

多執行緒用於網路程式

# 通訊常用的Client-Server模式

- ▶ 客戶端發起對話，伺服器端回應
- ▶ 網際網路節點以IP位址識別位置
  - ▶ IP地址如同門牌號碼
- ▶ 連線一端由“IP位址:埠號”定義
  - ▶ 埠號如同一棟大公寓中的房間號碼，它實為網路程式的識別碼

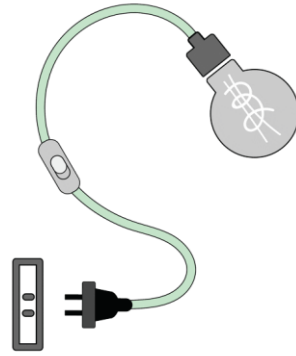
You Tube



140.125.20.80:20001

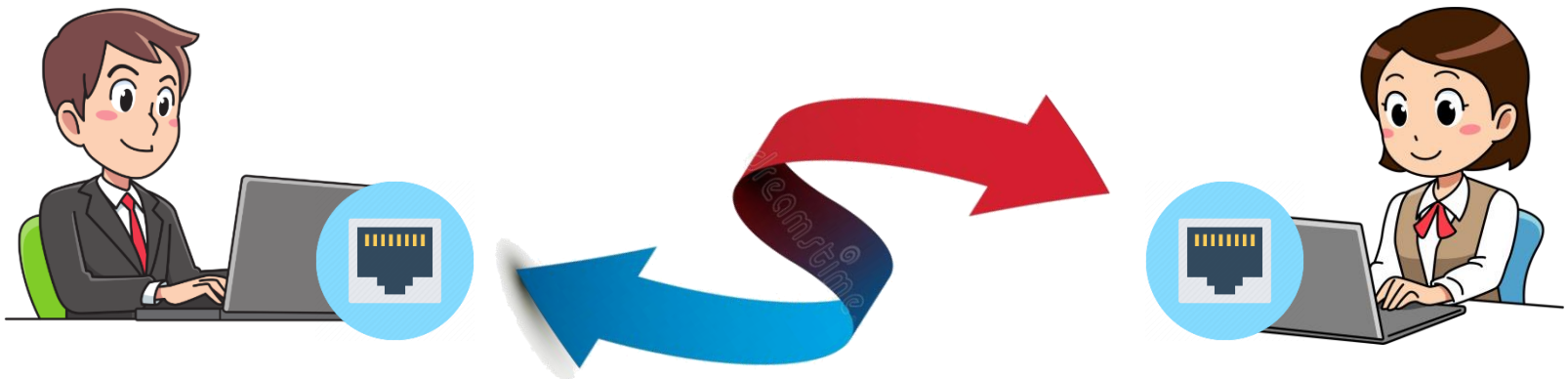


172.217.160.78:30001



# Socket Programming

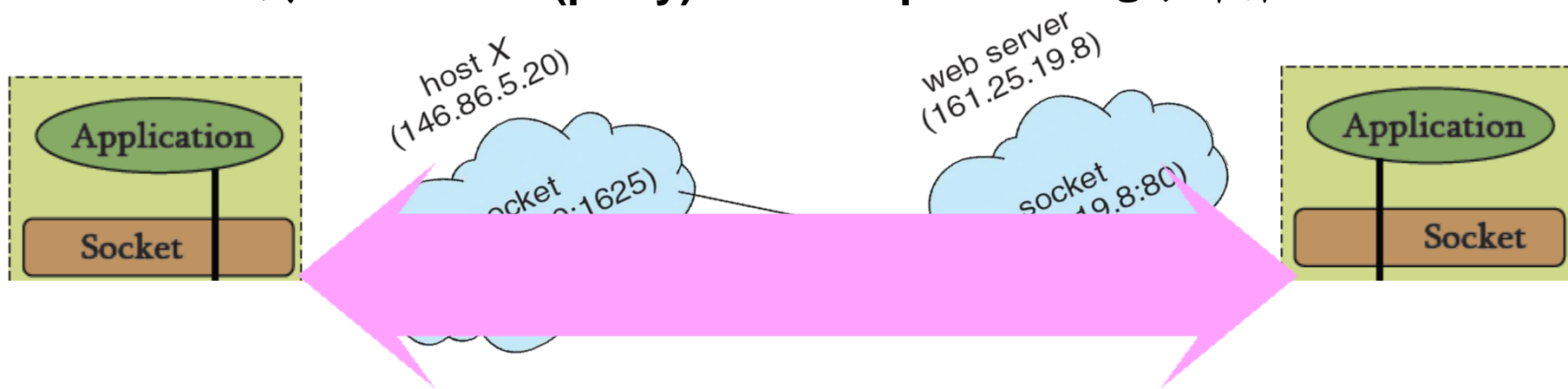
網際網路裝置彼此收送資料常用的函式庫與程式開發方式



# Socket是啥東東？



- ▶ 如同電線與插座連接，燈泡即亮起來，源源不絕得到電力
- ▶ 一個socket由通訊端點(IP位址:埠號)定義
  - ▶ Socket是標準化通訊介面，網路程式可透過它傳送/接收資料，程式設計師經由函數呼叫的形式執行類似I/O的動作，無需打理通訊傳輸底層許多細節，易於簡化網路程式的開發
  - ▶ 埠號用於識別網路主機內特定的應用程式
    - 例：21：SSH (putty)、80：http、110：電子郵件



# 通訊端點 (IP位址:埠號)

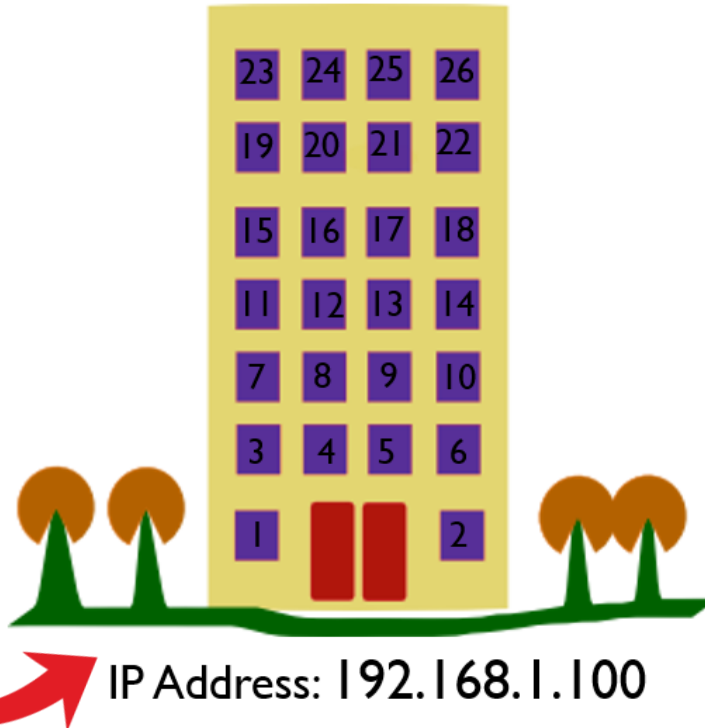
- ▶ 通訊端點(endpoint)指示網際網路一個獨特(不重複)的接口
  - ▶ 主機的IP位址如同網際網路世界的門牌號碼，郵差依據門牌號碼遞送信件，該主機執行的所有網路程式共用此門牌作為對外聯繫管道
  - ▶ 主機收到信件之後，依信件上註明的埠號(類似公寓內部聯繫的房號)轉交給真正目的地接收端(恆常聆聽該埠號的程式)作後續處理

IPEndPoint = IP + Port

192.168.1.100:7

192.168.1.100:16

192.168.1.100:22



IP Address: 192.168.1.100

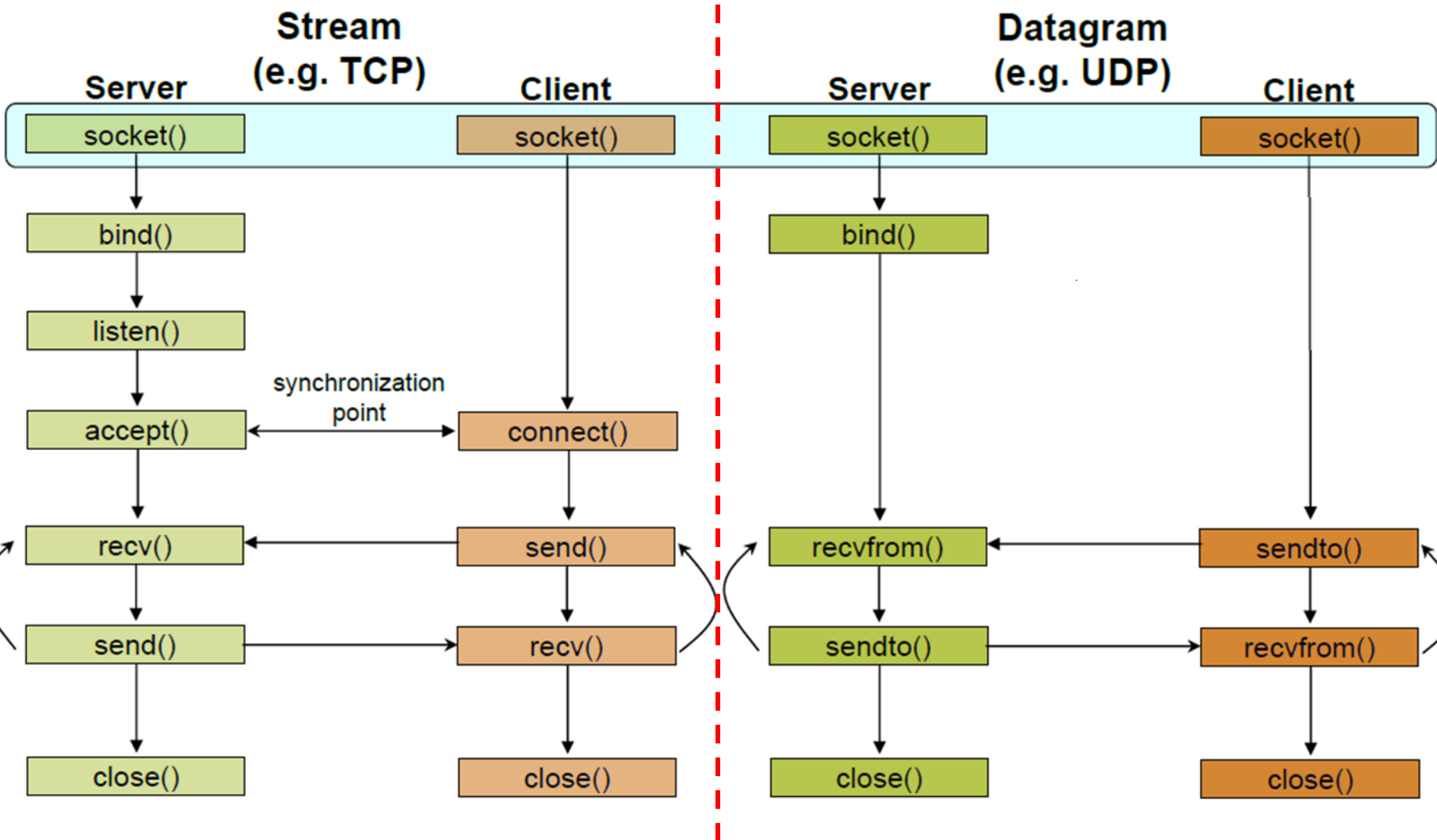
65536 Ports

0 – 1023: System Ports

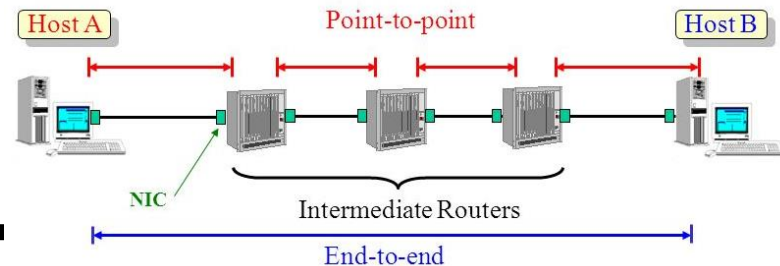
圖來源：[shorturl.at/dhkw4](http://shorturl.at/dhkw4)

# Socket常用函式

通訊模式主要有二種：連線導向TCP模式或無連線UDP模式



# TCP vs UDP



- ▶ 兩者皆提供「端對端」(end-to-end)的傳輸
  - ▶ 資料傳輸的起點和終點之間進行通訊，途中可經過多個中繼節點
- ▶ TCP (Transmission Control Protocol)
  - ▶ **可靠**的位元組串流傳輸協定，依傳送順序收到資料、資料不遺漏、無重複
  - ▶ 內嵌流量控制(flow control)機制
  - ▶ **連線導向**(connection-oriented)方式：在傳送訊息之前，需先通知對方，令對方有所準備
  - ▶ 可雙向傳輸
- ▶ UDP (User Datagram Protocol)
  - ▶ 收端將不會回應是否收到發送端傳送的資料
  - ▶ 無自動重傳機制
  - ▶ 收到資料的順序可能錯亂，可能重複
  - ▶ 無連線(connectionless)方式：每個封包目的地由發送端程式指定