

## HTTP 協議

HTTP (HyperText Transfer Protocol) 是一種在 OSI 模型中的第 7 層應用層作用的協議，為了處理伺服器主機和使用者瀏覽器之間溝通的協定。大多是通過 PORT 80 來傳輸 HTTP 的封包，基本上 HTTP 是一種基於客戶端—伺服器 (Client/Server) 的應用，或者是說是一種請求—回復的運作方式，一定是要 Client 發送請求 (Request) 後，Server 才能傳回復 (Response) 給接收端，發送請求可以透過網址，就是連到該網站、超連結來下達請求在 Web Server 的資源，比如 Html 等，處理完請求後，使用 MIME 的格式回覆給 Client 端。

目前有 3 個版本，分別為 HTTP/1.0、HTTP/1.1 和 HTTP/2.0。1 和 1.1 有幾點差異，最主要的改善是支援持久連線 (Connection : Keep-Alive)，不需要重複建立 TCP 連線，當客戶端與伺服器端下達請求後，並不會在回覆後馬上斷開連結，這樣就節省每次建立連線都要確認的那 3 個 RTT，這是 HTTP/1.0 最主要的缺點，連線無法復用，導致每次都要經過稱為 3 個握手的階段，在系統檔案較大或者延遲比較高時，會有明顯的影響，此外 HTTP/1.1 還有流水線 (pipelining) 的機制讓客戶端可以一次發送多個請求，這些請求可以是不按照順序，但是伺服器端必須按照順序回復 Client 端，在下載方面會有明顯的效能提升，但主要的處理情境是發送多個請求時，如果請求一卡住了，或是因為甚麼原因沒有到達伺服器，那所有其他請求都必須等待，後續的請求都會被卡住，這問題被稱為隊頭阻塞 (head of line blocking)。

在 HTTP/2.0 之前，還有一個前身 SPDY 試圖解決 HTTP 1.0 系列的問題，其實 pipelining 存在許多問題，因為不是所有請求類型都可以使用，此外隊頭阻塞也沒有完全解決，因為伺服器回復採用 FIFO 先發先回的演算法，需要照順序回復，所以 SPDY 提出了請求優先順序 (request prioritization) 允許給每個請求設定優先順序，重要的請求就會被優先得到回應。SPDY 還有做 Header 的壓縮，因為網頁包含 Cookie 後，檔頭容易變得太大，SPDY 相較於 HTTP 1.0 對於檔頭的壓縮率在 60% 以上。

HTTP/2.0 處理的問題和 SPDY 類似，但是更加考慮了普遍和適應性問題，所以設計架構更加嚴謹和慎重。在處理隊頭阻塞方面，使用了連線共享 (MultiPlexing)，客戶端和伺服器之間只會有一個連線，就算是不同的資料都能夠混和在一起通過和傳送，並透過給定的不同 stream id 辨別不同的資料，並進行組裝。