

HTTP/1.0

這是第一個在通訊中指定版本號的 HTTP 協定版本。

HTTP/1.1

預設採用持續連接 (Connection: keep-alive)，能很好地配合代理伺服器工作。

還支援以管道方式在同時傳送多個請求，以便降低線路負載，提高傳輸速度。

HTTP/1.1 相較於 HTTP/1.0 協定的區別主要體現在：

- 快取處理

- 頻寬最佳化及網路連接的使用

- 錯誤通知的管理

- 訊息在網路中的傳送

- 網際網路位址的維護

- 安全性及完整性

HTTP/2

目前版本，於 2015 年 5 月作為網際網路標準正式發布。

HTTP/2 相比 HTTP/1.1 的修改並不會破壞現有程式的工作，但是新的程式可以藉由新特性得到更好的速度。

HTTP/2 保留了 HTTP/1.1 的大部分語意，例如請求方法、狀態碼乃至 URI 和絕大多數 HTTP 頭欄位一致。而 HTTP/2 採用了新的方法來編碼、傳輸客戶端和伺服器之間的資料

幾年前 Google 曾發佈了令大家眼睛為之一亮的 SPDY 協議，標榜能加速 Web 的載入速度，並讓網站上的多重檔案可以併發下載。SPDY 因為具備優異的網頁傳輸及處理效能，因此後來從眾多 HTTP/2 提案版本中脫穎而出，最終成為 HTTP/2 標準草案的原型。其競爭對手包含微軟的 HTTP Speed+Mobility 等等協議。

HTTP/2 的許多關鍵功能也都來自於 SPDY，最大的改變就是加入一個多工（Multiplexing）的功能，可以允許瀏覽器在同時間內對多個伺服器發送請求，並採用更高效率的標頭壓縮技術，整體而言，HTTP/2 讓用戶端能以較少的連接數從伺服器端取得資料，大幅增加網路傳輸速度。

HTTP/2 的一項新特色是加上了伺服器推送功能，伺服器可以主動推送內容到瀏覽器上。這增加了許多特別的新應用，例如可以在瀏覽器尚未發出請求前，預先推送 CSS 或頁面 Layout 到瀏覽器上，增加之後的頁面載入速度。

HEAD 在傳輸的時候，有蠻多重複或冗餘的資訊，這些資訊可藉由 Haffman 演算法壓縮 HEAD 來增加傳輸速度

藉由控制下載流程的優先級，可以讓 HTTP/2 的傳輸過程中，將最重要的內容優先下載，避免大量資訊堵在一起。

相較於 SPDY 強制要採用 https 協議，HTTP/2 並未強制傳輸要加密，不過在 HTTP/2 協議下，將更容易實現 TLS 傳輸加密。

基本上，主流瀏覽器大多已經支援。不過您可以在自己的伺服器上加裝 HTTP/2 模組來支援一些新的 HTTP/2 特性，例如 NodeJS 就有 `node-http2` 模組可以安裝

對大多數的使用者來說，HTTP/2 的影響力不是那麼的大，因為瀏覽器很快就支援了，也不會有任何瀏覽網站上的影響。但對於工程師而言，就有很多眉眉角角要注意。例如前端過去常常依靠 CSS sprites 與 Assets 合併來優化頁面載入速度，也許未來的影響就沒有那麼大。後端則要考慮是否壓縮 HEAD 或採用 TLS 等等。整體而言，對網路世界依舊是好事，畢竟從 1.1 版推出以來已經 16 年沒有更新了，是時候邁向新時代的網路協議了。