WWW 伺服器軟體得到你所想要的資料，該伺服器軟體再根據你的要求，取得正確的資料後，又依循上述的流程，一層一層的包裝起來， 最後傳送到你的手上！就是這樣囉！

根據這樣的流程，我們就得要知道每個分層所需要瞭解的基礎知識，這樣才算學習網路基礎嘛！所以底下我們會依據 TCP/IP 的鏈結層、網路層、傳送層來進行說明，應用層的協定則在後續章節中有對應的協定再來談囉！同時我們也知道， 網路媒體一次傳輸的資料量是有限的，因此如果要被傳輸的資料太大時，我們在分層的包裝中，就得要將資料先拆開放到不同的包裹中， 再給包裹一個序號，好讓目的端的主機能夠藉由這些序號再重新將資料整合回來！很有趣吧！接下來就讓我們一層一層來介紹囉！

Tips:

一般來說，因為應用程式與程式設計師比較有關係，而網路層以下的資料則主要是作業系統提供的，因此， 我們又將 TCP/IP 當中的應用層視為使用者層，而底下的三層才是我們主要談及的網路基礎！所以這個章節主要就是介紹這三層啦！

2.2 TCP/IP 的鏈結層相關協定

TCP/IP 最底層的鏈結層主要與硬體比較有關係，因此底下我們主要介紹一些 WAN 與 LAN 的硬體。 同時會開始介紹那重要的 CSMA/CD 的乙太網路協定，以及相關的硬體與 MAC 訊框格式等。那就開始來聊聊囉！

2.2.1 廣域網路使用的設備

在 2.1.3 節我們有提到過，廣域網路使用的設備價格較為低廉。 不過廣域網路使用到的設備非常的多，一般用戶通常會接觸到的主要是 ADSL 數據機或者是光纖到大廈，以及第四台的 Cable 寬頻等。在這裡我們先介紹一些比較常見的設備，如果以後你有機會接觸到其他設備，再請你依據需求自行查閱相關書籍吧！

- 傳統電話撥接：透過 ppp 協定

  早期網路大概都只能透過數據機加上電話線以及電腦的九針序列埠 (以前接滑鼠或搖桿的插孔)，然後透過 Point-to-Point Protocol (PPP 協定) 配合撥接程式來取得網路 IP 參數，這樣就能夠上網了。不過這樣的速度非常慢，而且當電話撥接後， 就不能夠講電話了！因為 PPP 支援 TCP/IP, NetBEUI, IPX/SPX 等通訊協定，所以使用度非常廣！

- 整合服務數位網路 (Integrated Services Digital Network, ISDN)

  也是利用現有的電話線路來達成網路連線的目的，只是連線的兩端都需要有 ISDN 的數據機來提供連線功能。 ISDN 的傳輸有多種通道可供使用，並且可以將多個通道整合應用，因此速度可以成倍成長。基本的 B 通道速度約為 64Kbps， 但如美國規格使用 23 個以上的通道來達成連線，此時速度可達 1.5Mbps 左右。不過台灣這玩意兒比較少見。

- 非對稱數位用路回路 (Asymmetric Digital Subscriber Line, ADSL)：透過 pppoe 協定

  也是透過電話線來撥接後取得 IP 的一個方法，只不過這個方式使用的是電話的高頻部分，與一般講電話的頻率不同。 因此妳可以一邊使用 ADSL 上網同時透過同一個電話號碼來打電話聊天。在台灣，由於上傳/下載的頻寬不同， 因此才稱為非對稱的回路。ADSL 同樣使用數據機，只是他透過的是 PPPoE (PPP over Ethernet) 的方法！ 將 PPP 模擬在乙太網路卡上，因此你的主機需要透過一張網路卡來連接到數據機，並透過撥接程式來取得新的介面 (ppp0) 喔！

- 纜線數據機 (Cable modem)

  主要透過有線電視 (台灣所謂的第四台) 使用的纜線作為網路訊號媒體，同樣需要具備數據機來連接到 ISP，以取得網路參數來上網。 Cable modem 的頻寬主要是分享型的，所以通常具有區域性，並不是你想裝就能裝的哩！

2.2.2 區域網路使用的設備-乙太網路

在區域網路的環境中，我們最常使用的就是乙太網路。當然啦，在某些超高速網路應用的環境中， 還可能會用到價格相當昂貴的光纖通道哩。只是如同前面提到的，乙太網路因為已經標準化了，設備設置費用相對低廉， 所以一般你會聽到什麼網路線或者是網路媒體，幾乎都是使用乙太網路來架設的環境啦！ 只是這裡還是要提醒您，整個網路世界並非僅有乙太網路這個硬體介面喔！ 事實上，想瞭解整個乙太網路的發展，建議你可以直接參考風信子與張民人先生翻譯的 『Switched & Fast 乙太網路』一書，該書內容相當的有趣，挺適合閱讀的吶。 底下我們僅做個簡單的介紹而已。

- 乙太網路的速度與標準

乙太網路的流行主要是它成為國際公認的標準所致。早先 IEEE 所制訂的乙太網路標準為 802.3 的 IEEE 10BASE5 ，這個標準主要的定義是：『10 代表傳輸速度為 10Mbps，BASE 表示採用基頻信號來進行傳輸，至於 5 則是指每個網路節點之間最長可達 500 公尺。』

由於網路的傳輸資訊就是 0 與 1 啊，因此，資料傳輸的單位為每秒多少 bit ， 亦即是 M bits/second, Mbps 的意思。那麼為何制訂成為 10Mbps 呢？ 這是因為早期的網路線壓製的方法以及相關的製作方法，還有乙太網路卡製作的技術並不是很好， 加上當時的資料傳輸需求並沒有像現在這麼高，所以 10Mbps 已經可以符合大多數人的需求了。

Tips:

我們看到的網路提供者 (Internet Services Provider, ISP) 所宣稱他們的 ADSL 傳輸速度可以達到 下行/上行 2Mbps/128Kbps (Kbits per second) 時，那個 Kb 指的可不是 bytes 而是 bits 喔！所以 2M/128K 在實際的檔案大小傳輸速度上面，最大理論的傳輸為 256KBps/16 KBps(KBytes per second)，所以正常下載的速度約在每秒 100~200 KBytes 之間吶！ 同樣的道理，在網路卡或者是一些網路媒體的廣告上面，他們都會宣稱自己的產品可以自動辨識傳輸速度為 10/100 Mbps (Mega-bits per second)，呵呵！該數值還是得再除以 8 才是我們一般常用的檔案容量計算的單位 bytes 喔！

早期的網路線使用的是舊式的同軸電纜線，這種線路在現在幾乎已經看不到了。取而代之的是類似傳統電話線的雙絞線 (Twisted Pair Ethernet) ，IEEE 並將這種線路的乙太網路傳輸方法制訂成為 10BASE-T 的標準。 10BASE-T 使用的是 10 Mbps 全速運作且採用無遮蔽式雙絞線 (UTP) 的網路線。此外，10BASE-T 的 UTP 網路線可以使用星形連線(star)， 也就是以一個集線器為中心來串連各網路設備的一個方法，圖 2.1-1 就是星形連線的一個示意圖。

不同於早期以一條同軸電纜線連結所有的電腦的 bus 連線，透過星形連線的幫助， 我們可以很簡單的加裝其他的設備或者是移除其他設備，而不會受到其他裝置的影響，這對網路設備的擴充性與除錯來說， 都是一項相當棒的設計！也因此 10BASE-T 讓乙太網路設備的銷售額大幅提昇啊！

後來 IEEE 更制訂了 802.3u 這個支援到 100Mbps 傳輸速度的 100BASE-T 標準，這個標準與 10BASE-T 差異不大， 只是雙絞線線材製作需要更精良，同時也已經支援使用了四對絞線的網路線了， 也就是目前很常見的八蕊網路線吶！這種網路線我們常稱為等級五 (Category 5, CAT5) 的網路線。 這種傳輸速度的乙太網路就被稱為 Fast ethernet 。至於目前我們常常聽到的 Gigabit 網路速度 1000 Mbps 又是什麼吶？那就是 Gigabit ethernet 哩！只是 Gigabit ethernet 的網路線就需要更加的精良。

名稱	速度	網路線等級
乙太網路(Ethernet)	10Mbps	-

高速乙太網路(Fast Ethernet)	100Mbps	CAT 5
超高速乙太網路(Gigabit Ethernet)	1000Mbps	CAT 5e/CAT 6

為什麼每當傳輸速度增加時，網路線的要求就更嚴格呢？這是因為當傳輸速度增加時，線材的電磁效應相互干擾會增強， 因此在網路線的製作時就得需要特別注意線材的質料以及內部線蕊心之間的纏繞情況配置等， 以使電子流之間的電磁干擾降到最小，才能使傳輸速度提升到應有的 Gigabit 。 所以說，在乙太網路世界當中，如果你想要提升原有的 fast ethernet 到 gigabit ethernet 的話， 除了網路卡需要升級之外，主機與主機之間的網路線， 以及連接主機線路的集線器/交換器等，都必須要提升到可以支援 gigabit 速度等級的設備才行喔！

- 乙太網路的網路線接頭 (跳線/平行線)

前面提到，網路的速度與線材是有一定程度的相關性的，那麼線材的接頭又是怎樣呢？ 目前在乙太網路上最常見到的接頭就是 RJ-45 的網路接頭，共有八蕊的接頭，有點像是胖了的電話線接頭， 如下所示：

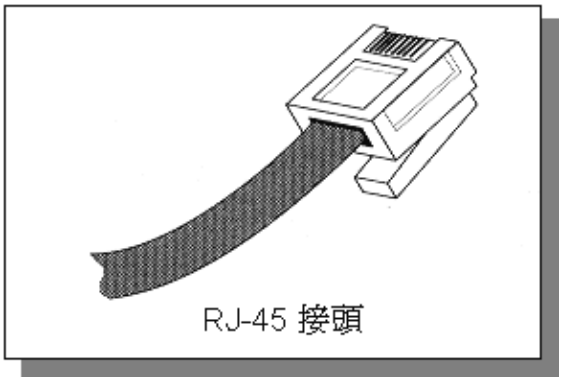

圖 2.2-1、RJ-45 接頭示意圖

而 RJ-45 接頭又因為每條蕊線的對應不同而分為 568A 與 568B 接頭，這兩款接頭內的蕊線對應如下表：

接頭名稱\蕊線順序	1	2	3	4	5	6	7	8
568A	白綠	綠	白橙	藍	白藍	橙	白棕	棕
568B	白橙	橙	白綠	藍	白藍	綠	白棕	棕

事實上，雖然目前的乙太網路線有八蕊且兩兩成對，但實際使用的只有 1,2,3,6 蕊而已， 其他的則是某些特殊用途的場合才會使用到。但由於主機與主機的連線以及主機與集線器的連線時， 所使用的網路線腳位定義並不相同，因此由於接頭的不同網路線又可分為兩種：

- 跳線：一邊為 568A 一邊為 568B 的接頭時稱為跳線，用在直接連結兩部主機的網路卡。
- 平行線：兩邊接頭同為 568A 或同為 568B 時稱為平行線，用在連結主機網路卡與集線器之間的線材；

2.2.3 乙太網路的傳輸協定：CSMA/CD

整個乙太網路的重心就是乙太網路卡啦！所以說，乙太網路的傳輸主要就是網路卡對網路卡之間的資料傳遞而已。 每張乙太網路卡出廠時，就會賦予一個獨一無二的卡號，那就是所謂的 MAC (Media Access Control) 啦！ 理論上，網卡卡號是不能修改的，不過某些筆記型電腦的網卡卡號是能夠修改的呦！ 那麼乙太網路的網卡之間資料是如何傳輸的呢？那就得要談一下 IEEE 802.3 的標準 CSMA/CD (Carrier Sense Multiple Access with Collision Detection) 了！我們以下圖來作為簡介，下圖內的中心點為集線器， 各個主機都是連線到集線器，然後透過集線器的功能向所有主機發起連線的。

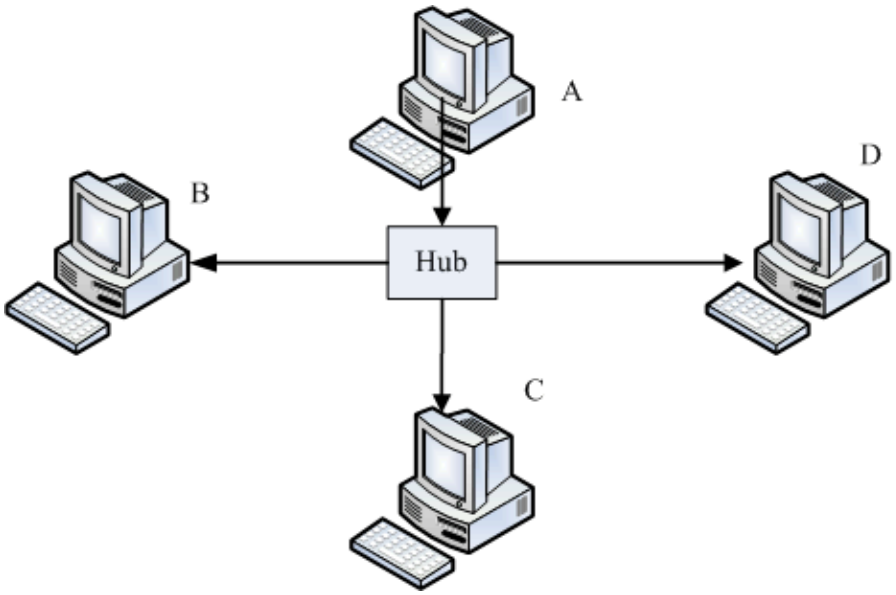

圖 2.2-2、CSMA/CD連線示意圖，由 A 發送資料給 D 時，注意箭頭方向

集線器是一種網路共享媒體，什麼是網路共享媒體啊？想像一下上述的環境就像一個十字路口，而集線器就是那個路口！ 這個路口一次只允許一輛車通過，如果兩輛車同時使用這個路口，那麼就會發生碰撞的車禍事件啊！那就是所謂的共享媒體。 也就是說，網路共享媒體在單一時間點內， 僅能被一部主機所使用。

理解了共享媒體的意義後，再來，我們就得要討論，那麼乙太網路的網卡之間是如何傳輸的呢？我們以上圖中的 A 要發給 D 網卡為例好了，簡單的說， CSMA/CD 搭配上述的環境，它的傳輸情況需要有以下的流程：

1. 監聽媒體使用情況 (Carrier Sense)：A 主機要發送網路封包前，需要先對網路媒體進行監聽，確認沒有人在使用後， 才能夠發送出訊框；
2. 多點傳輸 (Multiple Access)：A 主機所送出的資料會被集線器複製一份，然後傳送給所有連接到此集線器的主機！ 也就是說， A 所送出的資料， B, C, D 三部電腦都能夠接收的到！但由於目標是 D 主機，因此 B 與 C 會將此訊框資料丟棄，而 D 則會抓下來處理；
3. 碰撞偵測 (Collision Detection)：該訊框資料附有檢測能力，若其他主機例如 B 電腦也剛好在同時間發送訊框資料時， 那麼 A 與 B 送出的資料碰撞在一塊 (出車禍) ，此時這些訊框就是損毀，那麼 A 與 B 就會各自隨機等待一個時間， 然後重新透過第一步再傳送一次該訊框資料。

瞭解這個程序很重要嗎？我們就來談談：

- 網路忙碌時，集線器燈號閃個不停，但我的主機明明沒有使用網路：
  透過上述的流程我們會知道，不管哪一部主機發送出訊框，所有的電腦都會接收到！因為集線器會複製一份該資料給所有電腦。 因此，雖然只有一部主機在對外連線，但是在集線器上面的所有電腦燈號就都會閃個不停！

- 我的電腦明明沒有被入侵，為何我的資料會被隔壁的電腦竊取：
  透過上述的流程，我們只要在 B 電腦上面安裝一套監聽軟體，這套軟體將原本要丟棄的訊框資料捉下來分析，並且加以重組， 就能夠知道原本 A 所送出的訊息了。這也是為什麼我們都建議重要資料在網際網路上面得要『加密』後再傳輸！

- 既然共享媒體只有一個主機可以使用，為何大家可以同時上網：
  這個問題就有趣了，既然共享媒體一次只能被一個主機所使用，那麼萬一我傳輸 100MB 的檔案，集線器就得被我使用 80 秒 (以 10Mbps 傳輸時)，在這期間其他人都不可以使用嗎？不是的，由於標準的訊框資料在網路卡與其他乙太網路媒體一次只能傳輸 1500bytes，因此我的 100MB 檔案就得要拆成多個小資料包，然後一個一個的傳送，每個資料包傳送前都要經過 CSMA/CD 的機制。 所以，這個集線器的使用權是大家搶著用的！即使只有一部主機在使用網路媒體時，那麼這部主機在發送每個封包間， 也都是需要等待一段時間的 (96 bit time)！

- 訊框要多大比較好？能不能修改訊框？：
  如上所述，那麼訊框的大小能不能改變呢？因為如果訊框的容量能夠增大，那麼小資料包的數量就會減少， 那每個訊框傳送間的等待就可以減少了！是這樣沒錯，但是乙太網路標準訊框確實定義在 1500 bytes， 但近來的超高速乙太網路媒體有支援 Jumbo frame (巨型訊框,註10) 的話，那麼就能夠將訊框大小改為 9000bytes 哩！但不是很建議大家隨便修改啦！為什麼呢？2.2.5 MTU 那小節再說。

2.2.4 MAC 的封裝格式

上面提到的 CSMA/CD 傳送出去的訊框資料，其實就是 MAC 啦！MAC 其實就是我們上面一直講到的訊框 (frame) 囉！ 只是這個訊框上面有兩個很重要的資料，就是目標與來源的網卡卡號，因此我們又簡稱網卡卡號為 MAC 而已。 簡單的說，你可以把 MAC 想成是一個在網路線上面傳遞的包裹，而這個包裹是整個網路硬體上面傳送資料的最小單位了。 也就是說，網路線可想成是一條『一次僅可通過一個人』的獨木橋， 而 MAC 就是在這個獨木橋上面動的人啦！接下來，來看一看 MAC 這個訊框的內容吧！

前導碼 8 Bytes	目的位址 6 Bytes	來源位址 6 Bytes	資料欄位通訊 2 Bytes	主要資料 46-1500 Bytes	檢查碼 4 Bytes

圖 2.2-3、乙太網路的 MAC 訊框

上圖中的目的位址與來源位址指的就是網卡卡號 (hardware address, 硬體位址)，我們前面提到，每一張網卡都有一個獨一無二的卡號， 那個卡號的目的就在這個訊框的表頭資料使用到啦！硬體位址最小由 00:00:00:00:00:00 到 FF:FF:FF:FF:FF:FF (16 進位法)， 這 6 bytes 當中，前 3bytes 為廠商的代碼，後 3bytes 則是該廠商自行設定的裝置碼了。

在 Linux 當中，你可以使用 ifconfig 這個指令來查閱你的網路卡卡號喔！特別注意，在這個 MAC 的傳送中，他僅在區域網路內生效，如果跨過不同的網域 (這個後面 IP 的部分時會介紹)，那麼來源與目的的硬體位址就會跟著改變了。 這是因為變成不同網路卡之間的交流了嘛！所以卡號當然不同了！如下所示：

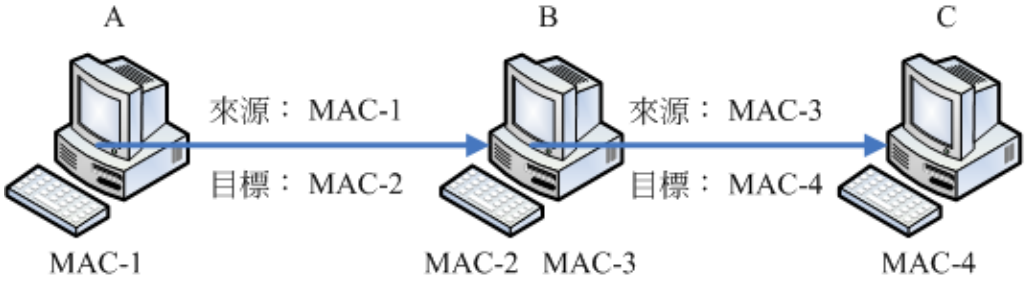

圖 2.2-4、同一訊框在不同網域的主機間傳送時，訊框的表頭變化

例如上面的圖示，我的資料要由電腦 A 通過 B 後才送達 C ，而 B 電腦有兩塊網路卡，其中 MAC-2 與 A 電腦的 MAC-1 互通，至於 MAC-3 則與 C 電腦的 MAC-4 互通。但是 MAC-1 不能與 MAC-3 與 MAC-4 互通，為啥？因為 MAC-1 這塊網路卡並沒有與 MAC-3 及 MAC-4 使用同樣的 switch/hub 相接嘛！所以，資料的流通會變成：

1. 先由 MAC-1 傳送到 MAC-2 ，此時來源是 MAC-1 而目的地是 MAC-2；
2. B 電腦接收後，察看該訊框，發現目標其實是 C 電腦，而為了與 C 電腦溝通， 所以他會將訊框內的來源 MAC 改為 MAC-3 ，而目的改為 MAC-4 ，如此就可以直接傳送到 C 電腦了。

也就是說，只要透過 B (就是路由器) 才將封包送到另一個網域 (IP 部分會講) 去的時候， 那麼訊框內的硬體位址就會被改變，然後才能夠在同一個網域裡面直接進行訊框的流通啊！

Tips:
由於網路卡卡號是跟著網路卡走的，並不會因為重灌作業系統而改變， 所以防火牆軟體大多也能夠針對網路卡來進行抵擋的工作喔！ 不過抵擋網卡僅能在區域網路內進行而已，因為 MAC 不能跨 router 嘛！！

- 為什麼資料量最小要 46 最大為 1500 bytes 呢？

  訊框內的資料內容最大可達 1500bytes 這我們現在知道了，那為何要規範最小資料為 46bytes 呢？這是由於 CSMA/CD 機制所算出來的！ 在這個機制上面可算出若要偵測碰撞，則訊框總資料量最小得要有 64bytes ，那再扣除目的位址、來源位址、檢查碼 (前導碼不算) 後， 就可得到資料量最小得要有 46bytes 了！也就是說，如果妳要傳輸的資料小於 46byes ，那我們的系統會主動的填上一些填充碼， 以補齊至少 46bytes 的容量才行！

2.2.5 MTU 最大傳輸單位

通過上面 MAC 封裝的定義，現在我們知道標準乙太網路訊框所能傳送的資料量最大可以到達 1500 bytes ， 這個數值就被我們稱為 MTU (Maximum Transmission Unit, 最大傳輸單位)。 你得要注意的是，每種網路介面的 MTU 都不相同，因此有的時候在某些網路文章上面你會看到 1492 bytes 的 MTU 等等。不過，在乙太網路上，標準的定義就是 1500 bytes。

在待會兒會介紹到的 IP 封包中，這個 IP 封包最大可以到 65535 bytes，比 MTU 還要大呢！既然禮物 (IP) 都比盒子 (MAC) 大，那怎麼可能放的進去啊？所以囉， IP 封包是可以進行拆解的，然後才能放到 MAC 當中啊！等到資料都傳到目的地， 再由目的地的主機將他組裝回來就是了。所以囉，如果 MTU 能夠大一些的話，那麼 IP 封包的拆解情況就會降低， 封包與封包傳送之間的等待時間 (前一小節提到的 96 bit time) 也會減少，就能夠增加網路頻寬的使用囉！

為了這個目的，所以 Gigabit 的乙太網路媒體才有支援 Jumbo frame 的嘛！這個 Jumbo frame 一般都定義到 9000bytes。 那你會說，既然如此，我們的 MTU 能不能改成 9000bytes 呢？這樣一來不就能夠減少資料封包的拆解，以增加網路使用率嗎？ 是這樣沒錯，而且，你也確實可以在 Linux 系統上更改 MTU 的！但是，如果考量到整個網路，那麼我們不建議你修改這個數值。 為什麼呢？

我們的封包總是需要在 Internet 上面跑吧？你無法確認所有的網路媒體都是支援那麼大的 MTU 對吧！如果你的 9000 bytes 封包通過一個不支援 Jumbo frame 的網路媒體時，好一點的是該網路媒體 (例如 switch/router 等) 會主動的幫你重組而進行傳送，差一點的可能就直接回報這個封包無效而丟棄了～這個時候可就糗大囉～ 所以， MTU 設定為 9000 這種事情，大概僅能在內部網路的環境中作～舉例來說，很多的內部叢集系統 (cluster) 就將他們的內部網路環境 MTU 設定為 9000，但是對外的介面卡可還是原本的標準 1500 喔！ ^_^

也就是說，不論你的網路媒體支援 MTU 到多大，你必須要考量到你的封包需要傳到目的地時， 所需要經過的所有網路媒體，然後再來決定你的 MTU 設定才行。就因為這樣，我們才不建議你修改標準乙太網路的 MTU 嘛！

Tips:

早期某些網路媒體 (例如 IP 分享器) 支援的是 802.2, 802.3 標準所組合成的 MAC 封裝，它的 MTU 就是 1492 ， 而且這些設備可能不會進行封包重組，因此早期網路上面常常有朋友問說，他們連上某些網站時，總是會連線逾時而斷線。 但透過修改用戶端的 MTU 成為 1492 之後，上網就沒有問題了。原因是什麼呢？讀完上頭的資料，您應該能理解了吧？^_^

2.2.6 集線器、交換器與相關機制

- 共不共享很重要，集線器還是交換器？ (註11)

剛剛我們上面提到了，當一個很忙碌的網路在運作時，集線器 (hub) 這個網路共享媒體就可能會發生碰撞的情況， 這是因為 CSMA/CD 的緣故。那有沒有辦法避免這種莫名其妙的封包碰撞情況呢？有的，那就使用非共享媒體的交換器即可啊！

交換器 (switch) 等級非常多，我們這裡僅探討支援 OSI 第二層的交換器。交換器與集線器最大的差異，在於交換器內有一個特別的記憶體， 這個記憶體可以記錄每個 switch port 與其連接的 PC 的 MAC 位址，所以，當來自 switch 兩端的 PC 要互傳資料時，每個訊框將直接透過交換器的記憶體資料而傳送到目標主機上！ 所以 switch 不是共享媒體，且 switch 的每個埠口 (port) 都具有獨立的頻寬喔！

舉例來說，10/100 的 Hub 上連結 5 部主機，那麼整個 10/100Mbps 是分給這五部主機的， 所以這五部主機總共只能使用 10/100Mbps 而已。那如果是 switch 呢？由於『每個 port 都具有 10/100Mbps 的頻寬』， 所以就看你當時的傳輸行為是如何囉！舉例來說，如果是底下的狀況時，每個連線都是 10/100 Mbps 的。

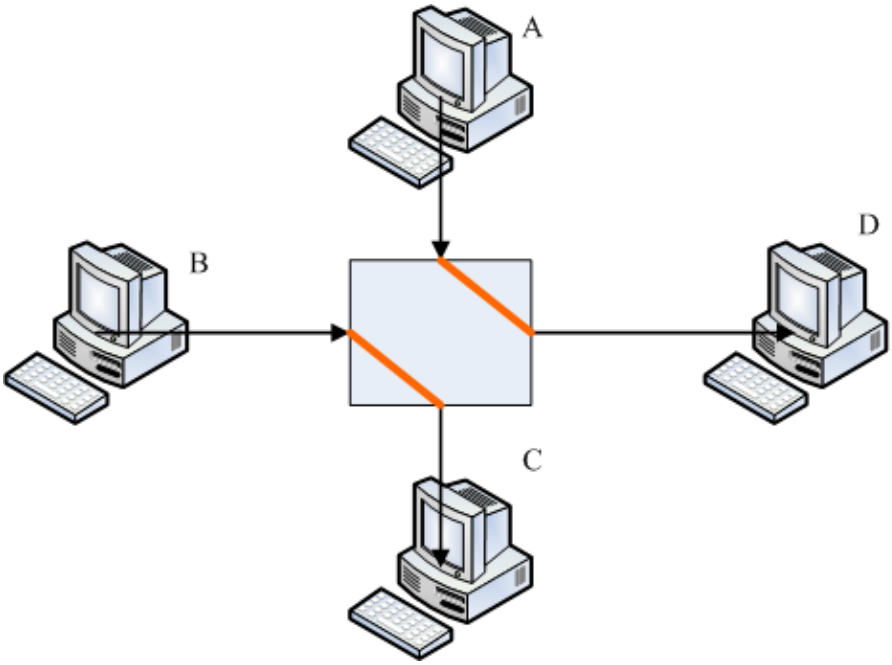

圖 2.2-5、交換器每個埠口的頻寬使用示意圖

A 傳送到 D 與 B 傳送到 C 都獨自擁有 10/100Mbps 的頻寬，兩邊並不會互相影響！ 不過，如果是 A 與 D 都傳給 C 時，由於 C port 就僅有 10/100Mbps ，等於 A 與 D 都需要搶 C 節點的 10/100Mbps 來用的意思。 總之，你就是得要記得的是，switch 已經克服了封包碰撞的問題，因為他有個 switch port 對應 MAC 的相關功能， 所以 switch 並非共享媒體喔！同時需要記得的是，現在的 switch 規格很多， 在選購的時候，千萬記得選購可以支援全雙工/半雙工，以及支援 Jumbo frame 的為佳！

- 什麼是全雙工/半雙工**(full-duplex, half-duplex)**

  前面談到網路線時，我們知道八蕊的網路線實際上僅有兩對被使用，一對是用在傳送，另一對則是在接收。 如果兩端的 PC 同時支援全雙工時，那表示 Input/Output 均可達到 10/100Mbps， 亦即資料的傳送與接收同時均可達到 10/100bps 的意思，總頻寬則可達到 20/200Mbps 囉 (其實是有點語病的，因為 Input 可達 10/100Mbps， output 可達 10/100Mbps ， 而不是 Input 可直接達到 20/200Mbps 喔！)如果你的網路環境想要達到全雙工時， 使用共享媒體的 Hub 是不可能的，因為網路線腳位的關係，無法使用共享媒體來達到全雙工的！ 如果你的 switch 也支援全雙工模式，那麼在 switch 兩端的 PC 才能達到全雙工喔！

- 自動協調速度機制 **(auto-negotiation)**：

  我們都知道現在的乙太網路卡是可以向下支援的，亦即是 Gigabit 網路卡可以與早期的 10/100Mbps 網路卡連結而不會發生問題。但是，此時的網路速度是怎樣判定呢？ 早期的 switch/hub 必須要手動切換速度才行，新的 hub/switch 因為有支援 auto-negotiation 又稱為 N-Way 的功能，他可自動的協調出最高的傳輸速度來溝通喔！如果有 Gigabit 與 10/100Mbps 在 switch 上面， 則 N-Way 會先使用最高的速度 (gigabit) 測試是否能夠全部支援，如果不行的話，就降速到下一個等級亦即 100 Mbps 的速度來運作的！

- 自動分辨網路線跳線或平行線 **(Auto MDI/MDIX)**：

  那麼我們是否需要自行分辨平行線與跳線呢？不需要啦！因為 switch 若含有auto MDI/MDIX 的功能時， 會自動分辨網路線的腳位來調整連線的，所以你就不需要管你的網路線是跳線還是平行線囉！

方便吧！ ^_^

- 訊號衰減造成的問題

由於電子訊號是會衰減的，所以當網路線過長導致電子訊號衰減的情況嚴重時， 就會導致連線品質的不良了。因此，連結各個節點的網路線長度是有限制的喔！ 不過，一般來說，現今的乙太網路 CAT5 等級的網路線大概都可以支援到 100 公尺的長度， 所以應該無庸擔心才是吶！

但是，造成訊號衰減的情況並非僅有網路線長度而已！如果你的網路線折得太嚴重(例如在門邊常常被門板壓，導致變形) ，或者是自行壓製網路線接頭，但是接頭部分的八蕊蕊線纏繞度不足導致電磁干擾嚴重， 或者是網路線放在戶外風吹日曬導致脆化的情況等等，都會導致電子訊號傳遞的不良而造成連線品質惡劣， 此時常常就會發現偶而可以連線、有時卻又無法連線的問題了！因此，當你需要針對企業內部來架設整體的網路時， 注意結構化佈線可是很重要的喔！

- 結構化佈線

所謂的結構化佈線指的是將各個網路的元件分別拆開，分別安裝與布置到企業內部， 則未來想要提升網路硬體等級或者是移動某些網路設備時，只需要更動類似配線盤的機櫃處， 以及末端的牆上預留孔與主機設備的連線就能夠達到目的了。例如底下的圖示：

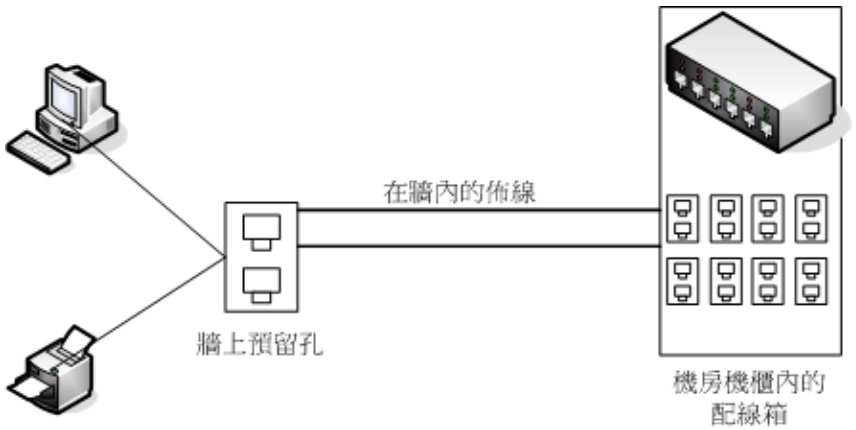

圖 2.2-6、結構化佈線簡易圖示

在牆內的佈線需要很注意，因為可能一佈線完成後就使用 5-10 年以上喔！那你需要注意的僅有末端牆上的預留孔以及配線端部分。 事實上，光是結構化佈線所需要選擇的網路媒體與網路線的等級，還有機櫃、機架，以及美化與隱藏網路線的材料等等的挑選， 以及實際施工所需要注意的事項，還有所有硬體、施工所需要注意的標準規範等等， 已經可以寫滿厚厚一本書，而鳥哥這裡的文章旨在介紹一個中小企業內部主機數量較少的環境， 所以僅提到最簡單的以一個或兩個交換器 (swtich) 串接所有網路設備的小型星形連線狀態而已。

如果你有需要相關硬體結構化佈線的資訊，可以參考風信子兄翻譯的『Swtich and Fast 乙太網路』一書的後半段！至於網路上的高手嗎？你可以前往酷學園請教 ZMAN (http://http://wordpress.morezman.com/) 大哥喔！

2.3 TCP/IP 的網路層相關封包與資料

我們現在知道要有網路的話，必須要有網路相關的硬體，而目前最常見的網路硬體介面為乙太網路，包括網路線、網路卡、Hub/Switch 等等。而乙太網路上面的傳輸使用網路卡卡號為基準的 MAC 訊框，配合 CSMA/CD 的標準來傳送訊框，這就是硬體部分。在軟體部分，我們知道 Internet 其實就是 TCP/IP 這個通訊協定的通稱，Internet 是由 InterNIC(註12) 所統一管理的， 但其實他僅是負責分配 Internet 上面的 IP 以及提供相關的 TCP/IP 技術文件而已。不過 Internet 最重要的就是 IP 啊！所以， 這個小節就讓我們來講講網路層的 IP 與路由吧！

2.3.1 IP 封包的封裝

目前網際網路社會的 IP 有兩種版本，一種是目前使用最廣泛的 IPv4 (Internet Protocol version 4, 網際網路協定第四版)， 一種則是預期未來會熱門的 IPv6 。IPv4 記錄的位址由於僅有 32 位元，預計在 2020 年前後就會分發完畢，如此一來， 新興國家或者是新的網路公司，將沒有網路可以使用。為了避免這個問題發生，因此就有 IPv6 的產生。 IPv6 的位址可以達到 128 位元，可以多出 2 的 96 次方倍的網址數量，這樣的 IP 數量幾乎用不完啦！雖然 IPv6 具有前瞻性，但目前主流媒體大多還是使用 IPv4 ，因此本文主要談到的 IP 都指 IPv4 而言喔！(註13)

我們在前一小節談到 MAC 的封裝，那麼 IP 封包的封裝也得要來瞭解一下，才能知道 IP 到底是如何產生的啊！ IP 封包可以達到 65535 bytes 這麼大，在比 MAC 大的情況下，我們的作業系統會對 IP 進行拆解的動作。至於 IP 封裝的表頭資料繪製如下：(下圖第一行為每個欄位的 **bit** 數)

<table>
<tr><td>4 bits</td><td>4 bits</td><td>8 bits</td><td>3 bits</td><td>13 bits</td></tr>
<tr><td>Version</td><td>IHL</td><td>Type of Service</td><td colspan="2">Total Length</td></tr>
<tr><td colspan="3">Identification</td><td>Flags</td><td>Fragmentation Offset</td></tr>
<tr><td colspan="2">Time To Live</td><td>Protocol</td><td colspan="2">Header Checksum</td></tr>
<tr><td colspan="5">Source Address</td></tr>
<tr><td colspan="5">Destination Address</td></tr>
<tr><td colspan="4">Options</td><td>Padding</td></tr>
<tr><td colspan="5">Data</td></tr>
</table>

圖 2.3-1、IP 封包的表頭資料

在上面的圖示中有個地方要注意，那就是『每一行所佔用的位元數為 32 bits』， 各個表頭的內容分別介紹如下：

- Version(版本)
  宣告這個 IP 封包的版本，例如目前慣用的還是 IPv4 這個版本就在這裡宣告。

- IHL(Internet Header Length, IP表頭的長度)
  告知這個 IP 封包的表頭長度，使用的單位應該是字組 (word) ，一個字組為 4bytes 大小喔。

- Type of Service(服務類型)
  這個項目的內容為『PPPDTRUU』，表示這個 IP 封包的服務類型，主要分為：
  PPP：表示此 IP 封包的優先度，目前很少使用；

D：若為 0 表示一般延遲(delay)，若為 1 表示為低延遲；
T：若為 0 表示為一般傳輸量 (throughput)，若為 1 表示為高傳輸量；
R：若為 0 表示為一般可靠度(reliability)，若為 1 表示高可靠度。
UU：保留尚未被使用。
舉例來說，gigabit 乙太網路的種種相關規格可以讓這個 IP 封包加速且降低延遲，某些特殊的標誌就是在這裡說明的。

- Total Length(總長度)
  指這個 IP 封包的總容量，包括表頭與內容 (Data) 部分。最大可達 65535 bytes。

- Identification(辨別碼)
  我們前面提到 IP 袋子必須要放在 MAC 袋子當中。不過，如果 IP 袋子太大的話，就得先要將 IP 再重組成較小的袋子然後再放到 MAC 當中。而當 IP 被重組時，每個來自同一個 IP 的小袋子就得要有個識別碼以告知接收端這些小袋子其實是來自同一個 IP 封包才行。 也就是說，假如 IP 封包其實是 65536 那麼大 (前一個 Total Length 有規定)， 那麼這個 IP 就得要再被分成更小的 IP 分段後才能塞進 MAC 訊框中。那麼每個小 IP 分段是否來自同一個 IP 資料，呵呵！那就是這個識別碼的功用啦！

- Flags(特殊旗標)
  這個地方的內容為『0DM』，其意義為：
  D：若為 0 表示可以分段，若為 1 表示不可分段
  M：若為 0 表示此 IP 為最後分段，若為 1 表示非最後分段。

- Fragment Offset(分段偏移)
  表示目前這個 IP 分段在原始的 IP 封包中所佔的位置。就有點像是序號啦，有這個序號才能將所有的小 IP 分段組合成為原本的 IP 封包大小嘛！透過 Total Length, Identification, Flags 以及這個 Fragment Offset 就能夠將小 IP 分段在收受端組合起來囉！

- **Time To Live(TTL, 存活時間)**
  表示這個 IP 封包的存活時間，範圍為 0-255。當這個 IP 封包通過一個路由器時， TTL 就會減一，當 TTL 為 0 時，這個封包將會被直接丟棄。說實在的，要讓 IP 封包通過 255 個路由器，還挺難的～ ^_^

- **Protocol Number(協定代碼)**
  來自傳輸層與網路層本身的其他資料都是放置在 IP 封包當中的，我們可以在 IP 表頭記載這個 IP 封包內的資料是啥， 在這個欄位就是記載每種資料封包的內容啦！在這個欄位記載的代碼與相關的封包協定名稱如下所示：

IP 內的號碼	封包協定名稱(全名)
1	ICMP (Internet Control Message Protocol)
2	IGMP (Internet Group Management Protocol)
3	GGP (Gateway-to-Gateway Protocol)
4	IP (IP in IP encapsulation)
6	TCP (Transmission Control Protocol)
8	EGP (Exterior Gateway Protocol)

17	UDP (User Datagram Protocol)

當然啦，我們比較常見到的還是那個 TCP, UDP, ICMP 說！

- Header Checksum(表頭檢查碼)
  用來檢查這個 IP 表頭的錯誤檢驗之用。

- **Source Address**
  還用講嗎？當然是來源的 IP 位址，從這裡我們也知道 IP 是 32 位元喔！

- **Destination Address**
  有來源還需要有目標才能傳送，這裡就是目標的 IP 位址。

- Options (其他參數)
  這個是額外的功能，提供包括安全處理機制、路由紀錄、時間戳記、嚴格與寬鬆之來源路由等。

- Padding(補齊項目)
  由於 Options 的內容不一定有多大，但是我們知道 IP 每個資料都必須要是 32 bits，所以，若 Options 的資料不足 32 bits 時，則由 padding 主動補齊。

你只要知道 IP 表頭裡面含有： TTL, Protocol, 來源位址與目標位址也就夠了！而這個 IP 表頭的來源與目標 IP ，以及那個判斷通過多少路由器的 TTL ，就能瞭解到這個 IP 將被如何傳送到目的端吶。後續各小節我們將介紹 IP 的組成與範圍，還有 IP 封包如何傳送的機制 (路由) 等等。

2.3.2 IP 位址的組成與分級

現在我們知道 IP (Internet Protocol) 其實是一種網路封包，而這個封包的表頭最重要的就是那個 32 位元的來源與目標位址！ 為了方便記憶，所以我們也稱這個 32 bits 的數值為 IP 網路位址就是了。因為網路是人類發明的，所以很多概念與郵務系統類似！ 那這個 IP 其實就類似所謂的『門牌號碼』啦！那麼這個 IP 有哪些重要的地方需要瞭解的呢？底下我們就來談一談吧！

既然 IP 的組成是 32 bits 的數值，也就是由 32 個 0 與 1 組成的一連串數字！那麼當我們思考所有跟 IP 有關的參數時，你就應該要將該參數想成是 32 位元的資料喔！ 不過，因為人類對於二進位實在是不怎麼熟悉，所以為了順應人們對於十進位的依賴性，因此，就將 32 bits 的 IP 分成四小段，每段含有 8 個 bits ，將 8 個 bits 計算成為十進位，並且每一段中間以小數點隔開，那就成了目前大家所熟悉的 IP 的書寫模樣了。如下所示：

```
IP 的表示式：
00000000.00000000.00000000.00000000    ==> 0.0.0.0
11111111.11111111.11111111.11111111    ==> 255.255.255.255
```

所以 IP 最小可以由 0.0.0.0 一直到 255.255.255.255 哩！但在這一串數字中，其實還可以分為兩個部分喔！ 主要分為 Net_ID (網域號碼)與 Host_ID (主機號碼) 兩部份。我們先以 192.168.0.0 ~ 192.168.0.255 這個 Class C 的網域當作例子來說明好了：

```
192.168.0.0~192.168.0.255 這個 Class C 的說明：
11000000.10101000.00000000.00000000
```

```
11000000.10101000.00000000.11111111
|----------Net_ID---------|-host--|
```

在上面的範例當中，前面三組數字 (192.168.0) 就是網域號碼，最後面一組數字則稱為主機號碼。 至於同一個網域的定義是『在同一個物理網段內，主機的 IP 具有相同的 Net_ID ，並且具有獨特的 Host_ID』，那麼這些 IP 群就是同一個網域內的 IP 網段啦！

Tips:
什麼是物理網段呢？當所有的主機都是使用同一個網路媒體串在一起， 這個時候這些主機在實體裝置上面其實是連線在一起的，那麼就可以稱為這些主機在同一個物理網段內了！ 同時並請注意，同一個物理網段之內，可以依據不同的 IP 的設定，而設定成多個『IP 網段』喔！

上面例子當中的 192.168.0.0, 192.168.0.1, 192.168.0.2, ..., 192.168.0.255 (共 256 個) 這些 IP 就是同一個網域內的 IP 群(同一個網域也稱為同一個網段！)，請注意，同一個 Net_ID 內，不能具有相同的 Host_ID ，否則就會發生 IP 衝突，可能會造成兩部主機都沒有辦法使用網路的問題！

- **IP** 在同一網域的意義

那麼同一個網域該怎麼設定，與將 IP 設定在同一個網域之內有什麼好處呢？

- Net_ID 與 Host_ID 的限制：
  在同一個網段內，Net_ID 是不變的，而 Host_ID 則是不可重複，此外，Host_ID 在二進位的表示法當中，不可同時為 0 也不可同時為 1 ，因為全為 0 表示整個網段的位址 (Network IP)，而全為 1 則表示為廣播的位址 (Broadcast IP)。例如上面的例子當中，192.168.0.0 (Host_ID 全部為 0)以及 192.168.0.255 (Host_ID 全部為 1) 不可用來作為網段內主機的 IP 設定，也就是說，這個網段內可用來設定主機的 IP 是由 192.168.0.1 到 192.168.0.254；

- 在區網內透過 IP 廣播傳遞資料
  在同物理網段的主機如果設定相同的網域 IP 範圍 (不可重複)，則這些主機都可以透過 CSMA/CD 的功能直接在區網內用廣播進行網路的連線，亦即可以直接網卡對網卡傳遞資料 (透過 MAC 訊框)；

- 設定不同區網在同物理網段的情況
  在同一個物理網段之內，如果兩部主機設定成不同的 IP 網段，則由於廣播位址的不同，導致無法透過廣播的方式來進行連線。 此時得要透過路由器 (router) 來進行溝通才能將兩個網域連結在一起。

- 網域的大小
  當 Host_ID 所佔用的位元越大，亦即 Host_ID 數量越多時，表示同一個網域內可用以設定主機的 IP 數量越多。

所以說，貴單位公司內的電腦群，或者是你宿舍或家裡面的所有電腦，當然都設定在同一個網域內是最方便的， 因為如此一來每一部電腦都可以直接透過 MAC 來進行資料的交流，而不必經由 Router (路由器) 來進行封包的轉遞呢！(Router 這部份在第八章才會提及)。

- **IP** 與門牌號碼的聯想

剛接觸到 IP 組成的朋友都很困擾，又分啥網域號碼與主機號碼，煩死了！其實，你不用煩惱啊！使用門牌號碼的概念來想即可。 既然 IP 是門牌，那拿我們崑山科技大學的門牌來說好了，我們的門牌是：『台南市永康區大灣路 949 號』， 假設整個大灣路是同一個巷弄，那麼我們這個門牌的網域號碼『台南市永康區大灣路』而我的主機號碼就是『 949 號』， 那麼整條大灣路上面只要是開頭為『台南市永康區大灣路』的，就是跟我們同一個網域囉！當然啦，門牌號碼不可能有第二個 949 號啊！這樣理解否？

另外，Host_ID 全為 0 與全為 1 (二進位的概念) 時，代表整條巷子的第一個與最後一個門牌，而第一個門牌我們讓他代表整條巷子， 所以又稱為 Network IP，就是巷子口那個 XXX 巷的立牌啦！至於最後一個 IP ，則代表巷子尾，亦即本條巷子的最後一個門牌， 那就是我們在巷子內廣播時的最後一個 IP ，又稱為 Broadcast IP 的囉。

在我們這個巷子內，我們可以透過大聲公用廣播的方式跟大家溝通訊息，例如前幾年很熱門的張君雅小妹妹的泡麵廣告， 在巷子內透過廣播告訴張君雅小妹妹，你阿嬤將泡麵煮好了，趕快回家吃麵去！那如果不是張君雅小妹妹呢？就將該訊息略過啊！ 這樣有沒有聯想到 CSMA/CD 的概念呢？

那如果你的資料不是要給本巷子內的門牌呢？此時你就得要將資料拿給巷子內的郵局 (路由器)，由郵局幫你傳送， 你只要知道巷子內的那間郵局在哪裡即可，其他的就讓郵局自己幫你把信件傳出去即可啊！這就是整個區網與門牌對應的想法！ 這樣有沒有比較清晰啊？

- **IP 的分級**

你應該要想到一個問題，那就是我的總門牌『台南市永康區大灣路 949 號』中，到哪裡是巷子而到哪裡是門牌？ 如果到『台南市』是巷子，那麼我的門牌將有好多鄉鎮的組成，如果巷子號碼到『台南市永康區』時， 那麼我們的門牌就又少了點。所以說，這個『巷子』的大小，將會影響到我們主機號碼的數量！

為了解決這個問題，以及為了 IP 管理與發放註冊的方便性，InterNIC 將整個 IP 網段分為五種等級， 每種等級的範圍主要與 IP 那 32 bits 數值的前面幾個位元有關，基本定義如下：

```
以二進位說明 Network 第一個數字的定義：
Class A : 0xxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx   ==> NetI_D 的開頭是 0
     |--net--|---------host-----------|
Class B : 10xxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx   ==> NetI_D 的開頭是 10
     |------net-------|------host------|
Class C : 110xxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx   ==> NetI_D 的開頭是 110
     |-----------net-----------|-host--|
Class D : 1110xxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx   ==> NetI_D 的開頭是 1110
Class E : 1111xxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx   ==> NetI_D 的開頭是 1111

五種分級在十進位的表示：
Class A :  0.xx.xx.xx ~ 127.xx.xx.xx
Class B : 128.xx.xx.xx ~ 191.xx.xx.xx
Class C : 192.xx.xx.xx ~ 223.xx.xx.xx
Class D : 224.xx.xx.xx ~ 239.xx.xx.xx
Class E : 240.xx.xx.xx ~ 255.xx.xx.xx
```

根據上表的說明，我們可以知道，你只要知道 IP 的第一個十進位數字，就能夠約略瞭解到該 IP 屬於哪一個等級， 以及同網域 IP 數量有多少。這也是為啥我們上頭選了 192.168.0.0 這一 IP 網段來說明時，會將巷子定義到第三個數字之故。 不過，上表中你只要記憶三種等級，亦即是 Class A, B, C 即可，因為 Class D 是用來作為群播 (multicast) 的特殊功能之用 (最常用在大批電腦的網路還原)，至於 Class E

則是保留沒有使用的網段。因此，能夠用來設定在一般系統上面的，就只有 Class A, B, C 三種等級的 IP 囉！

2.3.3 IP 的種類與取得方式

接下來要跟大家談一談也是很容易造成大家困擾的一個部分，那就是 IP 的種類！很多朋友常常聽到什麼『真實IP, 實體 IP, 虛擬 IP, 假的 IP....』煩都煩死了～其實不要太緊張啦！實際上，在 IPv4 裡面就只有兩種 IP 的類別，分別是：

- **Public IP** : 公共 IP ，經由 INTERNIC 所統一規劃的 IP，有這種 IP 才可以連上 Internet ；
- **Private IP** : 私有 IP 或保留 IP，不能直接連上 Internet 的 IP ， 主要用於區域網路內的主機連線規劃。

早在 IPv4 規劃的時候就擔心 IP 會有不足的情況，而且為了應付某些企業內部的網路設定，於是就有了私有 IP (Private IP) 的產生了。私有 IP 也分別在 A, B, C 三個 Class 當中各保留一段作為私有 IP 網段，那就是：

- **Class A：10.0.0.0 - 10.255.255.255**
- **Class B：172.16.0.0 - 172.31.255.255**
- **Class C：192.168.0.0 - 192.168.255.255**

由於這三段 Class 的 IP 是預留使用的，所以並不能直接作為 Internet 上面的連接之用，不然的話，到處就都有相同的 IP 囉！那怎麼行！網路豈不混亂？所以囉，這三個 IP 網段就只做為內部私有網域的 IP 溝通之用。簡單的說，他有底下的幾個限制：

- 私有 IP 的路由資訊不能對外散播 (只能存在內部網路)；
- 使用私有 IP 作為來源或目的地址的封包，不能透過 Internet 來轉送 (不然網路會混亂)；
- 關於私有 IP 的參考紀錄(如 DNS)，只能限於內部網路使用 (一樣的原理啦)

這個私有 IP 有什麼好處呢？由於他的私有路由不能對外直接提供資訊，所以，你的內部網路將不會直接被 Internet 上面的 Cracker 所攻擊！但是，你也就無法以私有 IP 來『直接上網』囉！因此相當適合一些尚未具有 Public IP 的企業內部用來規劃其網路之設定！否則當你隨便指定一些可能是 Public IP 的網段來規劃你企業內部的網路設定時，萬一哪一天真的連上 Internet 了，那麼豈不是可能會造成跟 Internet 上面的 Public IP 相同了嗎？

此外，在沒有可用的公開網路情況下，如果你想要跟同學玩連線遊戲怎辦？也就是說，在區網內自己玩自己的連線遊戲， 此時你只要規範好所有同學在同一段私有 IP 網段中，就能夠順利的玩你的網路啦！就這麼簡單呢！

那麼萬一你又要將這些私有 IP 送上 Internet 呢？這個簡單，設定一個簡單的防火牆加上 NAT (Network Address Transfer) 服務，你就可以透過 IP 偽裝 (不要急，這個在後面也會提到) 來使你的私有 IP 的電腦也可以連上 Internet 囉！

- 特殊的 **loopback IP** 網段

好了，那麼除了這個預留的 IP 網段的問題之外，還有沒有什麼其他的怪東西呢？當然是有啦！不然鳥哥幹嘛花時間來唬 XX 呢？沒錯，還有一個奇怪的 Class A 的網域，那就是 lo 這個奇怪的網域啦 (注

意：是小寫的 o 而不是零喔)！這個 lo 的網路是當初被用來作為測試作業系統內部迴圈所用的一個網域，同時也能夠提供給系統內部原本就需要使用網路介面的服務 (daemon) 所使用。

簡單的說，如果你沒有安裝網路卡在的機器上面， 但是你又希望可以測試一下在你的機器上面設定的伺服器環境到底可不可以順利運作，這個時候怎麼辦， 嘿嘿！就是利用這個所謂的內部迴圈網路啦！這個網段在 127.0.0.0/8 這個 Class A，而且預設的主機 (localhost) 的 IP 是 127.0.0.1 呦！所以囉，當你啟動了你的 WWW 伺服器，然後在你的主機的 X-Window 上面執行 http://localhost 就可以直接看到你的主頁囉！而且不需要安裝網路卡呢！測試很方便吧！

此外，你的內部使用的 mail 怎麼運送郵件呢？例如你的主機系統如何 mail 給 root 這個人呢？嘿嘿！也就是使用這一個內部迴圈啦！當要測試你的 TCP/IP 封包與狀態是否正常時，可以使用這個呦！(所以哪一天有人問你嘿！你的主機上面沒有網路卡， 那麼你可以測試你的 WWW 伺服器設定是否正確嗎？這個時候可得回答：當然可以囉！使用 127.0.0.1 這個 Address 呀！ ^_^)

- **IP 的取得方式**

談完了 IP 的種類與等級還有相關的子網域概念後，接下來我們得來瞭解一下，那麼主機的 IP 是如何設定的呢？ 基本上，主機的 IP 與相關網域的設定方式主要有：

- 直接手動設定(static)： 你可以直接向你的網管詢問可用的 IP 相關參數，然後直接編輯設定檔 (或使用某些軟體功能) 來設定你的網路。 常見於校園網路的環境中，以及向 ISP 申請固定 IP 的連線環境；

- 透過撥接取得： 向你的 ISP 申請註冊，取得帳號密碼後，直接撥接到 ISP ，你的 ISP 會透過他們自己的設定，讓你的作業系統取得正確的網路參數。 此時你並不需要手動去編輯與設定相關的網路參數啦。目前台灣的 ADSL 撥接、光纖到大樓、光纖到府等，大部分都是使用撥接的方式。 為因應用戶的需求，某些 ISP 也提供很多不同的 IP 分配機制。包括 hinet, seednet 等等都有提供 ADSL 撥接後取得固定 IP 的方式喔！ 詳情請向你的 ISP 洽詢。

- 自動取得網路參數 (DHCP)： 在區域網路內會有一部主機負責管理所有電腦的網路參數，你的網路啟動時就會主動向該伺服器要求 IP 參數， 若取得網路相關參數後，你的主機就能夠自行設定好所有伺服器給你的網路參數了。最常使用於企業內部、IP 分享器後端、 校園網路與宿舍環境，及纜線寬頻等連線方式。

不管是使用上面哪種方式取得的 IP ，你的 IP 都只有所謂的『Public 與 Private IP 』而已！而其他什麼浮動式、固定制、 動態式等等有的沒有的，就只是告訴你這個 IP 取得的方式而已。舉例來說，台灣地區 ADSL 撥接後取得的 IP 通常是 public IP， 但是鳥哥曾接到香港網友的來信，他們 ADSL 撥接後，取得的 IP 是 Private ，所以導致無法架設網站喔！

2.3.4 Netmask, 子網路與 CIDR (Classless Interdomain Routing)

我們前面談到 IP 是有等級的，而設定在一般電腦系統上面的則是 Class A, B, C。現在我們來想一想，如果我們設定一個區網， 使用的是 Class A ，那麼我們很容易就會想到，哪有這麼多電腦可以設定在同一個 Class A 的區段內 (256x256x256-2=16777214) ？ 而且，假設真有這麼多電腦好了，回想一下 CSMA/CD 吧，你的網路恐怕會一直非常停頓，因為妳得要接到一千多萬台電腦對你的廣播... 光是想到一千多萬台的廣播，你的網路還能使用嗎？真沒效率！

此外，分為 Class 的 IP 等級，是為了管理方面的考量，事實上，我們不可能將一個 Class A 僅劃定為一個區網。舉例來說， 我們崑山取得的 Public IP 是 120.xxx 開頭的，但是其實我們只有 120.114.xxx.xxx 而已，並沒有取得整個 Class A 喔！ 因為我們學校也用不了這麼多嘛！這個時候，我們就得要理解一下囉，就是，怎麼將 Class A 的網段變小？換句話說， 我們如何將網域切的更細呢？這樣不就可以分出更多段的區網給大家設定了？

前面我們提到 IP 這個 32 位元的數值中分為網域號碼與主機號碼，其中 Class C 的網域號碼佔了 24 位元，而其實我們還可以將這樣的網域切的更細，就是讓第一個 Host_ID 被拿來作為 Net_ID ，所以，整個 Net_ID 就有 25 bits ，至於 Host_ID 則減少為 7 bits 。在這樣的情況下，原來的一個 Class C 的網域就可以被切分為兩個子網域，而每個子網域就有『256/2 - 2 = 126』個可用的 IP 了！這樣一來，就能夠將原本的一個網域切為兩個較細小的網域，方便分門別類的設計喔。

- **Netmask, 或稱為 Subnet mask (子網路遮罩)**

那到底是什麼參數來達成子網路的切分呢？那就是 Netmask (子網路遮罩) 的用途啦！這個 Netmask 是用來定義出網域的最重要的一個參數了！不過他也最難理解了～@_@。為了幫助大家比較容易記憶住 Netmask 的設定依據，底下我們介紹一個比較容易記憶的方法。同樣以 192.168.0.0 ~ 192.168.0.255 這個網域為範例好了，如下所示，這個 IP 網段可以分為 Net_ID 與 Host_ID，既然 Net_ID 是不可變的，那就假設他所佔據的 bits 已經被用光了 (全部為 1)，而 Host_ID 是可變的，就將他想成是保留著 (全部為 0)，所以， Netmask 的表示就成為：

```
192.168.0.0~192.168.0.255 這個 C Class 的 Netmask 說明
第一個 IP： 11000000.10101000.00000000.00000000
最後一個 ： 11000000.10101000.00000000.11111111
       |----------Net_ID---------|-host--|
Netmask ： 11111111.11111111.11111111.00000000 <== Netmask 二進位
      ： 255 . 255 . 255 . 0    <== Netmask 十進位
特別注意喔，netmask 也是 32 位元，在數值上，位於 Net_ID 的為 1 而 Host_ID 為 0
```

將他轉成十進位的話，就成為『255.255.255.0』啦！ 這樣記憶簡單多了吧！照這樣的記憶方法，那麼 A, B, C Class 的 Netmask 表示就成為這樣：

```
Class A, B, C 三個等級的 Netmask 表示方式：
Class A : 11111111.00000000.00000000.00000000 ==> 255.  0.  0.  0
Class B : 11111111.11111111.00000000.00000000 ==> 255.255.  0.  0
Class C : 11111111.11111111.11111111.00000000 ==> 255.255.255.  0
```

所以說， 192.168.0.0 ~ 192.168.0.255 這個 Class C 的網域中，他的 Netmask 就是 255.255.255.0 ！再來，我們剛剛提到了當 Host_ID 全部為 0 以及全部為 1 的時後該 IP 是不可以使用的，因為 Host_ID 全部為 0 的時後，表示 IP 是該網段的 Network ，至於全部為 1 的時後就表示該網段最後一個 IP ，也稱為 Broadcast ，所以說，在 192.168.0.0 ~ 192.168.0.255 這個 IP 網段裡面的相關網路參數就有：

```
Netmask:   255.255.255.0    <==網域定義中，最重要的參數
Network:   192.168.0.0      <==第一個 IP
Broadcast: 192.168.0.255    <==最後一個 IP
可用以設定成為主機的 IP 數：
192.168.0.1 ~ 192.168.0.254
```

- 子網路切分

好了，剛剛提到 Class C 還可以繼續進行子網域 (Subnet) 的切分啊，以 192.168.0.0 ~192.168.0.255 這個情況為例，他要如何再細分為兩個子網域呢？我們已經知道 Host_ID 可以拿來當作 Net_ID，那麼 Net_ID 使用了 25 bits 時，就會如下所示：

```
原本的 C Class 的 Net_ID 與 Host_ID 的分別
11000000.10101000.00000000.00000000        Network:   192.168.0.0
11000000.10101000.00000000.11111111        Broadcast: 192.168.0.255
|----------Net_ID---------|-host--|

切成兩個子網路之後的 Net_ID 與 Host_ID 為何？
11000000.10101000.00000000.0 0000000   多了一個 Net_ID 了, 為 0 (第一個子網)
11000000.10101000.00000000.1 0000000   多了一個 Net_ID 了, 為 1 (第二個子網)
|----------Net_ID-----------|-host--|

第一個子網路
Network:    11000000.10101000.00000000.0 0000000    192.168.0.0
Broadcast:  11000000.10101000.00000000.0 1111111    192.168.0.127
            |----------Net_ID-----------|-host-|
Netmask:    11111111.11111111.11111111.1 0000000    255.255.255.128

第二個子網路
Network:    11000000.10101000.00000000.1 0000000    192.168.0.128
Broadcast:  11000000.10101000.00000000.1 1111111    192.168.0.255
            |----------Net_ID-----------|-host-|
Netmask:    11111111.11111111.11111111.1 0000000    255.255.255.128
```

所以說，當再細分下去時，就會得到兩個子網域，而兩個子網域還可以再細分下去喔 (Net_ID 用掉 26 bits ....)。呵呵！如果你真的能夠理解 IP, Network, Broadcast, Netmask 的話，恭喜你，未來的伺服器學習之路已經順暢了一半啦！ ^_^

例題：

試著計算出 172.16.0.0，但 Net_ID 佔用 23 個位元時，這個網域的 Netmask, Network, Broadcast 等參數

答：

由於 172.16.xxx.xxx 是在 Class B 的等級當中，亦即 Net_ID 是 16 位元才對。不過題目給的 Net_ID 佔用了 23 個位元喔！ 等於是向 Host_ID 借了 (23-16) 7 個位元用在 Net_ID 當中。所以整個 IP 的位址會變成這樣：

```
預設：        172   .  16    .0000000 0.00000000
           |----Net_ID--------------|--Host---|
Network:     172   .  16    .0000000 0.00000000   172.16.0.0
Broadcast:   172   .  16    .0000000 1.11111111   172.16.1.255
Netmask:  11111111.11111111.1111111 0.00000000   255.255.254.0
```

鳥哥在這裡有偷懶，因為這個 IP 段的前 16 個位元不會被改變，所以並沒有計算成二進位 (172.16)， 真是不好意思啊～至於粗體部分則是代表 host_ID 啊！

其實子網路的計算是有偷吃步的，我們知道 IP 是二進位，每個位元就是 2 的次方。又由於 IP 數量都是平均分配到每個子網路去， 所以，如果我們以 192.168.0.0 ~ 192.168.0.255 這個網段來說，要是給予

Net_ID 是 26 位元時，總共分為幾段呢？ 因為 26-24=2 ，所以總共用掉兩個位元，因此有 2 的 2 次方，得到 4 個網段。再將 256 個 IP 平均分配到 4 個網段去， 那我們就可以知道這四個網段分別是：

- 192.168.0.0~192.168.0.63
- 192.168.0.64~192.168.0.127
- 192.168.0.128~192.168.0.191
- 192.168.0.192~192.168.0.255

有沒有變簡單的感覺啊？那你再想想，如果同樣一個網段，那 Net_ID 變成 27 個位元時，又該如何計算呢？自己算算看吧！

- **無層級 IP： CIDR (Classless Interdomain Routing)**

一般來說，如果我們知道了 Network 以及 Netmask 之後，就可以定義出該網域的所有 IP 了！因為由 Netmask 就可以推算出來 Broadcast 的 IP 啊！因此，我們常常會以 Network 以及 Netmask 來表示一個網域，例如這樣的寫法：

```
Network/Netmask
192.168.0.0/255.255.255.0
192.168.0.0/24        <==因為 Net_ID 共有 24 個 bits
```

另外，既然 Netmask 裡面的 Net_ID 都是 1 ，那麼 Class C 共有 24 bits 的 Net_ID ，所以啦，就有類似上面 192.168.0.0/24 這樣的寫法囉！這就是一般網域的表示方法。 同理可證，在上述的偷吃步計算網域方法中，四個網段的寫法就可以寫成：

- 192.168.0.0/26
- 192.168.0.64/26
- 192.168.0.128/26
- 192.168.0.192/26

事實上，由於網路細分的情況太嚴重，為了擔心路由資訊過於龐大導致網路效能不佳，因此，某些特殊情況下， 我們反而是將 Net_ID 借用來作為 Host_ID 的情況！這樣就能夠將多個網域寫成一個啦！舉例來說，我們將 256 個 Class C 的私有 IP (192.168.0.0~192.168.255.255) 寫成一個路由資訊的話，那麼這個網段的寫法就會變成： 192.168.0.0/16，反而將 192 開頭的 Class C 變成 class B 的樣子了！ 這種打破原本 IP 代表等級的方式 (透過 Netmask 的規範) 就被稱為無等級網域間路由 (CIDR) 囉！ (註14)

老實說，你無須理會啥是無等級網域間路由啦！只要知道，那個 Network/Netmask 的寫法，通常就是 CIDR 的寫法！ 然後，你也要知道如何透過 Netmask 去計算出 Network, Broadcast 及可用的 IP 等，那你的 IP 概念就相當完整了！^_^

2.3.5 路由概念

我們知道在同一個區網裡面，可以透過 IP 廣播的方式來達到資料傳遞的目的。但如果是非區網內的資料呢？ 這時就得要透過那個所謂的郵局 (路由器) 的幫忙了！這也是網路層非常重要的概念喔！先來看看什麼是區網吧！

例題：

請問 192.168.10.100/25 與 192.168.10.200/25 是否在同一個網域內？

答：

如果經過計算，會發現 192.168.10.100 的 Network 為 192.168.10.0 ，但是 192.168.10.200 的 Network 卻是 192.168.10.128，由於 Net_ID 不相同，所以當然不在同一個網段內！ 關於 Network 與 Netmask 的算法則請參考上一小節。

如上題所述，那麼這兩個網段的資料無法透過廣播來達到資料的傳遞啊，那怎辦？ 此時就得要經過 IP 的路徑選擇 (routing) 功能啦！我們以下面圖示的例子來做說明。 下列圖示當中共有兩個不同的網段，分別是 Network A 與 Network B，這兩個網段是經由一部路由器 (Server A) 來進行資料轉遞的，好了，那麼當 PC01 這部主機想要傳送資料到 PC11 時， 他的 IP 封包該如何傳輸呢？

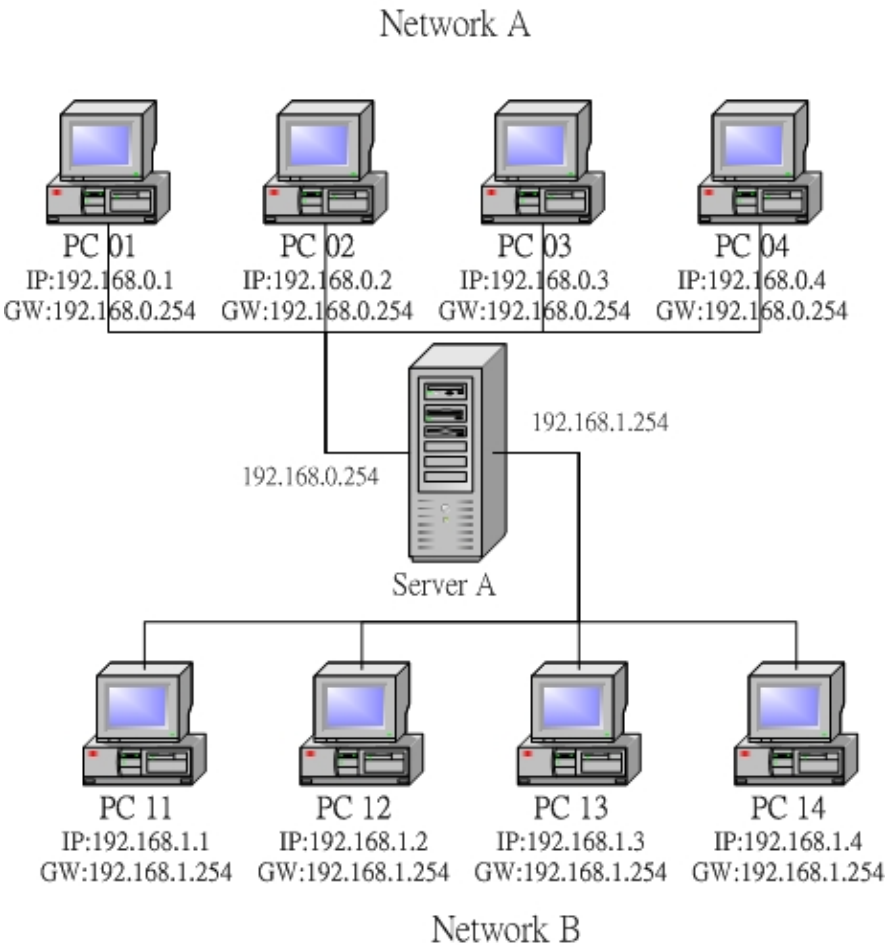

圖 2.3-2、簡易的路由示意圖

我們知道 Network A(192.168.0.0/24) 與 Network B(192.168.1.0/24) 是不同網段，所以 PC01 與 PC11 是不能直接互通資料的。不過， PC01 與 PC11 是如何知道他們兩個不在同一個網段內？這當然是透過 Net_ID 來發現的！那麼當主機想要傳送資料時，他主要的參考是啥？ 很簡單！是『路由表 (route table)』，每部主機都有自己的路由表』， 讓我們來看一看預設的情況下， PC01 要如何將資料傳送到 PC02 呢？

1. 查詢 IP 封包的目標 IP 位址：
   當 PC01 有 IP 封包需要傳送時，主機會查閱 IP 封包表頭的目標 IP 位址；

2. 查詢是否位於本機所在的網域之路由設定：
   PC01 主機會分析自己的路由表，當發現目標 IP 與本機 IP 的 Net_ID 相同時(同一網域)，則 PC01

會直接透過區網功能，將資料直接傳送給目的地主機。

3. 查詢預設路由 (default gateway)：
   但在本案例中， PC01 與 PC11 並非同一網域，因此 PC01 會分析路由表當中是否有其他相符合的路由設定， 如果沒有的話，就直接將該 IP 封包送到預設路由器 (default gateway) 上頭去，在本案例當中 default gateway 則是 Server A 這一部。

4. 送出封包至 gateway 後，不理會封包流向：
   當 IP 由 PC01 送給 Server A 之後， PC01 就不理會接下來的工作。而 Server A 接收到這個封包後， 會依據上述的流程，也分析自己的路由資訊，然後向後繼續傳輸到正確的目的地主機上頭。

Tips:

Gateway / Router ：網關/路由器的功能就是在負責不同網域之間的封包轉遞 (IP Forwarding)，由於路由器具有 IP Forwarding 的功能，並且具有管理路由的能力， 所以可以將來自不同網域之間的封包進行轉遞的功能。此外，你的主機與你主機設定的 Gateway 必定是在同一個網段內喔！

大致的情況就是這樣，所以每一部主機裡面都會存在著一個路由表 (Route table)，資料的傳遞將依據這個路由表進行傳送！而一旦封包已經經由路由表的規則傳送出去後， 那麼主機本身就已經不再管封包的流向了，因為該封包的流向將是下一個主機 (也就是那部 Router) 來進行傳送，而 Router 在傳送時，也是依據 Router 自己的路由表來判斷該封包應該經由哪裡傳送出去的！整體來說，資料傳送有點像這樣：

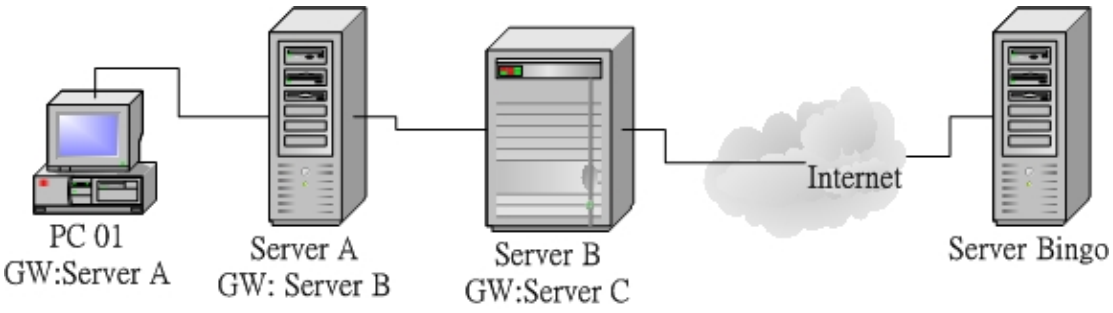

圖 2.3-3、路由的概念

PC 01 要將資料送到 Server Bingo 去，則依據自己的路由表，將該封包送到 Server A 去，Server A 再繼續送到 Server B ，然後在一個一個的接力給他送下去，最後總是可以到達 Server Bingo 的。

上面的案例是一個很簡單的路由概念，事實上， Internet 上面的路由協定與變化是相當複雜的，因為 Internet 上面的路由並不是靜態的，他可以隨時因為環境的變化而修訂每個封包的傳送方向。 舉例來說，數年前在新竹因為土木施工導致台灣西部整個網路纜線的中斷。 不過南北的網路竟然還是能通，為什麼呢？因為路由已經判斷出西部纜線的終止， 因此他自動的導向台灣東部的花蓮路線，雖然如此一來繞了一大圈，而且造成網路的大塞車， 不過封包還是能通就是了！這個例子僅是想告訴大家，我們上面提的路由僅是一個很簡單的靜態路由情況， 如果想要更深入的瞭解 route ，請自行參考相關書籍喔！ ^_^ 。

此外，在屬於 Public 的 Internet 環境中，由於最早時的 IP 分配都已經配置妥當， 所以各單位的路由一經設定妥當後，上層的路由則無須擔心啊！IP 的分配可以參考底下的網頁：

- 台灣地區 IP 核發情況：http://rms.twnic.net.tw/twnic/User/Member/Search/main7.jsp?Order=inet_aton(Startip)

2.3.6 觀察主機路由： route

既然路由是這麼的重要，而且『路由一旦設定錯誤， 將會造成某些封包完全無法正確的送出去！』 所以我們當然需要好好的來觀察一下我們主機的路由表啦！還是請再注意一下， 每一部主機都有自己的路由表喔！觀察路由表的指令很簡單，就是 route ，這個指令挺難的，我們在後面章節再繼續的介紹，這裡僅說明一些比較簡單的用法：

```
[root@www ~]# route [-n]
選項與參數：
-n ： 將主機名稱以 IP 的方式顯示

[root@www ~]# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.0.0     *               255.255.255.0   U     0      0        0 eth0
127.0.0.0       *               255.0.0.0       U     0      0        0 lo
default         192.168.0.254   0.0.0.0         UG    0      0        0 eth0

[root@www ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.0.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
127.0.0.0       0.0.0.0         255.0.0.0       U     0      0        0 lo
0.0.0.0         192.168.0.254   0.0.0.0         UG    0      0        0 eth0

# 上面輸出的資料共有八個欄位，你需要注意的有幾個地方：
# Destination ：其實就是 Network 的意思；
# Gateway     ：就是該介面的 Gateway 那個 IP 啦！若為 0.0.0.0 表示不需要額外的 IP；
# Genmask     ：就是 Netmask 啦！與 Destination 組合成為一部主機或網域；
# Flags       ：共有多個旗標可以來表示該網域或主機代表的意義：
#           U：代表該路由可用；
#           G：代表該網域需要經由 Gateway 來幫忙轉遞；
#           H：代表該行路由為一部主機，而非一整個網域；
# Iface       ：就是 Interface (介面) 的意思。
```

在上面的例子當中，鳥哥是以 PC 01 這部主機的路由狀態來進行說明。由於 PC 01 為 192.168.0.0/24 這個網域，所以主機已經建立了這個網域的路由了，那就是『 192.168.0.0 * 255.255.255.0 ... 』那一行所顯示的訊息！當你下達 route 時， 螢幕上說明了這部機器上面共有三個路由規則，第一欄為『目的地的網域』，例如 192.168.0.0 就是一個網域咯，最後一欄顯示的是 『要去到這個目的地要使用哪一個網路介面！』例如 eth0 就是網路卡的裝置代號啦。如果我們要傳送的封包在路由規則裡面的 192.168.0.0/255.255.255.0 或者 127.0.0.0/255.0.0.0 裡面時，因為第二欄 Gateway 為 * ，所以就會直接以後面的網路介面來傳送出去，而不透過 Gateway 咯！

萬一我們要傳送的封包目的地 IP 不在路由規則裡面，那麼就會將封包傳送到『default』所在的那個路由規則去，也就是 192.168.0.254 那個 Gateway 喔！所以，幾乎每一部主機都會有一個 default gateway 來幫他們負責所有非網域內的封包轉遞！這是很重要的概念喔！^_^！ 關於更多的路由功能與設定方法，我們在第八章當中會再次的提及呢！

2.3.7 IP 與 MAC：鏈結層的 ARP 與 RARP 協定

現在我們知道 Internet 上面最重要的就是那個 IP 了，也會計算所謂的區域網路與路由。 但是，事實上用在傳遞資料的明明就是乙太網路啊！乙太網路主要是用網卡卡號 (MAC) 的嘛！這就有問題啦！那這兩者 (IP 與 MAC) 勢必有一個關連性存在吧？沒錯！那就是我們要談到的 ARP (Address Resolution Protocol, 網路位址解析) 協定，以及 RARP (Revers ARP, 反向網路位址解析)

當我們想要瞭解某個 IP 其實是設定於某張乙太網路卡上頭時，我們的主機會對整個區網發送出 ARP 封包， 對方收到 ARP 封包後就會回傳他的 MAC 給我們，我們的主機就會知道對方所在的網卡，那接下來就能夠開始傳遞資料囉。 如果每次要傳送都得要重新來一遍這個 ARP 協定那不是很煩？因此，當使用 ARP 協定取得目標 IP 與他網卡卡號後， 就會將該筆記錄寫入我們主機的 ARP table 中 (記憶體內的資料) 記錄 20 分鐘 (註14)。

例題：

如何取得自己本機的網卡卡號 (MAC)

答：

```
在 Linux 環境下
[root@www ~]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:01:03:43:E5:34
          inet addr:192.168.1.100  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::201:3ff:fe43:e534/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
.....

在 Windows 環境下
C:\Documents and Settings\admin..> ipconfig /all
....
        Physical Address. . . . . . . . . : 00-01-03-43-E5-34
....
```

那如何取得本機的 ARP 表格內的 IP/MAC 對應資料呢？就透過 arp 這個指令吧！

```
[root@www ~]# arp -[nd] hostname
[root@www ~]# arp -s hostname(IP) Hardware_address
選項與參數：
-n ：將主機名稱以 IP 的型態顯示
-d ：將 hostname 的 hardware_address 由 ARP table 當中刪除掉
-s ：設定某個 IP 或 hostname 的 MAC 到 ARP table 當中

範例一：列出目前主機上面記載的 IP/MAC 對應的 ARP 表格
[root@www ~]# arp -n
Address            HWtype  HWaddress           Flags Mask    Iface
192.168.1.100      ether   00:01:03:01:02:03     C           eth0
192.168.1.240      ether   00:01:03:01:DE:0A     C           eth0
192.168.1.254      ether   00:01:03:55:74:AB     C           eth0

範例二：將 192.168.1.100 那部主機的網卡卡號直接寫入 ARP 表格中
[root@www ~]# arp -s 192.168.1.100  01:00:2D:23:A1:0E
# 這個指令的目的在建立靜態 ARP
```

如同上面提到的，當你發送 ARP 封包取得的 IP/MAC 對應，這個記錄的 ARP table 是動態的資訊 (一般保留 20 分鐘)，他會隨時隨著你的網域裡面電腦的 IP 更動而變化，所以，即使你常常更動你的電腦

IP，不要擔心，因為 ARP table 會自動的重新對應 IP 與 MAC 的表格內容！但如果你有特殊需求的話，也可以利用『 arp -s 』這個選項來定義靜態的 ARP 對應喔！

2.3.8 ICMP 協定

ICMP 的全名是『 Internet Control Message Protocol, 網際網路訊息控制協定 』。 基本上，ICMP 是一個錯誤偵測與回報的機制，最大的功能就是可以確保我們網路的連線狀態與連線的正確性！ ICMP 也是網路層的重要封包之一，不過，這個封包並非獨立存在，而是納入到 IP 的封包中！也就是說， ICMP 同樣是透過 IP 封包來進行資料傳送的啦！因為在 Internet 上面有傳輸能力的就是 IP 封包啊！ ICMP 有相當多的類別可以偵測與回報，底下是比較常見的幾個 ICMP 的類別 (Type)：

類別代號	類別名稱與意義
0	**Echo Reply (代表一個回應信息)**
3	Destination Unreachable (表示目的地不可到達)
4	Source Quench (當 router 的負載過高時，此類別碼可用來讓發送端停止發送訊息)
5	Redirect (用來重新導向路由路徑的資訊)
8	**Echo Request (請求回應訊息)**
11	Time Exceeded for a Datagram (當資料封包在某些路由傳送的現象中造成逾時狀態，此類別碼可告知來源該封包已被忽略的訊息)
12	Parameter Problem on a Datagram (當一個 ICMP 封包重複之前的錯誤時，會回覆來源主機關於參數錯誤的訊息)
13	Timestamp Request (要求對方送出時間訊息，用以計算路由時間的差異，以滿足同步性協定的要求)
14	Timestamp Reply (此訊息純粹是回應 Timestamp Request 用的)
15	Information Request (在 RARP 協定應用之前，此訊息是用來在開機時取得網路信息)
16	Information Reply (用以回應 Infromation Request 訊息)
17	Address Mask Request (這訊息是用來查詢子網路 mask 設定信息)
18	Address Mask Reply (回應子網路 mask 查詢訊息的)

那麼我們是如何利用 ICMP 來檢驗網路的狀態呢？最簡單的指令就是 ping 與 traceroute 了， 這兩個指令可以透過 ICMP 封包的輔助來確認與回報網路主機的狀態。在設定防火牆的時候， 我們最容易忽略的就是這個 ICMP 的封包了，因為只會記住 TCP/UDP 而已～事實上， ICMP 封包可以幫助連線的狀態回報，除了上述的 8 可以考慮關閉之外，基本上，ICMP 封包也不應該全部都擋掉喔！

2.4 TCP/IP 的傳輸層相關封包與資料

網路層的 IP 封包只負責將資料送到正確的目標主機去，但這個封包到底會不會被接受，或者是有沒有被正確的接收， 那就不是 IP 的任務啦！那是傳送層的任務之一。從 圖 2.1-4 我們可以看到傳送層有兩個重點，

一個是連接導向的 TCP 封包，一個是非連接導向的 UDP 封包，這兩個封包很重要啊！資料能不能正確的被送達目的， 與這兩個封包有關喔！

2.4.1 可靠連線的 TCP 協定

在前面的 OSI 七層協定當中，在網路層的 IP 之上則是傳送層，而傳送層的資料打包成什麼？ 最常見的就是 TCP 封包了。這個 TCP 封包資料必須要能夠放到 IP 的資料袋當中才行喔！ 所以，我們將圖 2.1-4 簡化一下，將 MAC, IP 與 TCP 的封包資料這樣看：

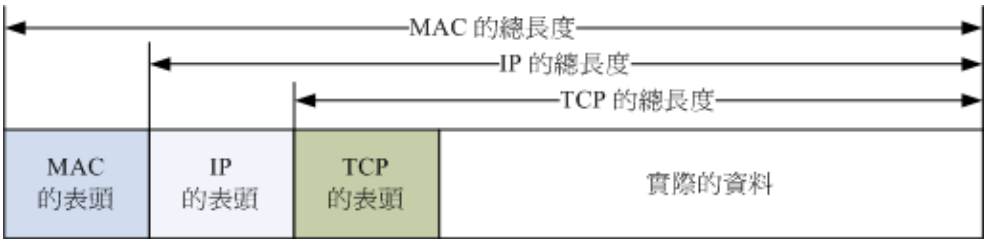

圖 2.4-1、各封包之間的相關性

想當然爾，TCP 也有表頭資料來記錄該封包的相關資訊囉？沒錯啦～ TCP 封包的表頭是長這個樣子的：

4 bits	6 bits	6 bits	8 bits	8 bits
Source Port			Destination Port	
Sequence Number				
Acknowledge Number				
Data Offset	Reserved	Code	Window	
Checksum			Urgent Pointer	
Options			Padding	
Data				

圖 2.4-2、TCP 封包的表頭資料

上圖就是一個 TCP 封包的表頭資料，各個項目以 Source Port, Destination Port 及 Code 算是比較重要的項目，底下我們就分別來談一談各個表頭資料的內容吧！

- **Source Port & Destination Port (來源埠口 & 目標埠口)**
  什麼是埠口(port)？我們知道 IP 封包的傳送主要是藉由 IP 位址連接兩端， 但是到底這個連線的通道是連接到哪裡去呢？沒錯！就是連接到 port 上頭啦！ 舉例來說，鳥哥的網站有開放 WWW 伺服器，這表示鳥站的主機必須要啟動一個可以讓 client 端連接的端口，這個端口就是 port (中文翻譯成為埠口)。同樣的，用戶端想要連接到鳥哥的鳥站時，就必須要在 client 主機上面啟動一個 port ，這樣這兩個主機才能夠利用這條『通道』來傳遞封包資料喔！這個目標與來源 port 的紀錄，可以說是 TCP 封包上最重要的參數了！

- **Sequence Number (封包序號)**
  由於 TCP 封包必須要帶入 IP 封包當中，所以如果 TCP 資料太大時(大於 IP 封包的容許程度)，

就得要進行分段。這個 Sequence Number 就是記錄每個封包的序號，可以讓收受端重新將 TCP 的資料組合起來。

- **Acknowledge Number (回應序號)**
  為了確認主機端確實有收到我們 client 端所送出的封包資料，我們 client 端當然希望能夠收到主機方面的回應，那就是這個 Acknowledge Number 的用途了。 當 client 端收到這個確認碼時，就能夠確定之前傳遞的封包已經被正確的收下了。

- Data Offset (資料補償)
  在圖 2.4-2 倒數第二行有個 Options 欄位對吧！那個 Options 的欄位長度是非固定的，而為了要確認整個 TCP 封包的大小，就需要這個標誌來說明整個封包區段的起始位置。

- Reserved (保留)
  未使用的保留欄位。

- **Code (Control Flag, 控制標誌碼)**
  當我們在進行網路連線的時候，必須要說明這個連線的狀態，好讓接收端瞭解這個封包的主要動作。 這可是一個非常重要的控制碼喔！這個欄位共有 6 個 bits ，分別代表 6 個控制碼，若為 1 則為啟動。分別說明如下：

  - URG(Urgent)：若為 1 則代表該封包為緊急封包， 接收端應該要緊急處理，且圖 2.4-1 當中的 Urgent Pointer 欄位也會被啟用。
  - ACK(Acknowledge)：若為 1 代表這個封包為回應封包， 則與上面提到的 Acknowledge Number 有關。
  - PSH(Push function)：若為 1 時，代表要求對方立即傳送緩衝區內的其他對應封包，而無須等待緩衝區滿了才送。
  - RST(Reset)：如果 RST 為 1 的時候，表示連線會被馬上結束，而無需等待終止確認手續。這也就是說， 這是個強制結束的連線，且發送端已斷線。
  - SYN(Synchronous)：若為 1，表示發送端希望雙方建立同步處理， 也就是要求建立連線。通常帶有 SYN 標誌的封包表示『主動』要連接到對方的意思。
  - FIN(Finish)：若為 1 ，表示傳送結束，所以通知對方資料傳畢， 是否同意斷線，只是發送者還在等待對方的回應而已。

  其實每個項目都很重要，不過我們這裡僅對 ACK/SYN 有興趣而已，這樣未來在談到防火牆的時候，你才會比較清楚為啥每個 TCP 封包都有所謂的『狀態』條件！那就是因為連線方向的不同所致啊！底下我們會進一步討論喔！ 至於其他的資料，就得請您自行查詢網路相關書籍了！

- Window (滑動視窗)
  主要是用來控制封包的流量的，可以告知對方目前本身有的緩衝器容量(Receive Buffer) 還可以接收封包。當 Window=0 時，代表緩衝器已經額滿，所以應該要暫停傳輸資料。 Window 的單位是 byte。

- Checksum(確認檢查碼)
  當資料要由發送端送出前，會進行一個檢驗的動作，並將該動作的檢驗值標注在這個欄位上；而接收者收到這個封包之後，會再次的對封包進行驗證，並且比對原發送的 Checksum 值是否相符，如果相符就接受，若不符就會假設該封包已經損毀，進而要求對方重新發送此封包！

- Urgent Pointer(緊急資料)
  這個欄位是在 Code 欄位內的 URG = 1 時才會產生作用。可以告知緊急資料所在的位置。

- Options(任意資料)
  目前此欄位僅應用於表示接收端可以接收的最大資料區段容量，若此欄位不使用， 表示可以使用任意資料區段的大小。這個欄位較少使用。

- Padding(補足欄位)
  如同 IP 封包需要有固定的 32bits 表頭一樣， Options 由於欄位為非固定， 所以也需要 Padding 欄位來加以補齊才行。同樣也是 32 bits 的整數。

談完了 TCP 表頭資料後，再來讓我們瞭解一下這個表頭裡面最重要的埠口資訊吧！

- 通訊埠口

在上圖的 TCP 表頭資料中，最重要的就屬那 16 位元的兩個咚咚，亦即來源與目標的埠口。由於是 16 位元，因此目標與來源埠口最大可達 65535 號 (2 的 16 次方)！ 那這個埠口有什麼用途呢？上面稍微提到過，網路是雙向的，伺服器與用戶端要達成連線的話， 兩邊應該要有一個對應的埠口來達成連線通道，好讓資料可以透過這個通道來進行溝通。

那麼這個埠口怎麼打開呢？就是透過程式的執行！舉例來說，鳥哥的網站上，必須要啟動一個 WWW 伺服器軟體， 這個伺服器軟體會主動的喚起 port 80 來等待用戶端的連線。你想要看我網站上的資料，就得要利用瀏覽器， 填入網址，然後瀏覽器也會啟動一個埠口，並將 TCP 的表頭填寫目標埠口為 80 ，而來源埠口是你主機隨機啟動的一個埠口， 然後將 TCP 封包封裝到 IP 後，送出到網路上。等鳥站主機接收到你這個封包後，再依據你的埠口給予回應。

這麼說你或許不好理解，我們換個說法好了。假如 IP 是網路世界的門牌，那麼這個埠口就是那個門牌號碼上建築物的樓層！ 每個建築物都有 1~65535 層樓，你需要什麼網路服務，就得要去該對應的樓層取得正確的資料。但那個樓層裡面有沒有人在服務你呢？ 這就得要看有沒有程式真的在執行啦。所以，IP 是門牌，TCP 是樓層，真正提供服務的， 是在該樓層的那個人 (程式)！

Tips:

曾經有一個朋友問過我說：『一部主機上面這麼多服務，那我們跟這部主機進行連線時，該主機怎麼知道我們要的資料是 WWW 還是 FTP 啊？』就是透過埠口啊！因為每種 Client 軟體他們所需要的資料都不相同，例如上面提到的瀏覽器所需要的資料是 WWW ，所以該軟體預設就會向伺服器的 port 80 索求資料；而如果你是使用 filezilla 來進行與伺服器的 FTP 資料索求時，filezilla 當然預設就是向伺服器的 FTP 相關埠口 (預設就是 port 21) 進行連接的動作啦！所以當然就可以正確無誤的取得 Client 端所需要的資料了

再舉個例子來說，一部主機就好像是一間多功能銀行，該銀行內的每個負責不同業務的窗口就好像是通訊埠口， 而我們民眾就好像是 Client 端來的封包。當你進入銀行想要繳納信用卡帳單時， 一到門口服務人員就會指示你

直接到該窗口去繳納，當然，如果你是要領錢，服務人員就會請你到領錢的窗口去填寫資料， 你是不會跑錯的對吧！ ^_^。萬一跑錯了怎麼辦？呵呵！當然該窗口就會告訴你『我不負責這個業務，你請回去！』， 呵呵！所以該次的連線就會『無法成功』咯！

- **特權埠口 (Privileged Ports)**

你現在瞭解了埠口的意義後，再來想想，網路既然是雙向的，一定有一個發起端。問題是，到底要連線到伺服器取得啥玩意兒？ 也就是說，哪支程式應該在哪個埠口執行，以讓大家都知道該埠口就是提供哪個服務，如此一來，才不會造成廣大用戶的困擾嘛！ 所以囉，Internet 上面已經有很多規範好的固定 port (well-known port)， 這些 port number 通常小於 1024 ，且是提供給許多知名的網路服務軟體用的。 在我們的 Linux 環境下，各網路服務與 port number 的對應預設給他寫在 /etc/services 檔案內喔！底下鳥哥列出幾個常見的 port number 與網路服務的對應：

連接埠口	服務名稱與內容
20	FTP-data，檔案傳輸協定所使用的主動資料傳輸埠口
21	FTP，檔案傳輸協定的命令通道
22	SSH，較為安全的遠端連線伺服器
23	Telnet，早期的遠端連線伺服器軟體
25	SMTP，簡單郵件傳遞協定，用在作為 mail server 的埠口
53	DNS，用在作為名稱解析的領域名稱伺服器
80	WWW，這個重要吧！就是全球資訊網伺服器
110	POP3，郵件收信協定，辦公室用的收信軟體都是透過他
443	https，有安全加密機制的WWW伺服器

另外一點比較值得注意的是，小於 1024 以下的埠口要啟動時， 啟動者的身份必須要是 root 才行，所以才叫做特權埠口嘛！這個限制挺重要的，大家不要忘記了喔！ 不過如果是 client 端的話，由於 client 端都是主動向 server 端要資料， 所以 client 端的 port number 就使用隨機取一個大於 1024 以上且沒有在用的 port number。

- **Socket Pair**

由於網路是雙向的，要達成連線的話得要伺服器與用戶端均提供了 IP 與埠口才行。 因此，我們常常將這個成對的資料稱之為 Socket Pair 了！

- 來源 IP + 來源埠口 (Source Address + Source Port)
- 目的 IP + 目的埠口 (Destination Address + Destination Port)

由於 IP 與埠口常常連在一起說明，因此網路定址常常使用『 IP:port 』來說明，例如想要連上鳥哥的網站時， 正確的鳥哥網站寫法應該是：『 linux.vbird.org:80 』才對！

2.4.2 TCP 的三向交握

TCP 被稱為可靠的連線封包，主要是透過許多機制來達成的，其中最重要的就是三向交握的功能。 當然， TCP 傳送資料的機制非常複雜，有興趣的朋友請自行參考相關網路書籍。 OK，那麼如何藉由 TCP 的表頭來確認這個封包有實際被對方接收，並進一步與對方主機達成連線？ 我們以底下的圖示來作為說明。

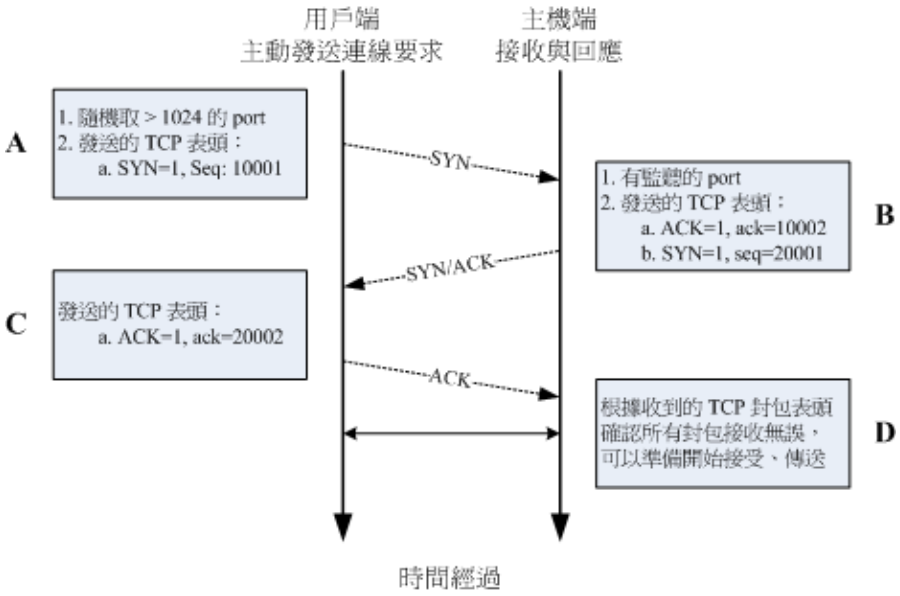

圖 2.4-3、三向交握之封包連接模式

在上面的封包連接模式當中，在建立連線之前都必須要通過三個確認的動作， 所以這種連線方式也就被稱為三向交握(Three-way handshake)。 那麼我們將整個流程依據上面的 A, B, C, D 四個階段來說明一下：

- A:封包發起
  當用戶端想要對伺服器端連線時，就必須要送出一個要求連線的封包，此時用戶端必須隨機取用一個大於 1024 以上的埠口來做為程式溝通的介面。然後在 TCP 的表頭當中，必須要帶有 SYN 的主動連線(SYN=1)，並且記下發送出連線封包給伺服器端的序號 (Sequence number = 10001) 。

- B:封包接收與確認封包傳送
  當伺服器接到這個封包，並且確定要接收這個封包後，就會開始製作一個同時帶有 SYN=1, ACK=1 的封包， 其中那個 acknowledge 的號碼是要給 client 端確認用的，所以該數字會比(A 步驟)裡面的 Sequence 號碼多一號 (ack = 10001+1 = 10002)， 那我們伺服器也必須要確認用戶端確實可以接收我們的封包才行，所以也會發送出一個 Sequence (seq=20001) 給用戶端，並且開始等待用戶端給我們伺服器端的回應喔！

- C:回送確認封包
  當用戶端收到來自伺服器端的 ACK 數字後 (10002) 就能夠確認之前那個要求封包被正確的收受了， 接下來如果用戶端也同意與伺服器端建立連線時，就會再次的發送一個確認封包 (ACK=1) 給伺服器，亦即是 acknowledge = 20001+1 = 20002 囉。

- D:取得最後確認
  若一切都順利，在伺服器端收到帶有 ACK=1 且 ack=20002 序號的封包後，就能夠建立起這次的連線了。

也就是說，你必須要瞭解『網路是雙向的』這個事實！ 所以不論是伺服器端還是用戶端，都必須要透過一次 SYN 與 ACK 來建立連線，所以總共會進行三次的交談！ 在設定防火牆或者是追蹤網路連線的問題時，這個『雙向』的概念最容易被忽略， 而常常導致無法連線成功的問題啊！切記切記！

Tips:

鳥哥上課談到 TCP 最常做的事就是，叫一個同學起來，實際表演三向交握給大家看！

1. 鳥哥說：A同學你在不在？
2. A同學說：我在！那鳥哥你在不在？
3. 鳥哥說：我也在

此時兩個人就確認彼此都可以聽到對方在講啥，這就是可靠連線啦！ ^_^

2.4.3 非連接導向的 UDP 協定

UDP 的全名是：『User Datagram Protocol, 用戶資料流協定』，UDP 與 TCP 不一樣， UDP 不提供可靠的傳輸模式，因為他不是連線導向的一個機制，這是因為在 UDP 的傳送過程中，接受端在接受到封包之後，不會回覆回應封包 (ACK) 給發送端，所以封包並沒有像 TCP 封包有較為嚴密的檢查機制。至於 UDP 的表頭資料如下表所示：

16 bits	16 bits
Source Port	Destination Port
Message Length	Checksum
Data	

圖 2.4-4、UDP 封包的表頭資料

TCP 封包確實是比較可靠的，因為通過三向交握嘛！不過，也由於三向交握的緣故， TCP 封包的傳輸速度會較慢。 至於 UDP 封包由於不需要確認對方是否有正確的收到資料，故表頭資料較少，所以 UDP 就可以在 Data 處填入更多的資料了。同時 UDP 比較適合需要即時反應的一些資料流，例如影像即時傳送軟體等， 就可以使用這類的封包傳送。也就是說， UDP 傳輸協定並不考慮連線要求、連線終止與流量控制等特性， 所以使用的時機是當資料的正確性不很重要的情況，例如網路攝影機！

另外，很多的軟體其實是同時提供 TCP 與 UDP 的傳輸協定的，舉例來說，查詢主機名稱的 DNS 服務就同時提供了 UDP/TCP 協定。由於 UDP 較為快速，所以我們 client 端可以先使用 UDP 來與伺服器連線。 但是當使用 UDP 連線卻還是無法取得正確的資料時，便轉換為較為可靠的 TCP 傳輸協定來進行資料的傳輸囉。 這樣可以同時兼顧快速與可靠的傳輸說！

Tips:

那麼上課時怎麼介紹 UDP 呢？很簡單喔！鳥哥就會說：『現在老師就是在進行 UDP 的傳送，因為老師一直講一直講， 俺也沒有注意到你有沒有聽到，也不需要等待你的回應封包！就這樣一直講！當然，你沒有聽到鳥哥講啥，我也不會知道...』

2.4.4 網路防火牆與 OSI 七層協定

由上面的說明當中，我們知道資料的傳送其實就是封包的發出與接受的動作啦！並且不同的封包上面都有不一樣的表頭 (header)，此外，封包上面通常都會具有四個基本的資訊，那就是 socket pair 裡面提到的『來源與目的 IP 以及來源與目的端的 port number』 。當然啦，如果是可靠性連線的 TCP 封包，還包含 Control Flag 裡面的 SYN/ACK 等等重要的資訊呢！好了，開始動一動腦筋，有沒有想到『網路防火牆』的字眼啊？

封包過濾式的網路防火牆可以抵擋掉一些可能有問題的封包， Linux 系統上面是怎麼擋掉封包的呢？其實說來也是很簡單，既然封包的表頭上面已經有這麼多的重要資訊， 那麼我就利用一些防火牆機制與軟體來進行封包表頭的分析，並且設定分析的規則，當發現某些特定的 IP 、特定的埠口或者是特定的封包資訊(SYN/ACK等等)，那麼就將該封包給他丟棄， 那就是最基本的防火牆原理了！

舉例來說，大家都知道 Telnet 這個伺服器是挺危險的，而 Telnet 使用的 port number 為 23 ，所以，當我們使用軟體去分析要送進我們主機的封包時， 只要發現該封包的目的地是我們主機的 port 23 ，就將該封包丟掉去！那就是最基本的防火牆案例啦！ 如果以 OSI 七層協定來說，每一層可以抵擋的資料有：

- 第二層：可以針對來源與目標的 MAC 進行抵擋；
- 第三層：主要針對來源與目標的 IP ，以及 ICMP 的類別 (type) 進行抵擋；
- 第四層：針對 TCP/UDP 的埠口進行抵擋，也可以針對 TCP 的狀態 (code) 來處理。

更多的防火牆資訊我們會在第九章防火牆與第七章認識網路安全當中進行更多的說明喔！

2.5 連上 Internet 前的準備事項

講了這麼多，其實我們最需要的僅是『連接上 Internet 』啦！那麼在 Internet 上面其實使用的是 TCP/IP 這個通訊協定，所以我們就需要 Public IP 來連接上 Internet 啊！你說對吧～ 不過，你有沒有發現一件事，那就是『為啥我不知道 Yahoo 的主機 IP ，但是俺的主機卻可以連到 Yahoo 主機上？』 如果你有發現這個問題的話，哈哈！你可以準備開始設定網路囉～ ^_^

2.5.1 用 IP 上網？主機名稱上網？DNS 系統？

講完了上頭的基本資料，現在你知道要連上 Internet 就得要有 TCP/IP 才行！尤其是那重要的 IP 啊！ 問題是，電腦網路是依據人類的需要來建立的，不過人類對於 IP 這一類的數字並不具有敏感性，即使 IP 已經被簡化為十進位了，但是人類就是對數字沒有辦法啊！怎麼辦？沒關係，反正電腦都有主機名稱嘛！ 那麼我就將主機名稱與他的 IP 對應起來，未來要連接上該電腦時，只要知道該電腦的主機名稱就好了，因為 IP 已經對應到主機名稱了嘛！所以人類也容易記憶文字類的主機名稱，電腦也可以藉由對應來找到他必須要知道的 IP ，啊！真是皆大歡喜啊！

這個主機名稱 (Hostname) 對應 IP 的系統，就是鼎鼎有名的 Domain Name System (DNS) 咯！也就是說， DNS 這個服務的最大功能就是在進行『主機名稱與該主機的 IP 的對應』的一項協定。 DNS 在網路環境當中是相當常被使用到的一項協定喔！舉個例子來說，像鳥哥我常常會連到奇摩雅虎的 WWW 網站去看最新的新聞，那麼我一定需要將奇摩雅虎的 WWW 網站的 IP 背下來嗎？天吶， 鳥哥的忘性這麼好，怎麼可能將 IP 背下來？！不過，如果是要將奇摩站的主機名稱背下來的話， 那就容易的多了！不就是 http://tw.yahoo.com 嗎？而既然電腦主機只認識 IP 而已， 因此當我在瀏覽器上面輸入了

『http://tw.yahoo.com』的時後，我的電腦首先就會藉由向 DNS 主機查詢 tw.yahoo.com 的 IP 後，再將查詢到的 IP 結果回應給我的瀏覽器， 那麼我的瀏覽器就可以藉由該 IP 來連接上主機啦！

發現了嗎？我的電腦必須要向 DNS 伺服器查詢 Hostname 對應 IP 的資訊 喔！那麼那部 DNS 主機的 IP 就必須要在我的電腦裡面設定好才行，並且必須要是輸入 IP 喔，不然我的電腦怎麼連到 DNS 伺服器去要求資料呢？呵呵！在 Linux 裡面，DNS 主機 IP 的設定就是在 **/etc/resolv.conf** 這個檔案裡面啦！

目前各大 ISP 都有提供他們的 DNS 伺服器的 IP 給他們的用戶，好設定客戶自己電腦的 DNS 查詢主機， 不過，如果你忘記了或者是你使用的環境中並沒有提供 DNS 主機呢？呵呵！沒有關係， 那就設定 Hinet 那個最大的 DNS 伺服器吧！ IP 是 168.95.1.1 咯！要設定好 DNS 之後，未來上網瀏覽時，才能使用主機名稱喔！不然就得一定需要使用 IP 才能上網呢！DNS 是很重要的，他的原理也頂複雜的，更詳細的原理我們在第十九章 DNS 伺服器裡面進行更多更詳細的說明喔！這裡僅提個大綱！

2.5.2 一組可以連上 **Internet** 的必要網路參數

從上面的所有說明當中，我們知道一部主機要能夠使用網路，必須要有 IP ，而 IP 的設定當中，就必須要有 IP, Network, Broadcast, Netmask 等參數，此外，還需要考慮到路由裡面的 Default Gateway 才能夠正確的將非同網域的封包給他傳送出去。 另外，考慮到主機名稱與 IP 的對應，所以你還必須要給予系統一個 DNS 伺服器的 IP 才行～ 所以說，一組合理的網路設定需要哪些資料呢？呵呵！就是：

- IP
- Netmask
- Network
- Broadcast
- Gateway
- DNS

其中，由於 Network 與 Broadcast 可以經由 IP/Netmask 的計算而得到，因此需要設定於你 PC 端的網路參數， 主要就是 IP, Netmask, Default Gateway, DNS 這四個就是了！

沒錯！就是這些資料！如果你是使用 ADSL 撥接來上網的話，上面這些資料都是由 ISP 直接給你的，那你只要使用撥接程式進行撥接到 ISP 的工作之後， 這些資料就自動的在你的主機上面設定完成了！但是如果是固定制 (如學術網路) 的話，那麼就得自行使用上面的參數來設定你的主機囉！缺一不可呢！以 192.168.1.0/24 這個 Class C 為例的話，那麼你就必須要在你的主機上面設定好底下的參數：

- IP: 由 192.168.1.1~192.168.1.254
- Netmask: 255.255.255.0
- Network: 192.168.1.0
- Broadcast: 192.168.1.255
- Gateway: 每個環境都不同，請自行詢問網路管理員
- DNS: 也可以直接設定成 168.95.1.1

2.6 重點回顧：

- 雖然目前的網路媒體多以乙太網路為標準，但網路媒體不只有乙太網路而已；
- Internet 主要是由 Internet Network Information Center (INTERNIC) 所維護；
- 乙太網路的 RJ-45 網路線，由於 568A/568B 接頭的不同而又分為平行線與跳線；
- 乙太網路上最重要的傳輸資料為 Carrier Sense Multiple Access with Collision Detect (CSMA/CD) 技術， 至於傳輸過程當中，最重要的 MAC 訊框內以硬體位址 (hardware address) 資料最為重要；

- 透過八蕊的網路線 (Cat 5 以上等級)，現在的乙太網路可以支援全雙工模式；
- OSI 七層協定為一個網路模型 (model) ，並非硬性規定。這七層協定可以協助軟硬體開發有一個基本的準則， 且每一分層各自獨立，方便使用者開發；
- 現今的網路基礎是架構在 TCP/IP 這個通訊協定上面；
- 資料鏈結層裡重要的資訊為 MAC (Media Access Control)，亦可稱為硬體位址，而 ARP Table 可以用來對應 MAC 與軟體位址 (IP) ；
- 在網路媒體方面， Hub 為共享媒體，因此可能會有封包碰撞的問題，至於 Switch 由於加入了 switch port 與 MAC 的對應，因此已經克服了封包碰撞的問題，也就是說，Switch 並不是共享媒體；
- IP 為 32 bits 所組成的，為了適應人類的記憶，因此轉成四組十進位的數據；
- IP 主要分為 Net ID 與 Host ID 兩部份，加上 Netmask 這個參數後，可以設定『網域』的概念；
- 根據 IP 網域的大小，可將 IP 的等級分為 A, B, C 三種常見的等級；
- Loopback 這個網段在 127.0.0.0/8 ，用在每個作業系統內部的迴圈測試中。
- 網域可繼續分成更小的網域 (subnetwork)，主要是透過將 Host_ID 借位成為 Net_ID 的技術；
- IP 只有兩種，就是 Public IP 與 Private IP ，中文應該翻譯為 公共 IP 與 私有(或保留) IP，私有 IP 與私有路由不可以直接連接到 Internet 上；
- 每一部主機都有自己的路由表，這個路由表規定了封包的傳送途徑，在路由表當中，最重要者為預設的通訊閘 (Gateway/Router)；
- TCP 協定的表頭資料當中，那個 Code (control flags) 所帶有的 ACK, SYN, FIN 等為常見的旗標， 可以控制封包的連線成功與否；
- TCP 與 IP 的 IP address/Port 可以組成一對 socket pair
- 網路連線都是雙向的，在 TCP 的連線當中，需要進行用戶端與伺服器端兩次的 SYN/ACK 封包發送與確認， 所以一次 TCP 連線確認時，需要進行三向交握的流程；
- UDP 通訊協定由於不需要連線確認，因此適用於快速即時傳輸且不需要資料可靠的軟體中，例如即時通訊；
- ICMP 封包最主要的功能在回報網路的偵測狀況，故不要使用防火牆將他完全擋掉；
- 一般來說，一部主機裡面的網路參數應該具備有：IP, Netmask, Network, Broadcast, Gateway, DNS 等；
- 在主機的 port 當中，只有 root 可以啟用小於 1024 以下的 port ；
- DNS 主要的目的在於進行 Hostname 對應 IP 的功能；

2.7 本章習題

- 在 ISP 提供的網路服務中，他們提到傳輸速度為 1.5M/382K ，請問這個數據的單位為何？

  數據單位為 bits/second, 與慣用的 bytes 差 8 倍。

- 什麼是 MAC (Media Access Control) ，MAC 主要的功能是什麼？

  Media Access Control 的縮寫，為乙太網路硬體訊框的規格，乙太網路就是以 MAC 訊框進行資料的傳送。 目前 MAC 也常被用為乙太網路卡卡號的代稱。

- 什麼是封包碰撞？為什麼會發生封包碰撞？

  當主機要使用網路時，必須要先進行 CSMA/CD 監聽網路，如果(1)網路使用頻繁 (2)網路間隔太大， 則可能會發生監聽時均顯示無主機使用，但發出封包後卻發生同步發送封包的情況，此時兩個封包就會產生碰撞， 造成資料損毀。

- ARP Table 的作用為何？如何在我的 Linux 察看我的 ARP 表格？

  ARP 協定主要在分析 MAC 與 IP 的對應，而解析完畢後的資料會存在系統的記憶體中， 下次要傳送到相同的 IP 時，就會主動的直接以該 MAC 傳送，而不發送廣播封包詢問整個網域了。
  利用 arp -n 即可

- 簡略說明 Netmask 的作用與優點；

Netmask 可以用來區分網域，且 Netmask 可以有效的增加網路的效率，這是因為 Netmask 可以定義出一個網域的大小，那麼 broadcast 的時間就可以降低很多！一般來說， 我們如果要將一個大網域再細分為小網域，也需要藉由 Netmask 來進行 subnet 的切割。

- 我有一組網域為： 192.168.0.0/28 ，請問這個網域的 Network, Netmask, Broadcast 各為多少？而可以使用的 IP 數量與範圍各是多少？

  因為共有 28 個 bits 是不可動的，所以 Netmask 位址的最後一個數字為 11110000，也就是 (128+64+32+16=240) ，所以：
  Network：192.168.0.0
  Netmask：255.255.255.240
  Broadcast：192.168.0.15
  IP：由 192.168.0.1 ~ 192.168.0.14 共 14 個可用 IP 喔！

- 承上題，如果網域是 192.168.0.128/29 呢？

  因為是 29 個 bits 不可動，所以最後一個 Netmask 的位址為： 11111000 也就是 (128+64+32+16+8=248)，所以：
  Network：192.168.0.128
  Netmask：255.255.255.248
  Broadcast：192.168.0.135
  IP：由 192.168.0.129 ~ 192.168.0.134 共 6 個可用的 IP 喔！

- 我要將 192.168.100.0/24 這個 Class C 的網域分為 4 個子網域，請問這四個子網域要如何表示？

  既然要分為四個網域，也就是還需要藉助 Netmask 的兩個 bits (2的2次方為4啊！)，所以 Netmask 會變成 255.255.255.192 ，每個子網域會有 256/4=64 個 IP ，而必須要扣除 Network 與 Broadcast ，所以每個子網域會有 62 個可用 IP 喔！因此，四個子網域的表示方法為：
  192.168.100.0/26, 192.168.100.64/26, 192.168.100.128/26, 192.168.100.192/26。

- 如何觀察 Linux 主機上面的路由資訊 (route table)？

  路由資訊的觀察可以下達 route 來直接察看！或者是下達 route -n 亦可

- TCP 封包上面的 SYN 與 ACK 標誌代表的意義為何？

  SYN 代表該封包為該系列連線的第一個封包，亦即是主動連線的意思；
  ACK 則代表該封包為確認封包，亦即是回應封包！

- 什麼是三向交握？在哪一種封包格式上面才會有三向交握？

  使用 TCP 封包才會有三向交握。TCP 封包的三向交握是一個確認封包正確性的重要步驟，通過 SYN, SYN/ACK, ACK 三個封包的確認無誤後，才能夠建立連線。至於 UDP 封包則沒有三向交握喔！

- 試說明何謂有網管？無網管的 switch ？此外，這些 switch 的硬體應算在 OSI 七層協定的第幾層？

  有網管者，會在 switch 內部加入其他的小型 OS，藉以控管 IP 或 MAC 的流通； 通常基礎的 switch 僅達控管 MAC ，故為 OSI 第二層(資料鏈結層)

- 為何 ISP 有時候會談到『申請固定 8 個 IP ，其中只有 5 個可以用』，你覺得問題出在哪裡？ 如果以網域的觀念來看，他的 netmask 會是多少？

  因為如果是一個網域的話，那麼八個 IP 前後(Host_ID 全為 0 與 1 的條件)為 Network 及 Broadcast ， 加上一個在 ISP 處的 Gateway ，所以僅有 5 個可以用。因為有 8 個 IP ，所以其 netmask 後八 bits 為 11111000 ，故為 255.255.255.248。

- Internet 協定中共包含 "Network Access Layer", "Internet Layer", "Transport Layer", "Application Layer"， 請將這四層與 OSI 七層協定的內容進行連結 (自行上網查詢相關文章說明)；

  Network Access Layer: 涵蓋 Data-Link 及 Physical Layer
  Internet Layer: 也是 Network Layer
  Transport Layer: 也是 Transport Layer
  Application Layer: 涵蓋 Application Layer, Persentatin Layer, Session Layer.

- 請自行上網查詢關於 NetBIOS 這個通訊協定的相關理論基礎，並請說明 NetBIOS 是否可以跨路由？

  請自行參考網中人的網路基礎文章

- 什麼是 Socket pair ？包含哪些基本資料？

  由 IP 封包的 IP address 與 TCP 封包的 port number 達成，分別為目的端的 IP/port 與本地端的 IP/port。

- IP 有一段 A Class 的網段分給系統做為測試用，請問該網段為？設定的名稱為？

  127.0.0.0/8, loopback

- ICMP 這個協定最主要的目的為？同時做為『回應』的類別為第幾類？

  做為網路檢測之用，為第 8 類 (echo request)

- IP 封包表頭有個 TTL 的標誌，請問該標誌的基本說明為何？其數據有何特性？

  為該封包的存活時間，該時間每經過一個 node 都會減少一，當 TTL 為 0 時，該封包會被路由器所丟棄。 該數據最大為 255。

- 在 Linux 當中，如何查詢每個 port number 對於服務的對應 (filename)

  /etc/services 檔案中有紀錄

- 什麼是星形連線？優點為何？

  利用一 hub/switch 連結所有的網路設備的一種連線方式，最大的好處是，每個『網路設備與 switch 之間』都是獨立的， 所以所以每個主機故障時均不會影響其他主機的連線。

- 請說明 CSMA/CD 的運作原理？

  發送流程
  1. 主機欲使用網路時，會先監聽網路，若網路沒有被使用時，才會準備傳送，否則繼續監聽；
  2. 當資料傳送鐘，發現有碰撞情況時，則會重新監聽網路，並且重新發送一次該封包；
  3. 若重複發生碰撞 16 次，則網路會癱瘓；

  接收流程
  1. 主機如果沒有在傳送資料，則會監聽網路，並且主動在接收的狀態下；
  2. 若接收到一個封包，並且該表頭所載 MAC 為本身的網卡卡號，則開始接收該封包，否則將該封包丟棄；
  3. 接收過程當中如果發生封包碰撞，則會通知原發送主機碰撞的資料；
  4. 封包接收完畢後，會以 MAC 表頭所載長度同時分析本封包長度，若發生問題，則會通知對方重新傳送。

2.8 參考資料與延伸閱讀

特別感謝：
本文在 2002/07 發出之後，收到相當多朋友的關心，也從而發現了自己誤會的一些基礎的網路理論，真的是感謝好朋友 Netman 兄與 ZMAN 兄的指導！這篇短文在第二版時 (2003/08/03) 做了相當大幅度的修訂，與原來的文章 (上次更新日期 2002/09) 已經有一定程度的差異了，第三版又針對整個內容與閱讀順序進行調整 (2010/08)，希望網友們如果有時間的話，能夠再次的閱讀，以釐清一些基本概念喔！

- 註1：粘添壽著，『Internet 網路原理與實務』，旗標出版社。
- 註2：Robert Breyer & Sean Riley 著，風信子,張民人譯，『Switched & Fast 乙太網路』，旗標出版社
- 註3：IEEE 標準的網站連結：http://standards.ieee.org/
- 註4：Request For Comment (RFC) 技術文件：http://www.rfc-editor.org/
- 註5：RFC-1122 標準的文件資料：ftp://ftp.rfc-editor.org/in-notes/rfc1122.txt
- 註6：粘添壽老師官網：http://www.tsnien.idv.tw/, 網際網路相關課程：http://120.118.165.46/tsnien/network/index.html(強烈建議前往參閱)
- 註7：台灣學術網路簡介 (TANET)：http://www.edu.tw/moecc/content.aspx?site_content_sn=1707
- 註8：Study Area 之網路基礎：http://www.study-area.org/network/network.htm
- 註9：維基百科對 OSI 協定的說明：http://en.wikipedia.org/wiki/OSI_model
- 註10：Phil Dykstra, Gigabit Ethernet Jumbo Frames： http://sd.wareonearth.com/~phil/jumbo.html

- 註11：Hub 與 Switch 的迷思：http://www.study-area.org/tips/hub_switch.htm
- 註12：管理 IP 的單位與相關說明：http://www.internic.org/, http://www.icann.org/, http://www.iana.org/, http://en.wikipedia.org/wiki/IPv4
- 註13：管理 IP 的單位：http://www.iana.org/, 台灣地區 IP 核發情況：http://rms.twnic.net.tw/twnic/User/Member/Search/main7.jsp?Order=inet_aton(Startip)
- 註14：相關參考資料
  『TCP/IP Illustrated, Volume 1 - The Protocols』，W. Richard Stevens ，資策會中文化部門譯；
  http://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing
- PPPoE http://en.wikipedia.org/wiki/Point-to-Point_Protocol_over_Ethernet

---

2002/07/18：第一次完成日期！
2002/09/26：修改了部分可能引起誤解的文章部分！
2003/08/03：重新編排版面，並且重新檢視文章內容，修訂文章！
2003/08/20：增加重點回顧與課後練習
2003/09/06：加入參考用解答
2004/03/16：修訂 N-Way 的錯誤，訂正為 Auto MDI/MDIX 的功能！
2006/02/09：將舊的文章移動到 此處
2006/07/12：參考了粘教授與風信子兄的書籍，修改了很多基礎資料喔！還有重點整理，不過，練習尚未更新
2006/07/16：加入習題練習囉！
2007/10/21：圖14那個 UDP 的表頭資料中，16 bits 誤植為 16 bytes，感謝討論區 ricky.liu 的告知！
2008/04/21：經由網友 chyanlong 兄的指點，IHL 的大小單位誤植為 byte，應該是字組 (word) 才對。
2010/07/22：將基於 CentOS4.x 所寫的資料放置於此處
2010/08/15：將章節依據 TCP/IP 相關的層級分別介紹，更改的幅度不小喔！
2011/07/15：將基於 CentOS 5.x 所撰寫的文章移動到此處
