

HTTP ( 超文字傳輸協議，HyperText Transfer Protocol)是網際網路上應用最為廣泛的一種網路協議。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，特別是在代理伺服器中。

HTTP/1.1 是當前版本，持久連線被預設採用，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/1.0

1.0 的 HTTP 版本，是一種無狀態，無連線的應用層協議。 HTTP1.0 規定瀏覽器和伺服器保持短暫的連結。

瀏覽器每次請求都需要與伺服器建立一個 TCP 連線，伺服器處理完成以後立即斷開 TCP 連線(無連線)，伺服器不跟蹤也每個客戶單，也不記錄過去的請求(無狀態)。

這種無狀態性可以藉助 cookie/session 機制來做身份認證和狀態記錄。

### 問題:

1. 無法複用連線
2. 每次傳送請求，都需要進行一次 TCP 連線，而 TCP 的連線釋放過程又是比較費事的。這種無連線的特性會使得網路的利用率變低。

3. 由於 HTTP1.0 規定下一個請求必須在前一個請求響應到達之前才能傳送，假設前一個請求響應一直不到達，那麼下一個請求就不傳送，後面的請求就阻塞了。

## HTTP/1.1

HTTP1.1 繼承了 HTTP1.0 的簡單，克服了 HTTP1.0 效能上的問題。

1. 增加 **Connection** 欄位，通過設定 **Keep-Alive** 保持 HTTP 連線不斷卡。避免每次客戶端與伺服器請求都要重複建立釋放建立 TCP 連線。提高了網路的利用率。
2. 如果客戶端想關閉 HTTP 連線，可以在請求頭中攜帶 **Connection:false** 來告知伺服器關閉請求
3. 基於 HTTP1.1 的長連線，使得請求管線化成為可能。管線化使得請求能夠“並行”傳輸。
4. HTTP1.1 加入了快取處理(強快取和協商快取)，新的欄位如 **cache-control**，支援斷點傳輸，以及增加了 **Host** 欄位(使得一個伺服器能夠用來建立多個 Web 站點)

客戶端同時發了兩個請求分別獲取 **html** 和 **css**，假如說伺服器的 **css** 資源先準備就緒，伺服器也會先發送 **html**，再發送 **css**。換句話來說，只有等到 **html** 響應的資源完全傳輸完畢後，**css** 響應的資源才開始傳輸，不允許同時存在兩個並行的響應。HTTP1.1 還是無法解決隊頭阻塞(head of line blocking)的問題。同時“管道化”技術存在各種各樣的問題，所以很多瀏覽器要麼根本不支援它，要麼直接預設關閉，並且開啟的條件很苛刻.....而且好像實際也沒有什麼用處。

## HTTP/2.0

HTTP2.0 通過在應用層和傳輸層之間增加一個二進位制分層幀，突破了 HTTP1.1 的效能限制，改進傳輸效能。

1. 建立連線上的雙向位元組流。
2. 與邏輯訊息對應的完整的一系列資料幀。
3. 通訊的最小單位，每個幀包含頭部，至少也會標識出當前所屬的流(stream\_id)

所有 HTTP2.0 通訊都在一個 TCP 連結上完成，這個連結可以承載任意流量的雙向資料流。每個資料流以訊息的形式傳送，而訊息由一或多個幀組成。這些幀

可以亂序傳送，然後再根據每個幀頭部的流識別符號(**Stream\_id**)重新封裝，優先順序高的資料流會被伺服器優先處理和返回客戶端，資料流還可以依賴其他的子資料流。