

HTTP(HyperText Transfer Protocol)超文字傳輸協議，是網際網路上應用最為廣泛的一種網路協議，最初的目的是為了提供一種釋出和接收 HTML 頁面的方法，是用於從 WWW 伺服器傳輸超文字到本地瀏覽器的傳輸協議，所有的 WWW 檔案都必須遵守這個標準。HTTP 連線使用的是“請求—響應”的方式，不僅在請求時需要先建立連線，而且需要客戶端向伺服器發出請求後，伺服器端才能回覆資料。

HTTP 1.0 規定瀏覽器與伺服器只保持短暫的連線，瀏覽器的每次請求都需要與伺服器建立一個 TCP 連線，伺服器完成請求處理後立即斷開 TCP 連線，伺服器不追蹤客戶也不記錄過去的請求，這也造成了一些效能上的缺陷。訪問一個包含有許多影像的網頁時，包含了多次請求和響應，每次都需要建立一個單獨連線，客戶端和伺服器端每次建立和關閉連線是一個相對比較費時的過程，嚴重影響效能。因此，Http1.0 最明顯的缺點就是連線無法複用，會導致每次請求都經歷三次握手和慢啟動。三次握手在高延遲的場景下影響較明顯，慢啟動則對檔案類大請求影響較大。另一個缺點是 head of line blocking，假設有多個請求同時發出，在第一個請求沒有收到回覆之前，後續發出的請求只能排隊，等第一個響應回來之後才能逐個發出，導致頻寬無法被充分利用，以及後續請求被阻塞。

HTTP/1.1 中，持久連線（Connection : Keep-Alive）被預設採用，在一個 TCP 連線上可以傳送多個 HTTP 請求和響應，減少了建立和關閉連線的消耗和延遲，通過增加更多的請求頭和響應頭來改進和擴充 HTTP 1.0 的功能。一段時間內的連線複用對 PC 端瀏覽器的體驗幫助很大，因為大部分的請求在集中在一小段時間以內，但對移動 app 來說，成效不大。

HTTP/2.0 在 HTTP 1.x 的基礎上，大幅度的提高了 web 效能，減少了網路延遲。HTTP/2 的多路複用(Multiplexing) 允許同時通過單一的 HTTP/2 連線發起多重的請求-響應訊息，把 HTTP 協議通訊的基本單位縮小為一個一個的幀，這些幀對應著邏輯流中的訊息，並行在同一個 TCP 連線上雙向交換訊息。此外，HTTP/2 多了使用了專門為首部壓縮（Header Compression）而設計的 HPACK 演算法。伺服器推送（Server Push）是一種在客戶端請求之前傳送資料的機制，在 HTTP/2 中，伺服器可以對客戶端的一個請求傳送多個響應，讓在遵循同源的情況下，不同頁面之間可以共享快取資源。

參考資料：

<https://codertw.com/%E7%A8%8B%E5%BC%8F%E8%AA%9E%E8%A8%80/496897/>

<https://iter01.com/548698.html>