

Http（超文字傳輸協議，HyperText Transfer Protocol）是一個簡單的請求-響應協議，它通常基於 TCP 進行連線。所有的 WWW 檔案都必須遵守這個標準。設計 HTTP 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。

HTTP 1.0 是第一個在通訊中指定版本號的 HTTP 協議版本，至今仍被廣泛採用。

HTTP 1.1 是當前版本，持久連線(Connection : Keep-Alive)被預設採用，並能很好地配合代理伺服器工作，還支援以管道方式同時傳送多個請求，以便降低線路負載，提高傳輸速度。

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。

HTTP1.0 和 1.1 在之後很長的一段時間內會一直並存，這是由於網路基礎設施更新緩慢所決定的。

HTTP 協議老的標準是 HTTP/1.0，為了提高系統的效率，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，伺服器不跟蹤每個客戶也不記錄過去的請求。缺點是由於每次客戶端向伺服器請求時都要重新建立 TCP 連線，當網頁中包含有外部的靜態資源的載入，像是圖片，CSS 和 JS 檔案等，客戶端還要根據這些資源的 URL 重新與伺服器進行 TCP 的連線，即使影像檔案都很小，但是客戶端和伺服器端每次建立和關閉連線卻是一個相對比較費時的過程，並且會嚴重影響效能。因此，Http1.0 最明顯的缺點之一就是連線無法複用，另外一個問題則是 head of line blocking: 會導致頻寬無法被充分利用，以及後續請求被阻塞。假設有 5 個請求同時發出,對於 HTTP1.0 的實現, 在第一個請求沒有收到回覆之前，後續從應用層發出的請求只能排隊。網路通暢的時候影響不大，一旦請求 1 的 request 因為什麼原因沒有抵達伺服器，或者伺服器響應因為網路阻塞沒有及時返回，影響的就是所有後續請求，問題就變得比較嚴重。

Http 1.1 和 Http 2.0 主要就是針對這兩個問題提出瞭解決方案和辦法：

解決連線無法複用問題，在 Http 1.1 和 Http 2.0 中：Connection 的預設值就是 Keep-Alive，如果要關閉連線複用需要顯式的設定 Connection:Close。如果 client 使用 http1.1 / 2.0 協議，但又不希望用長連結，則需要在 header 中指明 connection 的值為 close；如果 server 方也不想支援長連結，則在 response 中也需要明確說明 connection 的值為 close。不論 request 還是 response 的 header 中包含了值為 close 的 connection，都表明當前正在使用的 tcp 連結在當天請求處理完畢後會被斷掉。

解決 head of line blocking 問題：HTTP1.1 的解決方案 - http pipelining：請求 2，3 不用等請求 1 的 response 返回之後才發出，而是幾乎在同一時間把 request 發向了伺服器。2，3 及所有後續共用該連線的請求節約了等待的時間，極大的降低了整體延遲，缺點只能適用於 http 1.1。

Http 2.0 的解決方案:多路複用、請求優先順序、伺服器推送，多路複用允許同時通過單一的 HTTP 2 連線發起多重的請求-響應訊息。伺服器推送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。Server Push 讓 HTTP 1.x 時代使用內嵌資源的優化手段變得沒有意義；如果一個請求是由你的主頁發起的，伺服器很可能會響應主頁內容、logo 以及樣式表，因為它知道客戶端會用到這些東西。伺服器推送還有一個很大的優勢：可以快取，也讓在遵循同源的情況下，不同頁面之間可以共享快取資源成為可能。