

# Compare with HTTP1.0/1.1/2.0

我認為在比較 http1.0、1.1 和 2.0 之間的差異前，必須先了解 http 到底是甚麼。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 連線使用的是“請求—響應”的方式，不僅在請求時需要先建立連線，而且需要客戶端向伺服器發出請求後，伺服器端才能回覆資料。

## HTTP1.0

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

許多影像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像，上一次和下一次請求完全分離。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。

## **HTTP1.1**

HTTP1.1 的出現就是為了解決 1.0 每次連線後就馬上斷掉的問題。它支援持久連線 ( HTTP/1.1 的預設模式使用帶流水線的持久連線 )，在一個 TCP 連線上可以傳送多個 HTTP 請求和響應，減少了建立和關閉連線的消耗和延遲。一個包含有許多影像的網頁檔案的多個請求和應答可以在一個連線中傳輸，但每個單獨的網頁檔案的請求和應答仍然需要使用各自的連線。HTTP 1.1 還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，這樣也顯著地減少了整個下載過程所需要的時間。

## **HTTP2.0**

2.0 相較於之前的協議就提升了許多的效能。像是多路複用、二進位制分幀和首部壓縮等。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。二進位制分幀在不改動 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。而首部壓縮則是因為 HTTP/1.1 並不支援 HTTP 首部壓縮，為此 SPDY 和 HTTP/2 應運而生，SPDY 使用的是通用的 DEFLATE 演算法，而 HTTP/2 則使用了專門為首部壓縮而設計的 HPACK 演算法。