大標題的圖示2.1 網路是個什麼玩意兒

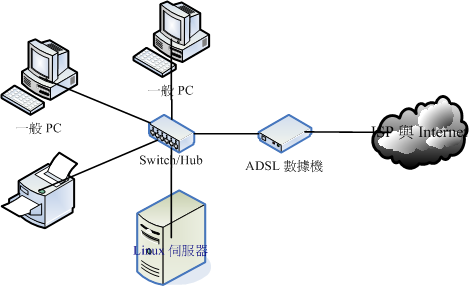
這一個章節旨在引導網路新鮮人快速進入網路的世界。

小標題的圖示2.1.1 什麼是網路

網路是幾部主機透過有線或無線網路連接主機與設備，使資料可透過網路媒體(網路線及其他網路卡等硬體)來傳輸的一種方式。

* 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，為解決資料傳輸問題，相關的公司都在研究各自的網路連接技術。其中比較有名的是全錄公司的 Ethernet，及IBM的Token-Ring。但是這些技術它們彼此不認識對方的網路技術！也就是說辦公室買了Ethernet的電腦主機，但其他的電腦卻用IBM的機器時，兩者間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  
  為解決網路硬體整合功能，在1960末美國國防部開始研究在不同的網路硬體上運作的軟體技術，使不同公司的電腦或資料可透過軟體來達成資料溝通。該研究由美國國防部尖端研究企畫署(DARPA)負責，稱該網路系統為ARPANET，這就是TCP/IP的雛形了！在1975年左右，ARPANET已可在Ethernet與Token-Ring等硬體平台下互通資料。DARPA在1980年正式推出TCP/IP後，想要推展，因此與柏克萊(Berkeley)大學合作，將TCP/IP植入的BSD Unix系統，TCP/IP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為Internet。
* 沒有任何王法的網際網路：Internet   
  現在Internet 是用 TCP/IP 的網路連接技術串聯起來的網路世界，Internet在1980後對email的需求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Internet是管理鬆散的所在。使用任何支援TCP/IP的硬體與作業系統，連接上網路後就進入Internet了。實際資料如果接上Internet，任何時刻都要保護自己！因為Internet 僅提供一個網路的連接介面，所以連接上Internet後，全世界都可以任你遨遊，因此『跨海』而來的攻擊就成了簡單的事件，電腦怪客(cracker)可在國外透過Internet對你的主機進行攻擊，我們的法律可管不到國外地區！
* 軟硬體標準制定的成功帶來的影響：IEEE標準規範  
  網路卡是一個介面卡，Internet則去向Hinet/Seednet或其他網路服務提供公司(Internet Service Provider, ISP)申請的帳號密碼。是否只有透過網路卡與Internet才能上網啊？當然不是！不過，最成功的是乙太網路(Ethernet)與Internet！這是因為這兩者都被『標準』所支援的緣故。  
  乙太網路最初是由全錄(Xerox PARC)建構出來，而後透過DEC, Intel與Xerox合作將乙太網路標準化。再經由 IEEE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，到了1989年國際標準組織 ISO將乙太網路編入IS88023標準！這表示乙太網路已是一項公認的標準介面，大家可依據這個標準來設定與開發硬體，只要硬體符合這個標準，理論上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。
* TCP/IP這Internet的通訊協定也是有標準的，大部分都以RFC(Request For Comments)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，就可發展出自己的TCP/IP軟體，並連接上Internet。

小標題的圖示2.1.2 電腦網路組成元件

  
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，上面圖示中的一般PC、Linux伺服器、ADSL數據機與網路印表機等，都可以稱為一個node！那中間那個集線器(hub)因不具有IP，因此hub不是節點。
* 伺服器(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。
* 用戶端(client)：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，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，主要提供網路位址(IP)的任務。一張網卡至少可搭配一個以上的網路介面；而每部主機內部也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，上圖中顯示的是星形連線(star)的方式，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(router)或通訊閘(gateway)：具兩個以上的網路介面，連接兩個以上不同網段的設備，如IP分享器。ADSL數據機不太能算網關，因為數據機通常視為一個在主機內的網卡設備，可在一般PC上面透過撥號軟體，將數據機模擬成為一張實體網卡(ppp)！

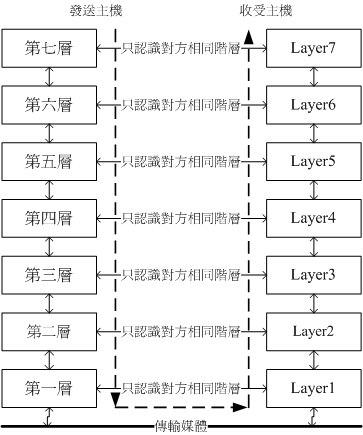
小標題的圖示2.1.3 電腦網路區域範圍

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，將網路的大小範圍定義如下：([註6](http://linux.vbird.org/linux_server/0110network_basic.php#ps6))

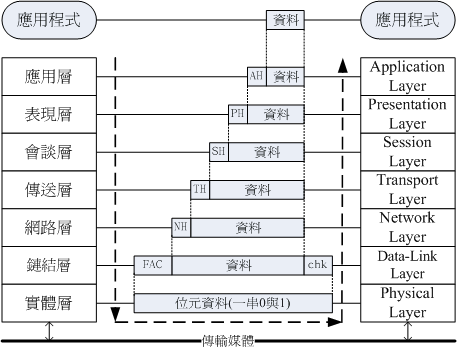
* 區域網路(Local Area Network, LAN)：點之間的傳輸距離較近，如一棟大樓內，或一個學校的校區內。可用較為昂貴的連線材料，如光纖或是高品質網路線(CAT 6)等。網路速度較快，連線品質較佳且可靠，可應用於科學運算的叢集式系統、分散式系統、雲端負荷分擔系統等。
* 廣域網路 (Wide Area Network, WAN)：傳輸距離較遠，例如城市與城市之間的距離，因此使用的便宜的設備為連線媒體，如電話線。由於線材品質較差，網路速度慢且可靠性較低，網路應用如email, FTP, WWW瀏覽等功能。

小標題的圖示2.1.4 電腦網路通訊協定： OSI 七層協定

網路分成數個階層(layer)，每個階層都有獨立的功能，且程式碼可獨立撰寫，使階層間的功能不會互相干擾。當某環節出現問題時，只要將該層級的程式碼重寫即可。OSI七層協定(Open System Interconnection) 的概念，以圖示來說：

  
圖 2.1-2、OSI 七層協定各階層的相關性

接近硬體的階層為底層(layer 1)，應用程式的則是高層(layer 7)，每層都有獨特的表頭資料，告知對方裡面的資訊是什麼，而真正的資料在後頭。不論接收還是發送端，每層只認識對方的同一階層資料。

  
圖 2.1-3、OSI 七層協定資料的傳遞方式

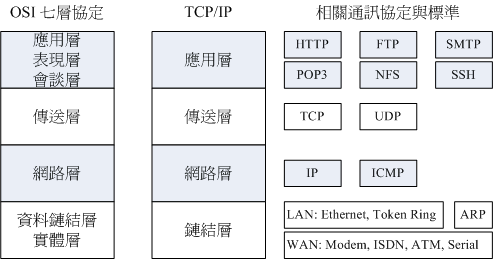
上圖中每個資料包的部分，上層的包裹是放入下層的資料中，而資料前面則是這個資料的表頭。第二層(資料鏈結層)位於軟體封包(packet)以及硬體訊框(frame)間的一個階層，必須要將軟體包裝的格式放到硬體能夠處理的格式中，因此這個階層又分為兩個子層在處理相對應的資料。因此第二層的資料格式比較不一樣，尾端有檢查碼～每一層負責的任務如下：

|  |  |
| --- | --- |
| 分層 | 負責內容 |
| Layer 1 實體層 Physical Layer | 網路媒體只能傳送0與1的位元串，因此實體層須定義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 | 下層是實體的定義，上層是軟體封裝的定義。因此第二層又分兩個子層。下層負責MAC(Media Access Control)，稱資料包裹為MAC訊框(frame)，MAC是網路媒體能處理的資料包裹，最終被實體層編碼成位元串。MAC須經由通訊協定來取得媒體的使用權，目前最常用的是IEEE 802.3的乙太網路協定。上層是由邏輯連結層(logical link control, LLC)，多工處理來自上層的封包資料(packet)並轉成MAC的格式，負責訊息交換、流量控制、失誤問題處理等。 |
| Layer 3 網路層 Network Layer | IP(Internet Protocol)在這一層定義。同時也定義出電腦間的連線建立、終止與維持等，資料封包的傳輸路徑選擇等等，除了最重要的IP外，就是封包能否到達目的地的路由(route)概念！ |
| Layer 4 傳送層 Transport Layer | 這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 | 這層級定義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 | 在應用程式上所製作的資料格式不一定符合網路傳輸的標準編碼格式！所以這層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，再交給底下傳送層等的協定來進行處理。這層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。 |
| 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使用OSI七層協定的聯網程式碼。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！

小標題的圖示2.1.5 電腦網路通訊協定： TCP/IP

OSI七層協定的架構太嚴謹，因此程式撰寫不容易。而TCP/IP是使用OSI七層協定的觀念，只是將它簡化為四層。後來在1990年代由於email, WWW的流行，造成TCP/IP為大家所接受，這也造就目前我們的網路社會！TCP/IP是由OSI七層協定簡化而來，它們的相關性圖示如下，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。傳送層沒變，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？拿常常連上的Yahoo入口網站來做個說明，整個連線的狀態可以這樣看：

1. 應用程式階段：打開瀏覽器輸入網址列，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；
2. 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；
3. 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；
4. 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)，向鏈結層丟去；
5. 鏈結層：如果使用乙太網路，IP會依據CSMA/CD的標準，包到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！

網路媒體一次傳輸的資料量是有限的，如果要被傳輸的資料太大時，在分層的包裝中，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！

|  |  |
| --- | --- |
| **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，因此又將 TCP/IP當中的應用層視為使用者層，而底下的三層才是主要談及的網路基礎！ |  |

大標題的圖示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。

小標題的圖示2.2.1 廣域網路使用的設備

廣域網路的設備非常多，一般用戶會接觸到的主要是ADSL數據機或者是光纖到大廈，以及第四台的Cable寬頻等。

* 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)配合撥接程式取得網路IP參數。速度非常慢，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)：用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，且可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，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)：用電話線撥接後取得IP，不過使用電話的高頻部分，與一般講電話的頻率不同。因此可同時用同個電話上網及打電話。在台灣由於上傳/下載的頻寬不同，因此稱為非對稱的回路。ADSL同樣使用數據機，透過PPPoE(PPP over Ethernet)的方法！將PPP模擬在乙太網路卡上，因此主機需透過一張網路卡連接到數據機，並透過撥接程式來取得新的介面(ppp0)！
* 纜線數據機(Cable modem)：透過有線電視(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的頻寬是分享型的，通常具有區域性，並不是想裝就能裝的！

小標題的圖示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，最常用的是乙太網路。在某些超高速網路環境中，還會用到價格昂貴的光纖通道。乙太網路已標準化，設備設置費用低廉，所以一般網路線或媒體，幾乎用乙太網路架設環境！只是網路世界並非僅有乙太網路這個硬體介面。



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制訂的乙太網路標準為802.3的IEEE 10BASE5，這個標準的定義是：『10表傳輸速度為10Mbps，BASE表採基頻信號傳輸，5指每個網路節點間最長達500公尺。』由於網路的傳輸資訊就是0與1，因此資料傳輸的單位為每秒多少bit，即M bits/second, Mbps。

|  |  |
| --- | --- |
| **Tips:** 網路提供者(Internet Services Provider, ISP) 稱ADSL傳輸速度可達到下行/上行2Mbps/128Kbps (Kbits per second)時，Kb不是指bytes而是bits ！2M/128K在實際傳輸速度上，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100~200KBytes間！ |  |

早期的網路線使用舊式的同軸電纜線。類似傳統電話線的雙絞線(Twisted Pair Ethernet)取而代之，IEEE將這種線路的乙太網路傳輸方法制訂成10BASE-T的標準。10BASE-T使用10 Mbps全速運作且採用無遮蔽式雙絞線(UTP)的網路線。10BASE-T的UTP網路線可用星形連線(star)，以一個集線器為中心串連各網路設備，[圖 2.1-1](http://linux.vbird.org/linux_server/0110network_basic.php#fig2.1-1) 就是星形連線的示意圖。

後來IEEE更制訂了802.3u支援100Mbps傳輸速度的100BASE-T標準，與10BASE-T差異不大，只是雙絞線線材製作要更精良，同時支援四對絞線的網路線，就是目前常見的八蕊網路線！這種網路線稱為等級五(Category 5, CAT5)的網路線。這種傳輸速度的乙太網路被稱為Fast ethernet。Gigabit網路速度1000 Mbps就是Gigabit ethernet！只是 Gigabit ethernet的網路線要更精良。

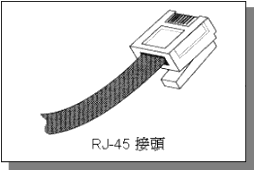
|  |  |  |
| --- | --- | --- |
| 名稱 | 速度 | 網路線等級 |
| 乙太網路(Ethernet) | 10Mbps | - |
| 高速乙太網路(Fast Ethernet) | 100Mbps | CAT 5 |
| 超高速乙太網路(Gigabit Ethernet) | 1000Mbps | CAT 5e/CAT 6 |

當傳輸速度增加時，線材的電磁效應相互干擾會增強，網路線的要求就更嚴格，因此在網路線的製作要注意線材的質料以及內部線蕊心的纏繞情況配置等，使電子流之間的電磁干擾降到最小，才能提升傳輸速度到Gigabit。在乙太網路世界當中，如果要提升fast ethernet到gigabit ethernet，除了網路卡需升級，主機與主機間的網路線，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！



乙太網路的網路線接頭 (跳線/平行線)

目前在乙太網路上最常見到的接頭是 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：

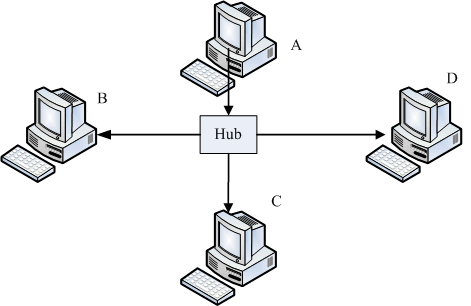
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 接頭名稱\蕊線順序 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 568A | 白綠 | 綠 | 白橙 | 藍 | 白藍 | 橙 | 白棕 | 棕 |
| 568B | 白橙 | 橙 | 白綠 | 藍 | 白藍 | 綠 | 白棕 | 棕 |

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，其他則是特殊用途才會用到。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；

小標題的圖示2.2.3 乙太網路的傳輸協定：CSMA/CD

乙太網路的傳輸就是網路卡對網路卡間的資料傳遞。每張乙太網路卡出廠時，會賦予一獨一無二的卡號，即MAC(Media Access Control)！理論上網卡卡號是不能修改的！乙太網路網卡間的資料傳輸就要談一下IEEE 802.3標準CSMA/CD (Carrier Sense Multiple Access with Collision Detection)！下圖的中心點為集線器，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，而集線器就是十字路口！一次只允許一輛車通過，如果兩輛車同時使用這個路口，會發生碰撞的事件！那就是所謂的共享媒體。網路共享媒體在單一時間點內僅能被一部主機使用。乙太網路的網卡傳輸使用CSMA/CD，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發送網路封包前，先監聽網路媒體，確認沒人使用後，才能發送出訊框；
2. 多點傳輸(Multiple Access)：A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！A送出的資料，B, C, D都能夠接收的到！但由於目標是D，因此B與C會將此訊框資料丟棄，而D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同時間發送訊框資料時，A與B送出的資料碰撞在一塊，訊框就是損毀，A與B會各自隨機等待一個時間，再透過第一步再傳送該訊框。

瞭解這個程序很重要嗎？我們就來談談：

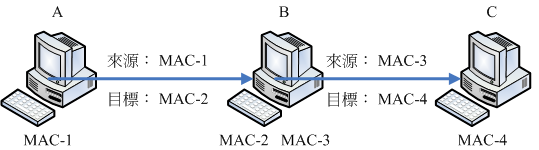
* 網路忙碌時，集線器燈號閃個不停，但主機明明沒有使用網路：透過上述的流程，不管哪一部主機發送出訊框，所有的電腦都會接收到！因為集線器會複製一份該資料給所有電腦。因此雖然只有一部主機在對外連線，但是在集線器上面的所有電腦燈號就都會閃個不停！
* 電腦明明沒有被入侵，為何資料會被竊取：透過上述的流程，只要在B上安裝監聽軟體，將原本要丟棄的訊框資料捉下並加以重組，就能知道A所送出的訊息了。這是為什麼重要資料在網際網路上面要『加密』後再傳輸！
* 既然共享媒體只有一個主機可以使用，為何大家可以同時上網：共享媒體一次只能被一個主機所使用，傳輸100MB的檔案，集線器就得被使用80秒(以10Mbps傳輸時)，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100MB檔案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，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？如果訊框的容量增大，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，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)，能將訊框大小改為9000bytes！但不建議隨便修改。

小標題的圖示2.2.4 MAC 的封裝格式

CSMA/CD傳的訊框資料就是MAC！MAC就是訊框(frame)！訊框上有兩個重要的資料:目標與來源的網卡卡號，因此簡稱網卡卡號為MAC。MAC是網路上傳送資料的最小單位。接下來看一看MAC這個訊框的內容吧！

乙太網路的 MAC 訊框  
圖 2.2-3、乙太網路的 MAC 訊框

目的與來源位址指的都是網卡卡號(hardware address, 硬體位址)，卡號會在訊框的表頭資料使用到！硬體位址由00:00:00:00:00:00到FF:FF:FF:FF:FF:FF(16進位法)，6 bytes中，前3bytes為廠商的代碼，後3bytes則是該廠商自行設定的裝置碼。Linux中用ifconfig查閱網路卡卡號！在MAC的傳送中，僅在區域網路內生效，如果跨過不同的網域，來源與目的的硬體位址就會跟著改變了。因為變成不同網卡間的交流了！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

上圖中，資料要由A經B後送達C，而B有兩塊網路卡，MAC-2與A的MAC-1互通，MAC-3與C的MAC-4互通。但MAC-1不能與MAC-3及MAC-4互通！所以，資料的流通會變成：

1. 由MAC-1傳送到MAC-2，此時來源是MAC-1而目的地是MAC-2；
2. B接收後察看訊框，發現目標是C，為了與C溝通，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。

要透過B(路由器)才將封包送到另一個網域，訊框內的硬體位址會被改變，才能在同一個網域裡面直接進行訊框的流通！

|  |  |
| --- | --- |
| **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能針對網路卡進行抵擋的工作！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！！ |  |

訊框內的資料內容最大可達1500bytes，那為何要規範最小資料為46bytes呢？CSMA/CD機制算出若要偵測碰撞，則訊框總資料量最小要有64bytes，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要有46bytes！如果傳輸的資料小於46byes，系統會主動的填上一些填充碼，以補齊至少46bytes的容量才行！

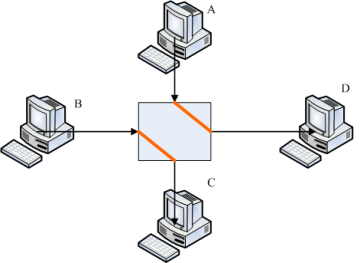
小標題的圖示2.2.5 MTU 最大傳輸單位

通過MAC封裝的定義，標準乙太網路訊框能傳送的資料量最大為1500bytes，這個數值被稱為MTU(Maximum Transmission Unit, 最大傳輸單位)。每種網路介面的MTU都不相同，因此在某些網路文章會看到1492 bytes的MTU等等。不過，在乙太網路上，標準的定義就是1500 bytes。

IP 封包最大到65535 bytes，比MTU還要大！IP封包拆解後才能放到MAC中！等到資料都傳到目的地，再由目的地的主機組裝回來。如果MTU大一些，IP封包的拆解情況就會降低，封包與封包傳送間的等待時間(96 bit time)也會減少，就能夠增加網路頻寬的使用！所以Gigabit的乙太網路媒體才有支援Jumbo frame！一般都定義到9000bytes。但如果考量到整個網路，不建議修改這個數值。因無法確認所有的網路媒體都是支援那麼大的MTU！如果9000 bytes封包通過一個不支援Jumbo frame的網路媒體時，好一點的是網路媒體(例如 switch/router 等)會主動的重組而進行傳送，差一點的可能就直接回報這個封包無效而丟棄了～所以MTU設定為9000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)就將內部網路環境MTU設定為9000，但對外的介面卡可還是標準的1500。

小標題的圖示2.2.6 集線器、交換器與相關機制

* 忙碌的網路運作時，集線器(hub)網路共享媒體會發生碰撞。使用非共享媒體的交換器可避免封包碰撞情況！換器(switch)等級非常多，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記憶體可記錄每個switch port與其連接的PC的MAC位址，所以來自switch兩端的PC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)都有獨立的頻寬！舉例來說，10/100的Hub上連結5部主機，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。由於switch是『每個port都有10/100Mbps的頻寬』，看當時的傳輸行為如何！

  
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都要搶C節點的10/100Mbps來用。switch克服了封包碰撞的問題，因為他有switch port對應MAC的相關功能，所以switch並非共享媒體！現在的switch規格很多，在選購的時候，記得選購可支援全雙工/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): 八蕊的網路線實際上僅有兩對被使用，一對在傳送，另一對則在接收。如果兩端的PC同時支援全雙工時，那示Input/Output均可達到10/100Mbps，即資料的傳送與接收均達到10/100bps，總頻寬則可達到20/200Mbps(Input達10/100Mbps，output達10/100Mbps，而不是Input可直接達到20/200Mbps)如果網路環境想要全雙工，使用共享媒體Hub是不可能的，因為網路線腳位的關係，無法使用共享媒體達到全雙工的！如果switch也支援全雙工模式，在switch兩端的PC才能達到全雙工喔！
* 自動協調速度機制(auto-negotiation)：現在的乙太網路卡是可向下支援的，即Gigabit網路卡可與早期的 10/100Mbps網路卡連結而不會發生問題。但網路速度怎樣判定呢？早期的switch/hub須要手動切換速度才行，新的hub/switch有支援auto-negotiation又稱為N-Way的功能，可自動的協調出最高的傳輸速度來溝通！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，就降速到下一個等級亦即100 Mbps的速度運作！
* 自動分辨網路線跳線或平行線(Auto MDI/MDIX)：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，所以就不需要管你的網路線是跳線還是平行線囉！
* 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時，會導致連線品質不良。因此連結各個節點的網路線長度是有限制的！但造成訊號衰減並非僅有網路線長度而已！如果網路線折得太嚴重，或是網路線接頭部分的八蕊蕊線纏繞度不足導致電磁干擾嚴重，或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不良而造成連線品質惡劣，此時常常會發現偶而可連線、有時卻又無法連線的問題了！

大標題的圖示2.3 TCP/IP 的網路層相關封包與資料

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。乙太網路上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，Internet其實是TCP/IP通訊協定的通稱，Internet是由InterNIC統一管理，其實他僅是分配Internet上面的IP以及提供相關的TCP/IP技術文件而已。

小標題的圖示2.3.1 IP 封包的封裝

目前網路的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記錄的位址僅有32位元，預計在2020年前後就會分發完畢，新興國家或者是新的網路公司，將沒有網路可以使用。因此就有IPv6的產生。IPv6的位址達128位元，多出2的96次方倍的網址數量！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！IP封包可達到65535 bytes，比MAC大，作業系統會對IP進行拆解。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)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4 bits | 4 bits | 8 bits | 3 bits | 13 bits |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | IHL | Type of Service | Total Length | |
| Identification | | | Flags | Fragmentation Offset |
| Time To Live | | Protocol | Header Checksum | |
| Source Address | | | | |
| Destination Address | | | | |
| Options | | | | Padding |
| Data | | | | |

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封包的表頭長度，使用的單位是字組(word)，一個字組為 4bytes。
* Type of Service(服務類型): 內容為『PPPDTRUU』，表示這個IP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65535bytes。
* Identification(辨別碼): 如果IP太大，得先將IP再重組再放到MAC中。而當IP被重組時，每個來自同一個IP就得要有個識別碼以告知接收端來自同一個IP封包。假如IP封包是65536(前一個Total Length 有規定)！
* 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，封包將會被直接丟棄。
* Protocol Number (協定代碼): 在IP表頭記載封包內是那一種協定的資料格式！代碼與對應封包協定名稱如下：

|  |  |
| --- | --- |
| 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) |

* 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: 來源的IP位址，從這裡也知道IP是32位元！
* Destination Address: 有來源還需要有目標才能傳送，這是目標的IP位址。
* Options (其他參數): 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。

重要的有：TTL, Protocol, 來源位址與目標位址也就夠了。

小標題的圖示2.3.2 IP 位址的組成與分級

為了順應人們對於十進位的依賴性，因此將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，每一段中間以小數點隔開，就成了所熟悉的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 |

IP 最小可以由0.0.0.0 一直到 255.255.255.255！在這一串數字中，可以分為兩個部分！主要分為Net\_ID(網域號碼)與Host\_ID(主機號碼)兩部份。先以192.168.0.0 ~ 192.168.0.255這個Class C的網域當作例子來說明：

|  |
| --- |
| 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)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有相同的Net\_ID，且有獨特的Host\_ID』！

|  |  |
| --- | --- |
| **Tips:** 什麼是物理網段呢？當所有的主機使用同一個網路媒體串在一起，這些主機在實體裝置是連線在一起的，就可稱為這些主機在同一個物理網段內了！同一個物理網段內，依據不同的IP的設定，而設定成多個『IP網段』！ |  |

上例中的192.168.0.0-255(共256個)，這些IP是同一個網域內的IP群(同一個網域/網段！)，同一個Net\_ID內，不能有相同的Host\_ID，否則IP會衝突，可能造成兩部主機都沒法使用網路！



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不可重複；Host\_ID二進位表示法中，不可同時為0或1，因為全0表整個網段的位址(Network IP)，而全為1則表示為廣播的位址 (Broadcast IP)。上例中，這個網段可用來設定主機的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)，則這些主機都可透過CSMA/CD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)；
* 不同區網在同物理網段的情況：在同一個物理網段內，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。要透過路由器(router)才能將兩個網域連結在一起。
* 網域的大小：當Host\_ID佔用的位元越大，即Host\_ID數量越多，表同一個網域內可用以設定主機的IP數量越多。

同公司內的所有電腦，設在同一個網域內時，每一部電腦可直接透過MAC進行資料交流，不必經由Router轉遞封包。



IP 的分級

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 |

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。上表中只要記憶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。因此能夠設定在一般系統上面的，就只有Class A, B, C三種等級的IP！

小標題的圖示2.3.3 IP 的種類與取得方式

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，這種IP才可連上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 規劃時就擔心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 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

這三個 IP 網段就只做為內部私有網域的IP溝通之用，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)

特殊的 loopback IP 網段

在沒安裝網路卡的機器，希望可以測試機器上設定的伺服器環境可不可以順利運作，就利用內部迴圈網路！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！啟動了WWW 伺服器，然後在X-Window上執行http://localhost就可直接看到主頁囉！不需要安裝網路卡！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？也就是使用這一個內部迴圈！當要測試你的 TCP/IP 封包與狀態是否正常時！(所以哪一天有人問你！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？當然可以囉！使用 127.0.0.1 這個 Address 呀！)



IP 的取得方式

主機的IP與相關網域的設定方式主要有：

* 直接手動設定(static)：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ISP申請註冊，取得帳號密碼後，直接撥接到ISP，ISP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。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。台灣地區ADSL撥接後取得的IP通常是public IP。
* 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。

小標題的圖示2.3.4 Netmask, 子網路與 CIDR (Classless Interdomain Routing)

IP是有等級的，一般電腦系統上使用Class A, B, C。如果用Class A設定一個區網，哪有這麼多電腦設定在同一個Class A的區段內(256x256x256-2=16777214)？而且回想一下CSMA/CD，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，真沒效率！

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24位元，其實還可以將這網域切的更細，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 。這樣原來的一個Class C的網域就可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，方便分類的設計。



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。以192.168.0.0 ~ 192.168.0.255 這個網域，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192.168.0.0 ~ 192.168.0.255的Class C 的網域中，Netmask就是 255.255.255.0！剛剛提到當Host\_ID全為 0及1時該IP是不可以使用的，Host\_ID全部為0表該網段的Network，全為1時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

剛剛提到 Class C 還可以繼續進行子網域 (Subnet) 切分，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 ....)。

|  |
| --- |
| 例題：計算172.16.0.0，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。不過題目的Net\_ID佔23個位元。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  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)至於粗體部分代表 host\_ID 啊！ |

無層級 IP： CIDR (Classless Interdomain Routing)

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 |

Netmask裡面的Net\_ID都是1，Class C共有24 bits的 Net\_ID，所以就有類似上面192.168.0.0/24的寫法！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

小標題的圖示2.3.5 路由概念

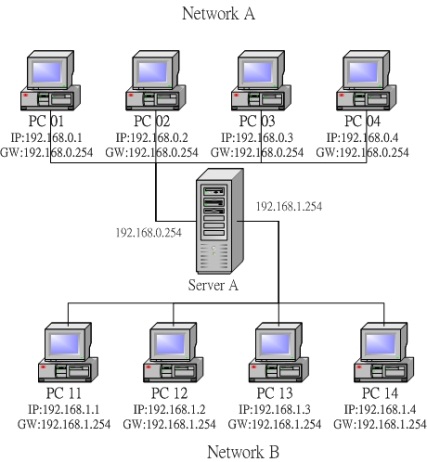
我們知道在同一個區網裡面，可以透過IP廣播的方式來達到資料傳遞的目的。但如果是非區網內的資料呢？這時就得要透過那個所謂的郵局(路由器)的幫忙了！這也是網路層非常重要的概念喔！先來看看什麼是區網吧！

|  |
| --- |
| 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： 192.168.10.100的Network為192.168.10.0，但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。 |

兩個網段的資料無法透過廣播傳遞資料，得經過IP的路徑選擇(routing)功能。下圖當中有兩個不同的網段，分別是Network A與B，這兩網段經一路由器(Server A)進行資料轉遞，當PC01要傳資料到PC11時，IP封包該如何傳輸呢？

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不能直接互通資料。PC01與PC11是如何知道他們兩個不在同一個網段內？是透過Net\_ID來發現的！主機想要傳送資料時是參考『路由表 (route table)』，每部主機都有自己的路由表』，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)：在本案例中，PC01與PC11非同一網域，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，就直接將該IP封包送到預設路由器(default gateway)，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，分析自己的路由資訊，向後繼續傳輸到正確的目的地主機上頭。

  
圖 2.3-2、簡易的路由示意圖

|  |  |
| --- | --- |
| **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)，路由器有IP Forwarding的功能，並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，主機與主機設定的Gateway必定是在同一個網段內喔！ |  |

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！

小標題的圖示2.3.6 觀察主機路由： route

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！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。 |

PC 01為192.168.0.0/24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，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。如果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！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！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！

小標題的圖示2.3.7 IP 與 MAC：鏈結層的 ARP 與 RARP 協定

IP與MAC有一個關連性就是ARP(Address Resolution Protocol, 網路位址解析)協定，及RARP(Revers ARP, 反向網路位址解析)。當要瞭解某個IP是設定於某張乙太網路卡上時，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，主機就會知道對方所在的網卡，接下來就能夠開始傳遞資料。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主機的ARP table中 (記憶體內的資料)記錄20分鐘以避免每次要傳送都得要重新來一遍ARP協定。

|  |  |
| --- | --- |
| 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，當發送ARP封包取得的IP/MAC對應，記錄的ARP table 是動態的資訊(一般保留20分鐘)，會隨時隨著網域裡面電腦的IP更動而變化，所以不要擔心常更動電腦IP，因為ARP table會自動的重新對應IP與 MAC 的表格內容！

小標題的圖示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』。ICMP是一個錯誤偵測與回報的機制，最大的功能是確保網路的連線狀態與連線的正確性！ICMP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！因為Internet上面有傳輸能力的就是IP封包！ 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) |

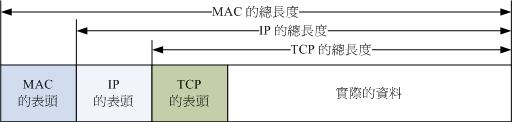
如何利用ICMP來檢驗網路的狀態呢？最簡單的指令就是ping與traceroute，這兩個指令透過ICMP封包的輔助來確認與回報網路主機的狀態。在設定防火牆的時候，最容易忽略ICMP的封包了，因為只會記住TCP/UDP而已～事實上，ICMP封包可以幫助連線的狀態回報，除了上述的8可以考慮關閉之外，基本上ICMP封包也不應該全部都擋掉喔！

大標題的圖示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，就不是 IP的任務！那是傳送層的任務之一。從[圖 2.1-4](http://linux.vbird.org/linux_server/0110network_basic.php#fig2.1-4)可看到一個是連接導向的TCP封包，一個是非連接導向的UDP封包！

小標題的圖示2.4.1 可靠連線的 TCP 協定

TCP封包資料要能夠放到IP的資料袋當中！所以將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

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

TCP封包的表頭資料，以Source Port, Destination Port及Code比較重要，底下就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): 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？就是連接到port上頭！舉例來說，網站有開放WWW伺服器，表示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。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)，就要分段。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)！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，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，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。想要我網站上的資料，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，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。

假如IP是網路世界的門牌，那麼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，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:**  一部主機上面這麼多服務，那跟這部主機進行連線時，該主機怎麼知道要的資料是WWW還是FTP？是透過埠口！因為每種Client軟體所需要的資料都不相同，如瀏覽器需要的資料是WWW，所以該軟體預設就會向伺服器的port 80索求資料；而如果使用filezilla來進行與伺服器的FTP資料索求時，filezilla預設就是向伺服器的FTP相關埠口(預設是21)進行連接的動作！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，而民眾就好像是Client端來的封包。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，如果是要領錢，服務人員就會請你到領錢的窗口去填寫資料，是不會跑錯的。萬一跑錯了怎麼辦？當然該窗口就會告訴你『我不負責這個業務，你請回去！』！所以該次的連線就會『無法成功』！ |  |

特權埠口 (Privileged Ports)

網路既然是雙向的，一定有一個發起端。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。Linux環境下各網路服務與port number的對應預設寫在/etc/services檔案內喔！底下列出幾個常見的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 |

小於1024以下的埠口要啟動時，啟動者的身份必須是root，所以才叫做特權埠口！不過如果是client端，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

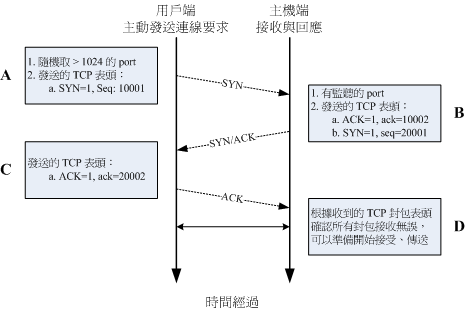
由於網路是雙向的，要達成連線的話得要伺服器與用戶端均提供了IP與埠口才行。因此將這成對的資料稱為Socket Pair！

* 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！

小標題的圖示2.4.2 TCP 的三向交握

TCP 被稱為可靠的連線封包，其中最重要的就是三向交握的功能。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，就被稱為三向交握(Three-way handshake)。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。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，就會製作一個帶有SYN=1, ACK=1的封包，其中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)，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: 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。

必須要瞭解『網路是雙向的』！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時，『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！

小標題的圖示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，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，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。UDP比較適合需要即時反應的資料流，例如影像即時傳送軟體等。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！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。

大標題的圖示2.5 連上 Internet 前的準備事項

在Internet上面其實使用的是 TCP/IP 這個通訊協定，所以需要Public IP來連接上Internet！不過『為啥不知道Yahoo 的主機IP，但是主機卻可連到Yahoo主機上？』如果你有發現這個問題，你可以準備開始設定網路囉～

小標題的圖示2.5.1 用 IP 上網？主機名稱上網？DNS 系統？

要連上Internet就要有TCP/IP！網路是依據人類的需要建立的，不過人類對於IP這一類的數字並不敏感！就將主機名稱與他的IP對應起來，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！

主機名稱(Hostname)對應IP的系統就是的Domain Name System (DNS)！DNS服務的最大功能就是進行『主機名稱與該主機的IP的對應』的一項協定。當在瀏覽器上面輸入了『http://tw.yahoo.com』時，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！

電腦必須要向DNS伺服器查詢Hostname對應IP！DNS主機的IP須先在我的電腦設定好，並且必須要是輸入IP，不然我的電腦怎麼連到 DNS 伺服器去要求資料呢？在Linux裡面，DNS主機IP的設定是在/etc/resolv.conf檔案裡！

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？那就設定Hinet那個最大的DNS伺服器吧！IP是168.95.1.1！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！

小標題的圖示2.5.2 一組可以連上 Internet 的必要網路參數

一組合理的網路設定需要的資料是：IP, Netmask, Network, Broadcast, Gateway, DNS.

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！但如果是固定制(如學術網路)，就得使用上面的參數來設定你的主機！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

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 封包：網路上資料的傳輸是以封包的方式傳遞，封包裡通常含有來源和目的的IP位址，來源和目的的MAC位址和資料。  
2. IP位址：要上網一定要有IP位址，如果IP位址不夠，那麼就需要IP分享器囉。  
3. MAC位址：每一張網路卡其實都有一個MAC位址，全世界不會有跟你相同的MAC位址的網路卡或設備。  
1. HUB(集線器)：一般而言，HUB有兩大特性，一個就是廣播，一個就是半雙工。  
廣播是指，當A電腦要透過HUB送資料給B電腦的時候，A送出來的資料其實連接在這台HUB上的電腦都會收到，但是只有B電腦會將資料收起來，其他電腦則是將封包丟掉。半雙工是指，收資料或送資料不能同時，你一次只能做其中一種。由於HUB的這種特性，所以當HUB連接非常多電腦時，網路就會變慢。  
2. Switch(交換器)：和HUB看起來一樣，但實際上差別很大。首先switch並不一直廣播，而且是全雙工的。主要是SWITCH會記錄封包中的MAC位址所以當電腦A傳送資料給電腦B時，其他電腦並不會也收到資料，而且這個時候別的電腦也可以同時互相傳送資料。雖然SWITCH有上述的好處，但是要傳送的資料封包每一個都必須經過SWITCH判斷決定要送往哪一台電腦，所以會有一些延遲，因此有時候電腦數少於五台，用HUB反而比SWITCH快。  
3. Router(路由器)：路由器通常最少會有兩個介面，而這兩個介面分別區隔不同的IP網段。例如IP分享器有WAN和LAN兩種介面，區隔WAN的實際IP與LAN的虛擬IP網段，所以說IP分享器是Router其實並沒有錯誤。

4. IP分享器：通常有一個WAN port和1~4個不等的Lan port(其實是Switch)。WAN port一般是接ADSL modem(也就是小烏龜啦)，而Lan port則是接到PC電腦。這個設備主要的功能是NAT，也就是做IP分享，他會將WAN port的真實IP(可以是固定IP或浮動IP)分享給LAN port的電腦使用。  
備註解釋一下NAT功能：一般IP分享器LAN port裡的電腦室使用虛擬IP，也就是俗稱的假IP，這個網段通常是192.168.X.X，最常見的是192.168.0.X與192.168.1.X。這個IP網段是保留的網段，在實際網際網路並不能使用。NAT這個功能負責記錄網卡MAC位址與假IP的關係並做轉換。  
舉例來說，假設使用者有兩台PC透過IP分享器上網  
A電腦IP為192.168.0.2，網路卡MAC位址是00046F12301A  
B電腦IP為192.168.0.50，網路卡MAC位址是000879215B09  
當A電腦想要連網站yahoo.com.tw時，A電腦會送出資料給yahoo，而IP分享器會記錄192.168.0.2的MAC是00046F12301A，並把資料內來源IP 192.168.0.2改成WAN port的真實IP，然後送往YAHOO。Yahoo收到以後會依照真實IP位址回傳資料給IP分享器，IP分享器再依照資料內的MAC位址判別出這封包是要給A電腦的，於是把目的IP改成192.168.0.2送往A電腦。  
簡單的說，IP分享器是透過傳送封包內的MAC位址來分辨這個封包資料是要給哪一台電腦的，進而達到分享IP的功能。