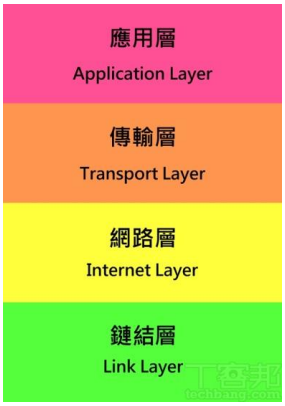
網路架構大概論 2－網路模型、封包架構、解析 OSI 7 層作用

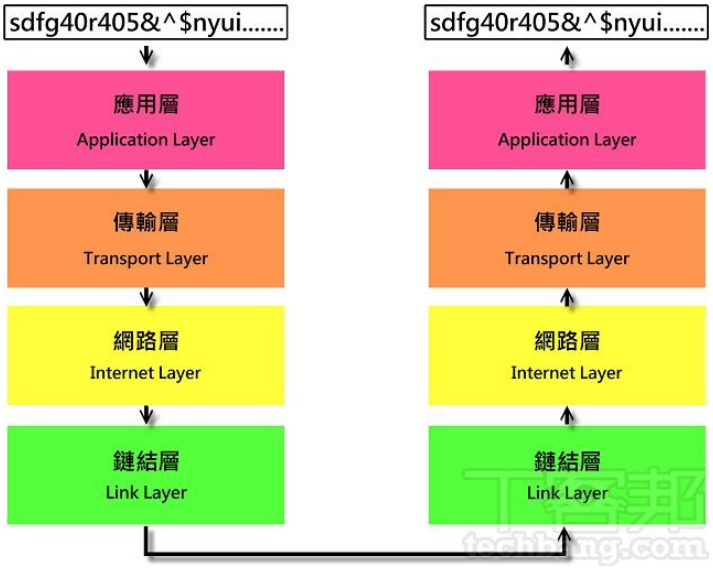
上一篇我們預告讀者，接下來將會介紹現今網路最重要的架構：TCP/IP，TCP/IP 從 ARPANet 後期發展以來，直至今日仍是網路主要使用的架構，如果你今天要設計能夠連到網路上的產品，支援 TCP/IP 就對了。

分層的意義

在正式進入主題之前，先來了解為何網路使用分層結構，而非從頭到腳一手全包的設計。由於網路需考量許多實際應用上的困難，如傳輸不同種類的資訊、不同版本以及作業系統的交流需求，甚至牽扯到更廣大的範圍，如遠距微波傳送、跨洲海底電纜等不同傳輸媒介。如果要設計能夠包山包海、全部通吃的網路架構是件吃力不討好的事，況且也無法預測未來會出現什麼需求，於是需要個具有彈性的網路架構，還要能夠兼顧未來發展。



▲網路模型的 4 層架構，工作彼此獨立卻又需要它層的協助。



▲資料經過網路模型的處理方式，先將傳送端的資料一層層打包完畢之後，送到網路上傳送。接收端接受到東西之後，再依相反順序拆開。

網路架構模型分為 4 層，也就是經常聽到的 DoD（Department of Defense）或是 TCP/IP 模型，如附圖所示，最上層為應用層（Application Layer），接下來依據為傳輸層（Transport Layer）、網路層（Internet Layer）、鏈結層（Link Layer）。這 4 層分工需要相互合作，卻又彼此獨立，好比寄信一般，需要郵局處理郵件，道路和交通工具負責運送，信才可送達目的地，但是你卻不用管郵局如何經營、交通工具如何設計、道路怎麼鋪設才會平坦又耐用。

將網路架構分層出來，便加入了許多可能性，只要符合這個網路模型，各層如何處理資訊的方式就各自獨立，不受其它外在條件的限制。舉個小小例子，筆電能夠透過有線和無線方式連上網路，但是這 2 種傳輸介質所看到的網頁有所差異嗎？應該不會發生有線介面看到 Dream Girls，換成無線介面就變閃亮三姊妹這種情形。同時，分層模型所具備的向後相容性也可以在無線傳輸規格上看到，從現今的 802.11 a/b/g/n，到未來的 802.11 ac/ad，均能夠傳送網頁資料，讀者應該也沒有聽過換個傳輸規格，網頁就要重寫這種荒唐事。

在實際傳送資料時，資料的路徑也是由上至下傳送，最後包成一包貨物交給網路中稱為路由器（router）或是閘道器（gateway）的機器傳送；到達目的地之後，再由下而上，層層拆開。

對於這 2 台相互交流的電腦來說，只有最原始的那份資料才是最重要的，猶如寄送包裹，裡面所攜帶的東西才是我們最關心的，氣泡紙或是外箱、地址，僅是確保東西正確送達目的地的手段。接下來將逐層介紹這個網路模型，了解各層到底在做什麼事。

其它的網路模型

部分讀者或許在其它地方看過不同名稱、不同層級的網路模型，本文採用 RFC 1122 中定義的模型。如果讀者有意了解其它的網路模型，請至英文版維基百科，搜尋「Internet protocol suite」。

應用層（Application Layer）

網路模型的最上層，也是想要利用網路傳輸資料的程式，能夠直接碰觸到的層級。無論是收發郵件的 POP3/SMTP/IMAP、或是網頁傳送標準 HTTP、檔案傳輸協定 FTP、現代網路不可或缺的網域/位址轉換伺服器 DNS，還是近年興起的 BitTorrent P2P（peer to peer）傳輸協定，均屬於這層的管轄範圍。

這層直接對應用程式開放，提供不同程式同一類型的服務，譬如你可以使用你所喜歡的程式開啟網頁，也可以使用 Outlook 或 Thunderbird 收發郵件。各種不同的應用層協定，會為該協定提供的服務提供必要的處理機制。同時也因為服務眾多，在這層的協定數量為 4

層之冠，光是想看個網頁，就可能有 3 種協定參與其中，包含取得私有 IP 的 DHCP、網域與位址轉換的 DNS、以及取得網頁資料的 HTTP。

傳輸層（Transport Layer）

資料經過了應用層之後，下一層便是傳輸層，此層對應用層傳送過來的資料進行處理，建立 2 台電腦之間可靠的傳輸。在此層常見的 2 種協定為 TCP 與 UDP，絕大多數應用層協定都會選擇 TCP，因為 TCP 提供一連串偵錯、重送、資料順序、流量控管等實作，確保 2 台電腦之間的連線正常，不會產生投桃報李的狀況。

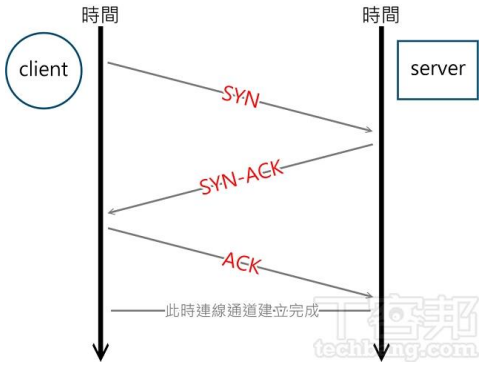
相較於 TCP 提供連線的可靠性，UDP 則專注於資料的傳遞，從 TCP（傳輸控制協定、Transmission Control Protocol）與 UDP（使用者資料電報協定：User Datagram Protocol）的名稱可看出端倪，UDP 幾乎不管資料是否正確送達，需要應用層自行處理。

但別認為 UDP 就是項垃圾，像是 DNS 查詢、DHCP 請求與配發都是使用 UDP 協定，另外像是部分影音串流服務，講求大量且迅速的資料傳遞，偶爾丟掉資料不會造成太大影響（因為最終觀賞或聆聽的終端是人，不易察覺畫面或聲音細微的缺陷）；即時通訊也是 UDP 的愛好者，因為人與人的溝通具時間性，上一句話漏掉的資訊，下一句話再傳來並沒有意義。除了上列 2 種常見的協定之外，前往中國旅遊出差，好用的翻牆方式 PPTP，也住在這層喔。

TCP 的三方交握

為了讓讀者有感體會 TCP 在建立穩定連線的努力，先偷渡 TCP 建立連線時的三方交握原理。當客戶端想向伺服器建立連線時，會先發出 SYN 資訊，接著伺服器回應 SYN-ACK 資訊，客戶端接受到之後，再回應 ACK，此時客戶端和伺服器的連線就算是建立完畢，可以開始傳送資料了。

UDP 則不進行此類手續，直接往對方送，途中就像是飛鴿傳書，小鳥變成焦仔巴也沒人知道。



網路層（Transport Layer）

前文中舉了許多郵寄的例子，此層就相當是郵件或包裹上方的地址欄位，無論你想要寄去世界上任何一角落，只要將地址寫好，郵件或包裹便會到達目的地。

網路層能夠提供實體介面網路卡 1 個邏輯上的位置，也就是我們常聽到的 IP 位址，此 IP 位址是唯一的（或至少在同層級的網路中是如此），所以資訊才能夠傳輸到正確的目的地。此層會將資料的目的地和來源地寫入，如同我們在信件上填入收件人和寄件者。

除了賦予每個網路介面卡邏輯 IP 位址外，IP 還定義了資料封包該如何繞送的規則，如同郵局有各地區的小據點，也有大型的轉運站，藉由這些規則，選擇較佳的傳送路徑，將東西正確傳遞。

網路層的 IP 協定，事實上並非完全可靠，有可能在途中因為某些原因（封包緩衝區溢位、TTL 為 0……），整個資料完全不見。IP 協定基本上是以「盡可能去做」為宗旨，有多少運算資源做多少事，超過負荷的資料並不提供其它方式修補，丟掉就是掉了。所以才需要傳輸層做這類確保工作。

鏈結層（Link Level）

位於整個網路架構的最底層，負責制定資料傳輸的「實體」規格。所謂「實體」，就是我們眼睛看的到，手可以摸到的部分。以乙太網路來說，凡是接頭規格、傳輸線材種類、不同電壓代表何種資訊，都是鏈結層負責的內容。

除了目前常見，採用 RJ-45 以及 4 對雙絞線的乙太網路之外，古老的乙太網路使用類似於電視同軸纜線的線材相互連結。以及目前大家比較常聽到的光纖、VDSL，也包含在其中。

鏈結層的運作也包含不同實體網路的轉接，像是從家中電腦的乙太網路轉換為對外的光纖環境，負責的光電轉換機就會先把乙太網路的標頭拿掉，加入為光纖制定的標頭，在光線上傳送。當光電轉換機從光纖上收到資料時，也會先把光纖標頭拿掉，加入乙太網路的標頭，再將資料送到電腦上。

資料層層包的術語

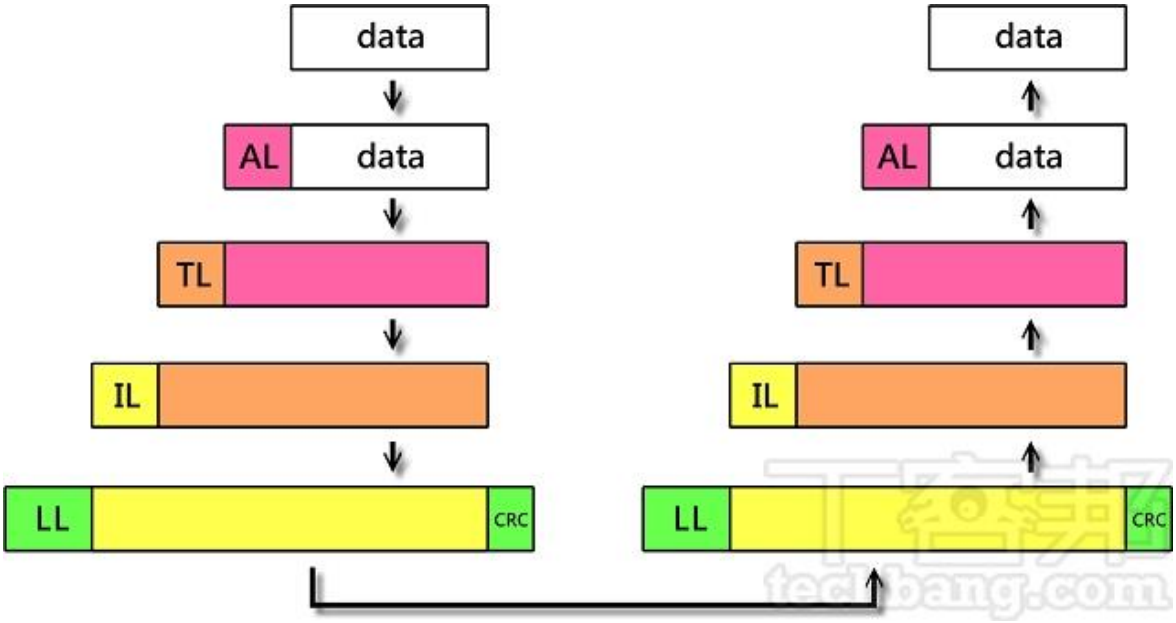
資料從應用程式丟出之後，經由應用、傳輸、網路、鏈結 4 層處理之後才可以在網路上遨遊。這 4 層分別會在資料的外圍加料，裡面標示著各層處理所需的資訊，也就是說，從資料到網路上這段期間，所需傳送位元總數會漸漸地變大。在此也先向讀者說明，由於網路

這種層層包的特性，網路的最大傳輸速度並不等於資料的淨傳輸量，所以別再怪 ISP 業者欺騙大眾，少給你頻寬。為了傳輸資料，這些標頭一定會佔去部分頻寬。

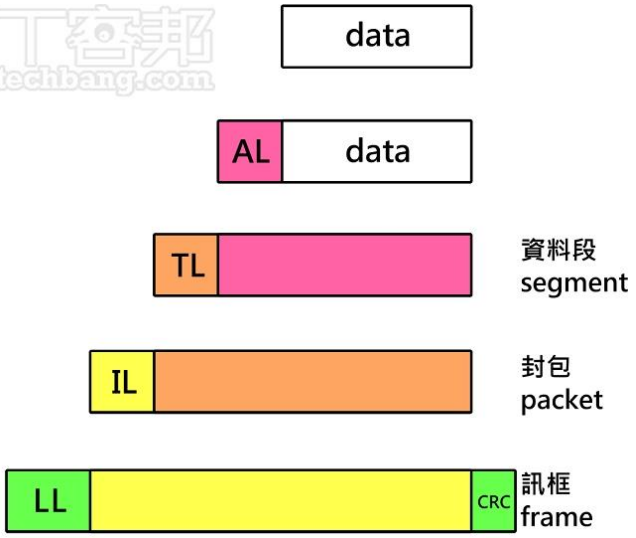
資料每經過 1 層，我們都歸給它名字，代表原始資料已經被這層處理過了，同時也可以藉由這些名稱，判斷目前東西在哪一層。

從應用程式傳來的資料，進入傳輸層，包入傳輸層表頭，此時整段東西稱為資料段（segment）。資料段進入網路層，包入表頭後稱為封包（packet）。進入鏈結層處理之後，會將封包前後分別加入表頭以及表尾，整個稱之為訊框（frame）；鏈結層也是唯一的一層，同時在資料的前後加入資訊。

有一點須請讀者注意一下，如果我們說「封包的資料」，指的就是封包表頭之後所有的資訊，包含應用層的表頭以及資料。因為當我們在討論某層的資訊時，所關心的就是那層的表頭，後面跟著的東西就不理它了。

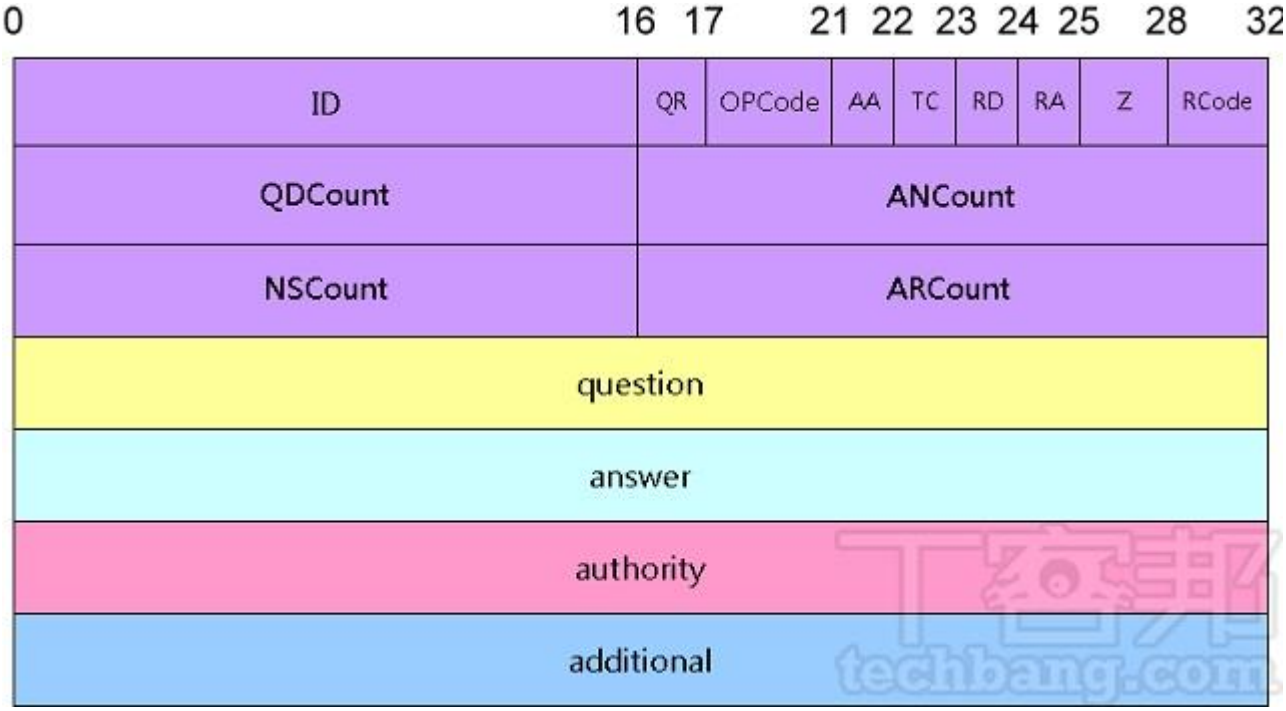


▲一段資料經過網路 4 層模型的傳輸情形，傳送端將資料一層層處理、打包；接收端則依相反順序拆開處理，最後得到傳送端所要傳送的訊息。



4 大層級的協定解析

如果前面的東西已經讓你頭昏腦脹、昏昏欲睡，筆者建議改天再來看下半段的內容；如果反而覺得精神抖擻，就請你把點心和飲料準備好，因為接下來是更難的東西，實際進入網路資料傳輸的過程，並了解有些什麼欄位，這些東西又是依據什麼製造出來的。



▲DNS 的訊息格式

應用層

應用層的協定非常多，像是網頁 HTTP、郵件 POP3、檔案 FTP……接下來將舉幾個常用的協定說明，避免腦袋超載燒毀。

DNS

在我們實際和網頁伺服器連結之前，都必須先向 DNS 伺服器發送出請求，詢問 www.techbang.com 這個網址的實際 IP 位址，才能夠獲得 60.199.208.218 這串數字，網頁瀏覽器才能向正確的伺服器請求網頁資料。

DNS 的訊息長度可變，主要分為 5 個區段：header、question、answer、authority、additional。其中 header 標明了其後 4 個區段相關的資訊，question 有自行的格式，而其後 3 個區段的格式相同。

header 長度固定為 12 Byte，其中包含 ID 16 bit，讓這個問題有個號碼，之後回傳過來的答案也會包含相同的號碼，讓電腦知道回傳的答案是對應哪個問題。QR 欄位僅 1 bit，數值為 0 時表示這個訊息為問題，為 1 時表示答案。OPCode（4 bit）為問題的型態，0 為標準查詢、1 為反向查詢、2 為要求伺服器狀態、4 為通知、5 為更新，剩餘的值為空白或是未定義。

AA（1 bit）為 1 時表示給予回應的 DNS 伺服器，有獲得網域擁有者的授權。TC 表示資料是否太大遭到截斷，RD 為請求以遞回方式查詢，RA 為回傳可以遞回方式查詢，Z 部分有 3 個位元保留不使用。RCode 為回應訊息的狀態，0 表示正常、1 表示問題有錯誤、2 為 DNS 伺服器發生問題、3 表示所查詢的網域不存在、5 表示 DNS 伺服器因為設定策略的緣故，不回應問題，其它數值還有其它意義，或是保留未定義。QDCount、ANCount、NSCount、ARCount 各為 16 bit，分別代表後面有幾個問題/答案/管轄/額外資訊。

question 欄位較為簡單，僅包含 question name、QType、QClass 3 種，question name 就是想要查詢的網域名稱、QType 表示回應的格式、QClass 為問題的等級，大部分設為 1。

剩下的 3 種欄位格式皆相同，name 擺放詢問的網域、type 指出 RData 的類型、class 為 RData 的等級、TTL 為此答覆有效的時間，電腦便可建立此 DNS 查詢快取的有效時間、RData length 為 RData 的資料長度、RData 當然就是存放回覆的答案啦。

DNS 查詢欄位內容，除 header 之外，並沒有固定的長度，但 name 會在結尾處加入 0 定位，其它 3 種欄位使用 RData length 得知資料長度。

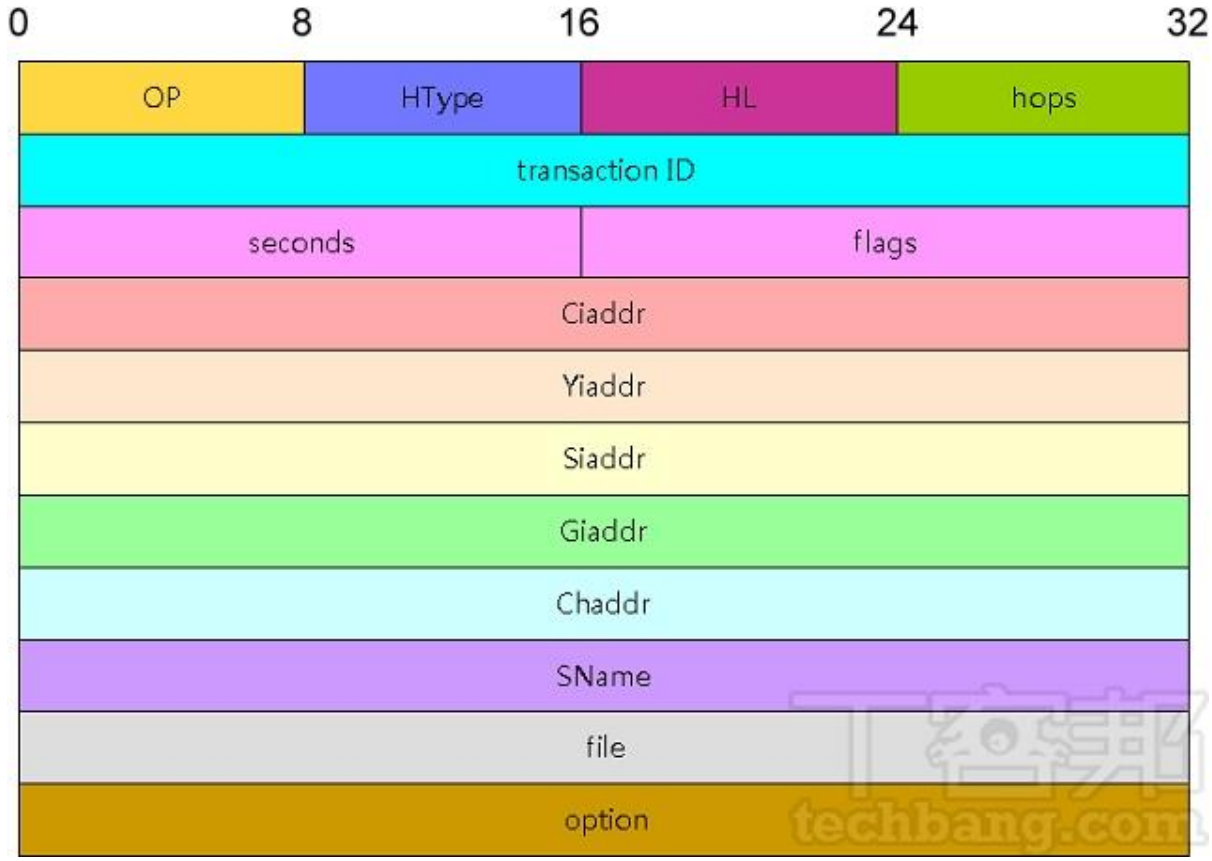
DHCP

DHCP 的發明給了網管莫大的好處，不必再逐台電腦設定 IP 位址這種煩人的東西。當有新電腦加入網路時，會向 1 個廣播位置發送查詢訊息，在此網路中的 DHCP 伺服器撈到這個查詢時，會回傳 1 個提供訊息（內含相關 IP 位址設定資料），表示我能夠提供 IP 位址給你。之後這台新加入網路的電腦繼續向廣播位址發送請求（因為可能不只 1 台 DHCP 伺服器可提供 IP 位址，所以要以廣播的方式告知沒被選中的 DHCP 伺服器），DHCP 再回應 1 個確認訊息，至此新電腦獲取 IP 位址的流程宣告結束。

OP 為封包傳遞方向，1 表示由客戶端傳給伺服器端，相反傳遞方向時為 2。HType 表示網路類型，常用的 Ethernet 為 1。HL 定義網路硬體位址（MAC）的長度，在 Ethernet 環境下，其 MAC 長度為 6 Byte，所以值為 6。

hops 開始由客戶端發送時的值為 0，但若是需要跨不同子網路，每經過 1 個路由器，此值就加 1。transsaction ID 是個隨機數字，用戶端發送時會任意填入，伺服器回應時便把此號碼照抄，用戶端便可得知回應的封包是針對哪個要求封包。seconds 丟出 DHCP 請求後經歷的時間，可讓伺服器在接收到大量請求後決定先回應誰（經過時間越久可能優先回覆）。flags 為 1 時表示此封包利用廣播方式傳送，其餘皆未定義。Ciaddr、Yiaddr、Siaddr、Giaddr、Chaddr 欄位都是填入位址，若是用戶端希望繼續使用上次的 IP 位址，則在 Ciaddr 中填入，Yiaddr 則是填入伺服器欲分配給用戶端的位址。Siaddr 為 DHCP 伺服器的自身 IP 位址，Giaddr 則是在跨網域時，路由器的位址，Chadder 則是用來填入用戶端網卡的 MAC 位址。

SName 放入 DHCP 伺服器的名稱，File 則是當用戶端需要經由網路開機時，等會兒將傳送的開機檔案名稱。options 是選用欄位，包含 DHCP 用戶端可接受的 DHCP 封包長度，或是用戶端租用 IP 位址的期限……



▲DHCP 的欄位資料

改變埠號逃過網管過濾

絕大多數的應用程式，都喜歡使用特定的埠號，因此網管人員可在防火牆中設立黑名單，將部分封包擋下（如 BitTorrent、Live Messenger、Skype……），便可達到阻止用戶使用這些程式的目的，因此我們可以手動更改埠號，繞過這些限制。

如果網管人員比較狠，只在防火牆裡使用白名單，僅允許 HTTP 常用的 80 埠，那麼換埠號也無濟於事。

HTTP

HTTP 超文本傳輸協定（HyperText Transfer Protocol），是現今網頁最常使用的協定。HTTP 的格式並不像 DNS 有欄位可以填，而是直接以文字方式要求、回應，像是人與人溝通的模式。但這並不代表電腦能夠識別自然語言，HTTP 還是有制定句型格式，以及什麼指令該做什麼事，整體來說偏向高階程式語言的感覺。HTTP 使用 method（方法）、path（路徑）、protocol（協定）下面就是 1 個簡單的 HTTP 協定範例：

GET /image/image001 HTTP/1.1
GET 是 HTTP 中經常使用的指令，代表向伺服器要求某件檔案，而這檔案就是在 image 資料夾之下的 image001。HTTP/1.1 代表所使用的 HTTP 協定版本，在網路中有著 1.0 和 1.1 版，其它指令還有 HEAD（不傳回整個資料，僅傳回 header 的訊息），DELETE（刪除特定檔案），POST（上傳資料請伺服器處理）等。

除了向伺服器發送檔案請求之外，還會附加一些 header 上去，例如下面這個例子：

User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)
User-Agent 代表之後跟著的字串為目前網頁瀏覽器使用的版本，在此例中為 Mozilla 4.0，還與 Microsoft 公司的 Internet Explorer 5.5 相容，以及使用 Windows NT 5.0 作業系統。所以以後遇到網頁上顯示你正在使用的網頁瀏覽器版本以及作業系統，不用再那麼訝異，以為自己的電腦被入侵了，其實是網頁瀏覽器發送出去的訊息。

以上列舉 3 個應用層的協定，若讀者對其它協定有興趣，可查閱 TCP/IP 相關書籍。

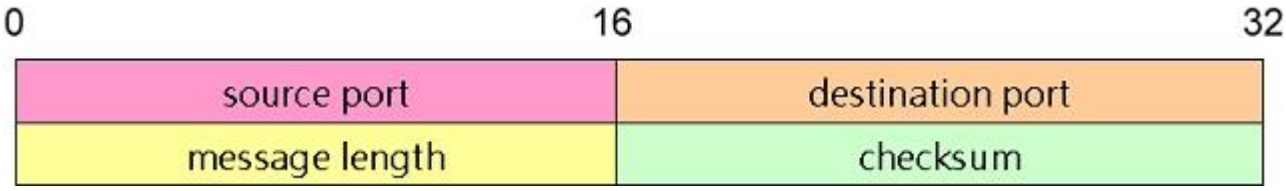
傳輸層

傳輸層負責打包由應用層傳來的資料，或是拆解由網路層傳來的東西，協定不若應用層那麼多，甚至可以說是網路 4 層模型中最少的，絕大多數僅使用其中 2 種傳輸協定：TCP 和 UDP。

UDP
使用者資料電報協定（User Datagram Protocol）的表頭格式相當簡單，僅包含 4 個部分：來源連接埠、目的連接埠、資料長度、校驗，全部只有 8 Byte 的長度。

當使用者的電腦使用許多網路應用程式時，便需要連接埠的幫忙，每個單一的連接埠負責服務單一應用程式，讓資料不混淆。由於埠的欄位有 2 Byte，數值可從 0～65535，其中有幾個埠是大家公認，提供給特定服務的專用埠，如 DNS 服務的 53、網路時間協定的 123……。當然，這只是公認，並非絕對，你想要使用這些埠也可以，但若發生連線問題可就怨不得人。

資料長度就是此份 UDP 資料的大小，最小 8 Byte（僅有 UDP 標頭），校驗碼負責檢查所攜帶的資料是否正確。UDP 和 TCP 的校驗計算中，有使用到「虛擬表頭」的功能，也就是將有關 IP 位址的資訊也加入計算之中，但實際上這些資訊並不隨 UDP 和 TCP 傳送，而是由更下 1 層的網路層獲得。



▲UDP 的表頭欄位定義

TCP

傳輸控制協定（Transmission Control Protocol）是現今網路主要的傳輸協定，重要性甚至與下 1 層的 IP 協定共稱為 TCP/IP，想要網路產品順暢傳輸資料，選擇這 2 種協定就對了。

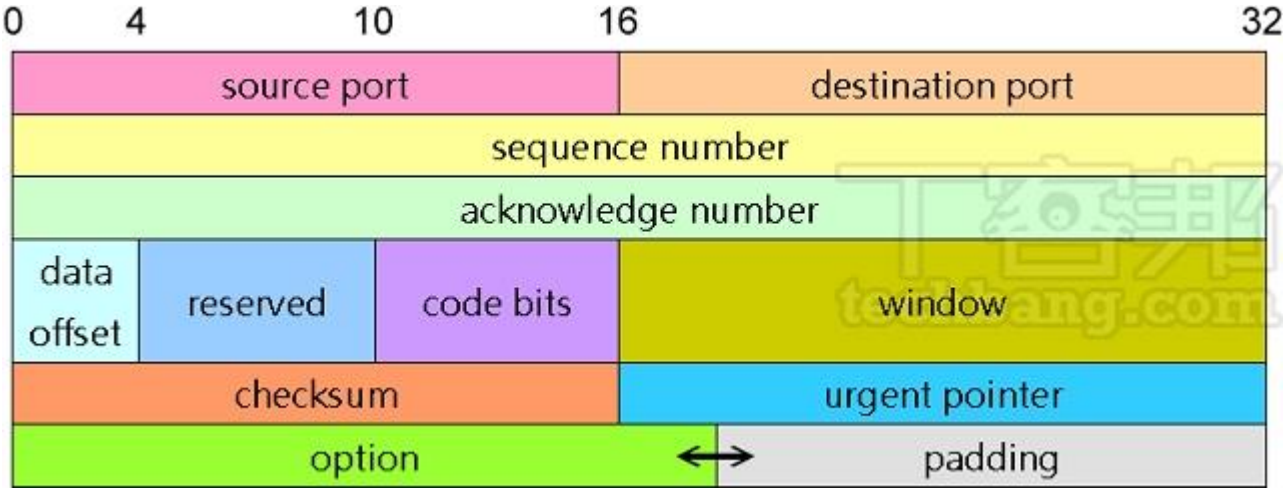
TCP 的表頭比起 UDP 複雜得多，因為 TCP 擁有更多控制以及復原功能，包含確認連線的 3 方交握程序、回傳確認封包（ACK）、限制流量的視窗大小。TCP 也有公認埠號的存在，FTP 資料埠 20、FTP 控制埠 21、SMTP 的 25、HTTP 的 80 等。

首先是來源埠與目的地埠，原理和 UDP 相同。sequence number 在傳送方首次連線時會設定隨機亂數，之後會逐漸加上已傳輸的資料量（以位元組為單位）。acknowledge number 主要是給接收方使用，將 sequence number 和接收位元組相加，傳回給發送方。以上的流程操作確認了 TCP 不會有遺失封包的情形發生，因為當發送方在一定時間內沒有收到接收方回傳的 ACK，就判斷封包傳送有問題無法送達，會將資料再傳送 1 次。

data offset 標明資料的起始處在哪裡，因為受到 TCP 表頭尾端有個 option 欄位，長度不固定所致。reservation 欄目保留未使用，而 code bit 長度僅 6 bit，但每個 bit 都有其實際意義。

code bit 的第一個 bit 為 UGR，此值設為 1 時，將中斷目前的資料傳輸，優先傳輸此資料。第二個 bit 為 ACK，表示 acknowledge number 有效。第三個為 PSH，一般來說，TCP 的資料會先進入傳送緩衝區，累積到一定的量時才傳送，但當 PSH 設為 1 時，緩衝區內的資料就會立即送出。第四個為 RST，可要求立即中斷連線，不經過確認斷線步驟。第五、第六個分別為 SYN 和 FIN，分別在請求連線和請求斷線時設定為 1。

window 可告知目前的接收緩衝區還有多少的空間，對方傳送資料時便不會送出超過此數值的資料，避免緩衝區溢位導致資料流失需要重傳，浪費頻寬。checksum 用來檢查資料是否正確，也有使用虛擬表頭的功能，將其它的資訊一同放入運算。urgent pointer 在 UGR 設定為 1 時，標出緊急資料的位置。option 則是選用欄位，可表示接收方能夠支援的最大資料區段大小，如果選用欄位的資料長度不是 32 bit 的整數時，會在其後填入 0，對齊邊界。



▲TCP 的表頭欄位定義

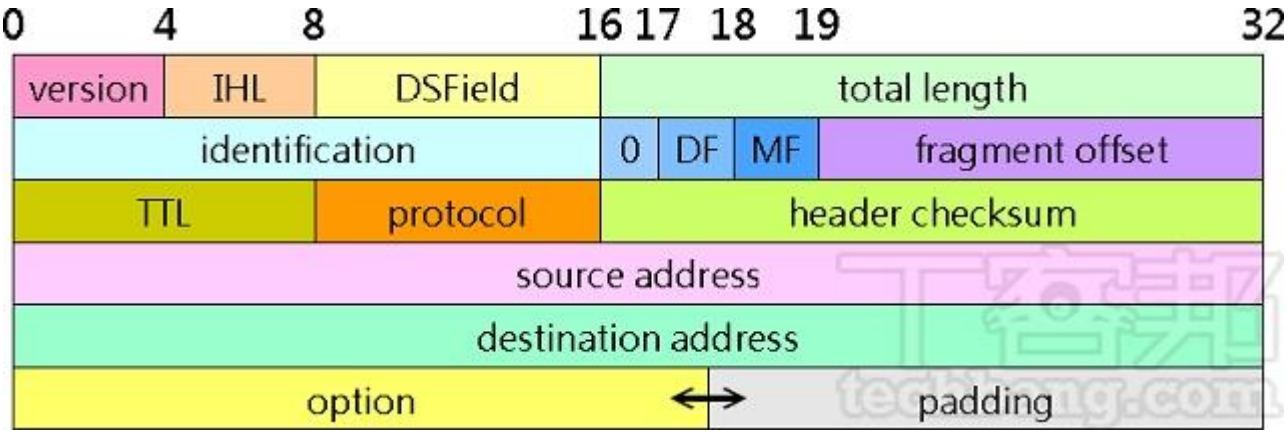
網路層（Internet Layer）

在網路層中，應用最廣的就是 IP 協定，然而 IP 協定存在著許多版本，目前世界上應用最廣的是第四版，第六版也已制定完成、實作，由於 IPv4 位址在可預見的未來中一定會配發完畢，各地也已開始佈建支援 IPv6 的硬體設施和軟體服務。

IPv4

IPv4 的表頭最小為 20 Byte，其中包含了 IP 位址、服務類型、存活時間等重要參數。首先由 4 bit 的 version 開頭，由於是 IPv4 版本，所以填入的值為 4。IHL 標出表頭大小，由於有 option 欄位，數值從 5～15 不定。DSField 欄位一開始為 type of service，但在 RFC 2474 中重新定義，前 6 bit 為 DSCP，標出這個資料的優先權，後 2 bit 目前保留未使用。total length 為 IP 封包長度（含標頭），identification 提供 1 個序列號，讓接收端能夠依照號碼，重組出原始的資料。flag 提供封包是否分段的訊息，而 fragment offset 就是用來指出目前這個資料分段在原始資料中的位置。

TTL 控制此封包能夠經過的網路節點數，每經過網路節點此值就減 1，數值為 0 時，網路裝置就捨棄此封包。protocol 指出需要由何種傳輸層協定處理，如 TCP 就設定為 6，UDP 設定為 17。header checksum 負責檢查 IP 表頭部分有沒有問題，若檢驗不合，此封包會被直接丟棄。source address 和 destination address 如同字面解釋，分別填入來源位址和目的地位址。最後的選項欄位並不一定要存在，但長度一定是 32 bit 的整數倍，不足的部分填入 0。



▲IPv4 的表頭欄位定義

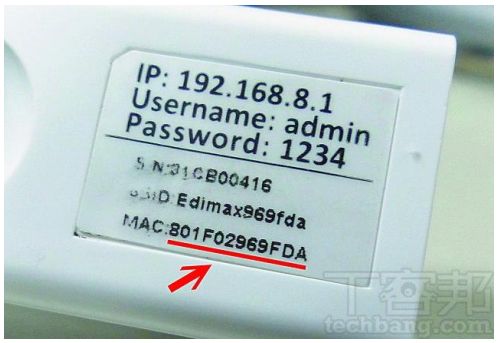
鏈結層（Link Layer）

終於到了最後 1 層，鏈結層終於要把資料，實際在媒體上傳輸、存取。下面這個例子就是目前比較常用的 Ethernet 範例：

Ethernet 的表頭相當簡單，只有目的地位址、來源位址、網路協定型態。目的地位址來源位址的長度都是 6 Byte，裡面放的是實際網路介面卡的 MAC 位址。type 指的是後面跟著的資料為何種網路協定，一般採用 TCP/IP 的協定，此欄就填入 2048，此外像是 Apple Talk 協定，就填入 32923。

最前端有個 preamble 的部分，主要是給接收端使用，讓接收端和發送端能夠同步。前 7 Byte 為 10101010，最後 1Byte 為 10101011。鏈結層也是唯一一層在資料尾端加入額外資訊，Ethernet 在尾端加入 4 Byte CRC 檢查碼。

▲Ethernet 的欄位定義

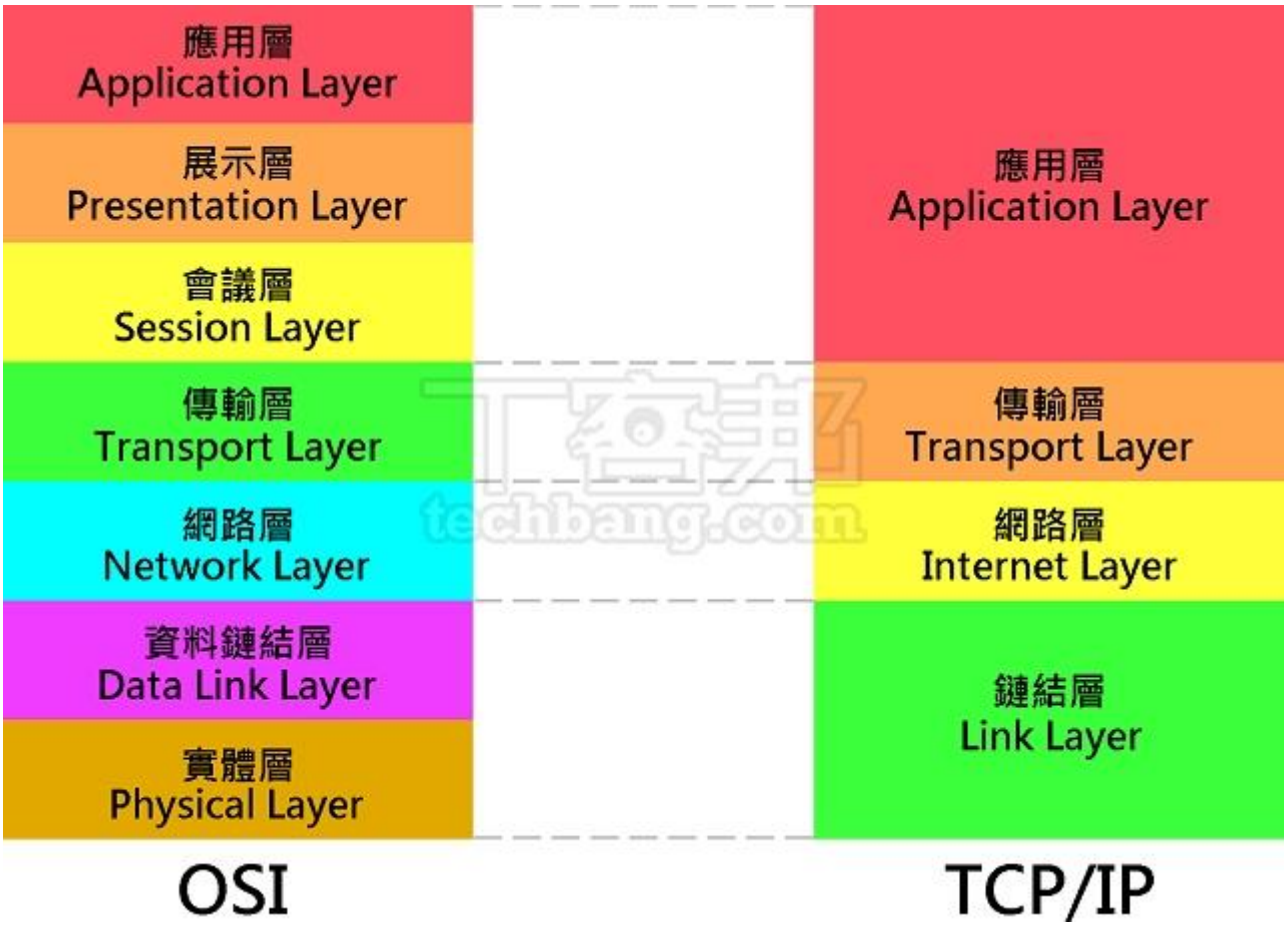


▲MAC（媒體存取控制，media access control）位址，每個網路介面都會有的獨立編號

OSI 7 層與網路 4 層對應

國際標準組織 ISO 提出 OSI（Open System Interconnection）參考模型，將網路分為 7 層，與網路模型比較起來，OSI 的定義較為細小，拆出更多層，但發展較晚，許多網路開發人員已習慣 TCP/IP 的分類方式。

不過 OSI 模型也不是一無是處，網路模型較常在實作方面被人提出，而 OSI 模型較常在觀念上提出。我們也利用 OSI 模型把網路設備分類，例如最簡單的集線器為 L1 設備、交換器為 L2 設備、路由器為 L3 設備，如果是更為強大的路由器，能夠過濾不同應用程式的封包，則為 L7 的路由器。



▲OSI 7 層與網路 4 層的對應關係

L7 應用層（Application Layer）

提供應用程式與其它主機上程式溝通的介面，也提供使用者應用程式存取網路上資源服務

L6 展示層（Presentation Layer）
定義與轉換資料格式，如 ASCII 和 EDCBIC 文字編碼的轉換，我們才能夠在不同的電腦上看到相同的資訊，加解密動作也由 OSI 指定在此層中處理。

L5 會議層（Session Layer）
負責交談如何開始、控制、結束。就像是有個會議主持人負責招集，制定開會流程，開會完後宣布散會。

L4 傳輸層（Transport Layer）
確保會議層與網路層之間傳遞的資料沒有遺失、重複，還有錯誤復原以及流量控管等功能。

L3 網路層（Network Layer）
邏輯定址、繞送、傳輸路徑選擇，維護 2 端點之間的連線。基本上就是 IP 協定所執行的工作。

L2 資料鏈結層（Data Link Layer）
決定何時資料可由實體傳輸媒介發送，也包含錯誤偵測機制。

L1 實體層（Physical Layer）
定義實際在媒介上的傳輸標準，大多直接拿取其它相關組織的規格直接使用，如 IEEE 電機電子工程師協會制定的一系列 802 標準。

到此為止，讀者應已了解為何網路模型需要分層，而各層之間又分別發揮什麼功能；內文也帶領大家走過 1 遍資料的包裝流程，理解資料經過網路傳輸需要哪些步驟，下一篇應該是介紹 IP 的分類以及繞送方式。