HTTP

超文本傳輸協定(HyperText Transfer Protocol，HTTP）是一種用於分佈式、協作式和超媒體訊息系統的應用層協定。設計 HTTP 最初的目的是為了提供一種發布和接收 HTML 頁面的方法。透過 HTTP 或者 HTTPS 協定請求的資源由統一資源識別碼（Uniform Resource Identifiers，URL）來標識。HTTP 的發展是由提姆·柏內茲-李於 1989 年在歐洲核子研究組織（CERN）所發起。HTTP 的標準制定由全球資訊網協會（World Wide Web Consortium，W3C）和網際網路工程任務組（Internet Engineering Task Force，IETF）進行協調，最終發布了一系列的 RFC，其中最著名的是 1999 年 6 月公佈的 RFC 2616，定義了 HTTP 協定中現今廣泛使用的一個版本——HTTP 1.1。2014 年 12 月，網際網路工程任務組（IETF）的 Hypertext Transfer Protocol Bis（httpbis）工作小組將 HTTP/2 標準提議遞交至 IESG 進行討論，於 2015 年 2 月 17 日被批准。 HTTP/2 標準於 2015 年 5 月以 RFC 7540 正式發表，取代 HTTP 1.1 成為 HTTP 的實作標準。

HTTP 協議中最老的標準是 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。訪問一個包含有許多影象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象，上一次和下一次請求完全分離。即使影象檔案都很小，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。

為了克服 HTTP 1.0 的這個缺陷，HTTP 1.1 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，減少了建立和關閉連線的消耗和延遲。一個包含有許多影象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 在繼承了 HTTP 1.0 優點的基礎上，也克服了 HTTP 1.0 的效能問題。HTTP 1.1 通過增加更多的請求頭和響應頭來改進和擴充 HTTP 1.0 的功能。如，HTTP 1.0 不支援 Host 請求頭欄位。但是雖然一段時間內的連線複用對 PC 端瀏覽器的體驗幫助很大，因為大部分的請求在集中在一小段時間以內。但對移動 app 來說，成效不大，app 端的請求比較分散且時間

跨度相對較大。

HTTP2.0 因為以下幾點特性，所以比前面兩個協議在效能上有很大的提升:

1.多路複用 (Multiplexing) : 允許同時通過單一的 HTTP/2 連線發起多重的請求-響應訊息。

2. 二進位制分幀 : HTTP/2 在 應用層(HTTP/2)和傳輸層(TCP or UDP)之間增加一個二進位制分幀層。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。

3. 首部壓縮（Header Compression） : 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。

4. 服務端推送（Server Push） : 服務端推送是一種在客戶端請求之前傳送資料的機制。在 HTTP/2 中，伺服器可以對客戶端的一個請求傳送多個響應。