傳輸層

TCP：telnet

UDP：SNMP、RTP、TFTP (UDP的性質：程序對程序的通訊)

傳輸層主要功能：流量控制、連線控制、錯誤控制、~~邏輯定址~~

------------------------------------------------------------------------------------------------------------

TCP開始：三路交握

TCP結束：四路交握

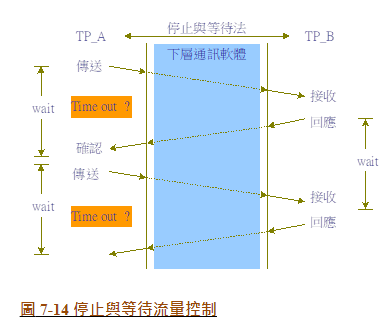
-------------------------------------------------------------------------------------------------------------

**流量控制方法**

[**http://www.tsnien.idv.tw/Internet\_WebBook/chap7/7-4%20TCP%20%E6%B5%81%E9%87%8F%E6%8E%A7%E5%88%B6.html**](http://www.tsnien.idv.tw/Internet_WebBook/chap7/7-4%20TCP%20%E6%B5%81%E9%87%8F%E6%8E%A7%E5%88%B6.html)

1. **停止並等待**

最簡單的流量控制法，此方法又稱為**『停止與等待』（Stop-and-Wait）**。其表示傳送端（TP\_A）將資料送出後必須等待接收端（TP\_B）回應，再決定是否繼續傳送下一筆資料。如果回應接收正常（Ack）便傳送下一筆資料；如果溢時（Time out）未收到回應或收到回應不確認訊息（Nak），則需重新傳送該筆資料。

****

**2.滑動視窗法**

TCP擁塞控制：乘法式減少、緩慢啟動、添加式增加

<https://zh.wikipedia.org/wiki/TCP%E6%8B%A5%E5%A1%9E%E6%8E%A7%E5%88%B6>

**擁塞窗口**

在TCP中，擁塞窗口（congestion window）是任何時刻內確定能被傳送出去的位元組數的控制因素之一，是阻止傳送方至接收方之間的鏈路變得擁塞的手段。他是由傳送方維護，通過估計鏈路的擁塞程度計算出來的，與由接收方維護的接收窗口大小並不衝突。

當一條連接建立後，每個主機獨立維護一個擁塞窗口並設定值為連接所能承受的MSS的最小倍數，之後的變化依靠**線增積減**機制來控制，這意味如果所有分段到達接收方和確認包準時地回到傳送方，擁塞窗口會增加一定數量。該窗口會保持指數增大，直到發生逾時或者超過一個稱為「慢啟動閾值（ssthresh）」的限值。如果傳送方到達這個閾值時，每收到一個新確認包，擁塞窗口只按照線性速度增加自身值的倒數。

當發生逾時的時候，慢啟動閾值降為逾時前擁塞窗口的一半大小、擁塞窗口會降為1個MSS，並且重新回到慢啟動階段。

系統管理員可以設定窗口最大限值，或者調整擁塞窗口的增加量，來對TCP調優。

在流量控制中，接收方通過TCP的「窗口」值（Window Size）來告知傳送方，由傳送方通過對擁塞窗口和接收窗口的大小比較，來確定任何時刻內需要傳輸的資料量。

-----------------------------------------------------------------------------------------------------------------

TCP錯誤偵測與改正：檢查和(1的補數)、回應、計時 ~~啟動緩慢~~

checksum

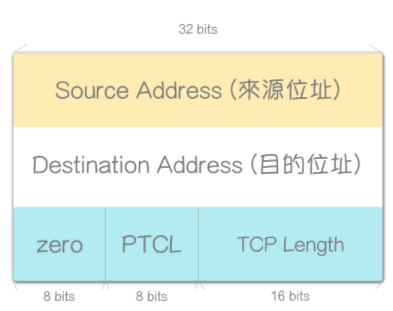
https://notfalse.net/21/tcp-checksum

TCP 本身沒有關於位址的訊息，  
這可能造成 區段 (segment) 被 錯誤的路由 (misrouted)。

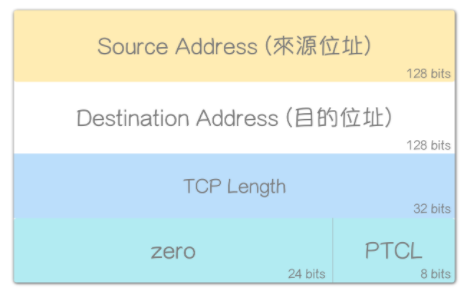
需提供足夠的資訊，讓 checksum 可檢測路由錯誤，  
這些資訊就是 — 虛擬表頭 (Pseudo-Header) 啦!  
總共有 12 個位元組 (octet) 也就是 96 (12\*8) 位元。

是因為這僅供 checksum 計算使用，並不實際傳送。  
(位址欄位來自 IP 表頭)

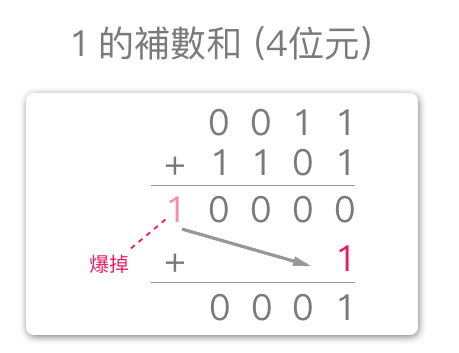
**虛擬表頭(IPv4)---12個位元組**



**虛擬表頭(IPv6)---40個位元組**

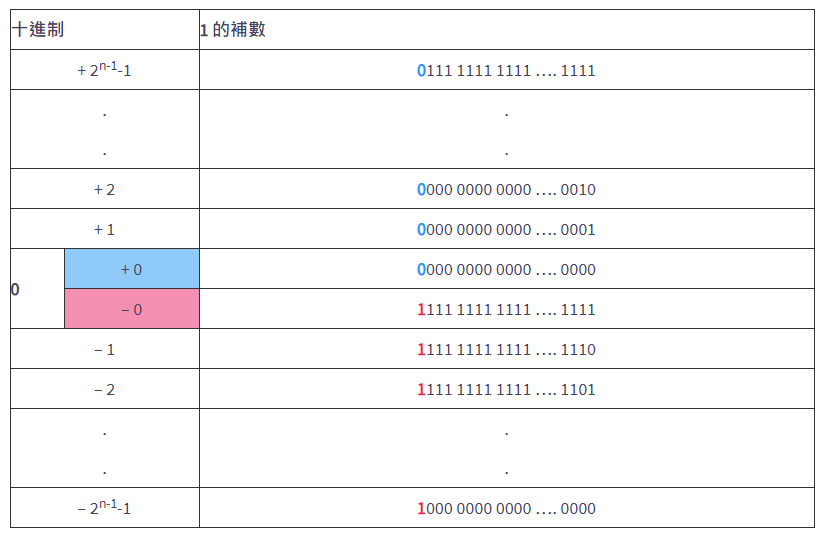
~~~~

\*\*補數

🡸3+(-2)=1

如果有『爆掉』的位數，要再將它加回，稱為 — — **端回進位 (end around carry)，**也就是將任何溢出的最高有效位元，加回最低有效位元。這是 1 的補數的重要性質!

<https://notfalse.net/20/signed-number-representations>



上圖為1的補數

而 2 的補數之『負數』，其實就是 1 的補數 + 1，

浮點數表示法標準 — — 符號大小、偏移表示法，  
網路通訊協定的[檢驗和](https://notfalse.net/21/tcp-checksum) — — 1 的補數，  
程式語言的整數、加法器的實作 — — 2 的補數，

TCP連線

\*TCP將資料視作Bytes Stream

STEP(一)：建立連線

STEP(二)：傳送資料

題目

P221(26)

**TCP端有執行流量控制**時，如果**接收端的一方送回回應號碼**(Acknowlefgementnumber)為1234時候，其意為何？

接收端**下一個想收到**的**位元組**是第1234個位元組