

HTTP 是基於 TCP 連線的傳輸協定，原因是一個網頁必須傳送大量的資料，TCP 協定提供傳輸控制，按順序組織資料和錯誤糾正。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（三次握手），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 1.0 是一種無狀態、無連線的應用層協議。HTTP1.0 規定瀏覽器與伺服器只能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（TCP 開始傳輸資料或發現數據丟失並開始重發時，先慢慢試探網路實際容量，避免因為傳送了過量的資料導致阻塞）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。HTTP1.0 存在無法複用連線的問題，每次傳送請求，都需要進行一次 TCP 連線，而 TCP 的連線釋放過程較為繁複。這種無連線的特性會使得網路的利用率變低。再來是線頭阻塞（Head of Line (HOL) Blocking），請求佇列的第一個請求因為伺服器忙碌中，導致後面的請求被阻塞。但 HTTP1.0 至今仍被廣泛使用。

- HTTP1.1

HTTP1.1 為了克服了 HTTP1.0 效能上的問題，支援持久連線，在一個 TCP 連線上可以傳送多個 HTTP 請求和響應，減少建立和關閉連線的消耗和延遲。

HTTP1.1 中，客戶端和伺服器都支援長連接，增加了緩存處理、Host 字段以及請求頭（header）和響應頭來擴充功能，Connection 的預設值為 Keep-Alive。此外，HTTP1.1 支援請求管線化（pipelining），使請求能夠並行傳輸，不用等待響應返回就可以發出新的請求，可以大幅降低整體的延遲時間。但是 pipelining 只適用於 HTTP1.1，而且請求併發是基於多個 TCP 連接，比較耗時，頭部訊息增加了每個請求的文本負荷，因此還是無法解決 HTTP1.0 的線頭阻塞問題，需要新的 HTTP 版本來改善。

- HTTP2.0

HTTP2.0 增加了二進位制分幀 (HTTP Frames)、多路複用、頭部壓縮以及服務端推送 (server push)。幀 (frame) 為 HTTP2.0 裡通訊的最小單位，每個幀有自己的格式，不同型別的幀負責傳輸不同的訊息。訊息 (message) 包含一個或多個幀。流 (stream) 建立連結後的一個雙向位元組流，用來傳輸訊息，每次傳輸的是一個或多個幀。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，訊息裡的幀可以亂序傳送，然後再根據每個幀頭部的流識別符號重新封裝。此功能在 HTTP2.0 實現了真正的並行傳輸，能在一個 TCP 上進行任意數量的 HTTP 請求。