# TCP/IP特性

1．Connectionless Packet Delivery Service

它是其它網路服務的基礎﹐幾乎所有封包交換網路都提供這種服務。TCP/IP 是根據信息中所含的位址資料來進行資料傳送﹐它不能確保每個獨立路由的封包是可靠和依序的送達目的地。在每一個連線過程中﹐線路都不是被“獨佔”的﹐而是直接映對到硬體位址上﹐因此特別有效。更重要的是﹐此種封包交換方式的傳送﹐使得 TCP/IP 能適應各種不同的網路硬體。

2. Reliable Stream Transport Service

因為封包交換並不能確保每一個封包的可靠性﹐因此我們就需要通訊軟體來自動偵測和修復傳送過程中可能出現的錯誤﹐和處理不良的封包。這種服務就是用來確保電腦程式之間能夠建立連接和傳送大量資料。關鍵的技術是將資料流進行切割﹐然後編號傳送﹐然後透過接收方的確認(acknowledgement)來保證資料的完整性。

3. Network Technology Independent

在封包交換技術中﹐TCP/IP 是獨立與硬體之上的。TCP/IP 有自己的一套資料包規則和定義﹐能應用在不同的網路之上。

4. Universal Interconnection

只要電腦用 TCP/IP 連接網路﹐都將獲得一個獨一無二的識別位址。資料包在交換的過程中﹐是以位址資料為依據的﹐不管封包所經過的路由之選擇如何﹐資料都能被送達指定的位址。

5. End-to-End Acknowledgements

TCP/IP 的確認模式是以“端到端”進行的。這樣就無需理會封包交換過程中所參與的其它設備﹐發送端和接收端能相互確認才是我們關心得。

5. Application Protocol Standards

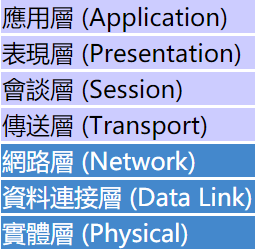
TCP/IP 除了提供基礎的傳送服務﹐它還提供許多一般應用標準﹐讓程式設計人員更有標準可依﹐而且也節省了許多不必要的重複開發。

## OSI模型

OSI 把數據通訊的各種功能分為七個層級﹐各司其職﹐但有相互依存﹑合作。但在功能上﹐它們又可以被劃分為兩組﹕

*網路群組*﹕由實體層﹑資料連接層﹑和網路層組成。

*使用者群組*﹕由傳送層﹑會談層﹑表現層﹑和應用層組成。



### 实体层(Physical Layer)

定义了应用在网路传输中的各种设备规格，以及如何将硬件所搭载的信号转换成电脑可以理解的电子信号。

### 数据链路层(Data Link Layer)

在這層指定了要採用的信息單元 (message unit﹐通常在 LAN上面的信息單元被稱為 frame﹐翻譯為“訊框”或“框包”)﹐還有它們的格式﹑以及如何通過網絡。每一個 frame 都會被賦予一個 MAC 位址碼和偵錯監測值(checksum)。

Ethernet 的 Data Link Layer Frame 看起來如下圖﹕



数据链路层通常管辖以下功能：

1. 网卡的MAC地址
2. 虚拟电路连接和逻辑连接的建立和结束。
3. 控制包的传送和错误检测方式。
4. 包的传送及接收顺序和传输方式。
5. 判定包的建立及重组分界。
6. 检测包的确认，以及在得不到回应或重新发送的情形下进行修复的程序。
7. 处理实体层的转换和管理
8. 对接收的包进行检错和确认
9. 检查包的MAC地址以确保数据能正确地抵达网络层。

**总的来说，这层的工作就是要保证无错误的物理上的数据传输。**

### 网络层(Network Layer)

這層就好比是一個中間人界乎於網絡功能和使用者功能之間。它會定義出封包在網路中移動的路由和其處理過程﹐這層還決定了網路是如何進行管理功能的﹐比如﹐發送狀態信息給接點和規範封包的流動等。

網路層的主要功能是讓封包(packet)在不同的網路之間成功地進行傳遞。它規定了網路的定址方式﹐及處理資料在不同網路之間的傳遞方式﹑處理子網路之間的傳遞﹑決定路由路徑﹑網路環境﹑資料處理順序﹑等等工作。

發送端電腦在封包被傳送出去之前﹐都會先為其建立 header ﹐作為在網路或子網間進行路由的依據。網路層在辨認和處理資料的時候﹐會忽略由高層協定制定的定義﹐只負責為數據在一個或多個網路間建立﹑維護﹑和終止連接。

網路層通常都有如下的這些功能﹕

如果封包不是屬於同一個網路的時候﹐會將之交由 router 處理

控制數據流量﹐當 router 的緩衝區飽和的時候﹐會通知數據傳輸設備使用其它路徑或暫停發送封包

當封包體積超過 router 的 MTU (Maximum Transmission Unit) 數值的時候﹐允許 router 對封包進行重組後再進行傳送。(一些所謂的增加 modem上網速度的軟體﹐就是因為可以對電腦的 MTU 數值進行最佳化﹐儘量減少 router 的封包重組﹐以達到最高的數據傳輸效率。)

負責 MAC 位址和網路位址(如 IP 位址﹑IPX 位址)之間的解釋和轉換

一個有趣的事情是﹐網路層還能將底層協定(網路功能)對上層協定(使用者功能)躲起來。這樣﹐在網路的使用者就可以使用不同種類的硬體了。這是非常好的事情﹐假如您用來建網的材料都不盡相同的話。

### 傳送層(Transport Layer)

在這層﹐將會設定如何控制節點之間的資料傳遞﹐還有錯誤檢測和修正的方法。

由於大多數網路﹐如 Ethernet 和 Token Ring等﹐由於物理上面的限制﹐一次所通過的數據流通常只有數千 byte 而已(IP 封包最大體積為 65536 bytes)﹐然而許多需要在網路中傳輸的資料都會超過這個數值。傳送層的主要功能是確保電腦資料正確的傳送到目的地。它的工作就是“打包”﹐也就是將電腦資料變成封包的形態﹐再賦以一定的檢測手段﹐將資料正確的傳到目的電腦﹐然後再將封包重組回資料。封包如果殘缺則進行重發﹐也能夠將重複的封包剔除。

傳送層可以等資料收集到足夠大的數量的時候才發送出去，並非應用程式每次產生一個數據就進行一次傳送，也就能減少了不必要的傳輸次數﹐以保證高效率的傳輸﹔反之，當應用程式產生大資料量數據時，則將之分拆成較小的封包再進行傳送。

傳送層的主要功能有﹕

1.接管由上層協定傳來的資料﹐並進行“分拆”和“打包”等工作。

2.利用點對點的方式進行資料傳送和回應的確認。

3.在得到接收端之資料緩衝區飽和信息之後﹐暫時停止資料發送。

4,能在單一位址上處理不同的程式協定(如ftp﹑http﹑telnet等)﹐並分別進行追蹤及轉換。

### 會談層(Session Layer)

這層定義了如何連接和掛斷連接﹐和在網路上面的數據如何交換。

這層所負責的是建立和管理電腦與電腦之間的溝通模式﹐也就是在資料真正進行傳送之前設定並建立好連線。這裡定義了連線的請求和結束﹑傳送和接收狀態的設定﹑等等動作。

當節點 A 要建立和節點 B 的連線的時候﹐ 會先發出“連線請求”訊息， 若對方接受連線，則回應“建立請求”訊息﹐然後雙方就建立好一個會談了﹔當會談結束的時候﹐也是先由節點 A 送出一個“結束請求”信息﹐等對方確認這個請求之後﹐那麼會談也就真正結束了。

會談層的功能主要有這些﹕

1.允許程式以電腦名稱註冊成為網路上唯一的位址。

2.在電腦之間建立﹑監測﹑和結束虛擬電路(Virtual Circuit)。

3．負責電腦之間的信息同步﹐監測資料溝通狀態﹐並對錯誤信息做出處理。

會談層為不同的程式以及與之對應的位於另一台電腦的程式﹐提供了可靠的溝通渠道。在某些溝通渠道中使用的是 half-duplex(半雙工﹐即單向) 的模式﹐有些則使用 full-duplex(全雙工﹐即雙向) 的模式﹐為程式之間協定好共同認可的模式﹐並進行同步和管理等責任﹐都是會談層所肩負起的。

### 表現層(Presentation Layer)

在這層﹐定義了數據的語法(syntax)﹑變更﹑和格式。當應用程式的語法和格式都不同的時候﹐這層還將定義了如何翻譯這些不同。

如果您想將您的 PC 通過網路連接 mainframe 電腦﹐那您就必須使用合適的 mainframe 文字串﹐通常會是 EBCDIS(External Binary Coded Decimal for Interchange Code)﹐然而您的 PC 使用的卻是 ASCII 碼(American Standard Code for Information Interchange)﹐這兩種編碼格式是截然不同的。表現層則是主要負責在不同機器之間進行編碼轉換。當應用程式產生資料要進行傳送的時候﹐表現層會將之換成網路的標準編碼格式再交由下層協定處理﹔然後當資料抵達目的地﹐表現層也會將網路的編碼換成對方應用程式所需的格式。

表現層的主要功能如下﹕

字元碼轉換﹐例如﹕ASCII 碼和 EDCDIC 碼之間的轉換。

資料形態轉換﹐例如﹕CR 碼和 CR-LF 碼﹑整數和浮點數之間的轉換。

對資料進行壓縮和加密﹐以提高速度和增加安全性。

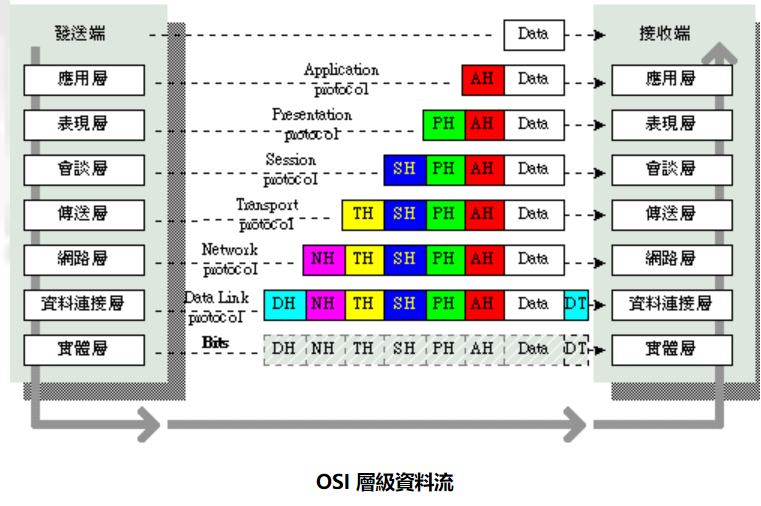
### 應用層(Application Layer)

剛才談到的表現層是負責將傳入來的資料種類轉換成 PC 的資料種類﹐不過﹐應用層則只轉換應用程式相關的檔案格式。表現層的轉換與應用層的轉換之間﹐最大的分別是﹕表現層是針對特定的主機的 CPU 類型﹐而應用層則針對特定應用程式。

在 TCP/IP 協定家族中，應用層所定義的還有一些應用程式功能﹐例如 FTP﹐HTTP﹐TELNET 等﹐主要負責客戶端和伺服端的連接。相信您對這些程式或多或少接觸過﹐比如您要從網上下載資料﹐可以通過 FTP 客戶程式﹐也可以通過流覽器將資料拿到。但在您的機器和在網路上面的伺服器﹐都必須使用相應的應用程式功能才可以做到﹐這在 TCP/IP 的網路環境裡面﹐通常都會是以不同的 port 號碼來讓不同的程式進行溝通。Port 號碼可以隨你喜歡來定﹐只要你知道號碼是多少就可以用相對的程式來進行交談了﹐不過﹐為了不至於太混亂﹐讓大家有規可尋﹐TCP/IP 也指定了一些常用的 port 來給常用的程式使用﹐例如﹐HTTP 使用 80﹑FTP使用 21﹑TELNET使用 23﹑諸如此類... 不過﹐需要知道的是﹕在 TCP/IP 協定中，進行 port 連接的功能卻應該是傳送層的事情。對應用程式而言，它是與 socket 作溝通，一個 socket 就是一個羅輯位址和 port 了。

### 封装

我們要知道，OSI 只是一個模形，其主要功能是定議了網路傳輸方面的參考模式。在其分層的結構中，可以從下圖看出 OSI 層級的資料流情況﹕



這裡我们发现OSI七层模型也体现了封装﹕每一個上層封包對下層來說﹐都是下層封包的數據﹔下層協定無需理會上層如何進行封裝﹐一律照單全收﹐然後加以自己的封裝﹐再把整個加封後的封包傳給更下一層。

### 分層協定的缺點

從前面的說明中﹐我們可以體會到分層協定設計的一個基本構思﹕讓協定設計者將複雜的問題分成數個子問題﹐然後再分別處理。然而﹐所付出的代價是﹕會令到分層的軟體變得非常低效。比方說﹕應用程式將一串位元組交到傳送層﹐經過封包切割之後﹐再傳到網路上。在此一過程中﹐傳送層會選擇最大的封包體積﹐以達到傳輸的最佳化﹐以配合訊框在實體網路中傳送的最佳化。倘若軟體的分層過嚴的話﹐傳送層就無法知道底層的路由狀況﹐也不知道與之直接相連的網路有哪些。更甚者﹐傳送層也不可能知道訊框的格式﹐也就不能知道如何界定封包體積的大小。所以﹐分層過嚴反而會成為傳送最佳化的障礙

因此﹐協定設計者在設計協定軟體時﹐通常會略為放寬限制﹐以允許路由選擇、MTU 之類的訊息廣播。同時在分配緩衝區的時候﹐會預先留下標頭空間讓低層填充資訊﹐或在高層協定傳送訊框時保留標頭資訊。如此設計便能在最佳化與分層結構中取得平衡。

## ARP协定

# 网络安全措施

防止他人使用公开URL伪造信息，可以采取一下安全措施：

1. 使用https协议传输
2. 增加token认证
3. 数据签名（DSA、RSA（非对称加密））