

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。但是，這也造成了一些效能上的缺陷，例如，一個包含有許多影象的網頁檔案中並沒有包含真正的影象資料內容，而只是指明瞭這些影象的 URL 地址，當 WEB 瀏覽器訪問這個網頁檔案時，瀏覽器首先要發出針對該網頁檔案的請求，當瀏覽器解析 WEB 伺服器返回的該網頁文件中的 HTML 內容時，發現其中的影象標籤後，瀏覽器將根據標籤中的 src 屬性所指定的 URL 地址再次向伺服器發出下載影象資料的請求。顯然，訪問一個包含有許多影象的網頁檔案的整個過程包含了多次請求和響應，每次請求和響應都需要建立一個單獨的連線，每次連線只是傳輸一個文件和影象，上一次和下一次請求完全分離。即使影象檔案都很小，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，並且會嚴重影響客戶機和伺服器的效能。當一個網頁檔案中包含 JavaScript 檔案，CSS 檔案等內容時，也會出現類似上述的情況。

同時，頻寬和延遲也是影響一個網路請求的重要因素。在網路基礎建設已經使得頻寬得到極大的提升的當下，大部分時候都是延遲在於響應速度。基於此會發現，http1.0 被抱怨最多的就是連線無法複用，和 head of line blocking 這兩個問題。理解這兩個問題有一個十分重要的前提：客戶端是依據域名來向伺服器建立連線，一般 PC 端瀏覽器會針對單個域名的 server 同時建立 6~8 個連線，手機端的連線數則一般控制在 4~6 個。顯然連線數並不是越多越好，資源開銷和整體延遲都會隨之增大。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。head of line blocking 會導致頻寬無法被充分利用，以及後續健康請求被阻塞。

HTTP 1.1 在繼承了 HTTP 1.0 優點的基礎上，也克服了 HTTP 1.0 的效能問題。HTTP 1.1 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。如，HTTP 1.0 不支援 Host 請求頭欄位，WEB 瀏覽器無法使用主機頭名來明確表示要訪問伺服器上的哪個 WEB 站點，這樣就無法使用 WEB 伺服器在同一個 IP 地址和埠號上配置多個虛擬 WEB 站點。在 HTTP 1.1 中增加 Host 請求頭欄位後，WEB 瀏覽器可以使用主機頭名來明確表示要訪問伺服器上的哪個 WEB 站點，這才實現了在一臺 WEB 伺服器上可以在同一個 IP 地址和埠號上使用不同的主機名來建立多個虛擬 WEB 站點。HTTP 1.1 的持續連線，也需要增加新的請求頭來幫助實現，例如，Connection 請求頭的值為 Keep-Alive 時，客戶端通知伺服器返回本次請求結果後保持連線；Connection 請求頭的值為 close 時，客戶端通知伺服器返回本次請求結果後關閉連線。HTTP 1.1 還提供了與身份認證、狀態管理和 Cache 快取等機制相關的請求頭和響應頭。HTTP/1.0 不支援檔案斷點續傳，`RANGE:bytes` 是 HTTP/1.1 新增內容，HTTP/1.0 每次傳送檔案都是從檔案頭開始，即 0 位元組處開始。`RANGE:bytes=XXXX` 表示要求伺服器從檔案 XXXX 位元組處開始傳送，這就是我們平時所說的斷點續傳！

HTTP/2 的特性，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。在 HTTP/1.1 協議中瀏覽器客戶端在同一時間，針對同一域名下的請求有一定數

量限制。超過限制數目的請求會被阻塞。這也是為何一些站點會有多個靜態資源 CDN 域名的原因之一，拿 Twitter 為例，目的就是變相的解決瀏覽器針對同一域名的請求限制阻塞問題。而 HTTP/2 的多路複用(Multiplexing) 則允許同時通過單一的 HTTP/2 連線發起多重的請求-響應訊息。因此 HTTP/2 可以很容易的去實現多流並行而不用依賴建立多個 TCP 連線，HTTP/2 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。並行地在同一個 TCP 連線上雙向交換訊息。HTTP/2 在 應用層(HTTP/2)和傳輸層(TCP or UDP)之間增加一個二進位制分幀層。在不改動 HTTP/1.x 的語義、方法、狀態碼、URI 以及首部欄位的情況下，解決了 HTTP1.1 的效能限制，改進傳輸效能，實現低延遲和高吞吐量。在二進位制分幀層中， HTTP/2 會將所有傳輸的資訊分割為更小的訊息和幀 (frame)，並對它們採用二進位制格式的編碼，其中 HTTP1.x 的首部資訊會被封裝到 HEADER frame，而相應的 Request Body 則封裝到 DATA frame 裡面。

HTTP/2 通訊都在一個連線上完成，這個連線可以承載任意數量的雙向資料流。在過去， HTTP 效能優化的關鍵並不在於高頻寬，而是低延遲。TCP 連線會隨著時間進行自我調諧，起初會限制連線的最大速度，如果資料成功傳輸，會隨著時間的推移提高傳輸的速度。這種調諧則被稱為 TCP 慢啟動。由於這種原因，讓原本就具有突發性和短時性的 HTTP 連線變的十分低效。HTTP/2 通過讓所有資料流共用同一個連線，可以更有效地使用 TCP 連線，讓高頻寬也能真正的服務於 HTTP 的效能提升。