

HTTP 是人們最為常見的網路協議，HTTP 最初是提供接收和釋出 html 頁面的方法，是用於從 WWW 服務器傳輸超文本到本地瀏覽器的傳輸協議。默認使用 80 端口，HTTP 客戶端發起一個請求，建立一個到服務器指定端口（默認是 80 端口）的 TCP 連接。HTTP 協議和 TCP 協議是不衝突的，HTTP 定義在七層協議中的應用層，TCP 解決的是傳輸層的邏輯。HTTP 使用 TCP 而不是 UDP 的原因在於（打開）一個網頁必須傳送很多數據，而 TCP 協議提供傳輸控制，按順序組織數據，和錯誤糾正。HTTP 協議的瓶頸及其優化技巧都是基於 TCP 協議本身的特性。如 TCP 建立連接時三次握手有 1.5 個 RTT（round-trip time）的延遲，為了避免每次請求的都經歷握手帶來的延遲，應用層會選擇不同策略的 http 長鏈接方案。又如 TCP 在建立連接的初期有慢啟動（slow start）的特性，所以連接的重用總是比新建連接性能要好。HTTP 連接使用的是“請求—響應”的方式，不僅在請求時需要先建立連接，而且需要客戶端向服務器發出請求后，服務器端才能回復數據。HTTP 協議老的標準是 HTTP/1.0，為了提高系統的效率，HTTP 1.0 規定瀏覽器與服務器只保持短暫的連接，瀏覽器的每次請求都需要與服務器建立一個 TCP 連接，服務器完成請求處理后立即斷開 TCP 連接，服務器不跟蹤每個客戶也不記錄過去的請求。http1.0 被抱怨最多的就是連接無法復用，和 head of line blocking 這兩個問題。客戶端是依據域名來向服務器建立連接，一般 PC 端瀏覽器會針對單個域名的 server 同時建立 6~8 個連接，手機端的連接數則一般控制在 4~6 個。顯然連接數並不是越多越好，資源開銷和整體延遲都會隨之增大。連接無法復用會導致每次請求都經歷三次握手和慢啟動。三次握手在高延遲的場景下影響較明顯，慢啟動則對文件類大請求影響較大。head of line blocking 會導致帶寬無法被充分利用，以及後續健康請求被阻塞。為了克服 HTTP 1.0 的這個缺陷，HTTP1.1 支持長連接（HTTP/1.1 的默認模式使用帶流水線的長連接），在一個 TCP 連接上可以傳送多個 HTTP 請求和響應，減少了建立和關閉連接的消耗和延遲。HTTP 1.1 還提供了與身份認證、狀態管理和 Cache 緩存等機制相關的請求頭和響應頭。HTTP2.0 支持明文 HTTP 傳輸，

而 SPDY 強制使用 HTTPS .HTTP2.0 可以支持非 HTTPS，但是現在主流的瀏覽器像 chrome，firefox 表示還是只支持基於 TLS 部署的 HTTP2.0 協議，所以要想升級成 HTTP2.0 還是先升級 HTTPS。

HTTP2.0 消息頭的壓縮算法採用 HPACK，而非 SPDY 採用的 DEFLATE