

# Computer Networking Notes Ch. 2

---

## How to create a network app?

---

- 軟體主要是開發在終端上面，也就是 application layer 相互運作。
- 在開發網路軟體時，不需要考慮到網路設備。

## Architecture

---

### Client - Server architecture

- Server
  - 向 client 提供資料。
  - 永遠都在開機狀態。
  - 具有固定的 IP Address。
- Client
  - 向伺服器做聯繫並取得資料。
  - 不會與其他 client 端做聯繫。

### P2P architecture

- 去中心化，client 也有可能是 server，也有可能兩者都是。
- 一個 peers 會像其他的 peers 請求資訊，因此會從其他的 peers 取得資料。
- peers 的 IP 是動態的。
- 最常使用的是 P2P File Sharing。

## TCP / UDP

---

### Application Requiriement

- 應用程式需要的四大：
  - data intergrity
  - throughput
  - timing
  - security
- 依照需求，有些應用程式要求網速、時間、不能有任何 data loss...
- 主要分成兩種不同的 transport protocols services：TCP、UDP

### TCP

- 主要有五大特點：
  - reliable transport：TCP 會去確認丟出去的封包是不是能夠正確的被 Receiver 所接收，故可以確保沒有 Data Loss，適合不允許有 Data Loss 的應用程式。
  - flow control
  - congestion control
  - connection-oriented
- 不提供這些服務：

### UDP

- 與 TCP 相反：
  - unreliable transport：UDP 不會去確認丟出去的封包是不是能夠被 receiver 所接收，不管對方是死是活就是砸。
- 不提供這些服務：

## Web and HTTP

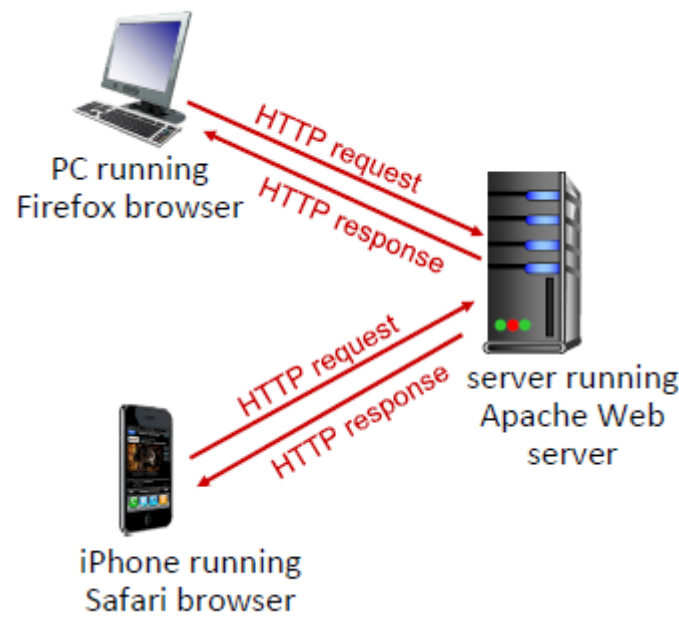
---

# HTTP

- 全名為 hypertext transfer protocol，使用 Client-Server Model。

## Client-Server Model

- 主要步驟：
  - client 會向 server 提出請求（Request），然後呈現 server 提供的網路物件。
  - server 會向發出請求的 client 端傳送物件（Response）。
- 過程中使用 TCP 的 Port 80。
- 是一個 stateless，不會記錄前一個使用者的請求。



## Two type of connection

- Non-persistent HTTP (HTTP 1.0)
  - 流程：開始連接 → 下載一個物件 → 關閉連接。
  - 通常如果要下載多個物件，就需要有多個連接。
  - 這樣的設計最簡單，但沒有效率。
- Persistent HTTP (HTTP 1.1 - beyond)
  - 流程：開始連接 → 下載多個物件 → 關閉連接。

## The example of Non-persistent HTTP

如果有一個連結 <http://uriqh-website.net/home.index>，含有文字與 10 個圖片：

- Client 初始化 TCP 連接，利用連結來連接 Server。
- Server 接受 Client 的連接，回傳 Accept。
- Client 利用 Socket 傳送請求訊息，請求 `home.index` 物件。
- Server 利用 Socket 回傳物件。
- Client 得到文字物件，Server 關閉連接。

用以上的方法來得到 10 個圖片，每次要求一個物件，故需要連接 10 次。

Here should have a image to describe the description above.

## The response time of Non-persistent HTTP

- RTT：小封包從 Client 往 Server 端的發送，並且回傳結果所需的時間。
- 使用 RTT 來估算 HTTP 的時間：
  - 對於 Non-persistent HTTP 來說，回傳 Accept 需要一個 RTT，回傳物件也需要一個 RTT，所以總共需要兩個 RTT。
  - 因此我們可以估算對於 Non-persistent HTTP 的反應時間為： $2RTT + \text{file transmission time}$ 。
- 若我們考慮傳輸距離，則這個 RTT 可能會很大或很小，對於一些需要 timing 的應用程式來說，可能會太久，故效率不好。

Here should have a image to describe the description above.