

# 計算機網路概論



## 基礎

© All rights reserved. No part of this publication and file may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior written permission of Professor Nen-Fu Huang (E-mail: [nfhuang@cs.nthu.edu.tw](mailto:nfhuang@cs.nthu.edu.tw)).

# 大綱

---

- 應用程式 (Applications)
- 網路連接性 (Network Connectivity)
- 網路架構 (Network Architecture)
- 網路效能 (Network Performance)

# 應用程式

---

- 一般大眾大多經由應用程式 (applications) 來認識互聯網(Internet)
  - 全球資訊網 (World Wide Web, WWW)
  - 線上遊戲 (On line games)
  - 電子郵件 (Gmail, hotmail,...)
  - 社群網站 (Facebook, twitter,...)
  - 影音串流服務 (Youtube, ppstream, kkbox, ...)
  - 檔案分享服務 (dropbox, ...)
  - 即時通訊軟體 (Skype, IM+, MSN, Line, WeChat,...)
  - ...

# 應用程式範例

The screenshot displays the ShareRoom interface for a meeting titled "ShareRoom展示會議". The main content area shows a document with the following text:

- It is not practical to compute  $T(x)$  if  $m$  exceeds 4 or 5.
- Based on the state diagram of a code, there is an efficient way [6] to find the free distance for large  $m$  ( up to about 20).
- No general solution to the problem of finding  $d_{free}$  for large values of  $m$  has yet been discovered.
- Some of good codes are referred in Table 1.1 .
- For Example,  $g^{(1)} = 54 = [1\ 0\ 1\ 1\ 0\ 0]$ ,  $g^{(2)} = 64 = [1\ 1\ 0\ 1\ 0\ 0]$

Below the list, the text "This is my meeting!" is displayed. A red circle highlights the first two bullet points. The interface also features a 2x2 grid of video thumbnails for participants: ChiaAn, 會員1, 黃小月, and ATO. A chat window at the bottom shows messages from ChiaAn: "Hello world" and "My name is andy lee".

一個視訊會議的多媒體應用程式

# 應用程式

---

- URL 網頁地址 俗稱
  - Uniform resource locator
  - <http://www.sharecourse.net/sharecourse/>
- HTTP 超文本傳輸協議
  - Hyper Text Transfer Protocol
- TCP 傳輸控制協議
  - Transmission Control Protocol
- 點擊一個URL可能包含17個訊息的傳送
  - 6個用來尋找主機的 IP 網址
  - 3個用來建立雙方的 TCP 連線
  - 4個用來 HTTP請求與回應
  - 4個用於解除雙方的 TCP 連線

# 大綱

---

- 應用程式 (Applications)
- 網路連接性 (Network Connectivity)
- 網路架構 (Network Architecture)
- 網路效能 (Network Performance)

# 網路連接性

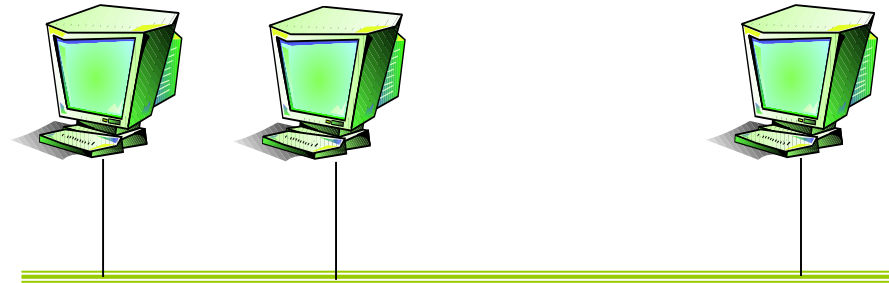
## ■ 重要的專用術語

- 鏈結 (Link)
- 節點 (Nodes)
- 點對點 (Point-to-point)
- 多方存取 (Multiple access)
- 交換網路 (Switched Network)
  - ▶ 電路交換 (Circuit Switched)
  - ▶ 封包交換 (Packet Switched)
- 封包, 訊息 (Packet, message)
- 儲存並傳送 (Store-and-forward)



點對點鏈結

Point-to-point



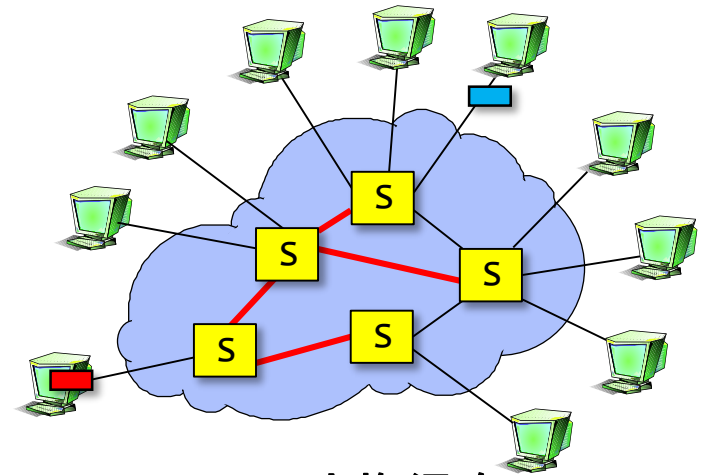
多方存取鏈結

Multiple access

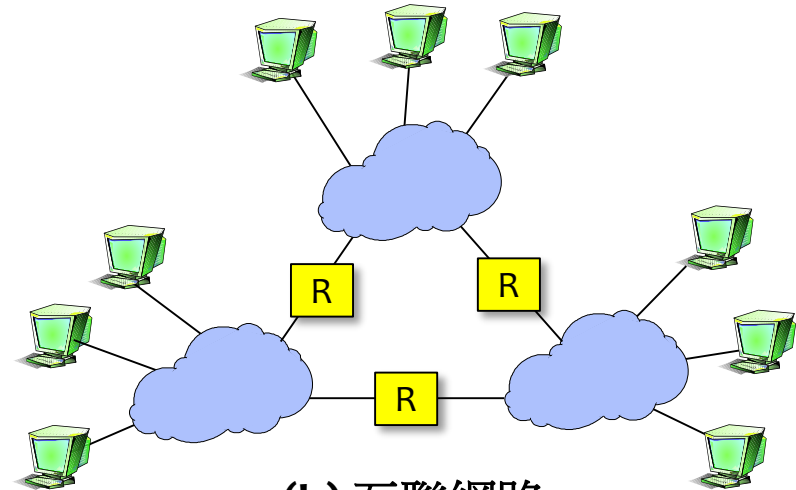
# 網路連接性

## ■ 重要的專用術語

- 主機 (Hosts)
- 交換機 (Switches)
- 擴張樹 (Spanning tree)
- 互聯網路 (Internetwork)
- 路由器/閘道 (Router/gateway)
- 主機與主機連接 (Host-to-host connectivity)
- 網址 Address
- 路由 Routing
- 單點傳送/廣播傳送/群播傳送  
Unicast/broadcast/multicast
- 區域網路 (Local Area Networks)
- 都會區網路 (Metropolitan Area Networks)
- 廣域網路 (Wide Area Networks)



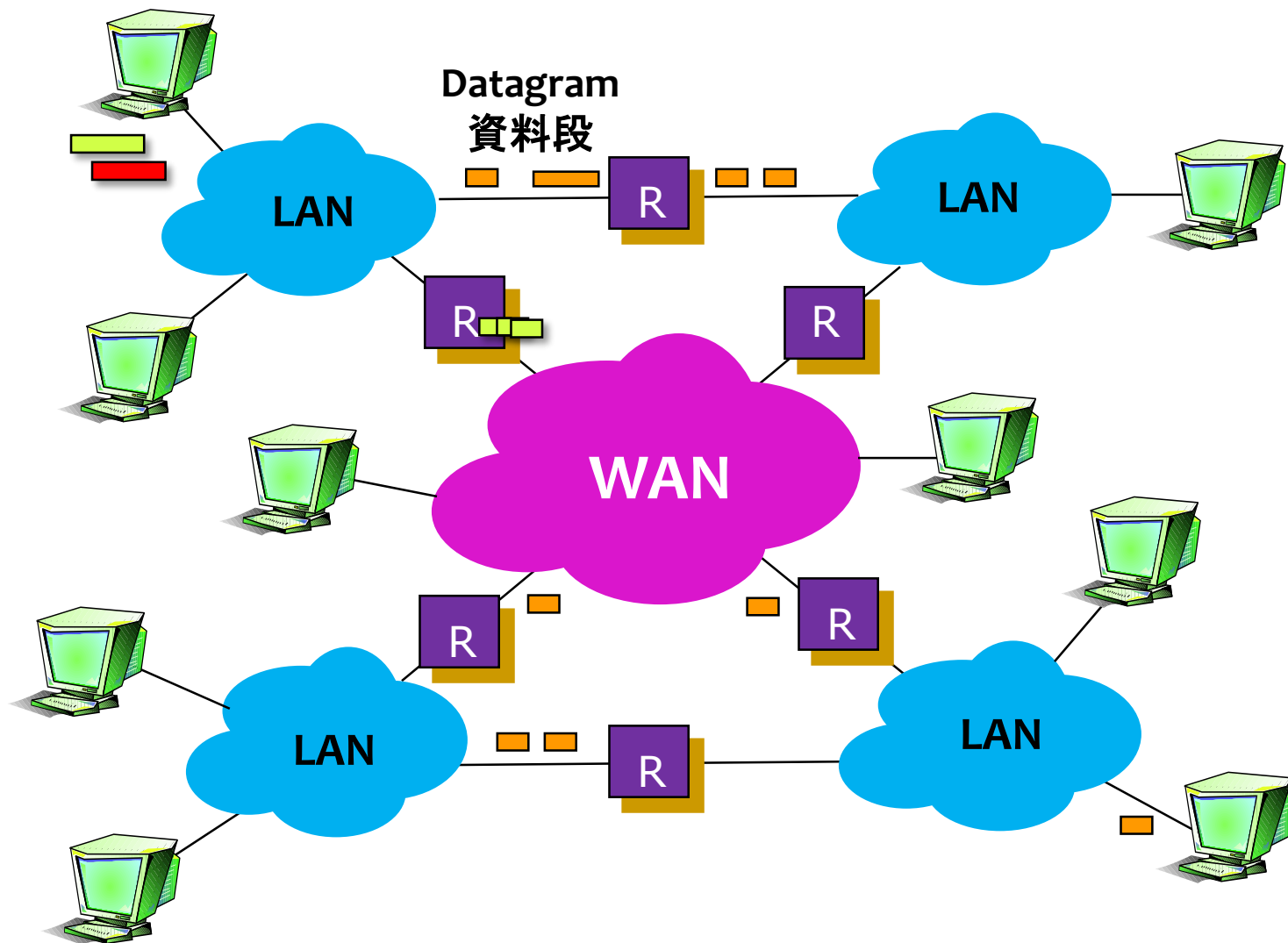
(a) 交換網路  
A switched network



(b) 互聯網路  
Interconnection of networks

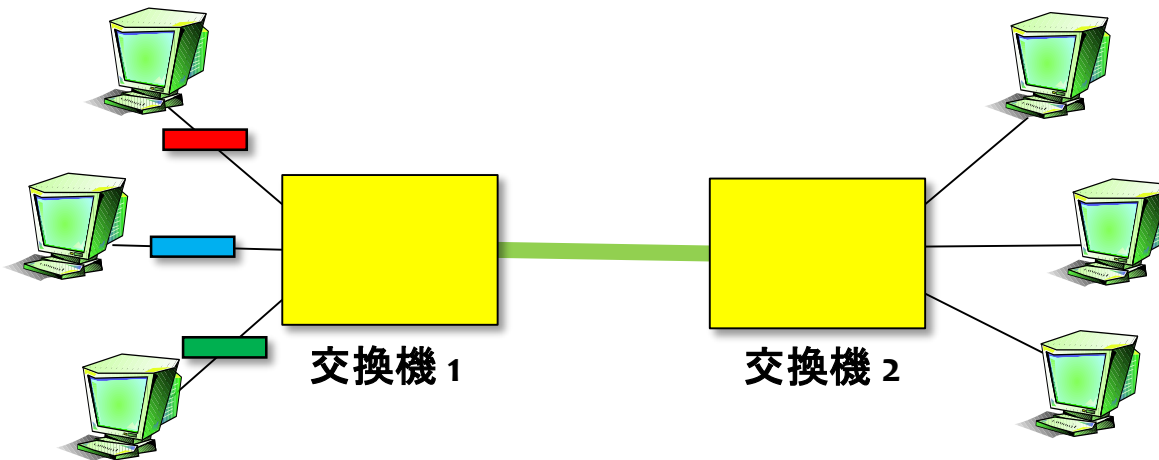


# 一個封包是如何在互聯網上被遞送的?



# 符合成本效益的資源共享

- 資源:鏈結與節點
- 如何分享一個鏈結?
  - 多工 (Multiplexing)
  - 解多工 (De-multiplexing)



在單一鏈結上多工處理數個邏輯資料流

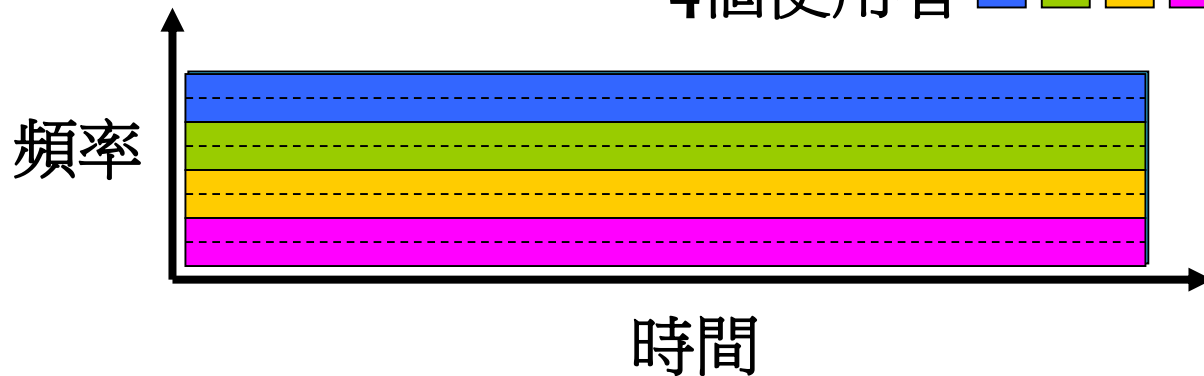
# 符合成本效益的資源共享

## ■ 分頻多工: Frequency Division Multiplexing (FDM)

分頻多工

範例:

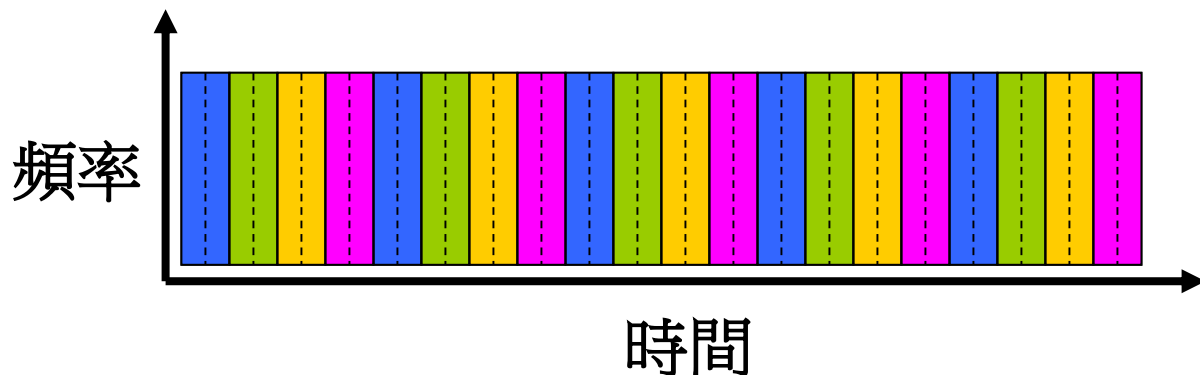
4個使用者 ■ ■ ■ ■



## ■ 同步分時多工: Synchronous Time-division Multiplexing (TDM)

- ▶ 使用預先設定好的時間區間進行資料傳輸

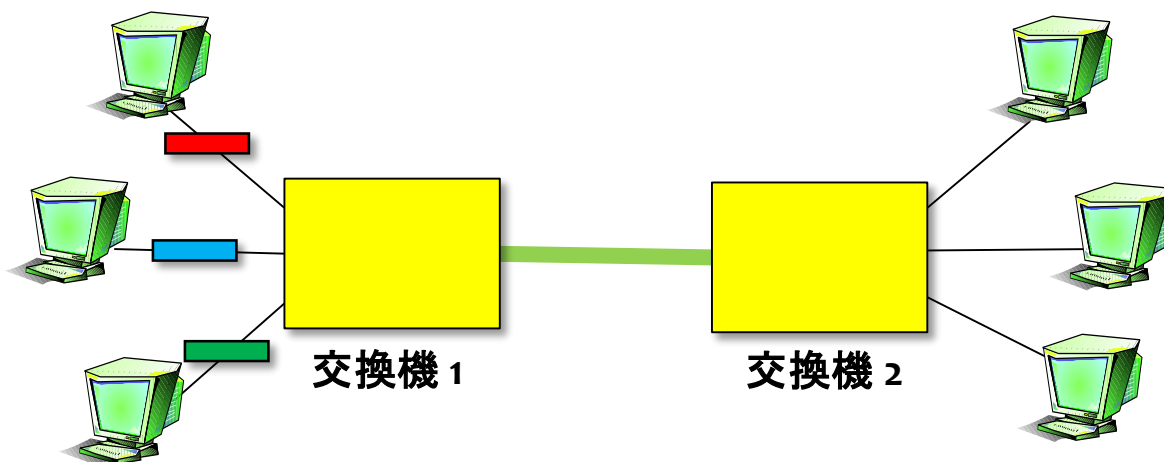
分時多工



# 符合成本效益的資源共享

## ■ 統計式多工 (Statistical Multiplexing)

- 資料依據資料流需求來傳送
- 何謂資料流 (flow)?
- 封包 V.S. 訊息
- 交換機排程法
  - ▶ 先進先出法(FIFO), 循環法(Round-Robin), 優先權法(Priorities), 服務品質(Quality-of-Service (QoS))
- 何謂壅塞?

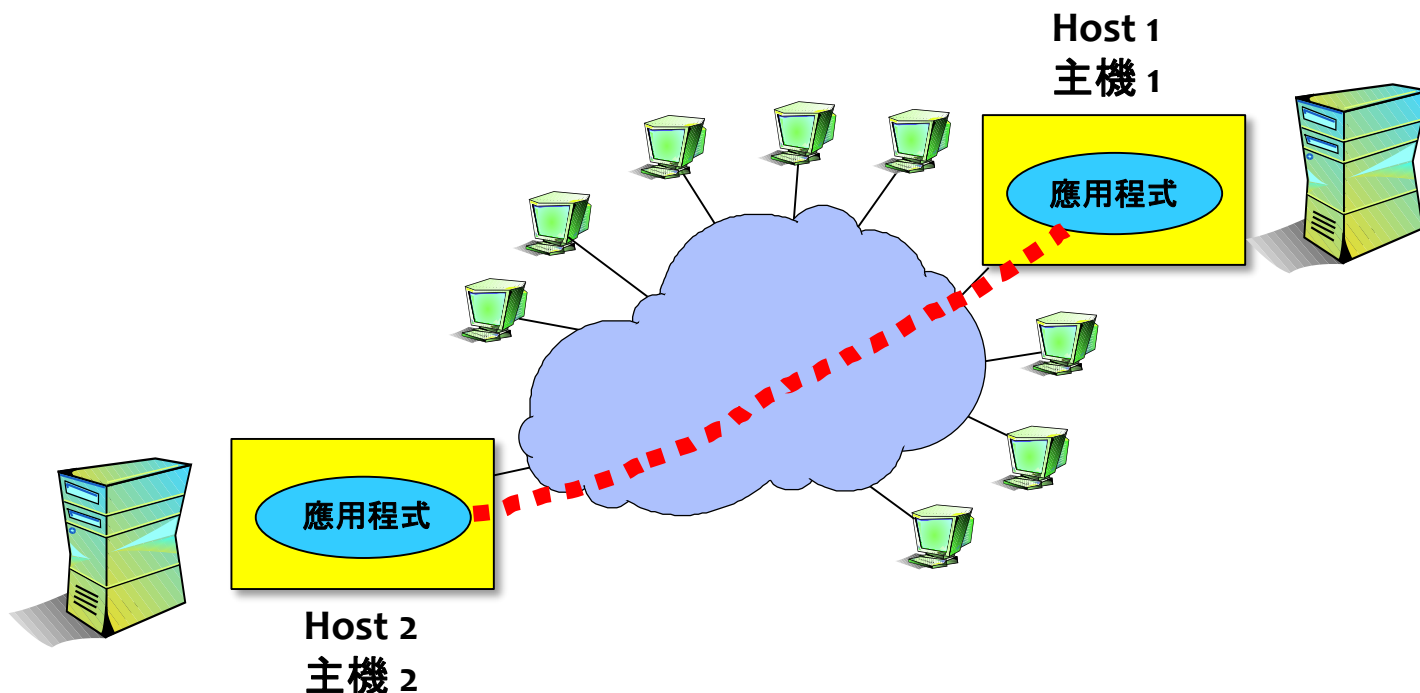


交換機將多個資料來源的封包多工分送到一個共用鏈結上

# 邏輯通道 (Logical Channels)

## ■ 邏輯通道

- 應用程式與應用程式之間的通訊路徑或管道



行程(processes)間透過邏輯通道進行通訊

# 網路可靠性

---

- 網路可能隱藏著各種因素造成傳輸發生錯誤
  - 位元遺失
    - ▶ 位元訊號錯誤 (從1變0 或是反過來)
    - ▶ 突爆錯誤-連續好幾個位元都發生錯誤
  - 封包遺失 (雍塞)
  - 鏈結或節點損壞
  - 訊息延遲
  - 訊息遞送順序錯誤
  - 惡意第三方竊聽或竄改

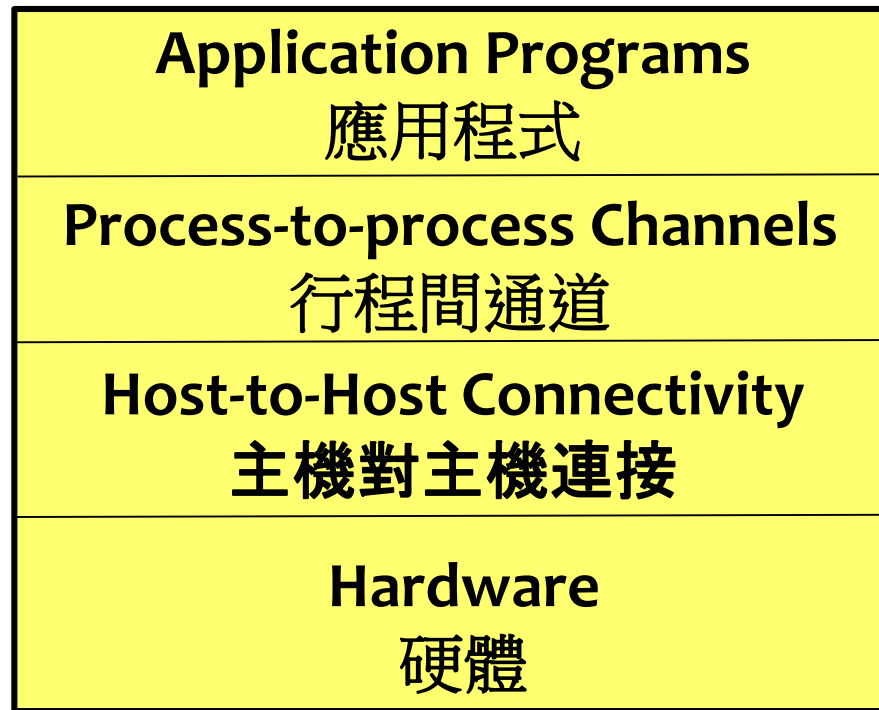
# 大綱

---

- 應用程式 (Applications)
- 網路連接性 (Network Connectivity)
- 網路架構 (Network Architecture)
- 網路效能 (Network Performance)

# 網路架構

---



分層網路系統範例

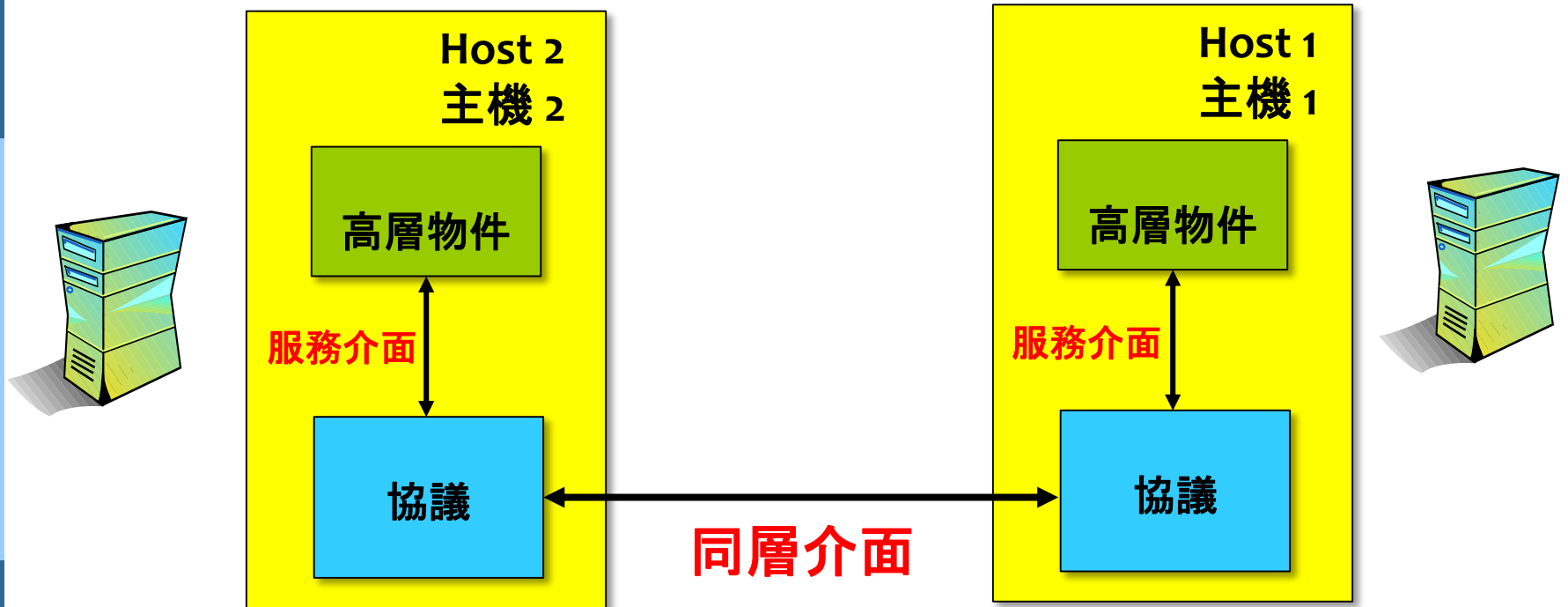


# 協議 (Protocols)

---

- 協議定義了介於不同層的介面
  - 在相同系統中的分層介面或是
  - 在對等系統中的同層介面
- 用來建立且區隔網路架構中的每個區塊模組
- 每種協議都定義了兩種不同介面
  - **服務介面**:定義本介面在協議中的執行業務
  - **同層介面**:定義用於同層對等者(peer)的訊息交換

# 協議的介面



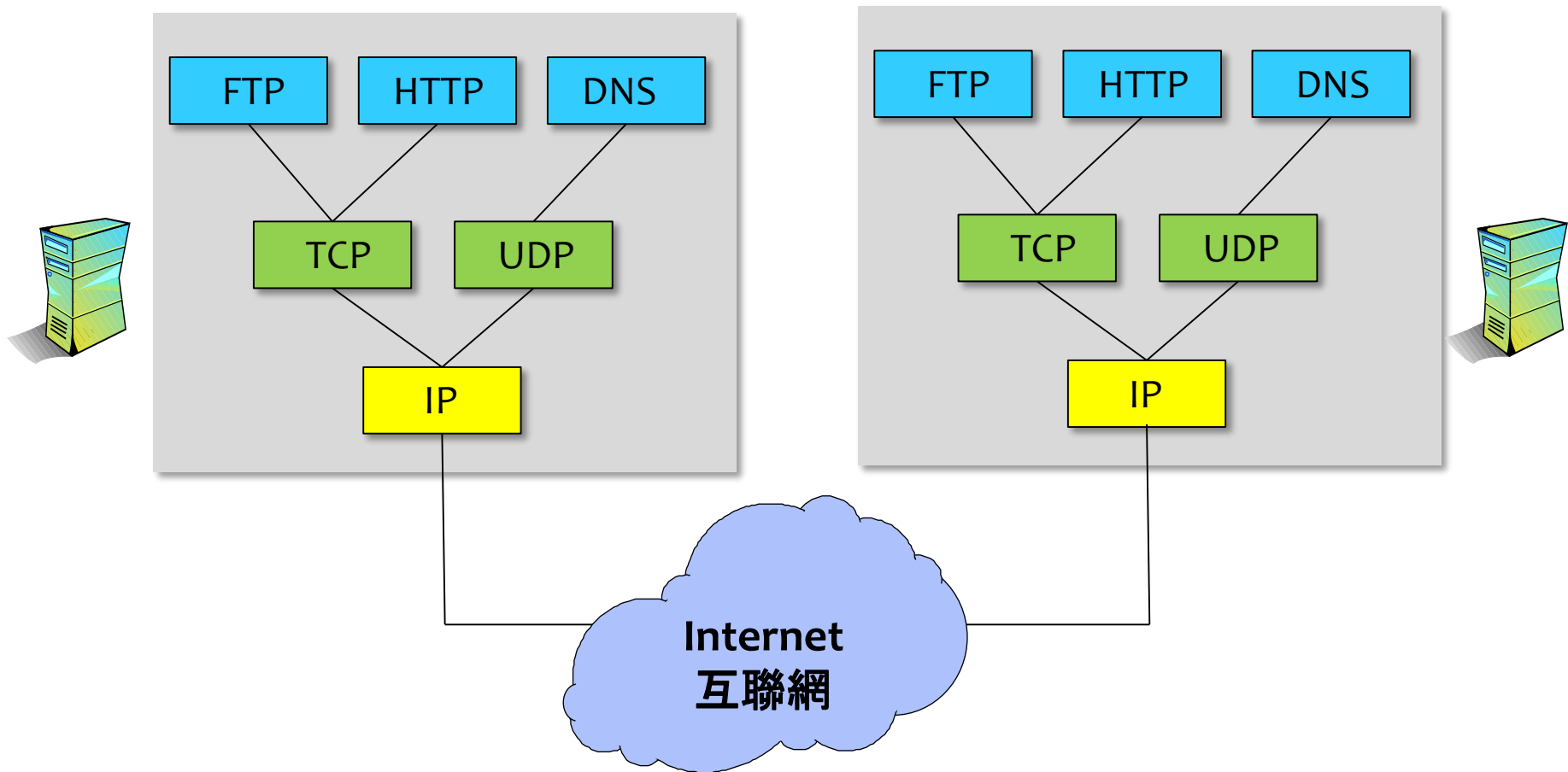
一個協議的**服務介面**與**同層介面**

# 協議

---

- **協議規格**: 如虛擬碼, 狀態轉移圖, 訊息格式
- **相互操作**: 建構兩個或多個正確的協議規格
- **互聯網工程任務組 (IETF: Internet Engineering Task Force)**
  - 負責定義互聯網標準與協議規格

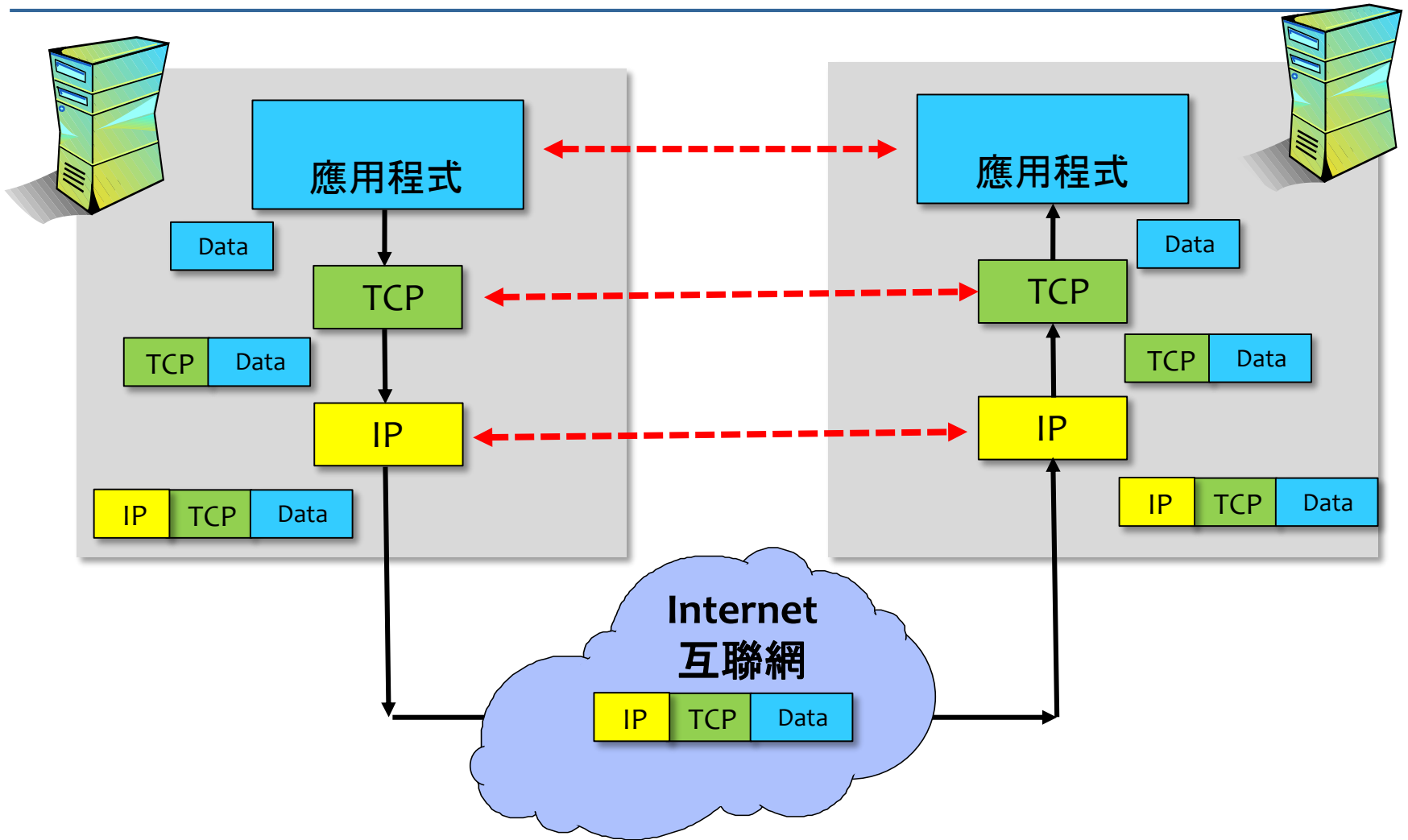
# 協議架構



協議架構範例圖

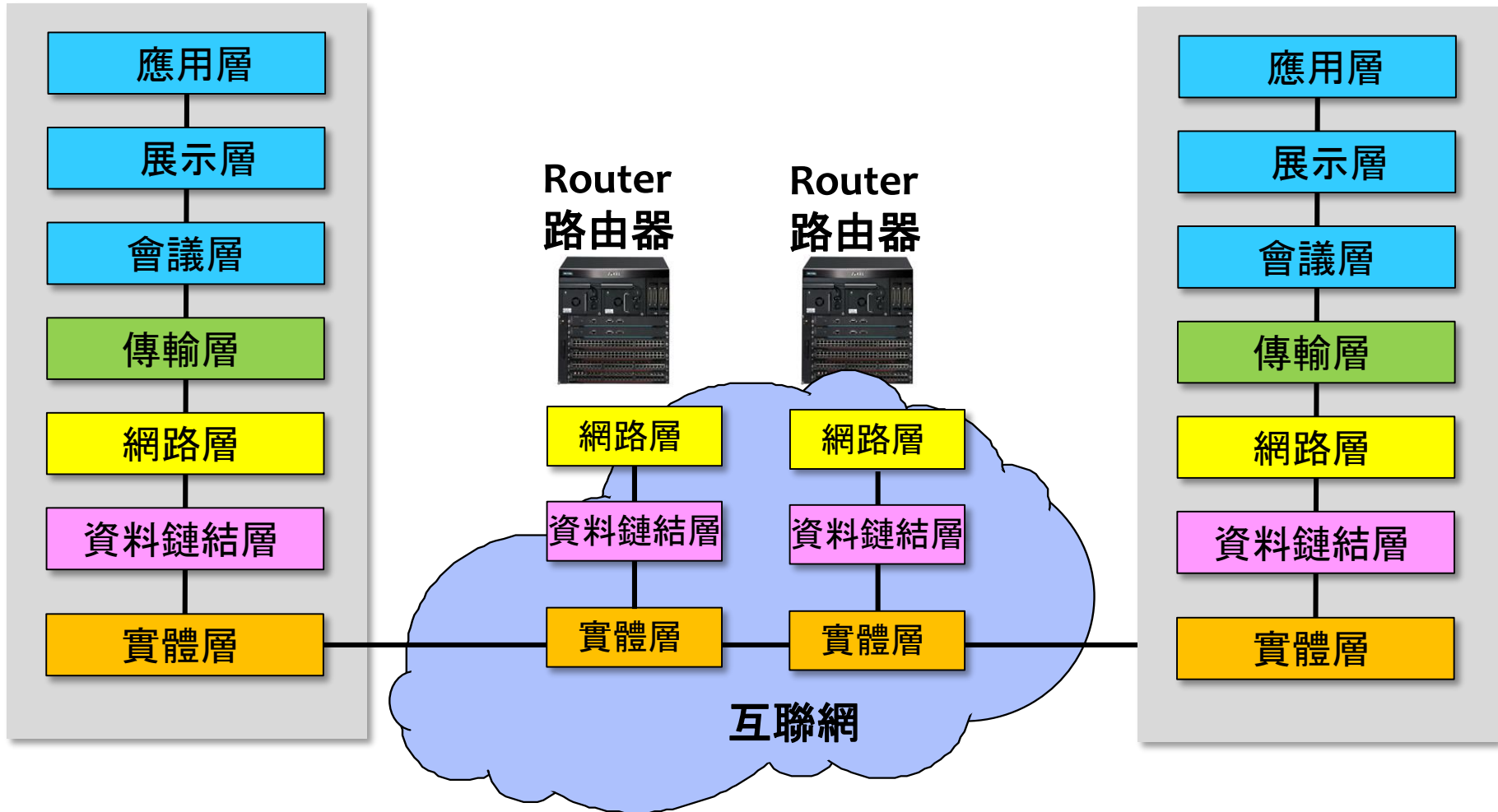
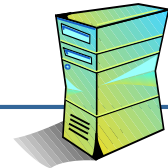
各節點透過協議與連結建立關聯性

# 訊息封裝 (encapsulation)



高層的訊息封裝在較低層的訊息中

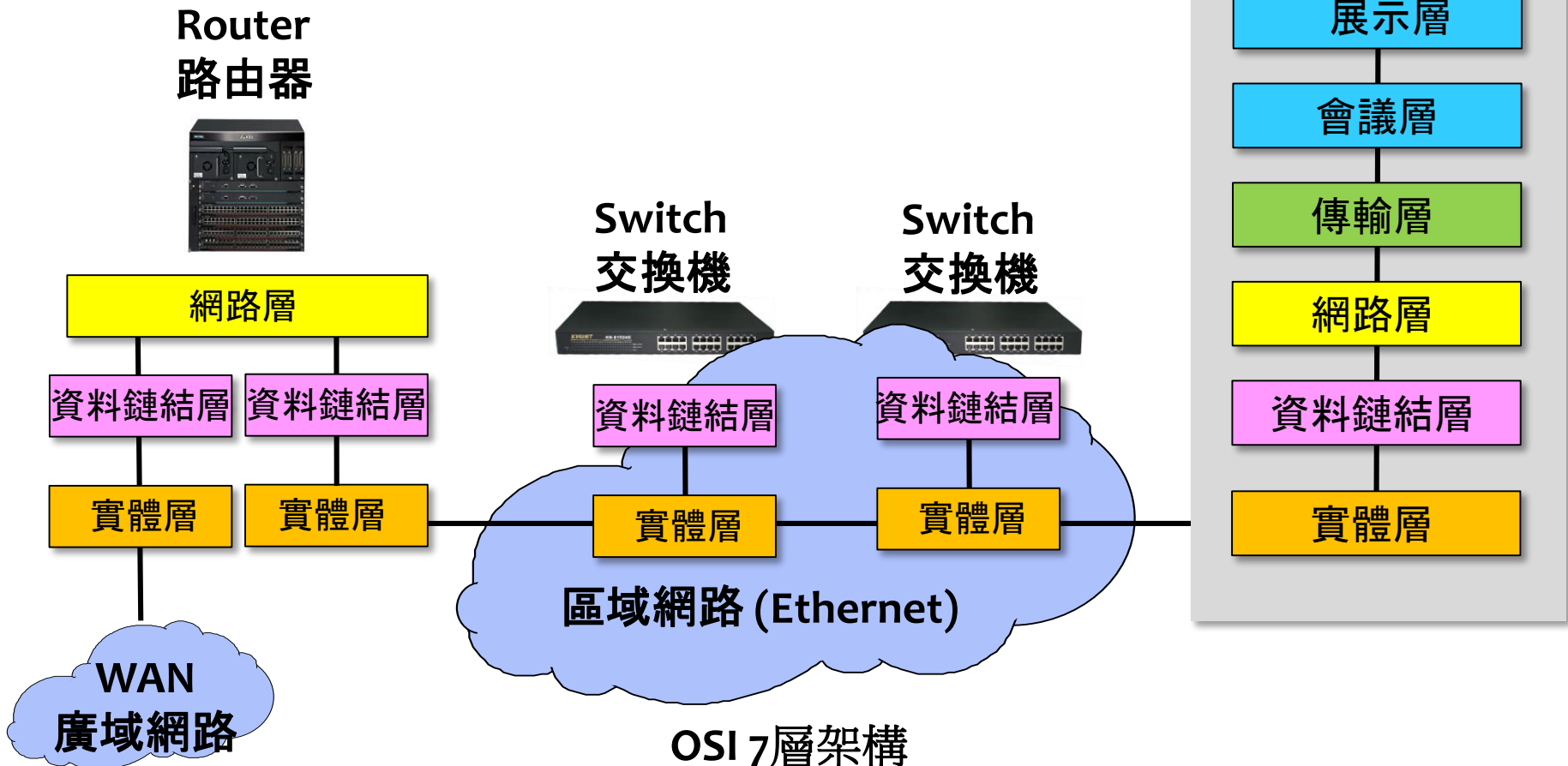
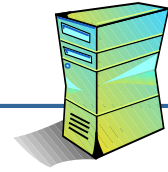
# OSI架構



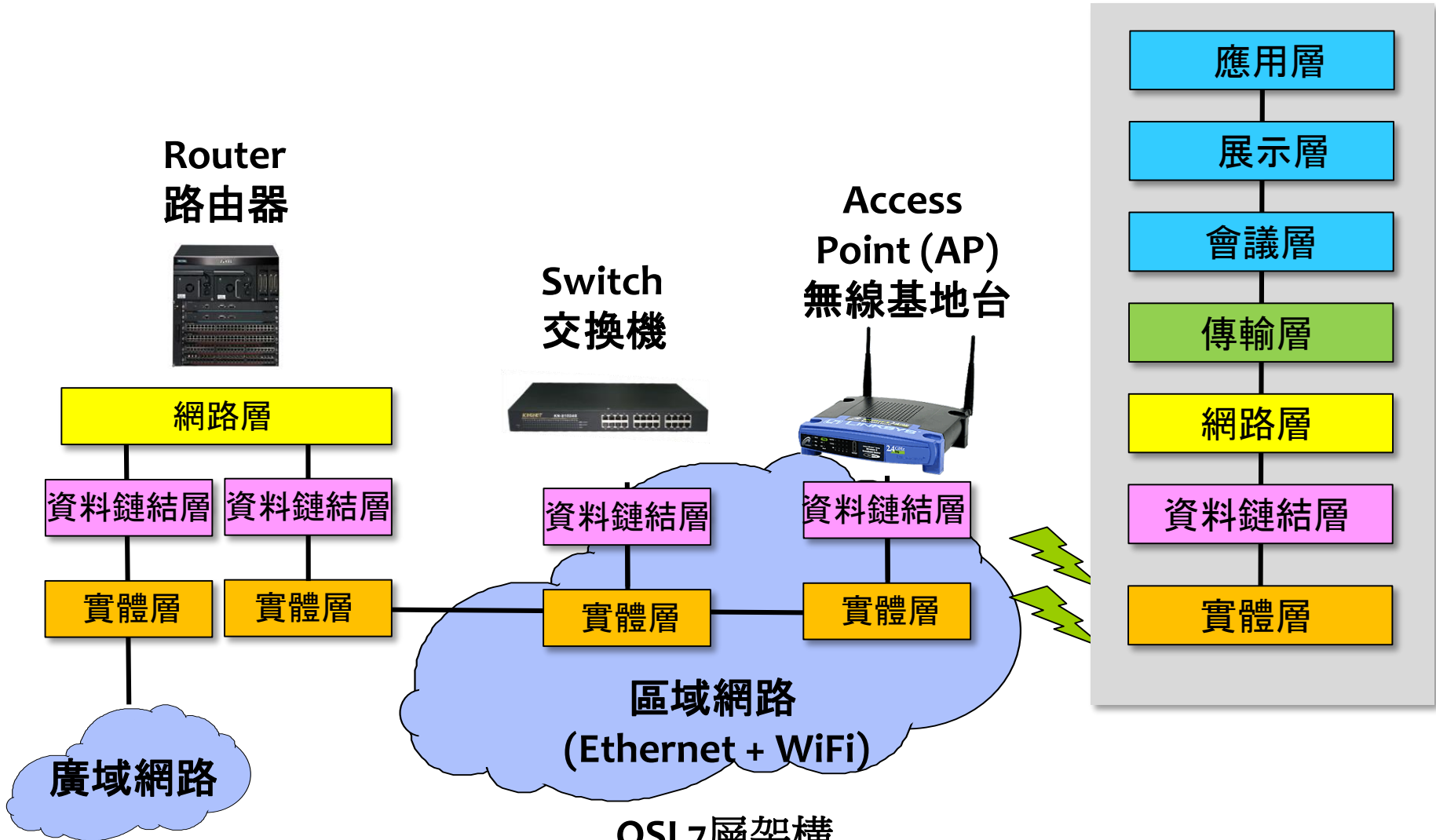
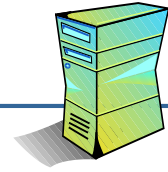
OSI 7 層架構

OSI -開放式系統互聯通訊

# OSI 架構



# OSI架構



OSI 7層架構

OSI -開放式系統互聯通訊

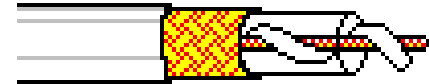


# 各層的詳述

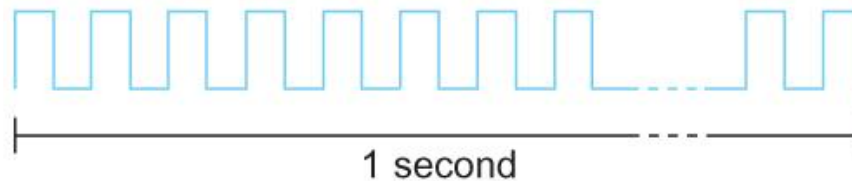
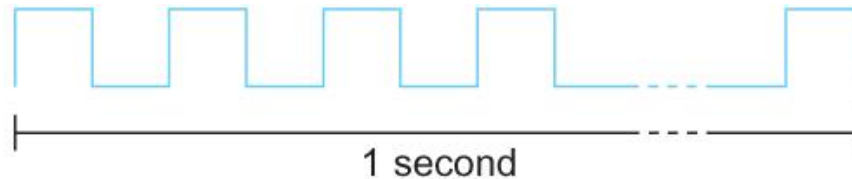
## ■ 實體層(如何將原始資料在鏈結上傳輸)

- 在一個通訊鏈結上處理**原始位元傳輸**

- ▶ 同軸電纜
- ▶ 雙絞線
- ▶ 光纖
- ▶ 無線傳輸



- 不同的訊號編碼方法



# 各層的詳述

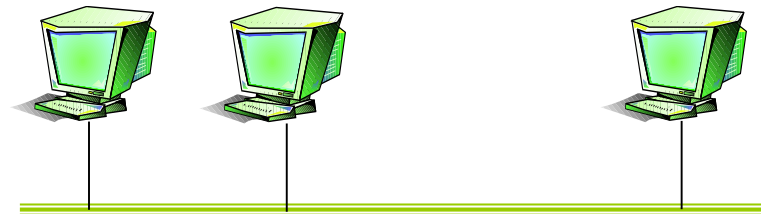
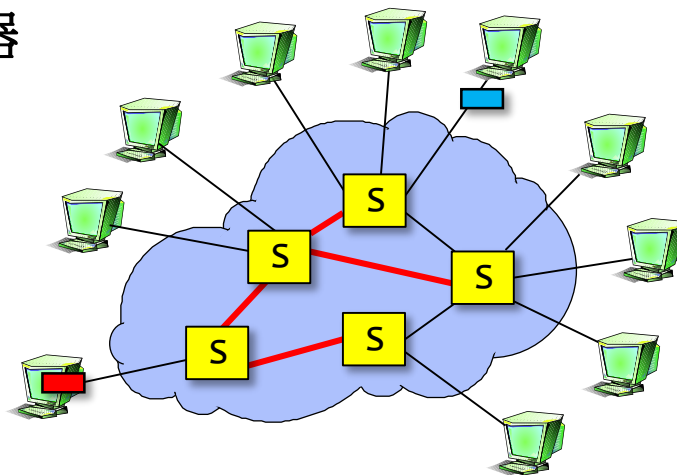
## ■ 資料鏈結層(如何將 訊框傳給直接相連的主機或設備)

- 將連續的位元流組成一個 訊框 (frame)
- 如何將一個訊框傳輸至一個直接連接的主機 (目的地)?
- 媒體存取控制 (Media Access Control Protocol)
  - ▶ CSMA/CD (IEEE 802.3 Ethernet)
  - ▶ CSMA/CA (IEEE 802.11 Wireless LAN)

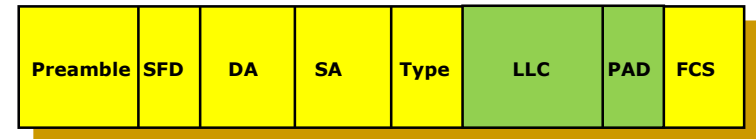
- 第二層裝置
- 交換機
- 橋接器



Point-to-point



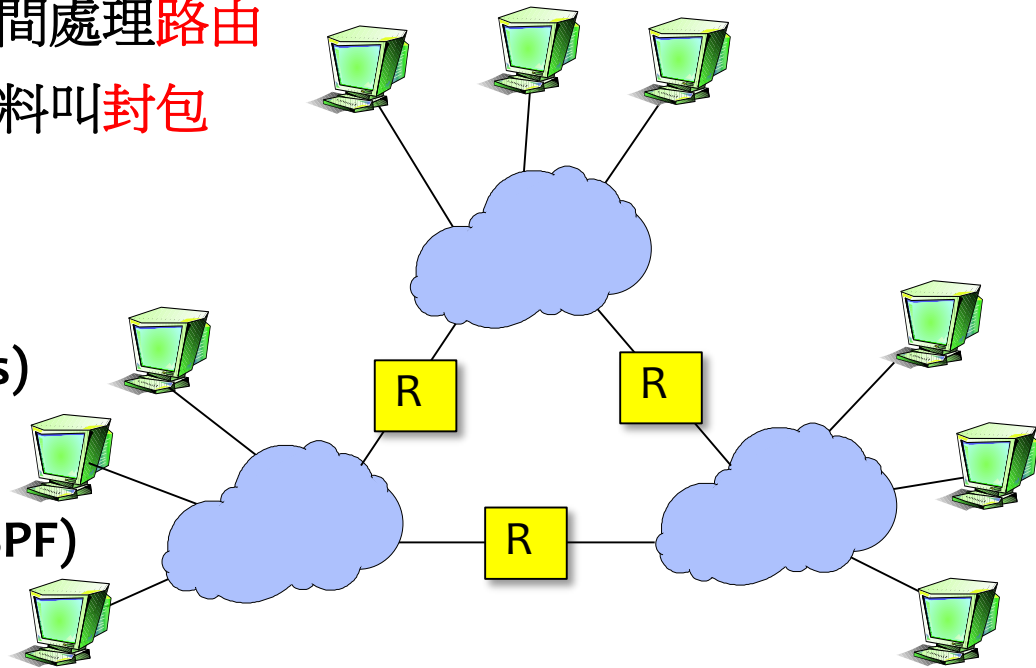
Multiple access



# 各層的詳述

## ■ 網路層 (如何將封包透過互聯網傳送給目的地主機)

- 如何將封包透過互聯網傳送至主機
- 在封包交換網路中的節點之間處理**路由**
- 在網路層節點進行交換的資料叫**封包**
- 互聯網協議 (IP)
- 路由器 (Router)
- 路由協議 (Routing protocols)
  - ▶ 路由信息協議 (RIP)
  - ▶ 開放式最短路徑優先 (OSPF)
  - ▶ 邊界閘道器協議 (BGP)
- 路由表 (Routing table)

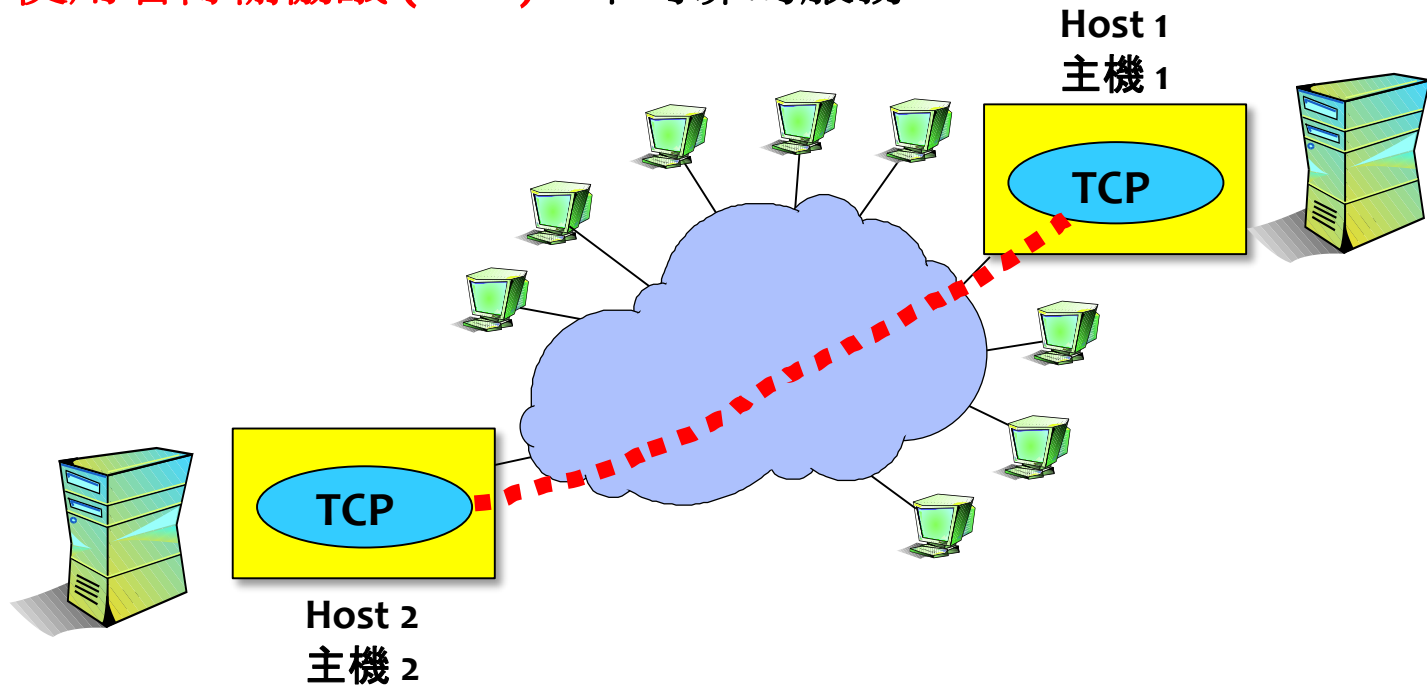


互聯的網路

所有網由器都實現較低的三層協議

# 各層的詳述

- 傳輸層(提供不同主機 行程 (processes) 間的資料傳送)
  - 實現行程間的通訊
  - 在這層傳送的每個資料稱為“訊息”
  - 傳輸控制協定議 (TCP) – 可靠的服務
  - 使用者傳輸協議 (UDP) – 不可靠的服務



# 各層的詳述

---

## ■ 會議層

- 用於將不同的傳輸流緊密結合成一個應用程式的一部分

## ■ 展示層

- 著重於網路資料交換格式的呈現

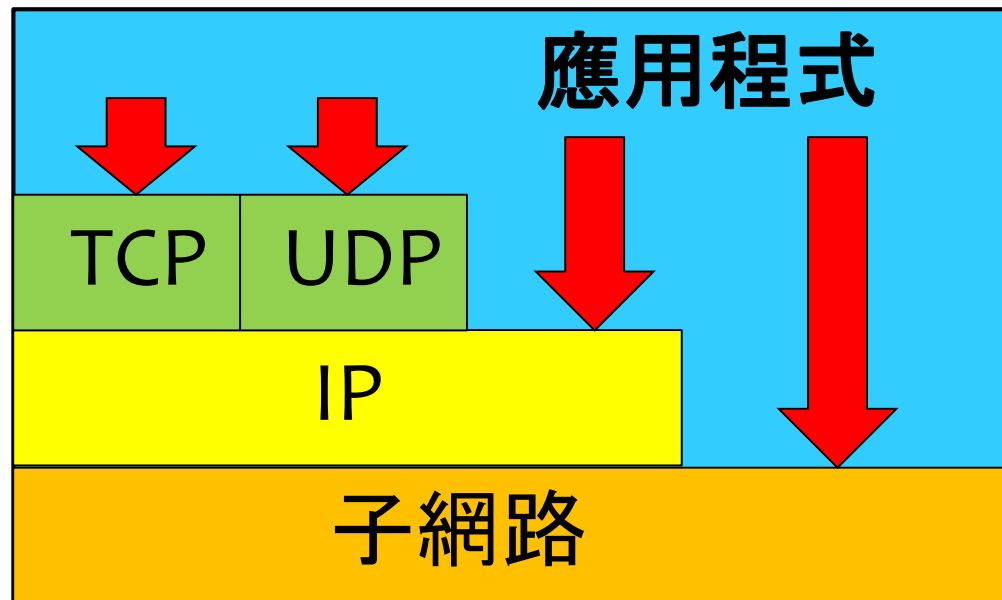
## ■ 應用層

- 提供共同的應用程式格式交換
- 檔案傳輸協定(FTP)/電子郵件(E-mail)/域名伺服器(DNS)/超文本傳輸協定(HTTP)/臉書(Facebook)

傳輸層以及較高的上層協議通常只實作在主機上，一般不會出現在網路中間的交換機或路由器等網路設備中

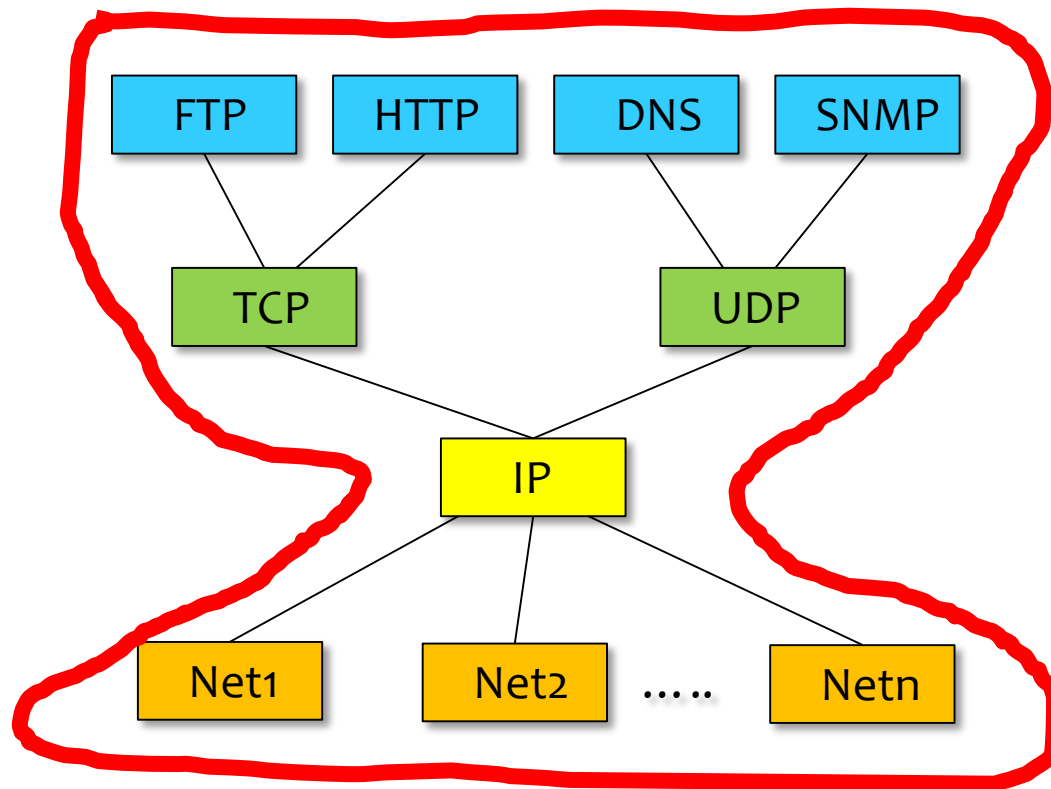
# 互聯網架構

- 由 IETF 制定
- 三個主要的功能
  - **寬鬆層與層關係**。應用程式可以**隨意略過**傳輸層以便直接使用 IP 或是其他更底層的網路服務



# 互聯網架構

- **沙漏型架構**-上層較寬，中層窄，下層又變寬。IP協議提供目前大部分上下層連接的一個重要的點



互聯網架構

# 互聯網架構

---

- 制定一個新的協議需要包含
  - ▶ 協議的規格 與
  - ▶ 至少一個(最好兩個)規格的實作



# 大綱

---

- 應用程式 (Applications)
- 網路連接性 (Network Connectivity)
- 網路架構 (Network Architecture)
- 網路效能 (Network Performance)

# 網路效能

---

## ■ 頻寬 (bandwidth)

- 頻譜的寬度
- 在一個通訊鏈結中每秒可以傳輸的位元數量

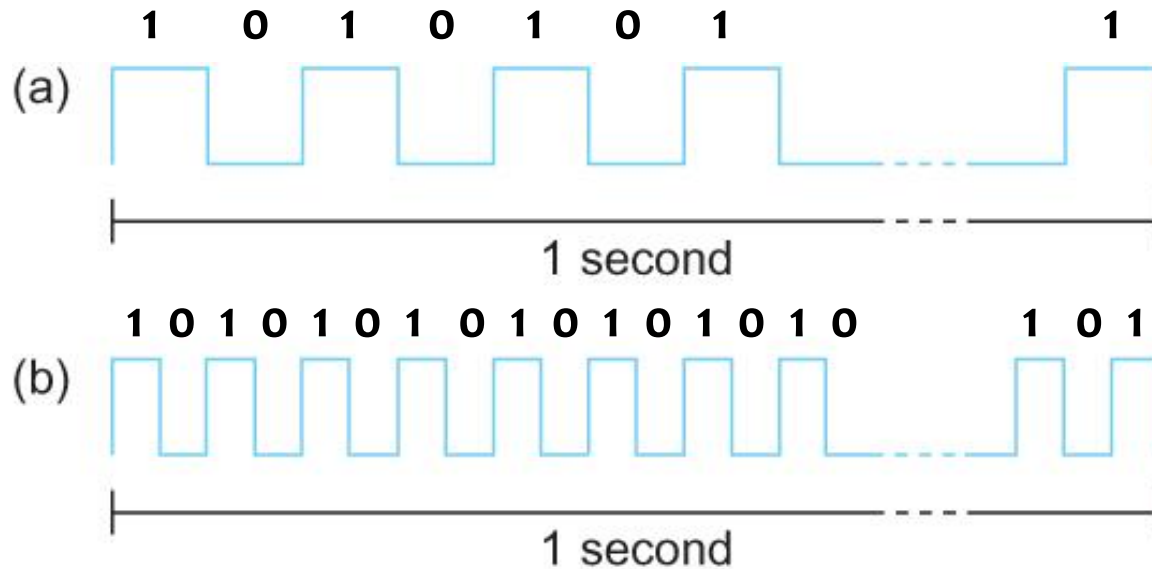
## ■ 1 Mbps: $1 \times 10^6$ 位元/秒

## ■ $1 \times 10^{-6}$ 秒傳輸一個位元，或是想像在時間軸上，一個位元將會佔據一個微秒的時間

## ■ 在2 Mbps連線中就代表訊號寬度 0.5 微秒

## ■ 越小的寬度則需要越大的單位傳輸時間

# 頻寬 (Bandwidth)



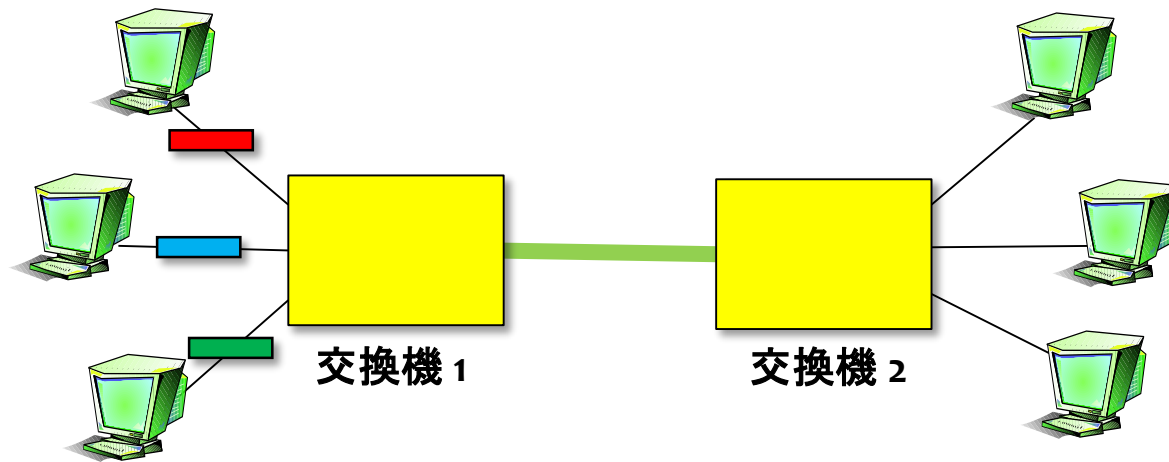
在特定的頻寬中傳輸位元將佔據一定的寬度

(a) 在1Mbps的傳輸位元(每個位元將使用1微秒的寬度)

(b) 在2Mbps的傳輸位元(每個位元將使用0.5微秒的寬度)

# 網路效能(Network Performance)

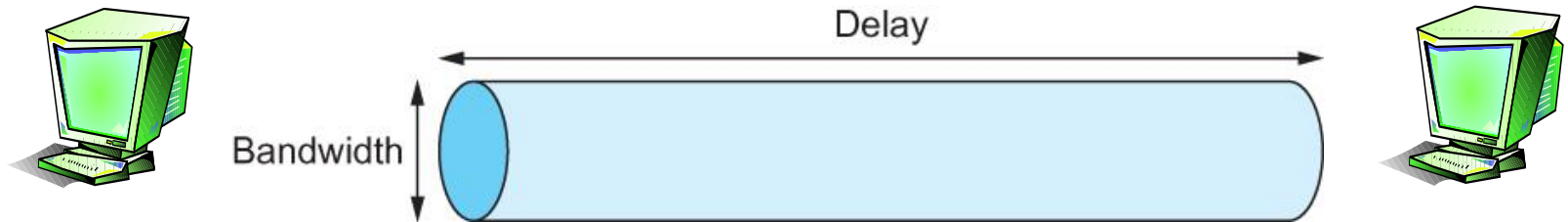
- **等待時間** = 傳播時間 + 傳輸時間 + 排隊時間
- **傳播時間** = 距離/光速
- **傳輸時間** = 傳輸封包大小/頻寬



- 一個位元傳輸 => 傳播時間將變得重要(短資料很快就送完, 但需要長時間才能傳到對方, **資料已送完, 但前導資料還未到達對方**)
  - 傳播時間 >> 傳輸時間
- 大量位元組傳輸 => 頻寬會變得重要(長資料很慢才能送完, **未送完前, 前導資料已到對方**)
  - 傳輸時間 >> 傳播時間

# 延遲 x 頻寬(Delay X Bandwidth)

- 一個在兩個行程 (processes) 間的頻道可視為一個管線
  - **延遲**:管線的**長度**
  - **頻寬**:管線的**寬度**
  - 延遲 x 頻寬 代表可以儲存在管線之中的位元數
  - 舉例來說, 假設延遲是80 毫秒 (ms) 與100 Mbps頻寬
- ⇒  $80 \times 10^{-3} \text{ 秒} \times 100 \times 10^6 \text{ 位元/秒}$
- ⇒  $8 \times 10^6 \text{ 位元} = 8 \text{ M 位元} = 1 \text{ MB 資料.}$



網路可視為一個管線

# 延遲 x 頻寬(Delay X Bandwidth)

---

- 頻寬與延遲的相對重要性取決於網路的應用
  - 對於大檔案的傳輸，頻寬是較為關鍵的因素
  - 對於小訊息的傳輸，延遲是較為關鍵的因素
  - 延遲的變異數 (**jitter**) 也將對網路應用造成影響 (e.g. 影音會議)

# 延遲 x 頻寬(Delay X Bandwidth)

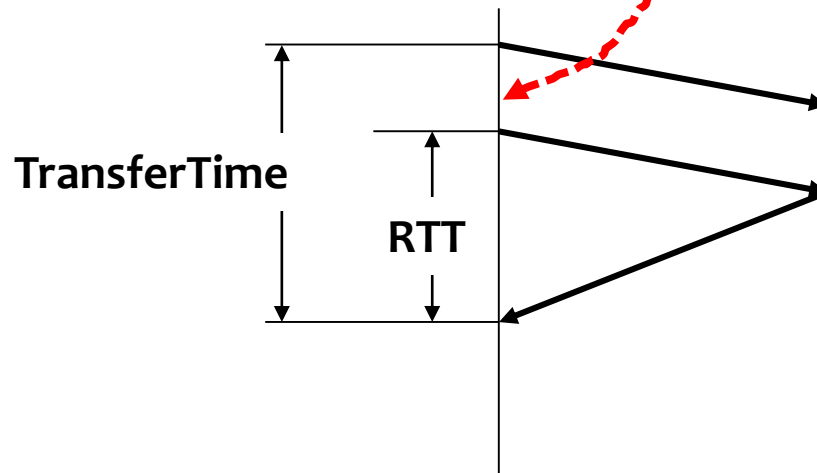
---

- 如果傳送端持續塞滿整個管線，**延遲x頻寬**就是傳送端在第一個位元抵達接收端前必須傳送的位元數
- 另外需要一個單向的延遲作為接受接收端回應的延遲
- 傳送端若無法將傳輸管線塞滿，那就不算是有效利用網路
  - 停下來等對方回應之前如果可以傳送 **延遲x頻寬** 的資料量是最佳的設計

# 吞吐量 (throughput)

## ■ 無限頻寬

- 來回通訊延遲 (RTT, Round Trip Time)
- 吞吐量 = 傳輸封包大小 / 傳輸時間
- 傳輸時間 = 來回通訊延遲 + 傳輸大小 / 頻寬
- 這些參數彼此為相對性關係, 如 1MB 檔案對於 1Gbps 連線看起來就像 1KB 封包之於 1Mbps 連線





# 總結

---

- **多層架構**是一個適用於電腦網路的架構
  - 實體層
  - 資料鏈結層
  - 網路層
  - 傳輸層
  - 會議層/展示層/應用層
- **兩個用於分析電腦網路效能的主要參數**
  - 頻寬 (Bandwidth)
  - 延遲 (Delay)