

網頁程式設計作業-HTTP

學號:B0929016

姓名:蘇稚鈞

系級:資工二甲

超文字傳輸通訊協定 (HTTP) 是針對在網路上傳送內容而設計的通訊協定。 HTTP 是簡單的通訊協定，利用可靠的傳輸控制通訊協定 (TCP) 服務來執行其內容傳輸功能。 因此，HTTP 是高度可靠的內容傳輸通訊協定。 HTTP 是其中一個最常使用的應用程式協定。 Web 上的所有作業都會使用 HTTP 通訊協定。HTTPS 是 HTTP 通訊協定的安全版本，會使用傳輸層安全性 (TLS) 來執行 HTTP，以保護基礎 TCP 連線。 除了設定 TLS 所需的其他設定以外，HTTPS 基本上與使用中的 HTTP 相同。

而若要讓 HTTPS 正常運作，NetX Web HTTP 封裝需要安裝 NetX (5.10 版或) 更新版本，以及 NetX 安全的 TLS (5.11 版或更新版本的)。 此外，也必須建立 IP 實例，而且必須在該相同 IP 實例上啟用 TCP，以搭配使用 TLS。 TLS 會話必須使用適當的密碼編譯常式來初始化，也就是信任的 CA 憑證，且必須配置給 TLS 信號交換期間，遠端伺服器主機將提供的憑證空間。 NetX Web HTTP 套件的 HTTPS 伺服器部分有幾個額外的需求。首先，它需要完整存取 TCP 知名埠 443 來處理所有用戶端 HTTPS 要求 (如同純文字 HTTP 一樣，此埠可以由應用程式) 變更。其次，TLS 會話必須使用適當的密碼編譯常式和伺服器身分識別憑證 (或預先共用的金鑰) 來初始化。

而上文中的 NetX Web HTTP 就會按照 HTTP 1.1 等標準實行。當然，除了 HTTP 1.1，常見的還有 HTTP 1.0 及 HTTP 2.0。

先來說明 HTTP 1.0，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。同時，頻寬和延遲也是影響一個網路請求的重要因素。在網路基礎建設已經使得頻寬得到極大的提升的當下，大部分時候都是延遲在於響應速度。

再來是 HTTP 1.1，為了克服 HTTP 1.0 的缺陷，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。除此之外，HTTP/1.1 相較於 HTTP/1.0 協議的區別主要體現在：快取處理、頻寬優化及網路連線的使用、錯誤通知的管理、訊息在網路中的傳送、網際網路地址的維護和安全性及完整性上。

最後是 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，突破了 HTTP1.1 的效能限制，改進傳輸效能。因此，與 HTTP1.1 相比，HTTP 2.0 多了下列四大特性，第一、多路複用，多路複用允許同時通過單一的 HTTP 2.0 連線發起多重的請求-響應訊息。在 HTTP1.1 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，<http://twimg.com>，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 2.0 的多路複用(Multiplexing) 則允許同時通過單一的 HTTP 2.0 連線發起多重的請求-響應訊息。因此 HTTP 2.0 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 2.0 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。第二、二進位制分幀，HTTP 2.0 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 HTTP 1.1 的效能限制，改進傳輸效能，實現低延遲和高吞吐量。在二進位制分幀層中，HTTP 2.0 會將所有傳輸的資訊分割為更小的訊息和幀 (frame)，並對它們採用二進位制格式的編碼，其中 HTTP 1.x 的首部資訊會被封裝到 HEADER frame，而相應的 Request Body 則封裝到 DATA frame 裡面。HTTP 2.0 通訊都在一個連線上完成，這個連線可以承載任意數量的雙向資料流。在過去，HTTP 效能優化的關鍵並不在於高頻寬，而是低延遲。TCP 連線會隨著時間進行自我調諧，起初會限制連線的最大速度，如果資料成功傳輸，會隨著時間的推移提高傳輸的速度。這種調諧則被稱為 TCP 慢啟動。由於這種原因，讓原本就具有突發性和短時性的 HTTP 連線變的十分低效。HTTP 2.0 通過讓所有資料流共用同一個連線，可以更有效地使用 TCP 連線，讓高頻寬也能真正的服務於 HTTP 的效能提升。第三、首部壓縮，HTTP 1.1 並不支援 HTTP 首部

壓縮，為此 SPDY 和 HTTP 2.0 應運而生，SPDY 使用的是通用的 DEFLATE 演算法，而 HTTP 2.0 則使用了專門為首部壓縮而設計的 HPACK 演算法。第四、服務端推送，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。Server Push 讓 HTTP1.x 時代使用內嵌資源的優化手段變得沒有意義；如果一個請求是由你的主頁發起的，伺服器很可能會響應主頁內容、logo 以及樣式表，因為它知道客戶端會用到這些東西。這相當於在一個 HTML 文件內集合了所有的資源，不過與之相比，伺服器推送還有一個很大的優勢：可以快取，也讓在遵循同源的情況下，不同頁面之間可以共享快取資源成為可能。

參考資料

1. <https://docs.microsoft.com/zh-tw/azure/rtos/netx-duo/netx-duo-web-http/chapter1>
2. <https://codertw.com/%E7%A8%8B%E5%BC%8F%E8%AA%9E%E8%A8%80/496897/>
3. <https://www.gushiciku.cn/pl/pujW/zh-tw>