

HTTP/1.0，1996 年發布，但其實在這之前，各家瀏覽器與服務器都是各自做自己的，並沒有任何共識。這時發布的 1.0 就有點像是參考各家的實作，來重新定義 HTTP 如何操作。HTTP/1.0 規定瀏覽器與伺服器保持較短時間的連結，瀏覽器每次請求都和伺服器經過三次握手和慢啟動（基本思想是當 TCP 開始傳輸數據或發現數據丟失並開始重發時，首先慢慢的對網路實際容量進行試探，避免由於發送了過量的數據而導致阻塞）建立一個 TCP 連結，伺服器完成請求處理後立即斷開 TCP 連結，而且不跟蹤每個瀏覽器的歷史請求。

HTTP/1.1，1997 年發布第一版 RFC 2068，1999 年發布第二版 RFC 2616 取代前一版。它在定義上保留了许多擴充空間（如 Header），讓建構在 HTTP/1.1 上的應用程式或其他基於 HTTP 的協定，更有發揮的空間。一個 TCP 連結可以傳送多個 HTTP 請求和相應，減少了 TCP 建立連結和關閉連結的消耗。另外 HTTP/1.1 允許客戶端不用等待上一次請求結果返回，就可以發出下一次請求，但伺服器端必須按照接收到客戶端請求的先後順序依次回送響應結果，以保證客戶端能夠區分出每次請求的響應內容。不使用管道的 http 請求，在使用持久連結時，必須嚴格滿足先進先出的隊列順序（FIFO），即發送請求，等待響應完成，再發送客戶端隊列中的下一個請求。管道可以讓我們把 FIFO 隊列從客戶端（請求隊列）遷移到伺服器（響應隊列），即客戶端可以並行，服務端串行。客戶端可以不用等待前一個請求返回，發送請求，但伺服器端必須順序的返回客戶端的請求響應結果。

HTTP/2，Google 2012 年發表了新的傳輸協定 SPDY，雖然當時不是標準，但該開發團隊全程參與了 2015 年發布的 HTTP/2。此協定與 HTTP/1.1 高度相容，並為了減少網路延遲，定義了很多方法來提升網頁載入速度。多路復用允許同時通過單一的 HTTP 2.0 連接發起多重的請求-響應消息，即所有 HTTP 2.0 連接都是持久化的，而且客戶端與伺服器之間也只需要一個連接即可，所有數據流共用同一個連接，減少了因 http 連結多而引起的網路擁塞（在 HTTP1.1 協議中，同一時間，瀏覽器會針對同一域名下的請求有一定數量限制），解決了慢啟動針對突發性和短時性的 http 連結低效的問題。將通信的基本單位縮小為幀，即應用層(HTTP)和傳輸層(TCP or UDP)之間增加一個二進位分幀層，因此在多向請求和響應時，客戶端和伺服器可以把 HTTP 消息分解為互不依賴的幀，然後亂序發送，最後再在另一端把它們重新組合起來，解決了 http 1.*的對手阻塞問題。

參考資料:

<https://kknews.cc/zh-tw/code/apbvzbv.html>

<https://ithelp.ithome.com.tw/articles/10217426>