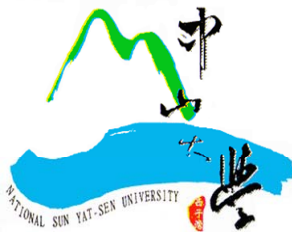

課程名稱：無線網路與行動計算

Wireless Internet and Mobile Computing

- 1) 本課程計分方式
- 2) Internet Overview
- 3) 本學期授課大綱



Instructor: 周孜燦 助理教授
國立中山大學電機系

Email: ztchou@ee.nsysu.edu.tw

課程目標



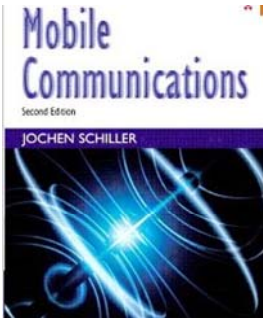
目前最常用的無線網際網路技術：WiFi



目前最常用的無線電話網路技術：GSM

目前全世界最大的二個網路分別為「電話網路」和「網際網路」。我們想探討無線通訊技術的發明對「電話網路」和「網際網路」的影響，特別著重在「通訊協定」（比較像軟體的部分）的介紹，而非無線電波的部分

Reference English Books

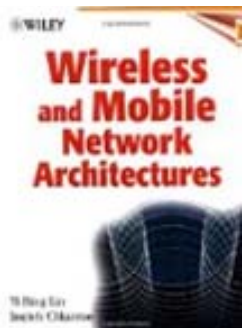


Jochen H. Schiller, *Mobile Communications* (2nd Edition), 2003

Dharma P. Agrawal, Qing-An Zeng, *Introduction to Wireless and Mobile Systems*, 2010



這二本書還不錯，有介紹無線通訊技術對網際網路的影響，和本課程內容比較接近，但卻沒有介紹最新技術（如 WiMAX、cognitive radio networks）



Yi-Bing Lin and Imrich Chlamtac, *Wireless and Mobile Network Architectures*, 2000

這本書由交大林一平教授所撰寫，也滿不錯的，可惜都 focus 在手機系統，並沒有介紹一般常見的 WiFi 技術

Reference Chinese Books



曾煜棋、潘孟鉉、林致宇，無線區域及個人網路：隨意及感測器網路之技術與應用，2007

這本書由交大曾煜棋教授撰寫，滿不錯的，主要是介紹無線隨意網路（沒有基地台的無線網路）。但我們日常所接觸的無線網路大部分是有基地台的無線網路，且此書對網際網路的探討不是很多。



簡榮宏、廖冠雄，無線區域網路，2007

這本書由交大簡榮宏教授撰寫，大概是有史以來介紹 WiFi 技術最好的教科書，可惜也只有介紹 WiFi，沒有介紹其他的無線網路技術（如 WiMAX 及手機系統）。

Prerequisite

<p>課本</p>	<p>總的來說，老師對於市面上所有的教科書都不滿意，所以自己編寫教材（就是上課投影片），因此沒有課本。投影片的內容盡量圖文並茂；此外，由於投影片的文字夠多，相信足以當講義閱讀。</p>
<p>基礎知識</p>	<p>上課內容很像講故事，投影片最多只用到高中數學。我們假設同學修過計算機概論及程式設計的課。</p>
<p>相輔相成的知識</p>	<p>如果同學修過電腦網際網路，那是最棒。如果沒修過，我是覺得沒關係，因為我們會先簡單介紹網際網路。但你如果一直覺得很吃力，那就退選。</p>
<p>考試的難易度</p>	<p>期中考（可能有期末考，也可能沒有期末考）題目大部分出自章節最後的自我評量。只要考前確實有練習（有背答案），相信至少可以有75~80分。</p>

Just Kidding



考題約有 75% 出自章節最後的自我評量，如果你沒有過，真的很對不起我。
拜託不要為了課後輔導而考 0 分！

課程計分方式

只要有點名沒到，
便採用這種計分方式

- ◆ 課堂表現：**20%**
(只要有一次點名沒到，該項評量以**0**分計算)
- ◆ 期中考：**40%**
- ◆ 期末考**或**論文簡報：**40%** (期末考不含期中考的範圍)

如果沒有點名 或 每次點名全到，
便採用這種計分方式

- ◆ 期中考：**50%**
- ◆ 期末考**或**論文簡報：**50%** (期末考不含期中考的範圍)

投影片下載

- ◆ 投影片下載網址 <http://wmi.ee.nsysu.edu.tw>
- ◆ 投影片密碼 **nsysu**
- ◆ 自己注意更新日期（如果有更新，幅度也是非常的小）

這個選項

中山電機所 無線行動網路實驗室

老師 我的對話頁 我的參數設置 我的監視列表 我的貢獻 登出

首頁 課程與教材 研究生須知 實驗室設備 簽到系統 每週論文簡報

首頁

頁面 討論 編輯 歷史 刪除 移動 保護 監視

無線行動網路實驗室

無線隨意網路之最佳動態省電協定與技術

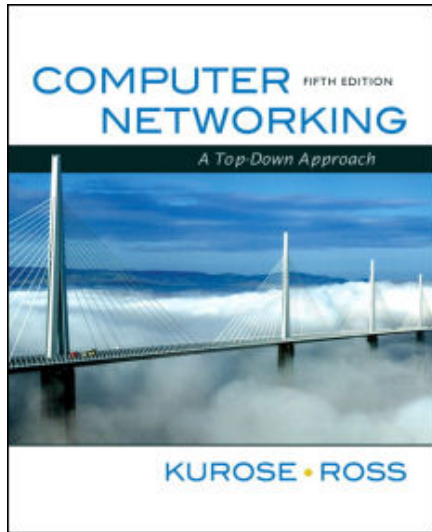
指導教授：周孜燦 研究生：賴嘉緯 研究助理：林鈺翔

技術簡介

無線行動隨意網路（俗稱 MANET）是由一群行動裝置所構成，它們是直接或間接透過其他的行動裝置來通訊，而沒有任何基地台的輔助。由於行動裝置是以電池作為電力來源，然而電池

- 歡迎光臨
- 公告
- 教授簡介
- 已畢業成員
- 教學教材
- 研究生學習態度
- 如何撰寫論文

References for this Chapter (Internet Overview)



原文書



中譯本

James F. Kurose and Keith W. Ross, *Computer Networking: A Top-Down Approach*, 5th Edition, 2009

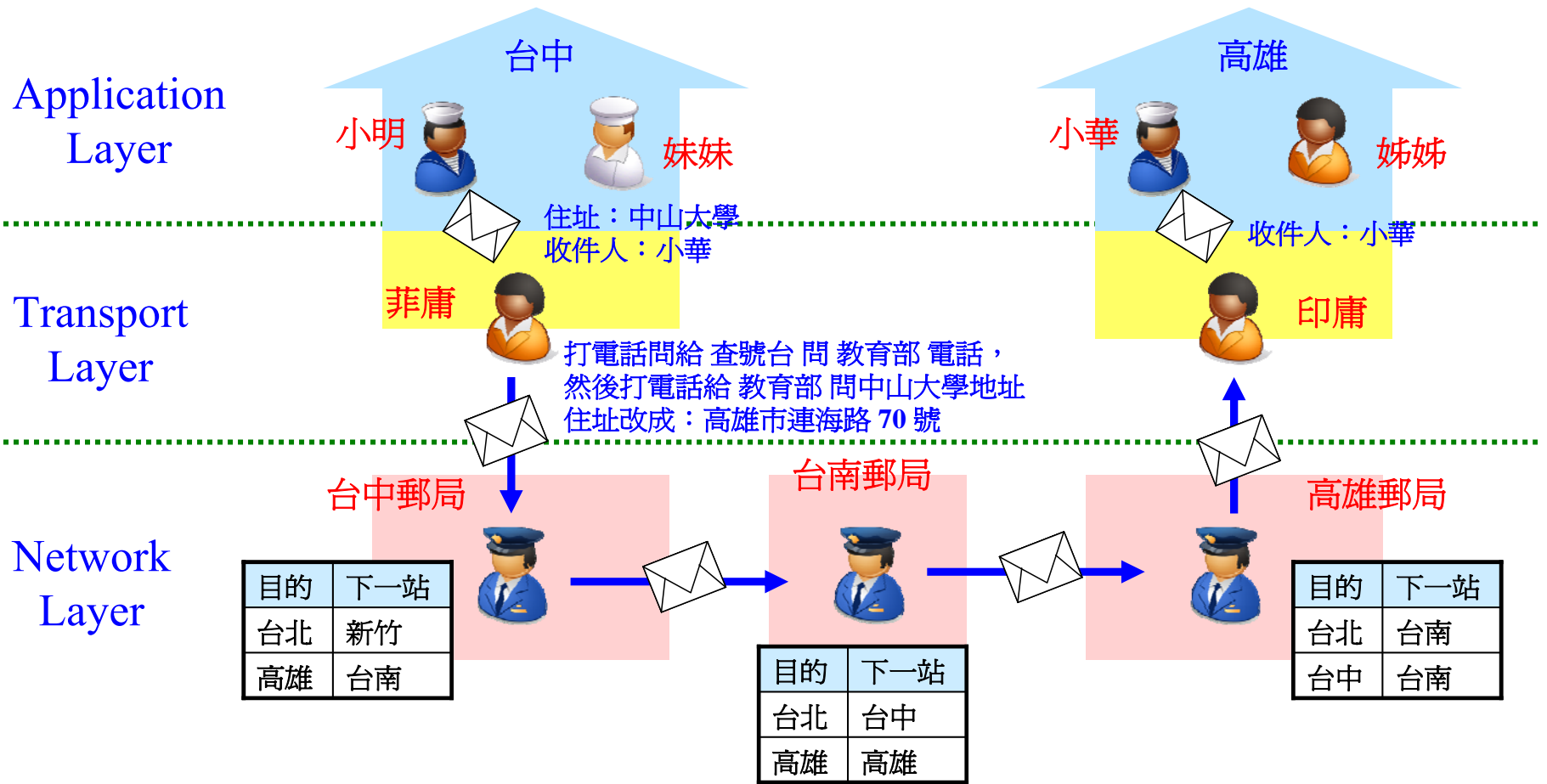
→ 這本書的第一章寫的比較差，但之後的所有章節堪稱有史以來介紹網際網路最好的書籍。

Quick Internet Overview



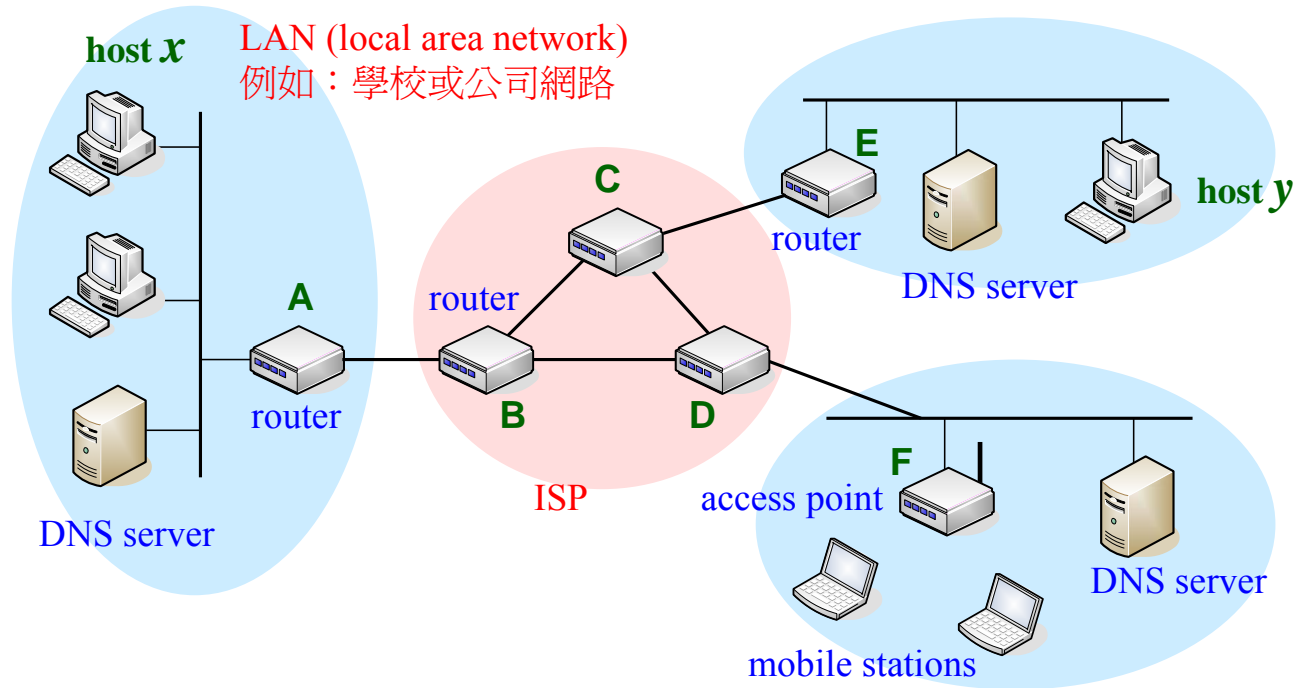
隨著網路技術的進步，你可以透過瀏覽器到處欣賞別人所架設的網站。或者透過搜尋站，找到你想要的資料。當你在瀏覽上的上方輸入 <http://www.ee.nsysu.edu.tw>，便會秀出中山電機系的網頁，你有沒有想過這一切是如何發生的？

Analogical Solution in Postal System



如果住在台中的小明想寄一封信給住在高雄的小華，他應該不會搭乘高鐵親自拿信給小華，而是透過很多人的**分工合作**，**分層負責**來達成目的。同樣地，當你想瀏覽某個網頁，這一切也是透過分工合作完成的。

Components of Internet



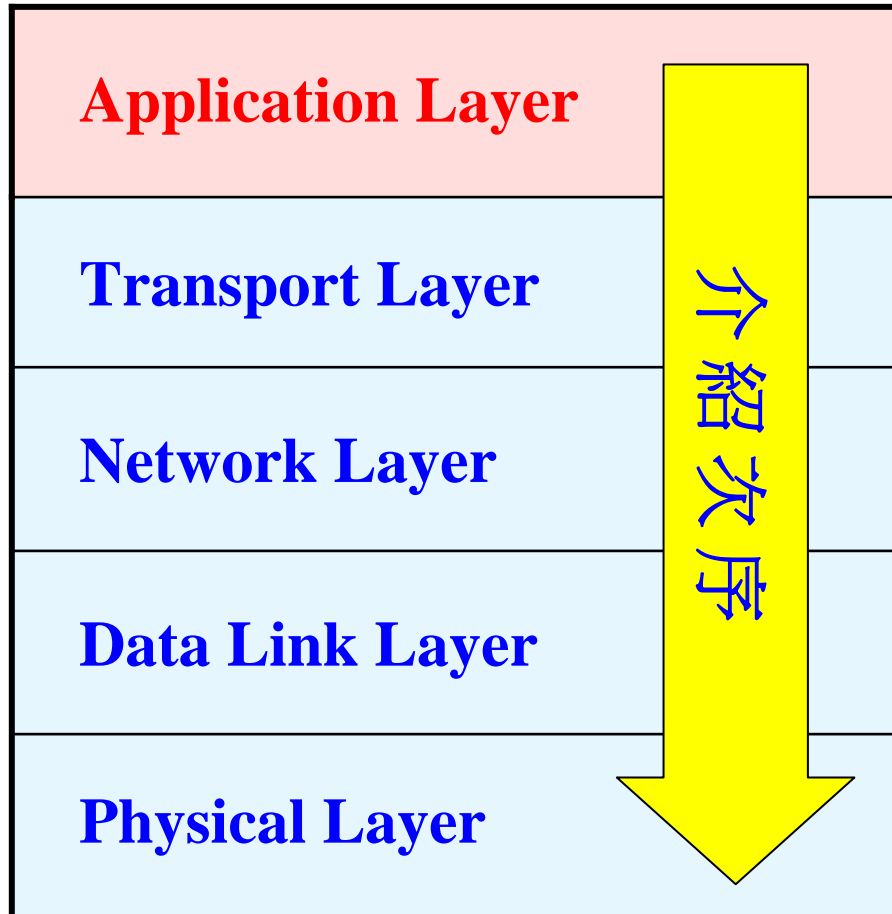
Internet 就是連結很多網路所構成的網路。類似郵政系統，當 Internet 裡頭的電腦 x （稱為 **host**）想送資料給 y ，會先透過 DNS (domain name system) server 查詢 y 的 address，然後透過 router A、B、C、E 將資料送給 y 。DNS server 相當於菲傭或查號台的角色，負責查出 y 的 address。大部分的 router 由 ISP（Internet service provider，如：中華電信）所擁有，只負責資料的轉送，相當於郵局或郵差的角色

TCP/IP Reference Model : Defined by IETF

Layer	簡稱	代表性技術	國際標準的制訂機構
Application Layer	L5	HTTP	IETF
Transport Layer	L4	TCP, UDP	IETF
Network Layer	L3	IP	IETF
Link Layer	L2	Ethernet (802.3)	IEEE
Physical Layer	L1	同軸電纜	IEEE

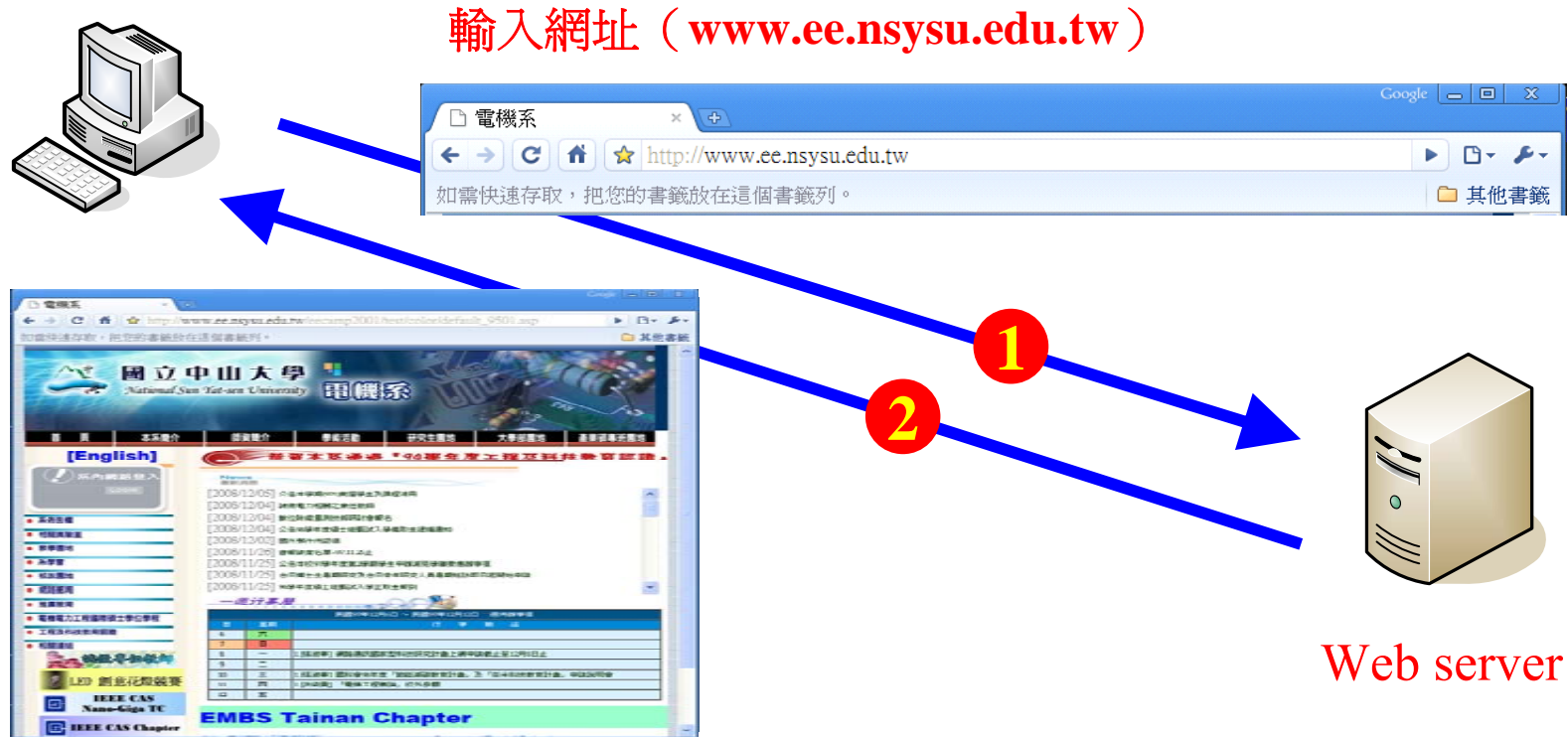
1960 年代，美國國防部首先開啓網路的研究。爲了串連美國主要大學的電腦設備，1974 年 Vinton G. Cerf 與 Bob Kahn 發表 TCP（transmission control protocol）及 IP（Internet Protocol）技術，用以串連個校際間的電腦設備，這就是網際網路（Internet）的雛形。1982 年起，民間組織 IETF（Internet Engineering Task Force）致力於 Internet 細部協定的開發與實作。上圖所示爲 IETF 所訂的分層負責架構，又稱爲 TCP/IP reference model。

Introduction to Internet via Top-Down Approach



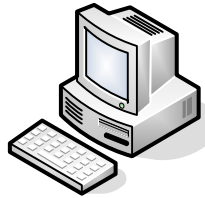
TCP/IP Reference Model

How to Get a Web Page ? (1/3)

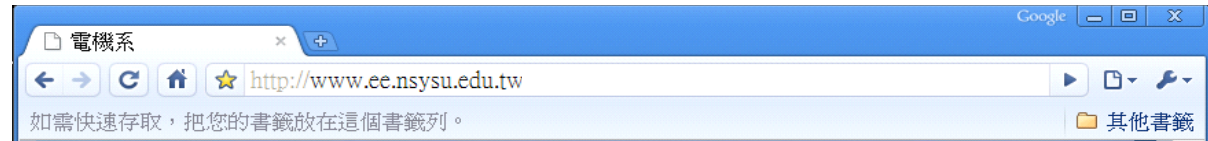


Application layer 的目的就是讓使用者享受網際網路所提供的服務。例如：你可以透過瀏覽器看中山電機系的網頁。Application layer 就是規範網路應用程式（例如：瀏覽器）所需遵守的規範（稱做 protocol，中文翻成「通訊協定」，這個字源自於外交禮儀，表示大家都必須遵守的規範）

How to Get a Web Page ? (2/3)



URL (uniform/universal resource locator)



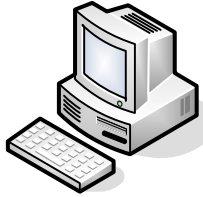
使用 **HTTP** 通訊協定發出控制命令，
要求傳回網頁的內容



Web server

1. 首先，每個網頁都有一個網址（正式學名稱爲 URL），你必須在網址的地方輸入 URL，你的瀏覽器才知道要幫你去哪兒抓取網頁。
2. 你的瀏覽器會使用 HTTP（Hypertext Transfer Protocol）來傳送控制命令給中山電機系 web server。最後 web server 會回傳網頁的內容給你。

How to Get a Web Page ? (3/3)



```
<HTML>
<HEAD>
<TITLE>電機系</TITLE>
..... (省略) .....
</HTML>
```

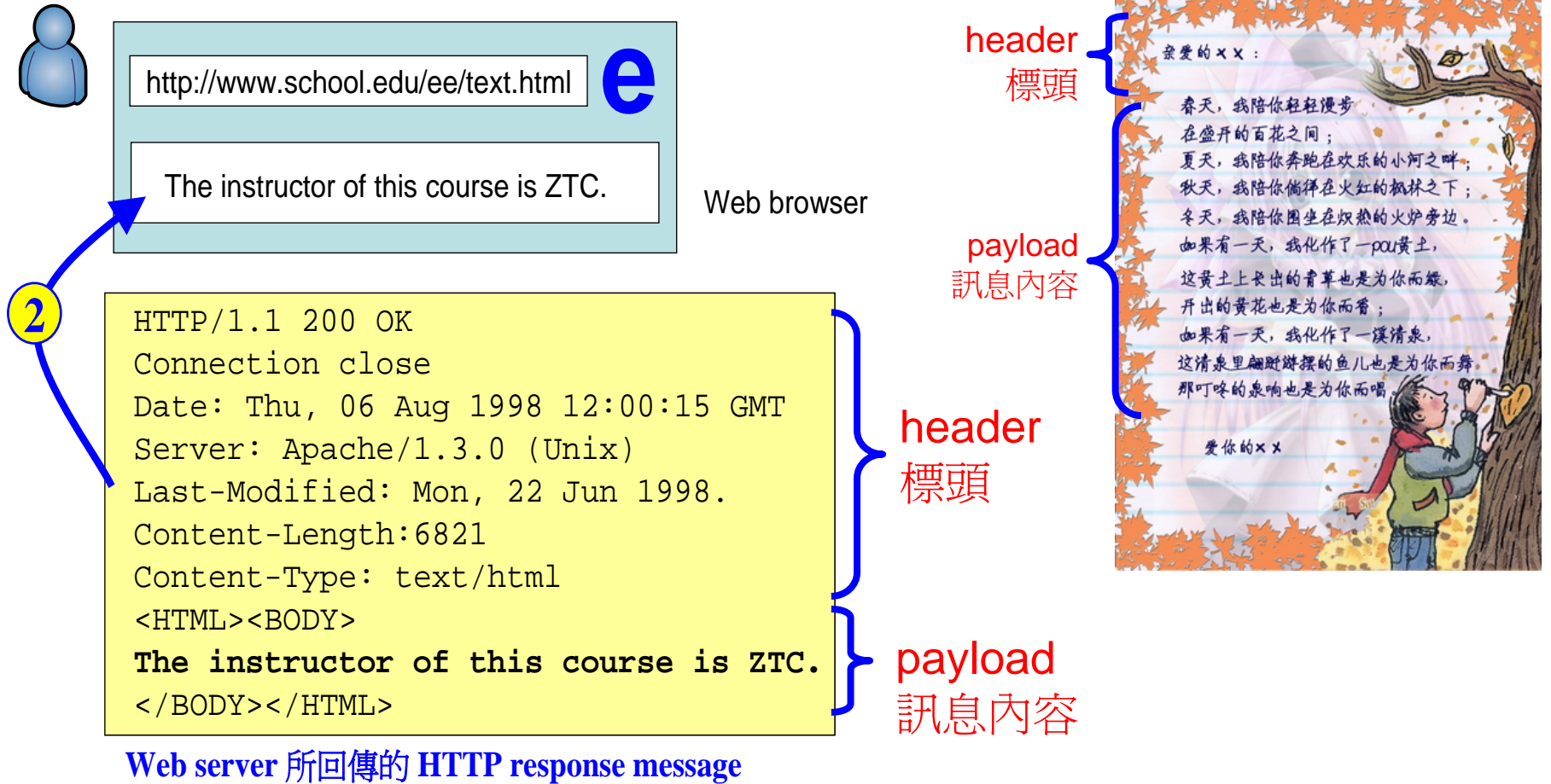
回傳網頁的內容（使用 HTML 撰寫）



Web server

網頁的內容是由 HTML（HyperText Markup Language）所撰寫。當你的瀏覽器收到網頁的內容之後，就會解析語法，並按照 HTML 的規範將網頁的內容呈現出來。

HTTP Response Message



Web server 所回傳的 HTTP response message

How to Know the IP Address of Web Server ?



HTTP 會產生 application layer 的 message，如下：

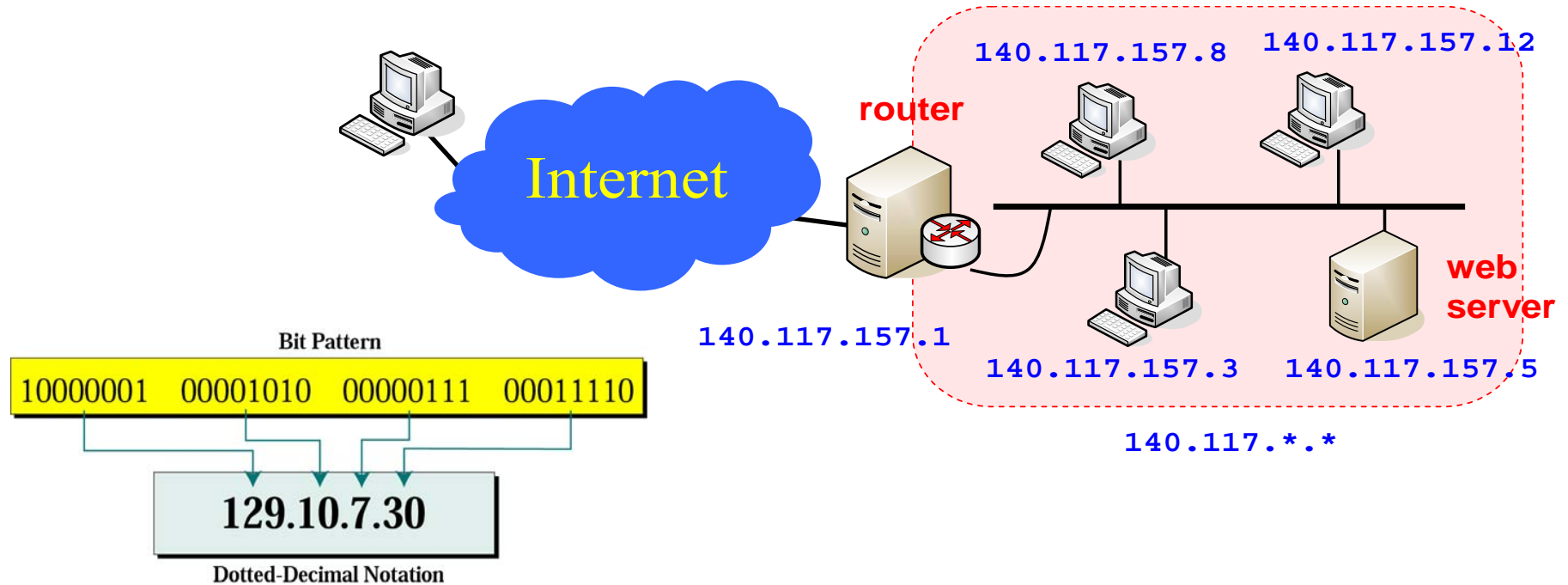
```
GET /ee/text.html HTTP/1.1  
Host: www.school.edu  
User-agent: Mozilla/4.0  
Connection: close  
Accept-language: fr
```

如何得知
www.school.edu
的 IP address?

HTTP request message

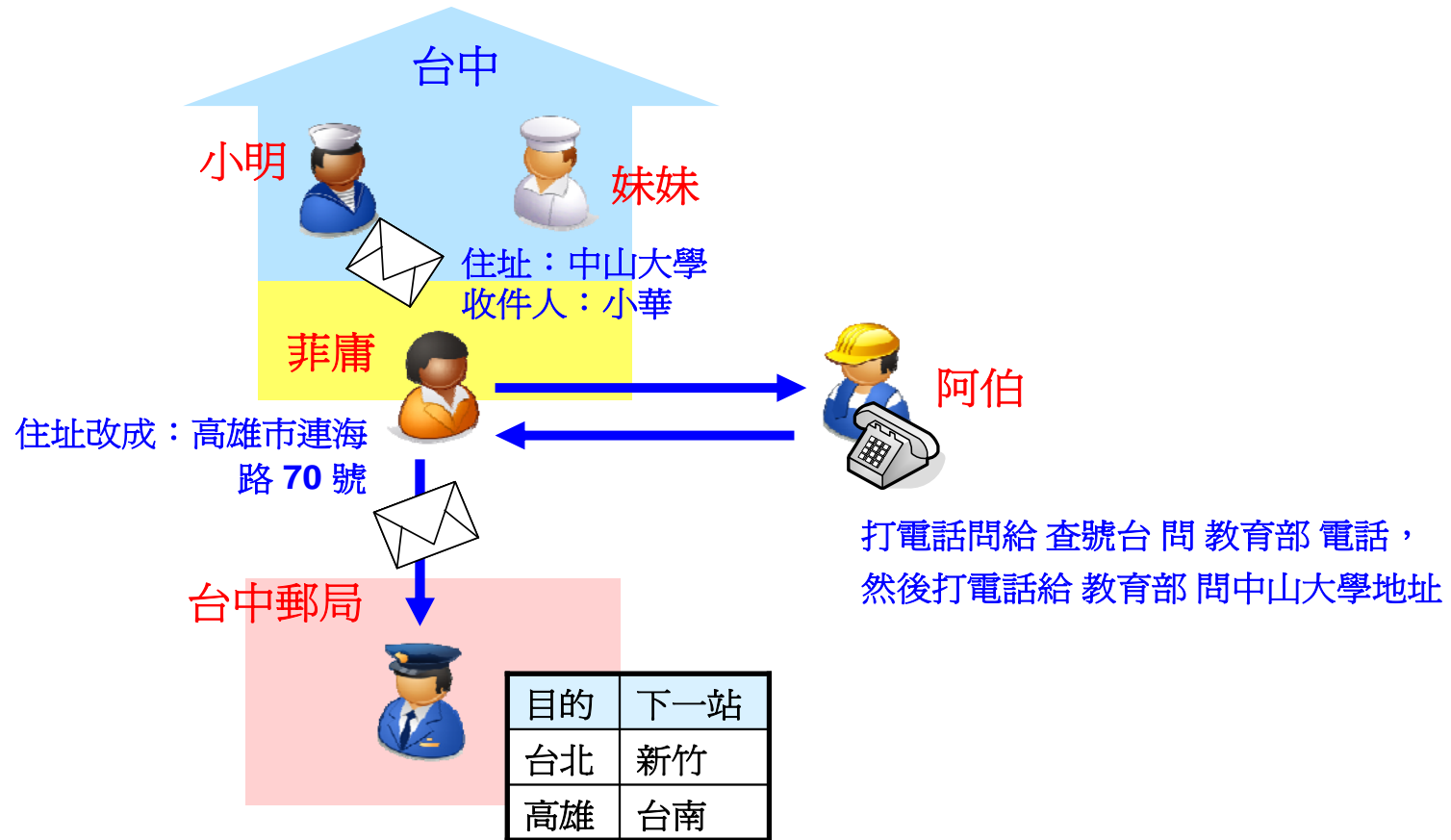
當你輸入網址 `www.school.edu/ee/text.html` 時，你的瀏覽器會產生一個 HTTP request 訊息，傳送給 web server (`www.school.edu`)，要求傳回網頁的內容。但你的瀏覽器怎麼知道 **web server** 在哪兒？或者你的瀏覽器怎麼知道 **web server** 的 IP address 是多少？

Aside : IP Address



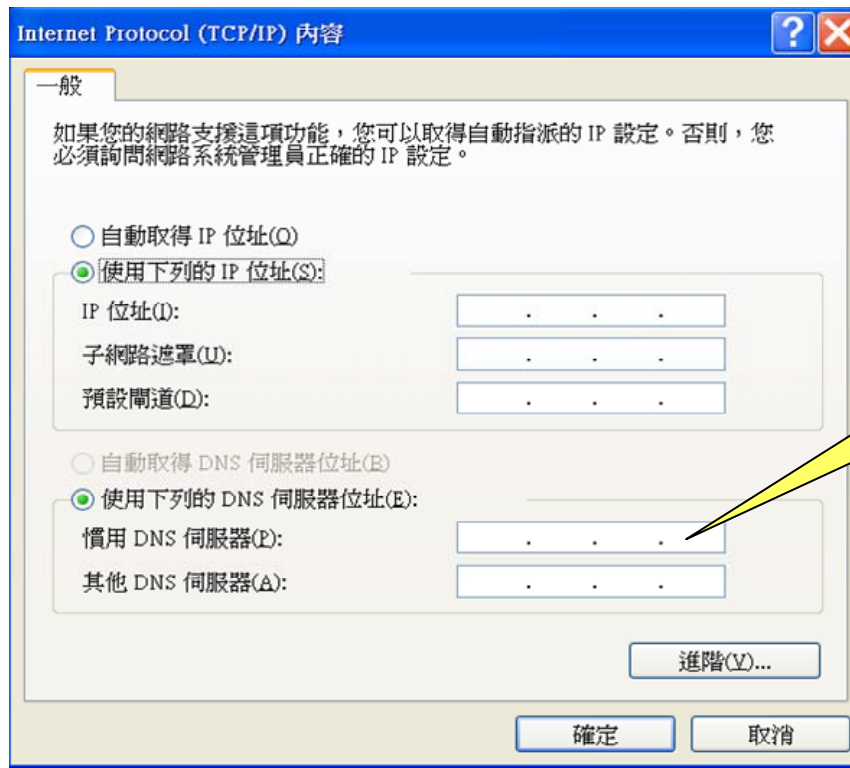
如果你要寄郵件給周老師，你會寄到中山大學。就像人類每個居住地方都有地址一樣，**每台連上網的電腦都需要有 32 bits 的 IP address (IP 是 Internet Protocol 的縮寫)**。就像人類的地址是由政府所分配，IP address 是由主管機關 Internet Assigned Numbers Authority (IANA) 所分配。需要 IP addresses 就必須跟 IANA 申請。例如中山大學申請到的 addresses 為 140.117.*.*

Analogical Solution in Human Query System



在人類的世界裡頭，菲庸如何得知中山大學的地址？她可以就近詢問大樓管理員阿伯；如果阿伯不知道，他一定知道查號台（假設大家都知道）阿伯可以透過查號台詢問教育部，最終可以問到中山大學的地址。

DNS (Domain Name System) Server



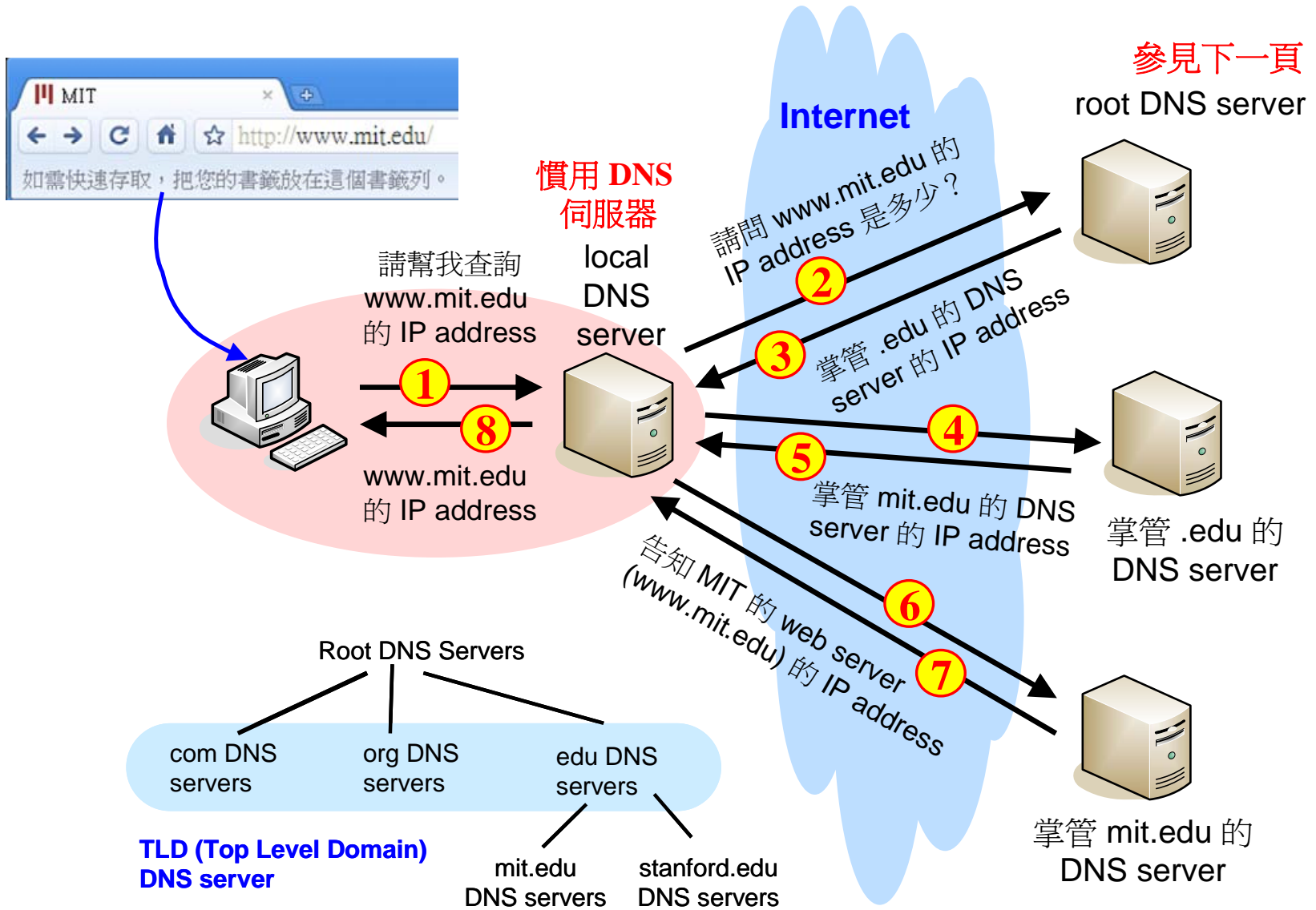
慣用 DNS 伺服器的
IP address 是多少，
可以去問系辦

想上網的使用者必須填寫「慣用 DNS 伺服器」的 IP address。

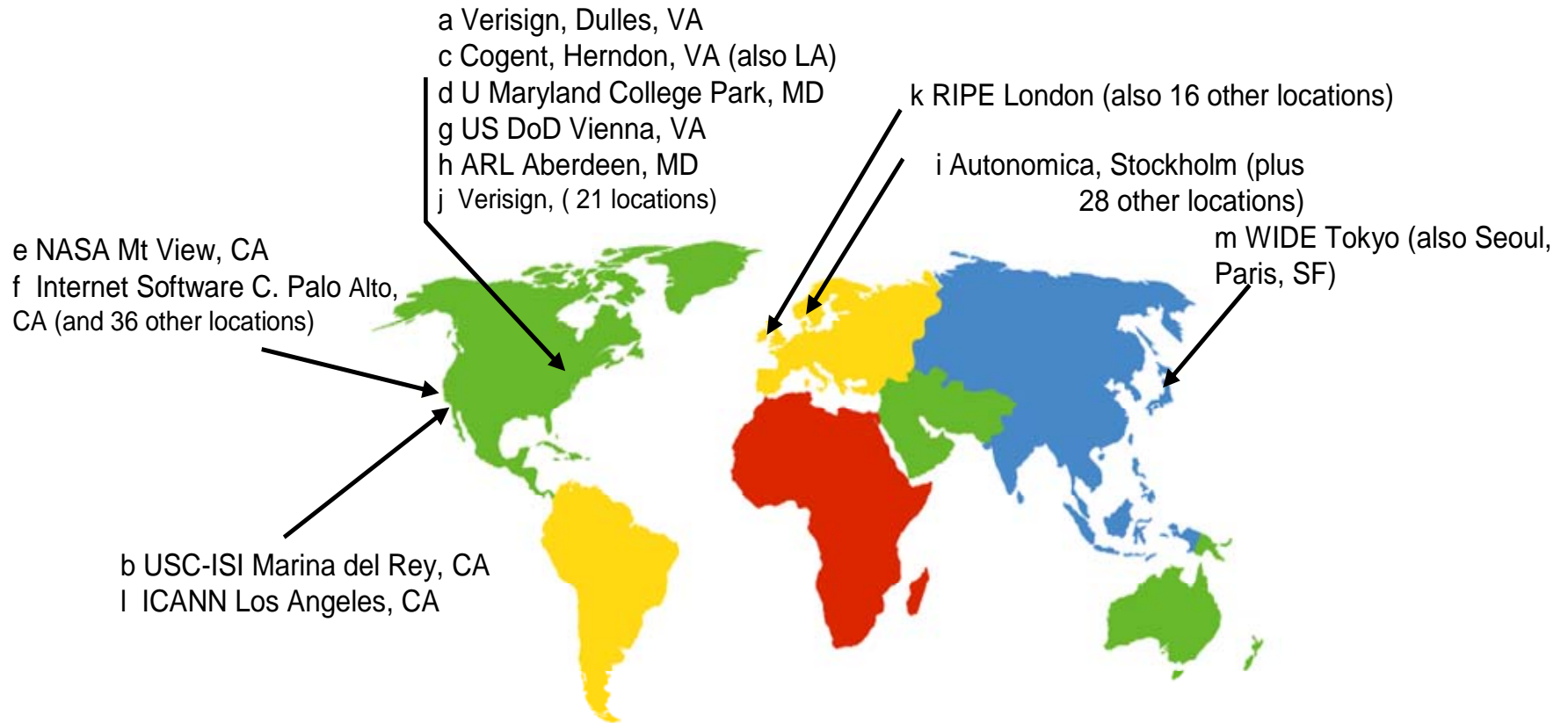
這個慣用 DNS 伺服器（**local DNS server**）就像大樓管理員阿伯，

可以讓你詢問 IP address。如果 local DNS server 不知道，他可以幫你詢問
查號台（**root DNS server**），最後一定可以問到 web server 的 IP address

How DNS Works ?

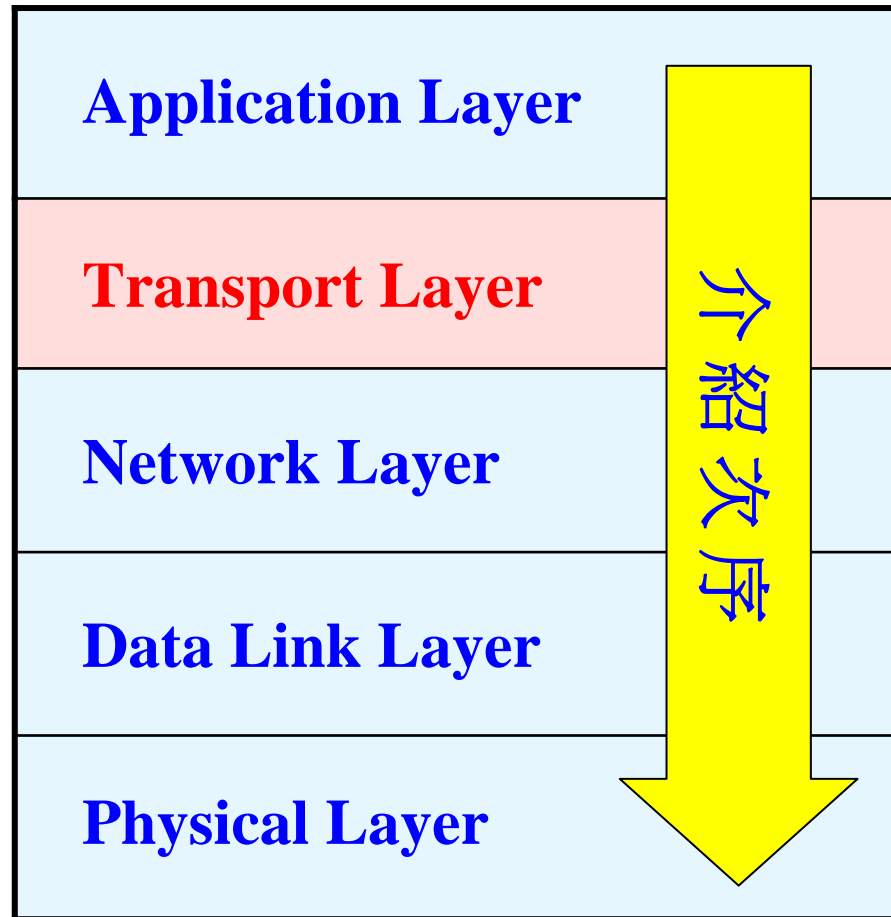


Root DNS Servers



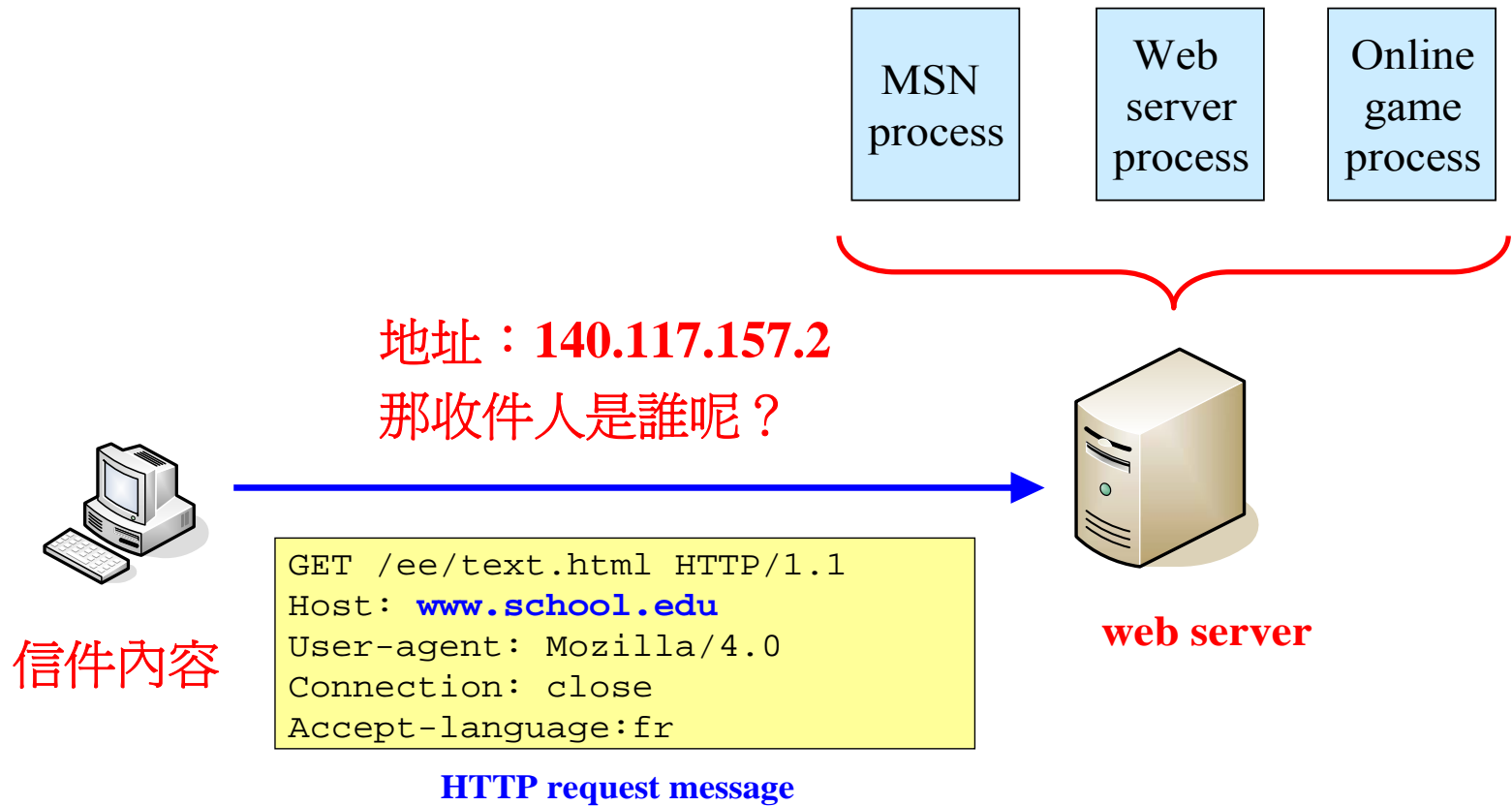
全世界大約有 20 幾台 root DNS servers，大部分在美國。每一台「慣用 DNS 伺服器」都必須知道 root DNS servers 的 IP addresses。

Agenda : Transport Layer



TCP/IP Reference Model

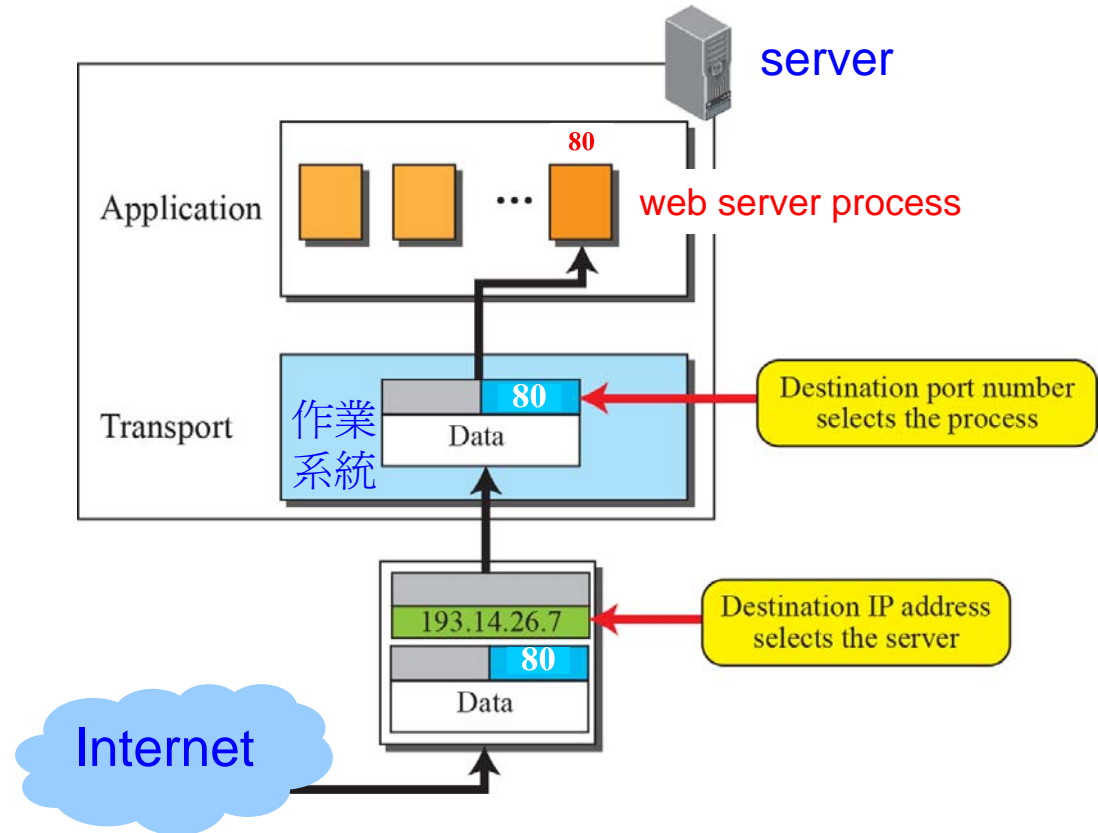
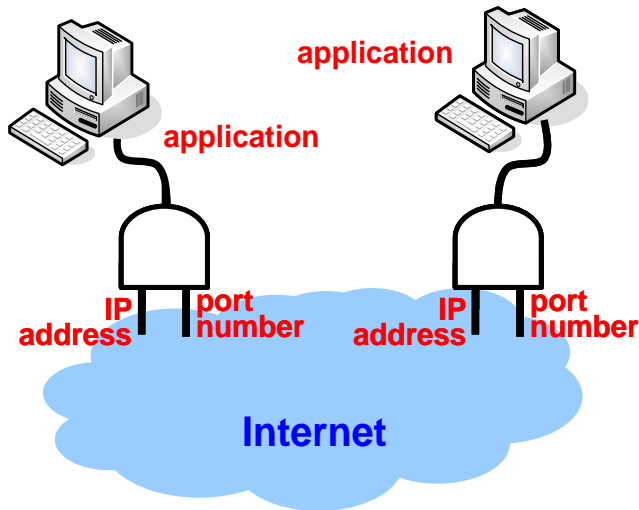
Which Process Should Handle the HTTP Request ?



一旦你的瀏覽器得知 web server 的 IP address，就可以傳送 HTTP request 訊息給 web server。但是 web server 上面可能有好多 processes 正在執行，web server 如何得知這個訊息是要傳給哪一個 process 呢？

The Purpose of Port Number

web service 的 port number 是眾所周知的 80



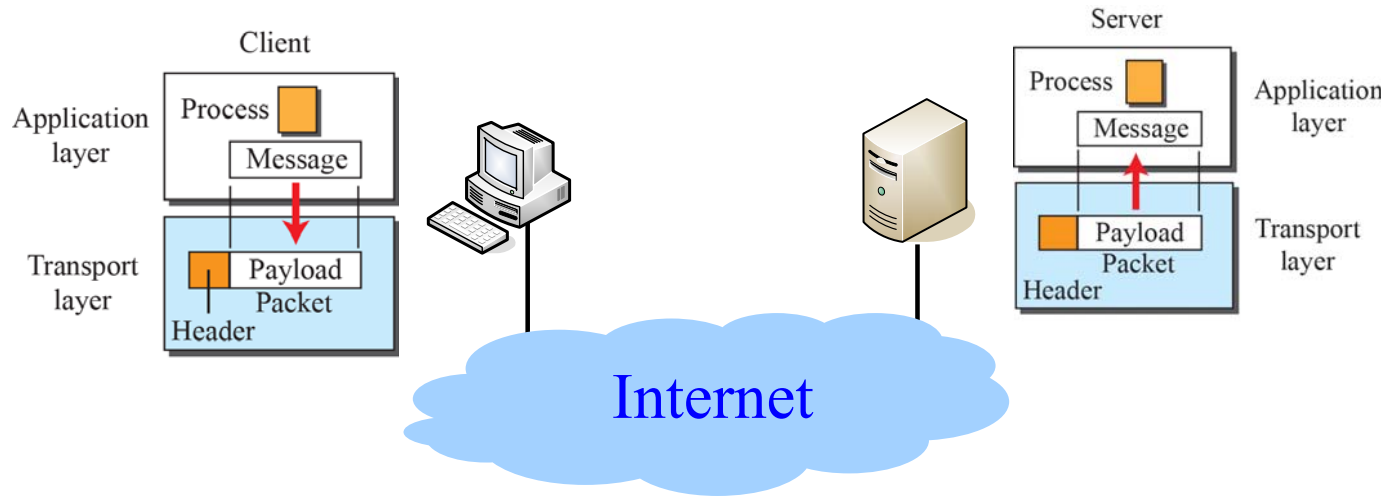
IP address + port number (合起來稱為 socket) 可以用來確定 HTTP request 這個訊息要傳給網際網路裡頭的哪一台電腦裡頭的哪一個正在執行的程式 (稱為 process)。

Some Well-Known Port Numbers

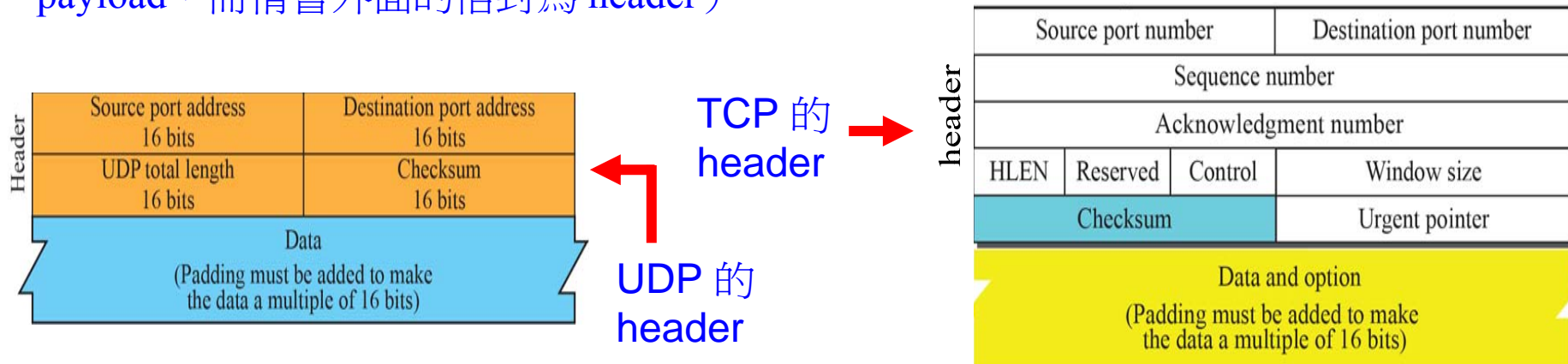
<i>Port</i>	<i>Protocol</i>	<i>Description</i>
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
20	FTP, Data	File Transfer Protocol (data connection)
21	FTP, Control	File Transfer Protocol (control connection)
23	TELNET	Terminal Network
25	SMTP	Simple Mail Transfer Protocol
53	DNS	Domain Name Server
67	BOOTP	Bootstrap Protocol
79	Finger	Finger
80	HTTP	Hypertext Transfer Protocol
111	RPC	Remote Procedure Call



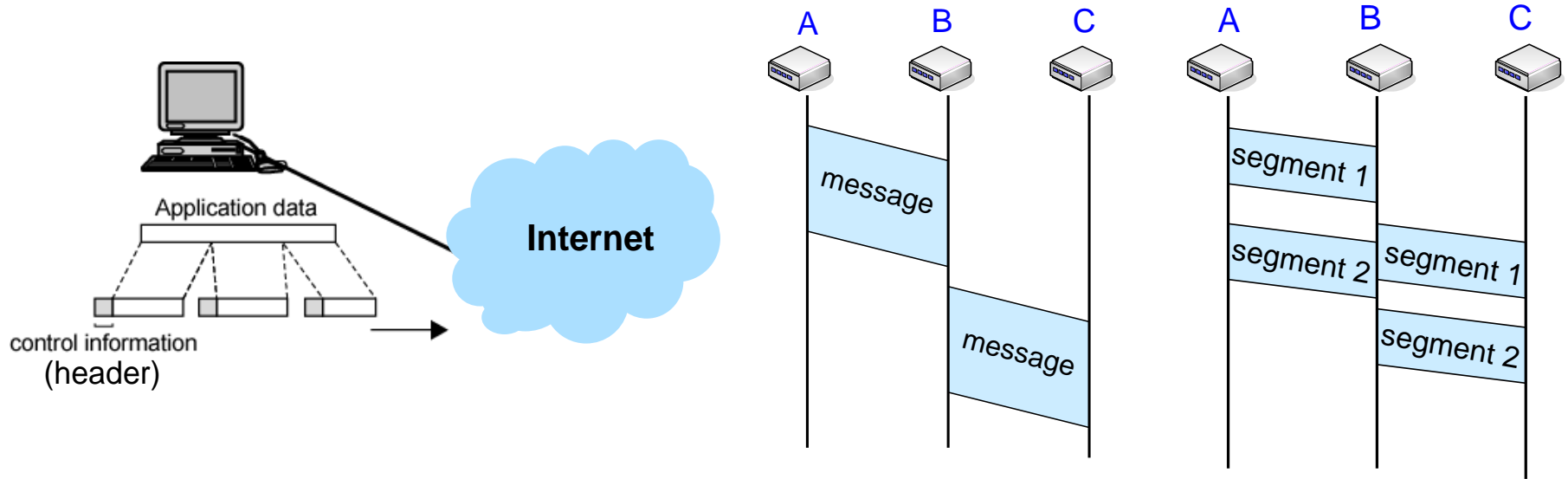
Aside : Headers of Transport Layers



就如同 application layer 所產生的 message 包含有 header 和 payload 的部分；一樣，transport layer 所產生的東西（稱為 segment）也包含 header 和 payload 的部分，其中 payload 的部分為 application layer 的 message（以人類的例子來說，整封情書變成 payload，而情書外面的信封為 header）

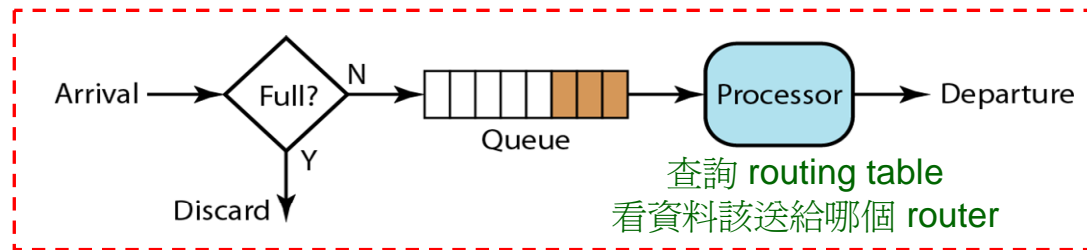
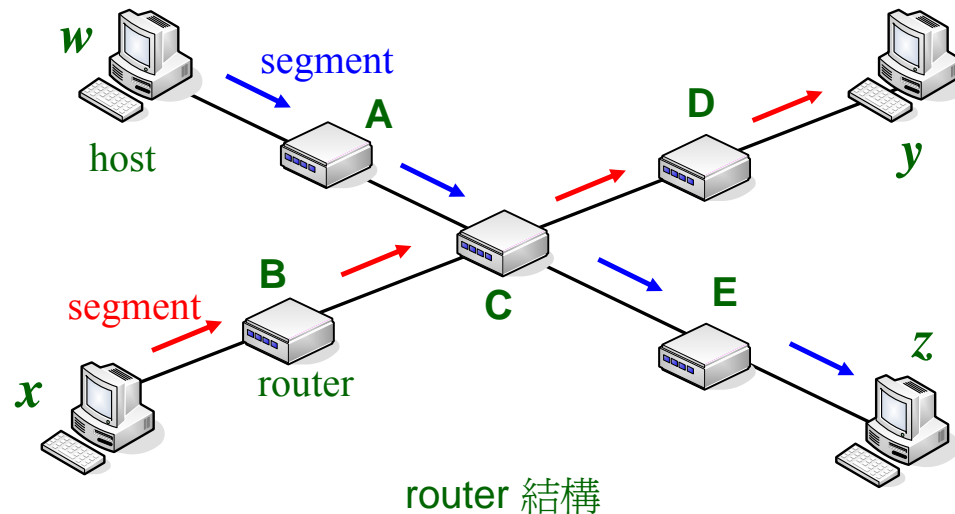


The Purpose of Fragmentation



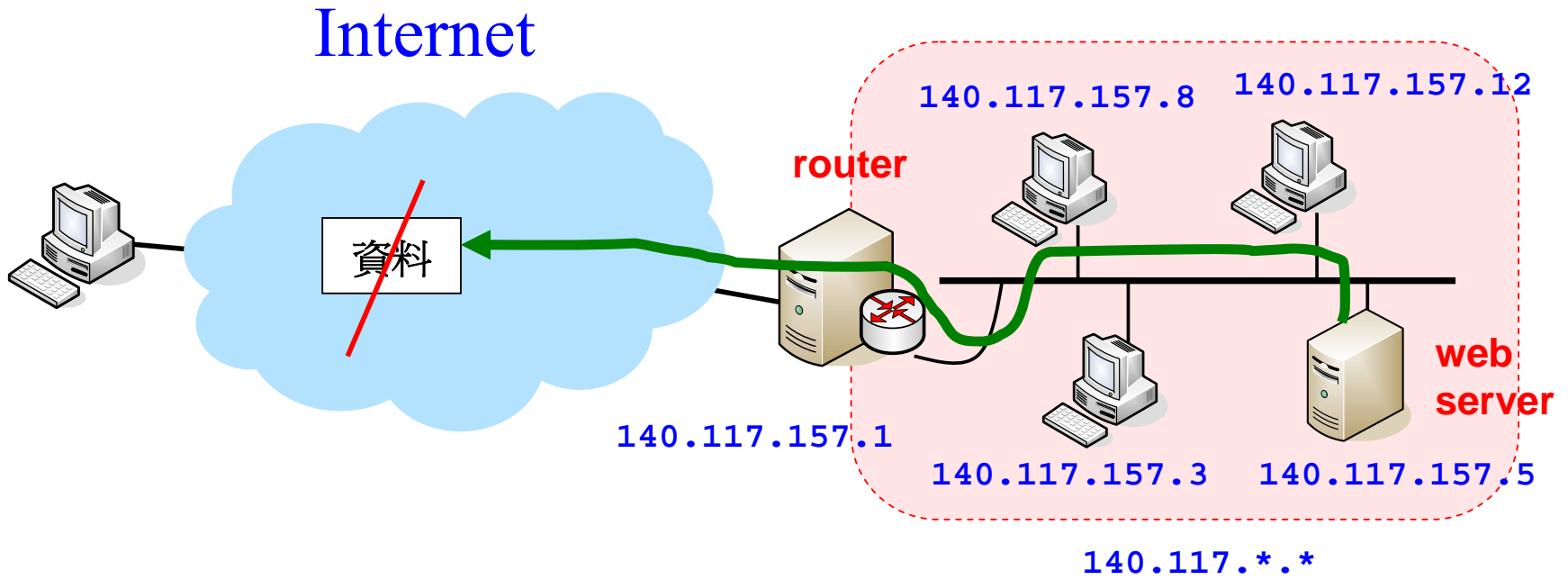
除了 application layer 之外，每個 layer 都可以適當地切割上層的資料，好處有二：(1) 資料在傳輸的過程中難免有干擾。如果 message 切割成 2 個 segments，且只有 segment 1 受到干擾而損毀，那麼只需重送 segment 1，並不需要整個 message 全部重送。(2) 如上方右圖顯示，切割成 2 個 segments 之後，利用 pipeline 的傳輸方式，可以較快地將整個 message 從 A 送到 C。

Data May Be Lost in a Congested Router



假設 w 和 x 分別一直傳送資料 (segments) 給 z 和 y 。當 router C 收到 w 和 x 的 segments 之後，需要一點時間來判斷這些 segments 該轉送給 D 還是 E ？如果 x 和 w 產生 segments 的速度太快，導致 router C 的 buffer 滿了 (發生 congestion)，此後 C 便會將最新收到的 segments 加以丟棄 (注意：收下來之後才丟棄)

How to Deal With the Lost Data ?



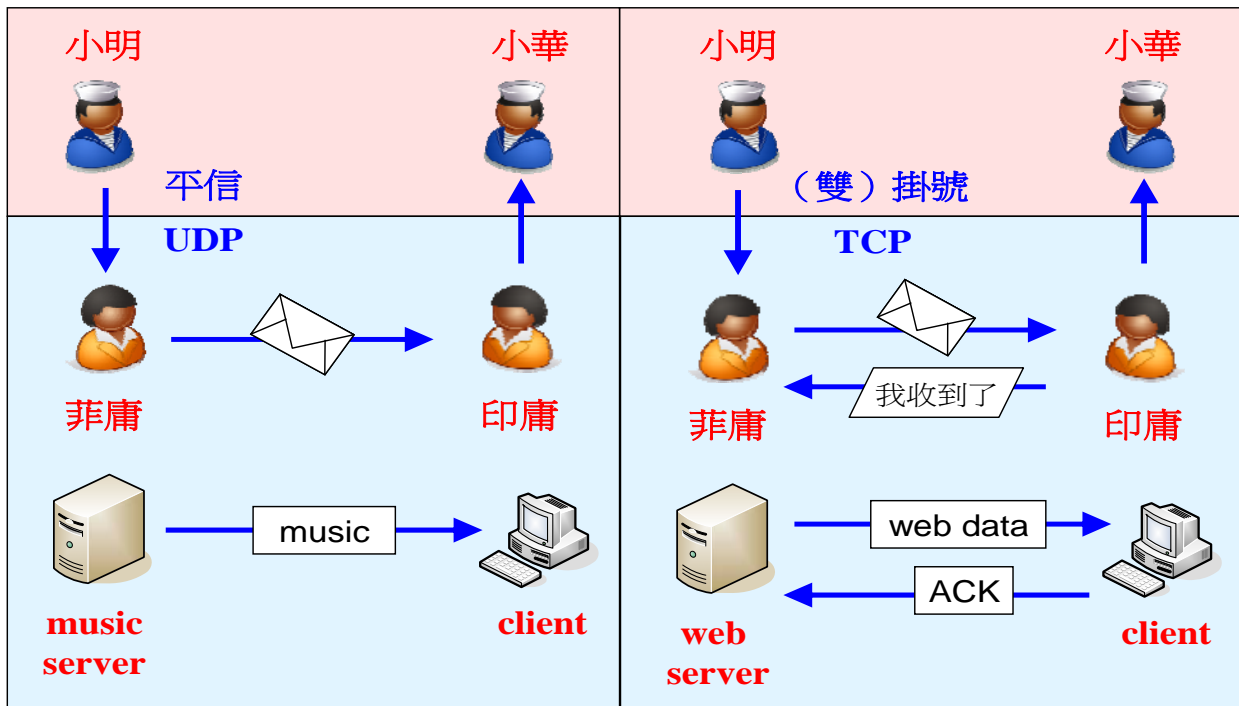
若 web server 將 HTTP response 訊息回傳給使用者的途中發生資料遺失（被 congested router 所丟棄）該怎辦？

Characteristics of Application Layer Data

性質	應用程式	對 delay 要求	對 data loss 的容忍度	適用協定
影音	YouTube	delay 必須很低	可容忍	UDP
資料	email	delay 高無所謂	不可容忍	TCP

並不是所有的應用程式對於資料發生遺失都很在乎，例如 video 的資料發生遺失，頂多畫面出現 lag 的現象。但有些應用程式無法容忍資料發生遺失，例如 email。為此，**IETF** 制訂了二種 **transport layer** 的 **protocols**：傳輸資料可靠度低的 **UDP (User Datagram Protocol)** 以及 傳輸資料可靠度高的 **TCP (Transmission Control Protocol)**。當 **application layer** 採用 **UDP** 方式來傳送資料時，這些資料就算遺失也無所謂（**UDP** 什麼事也不做，就只是幫 message 添加 header 而已）。因此我們將特別關注 **TCP** 是如何運作的。

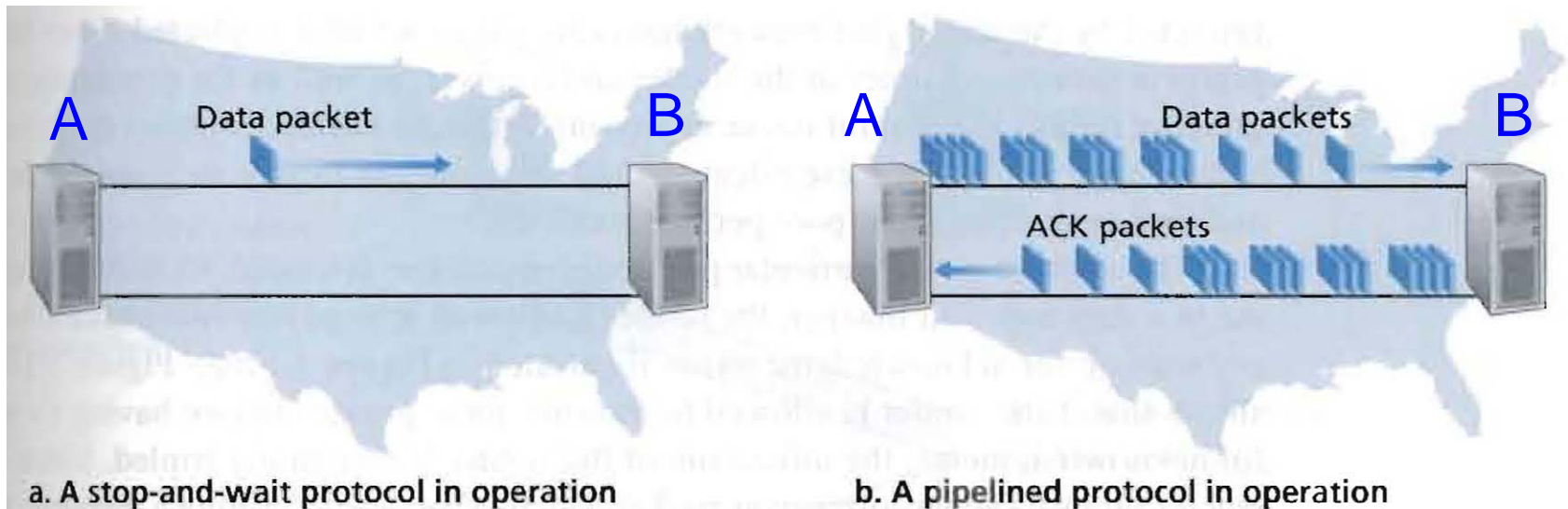
Two Transport-Layer Protocols : TCP and UDP



就像人類寄信對方的時候，可以選擇是否要採用可靠度較高的郵寄方式，例如：

(雙)掛號。若採用 **TCP**，那麼當資料有遺失時，就必須重送。但是菲庸如何得知信件遺失？菲庸可以設定 `timeout` 的值。若菲庸等很久（超過 `timeout` 時間）都沒收到來自印庸的 **ACK** (`acknowledge`)，就假設 `segment` 發生遺失，必須重送

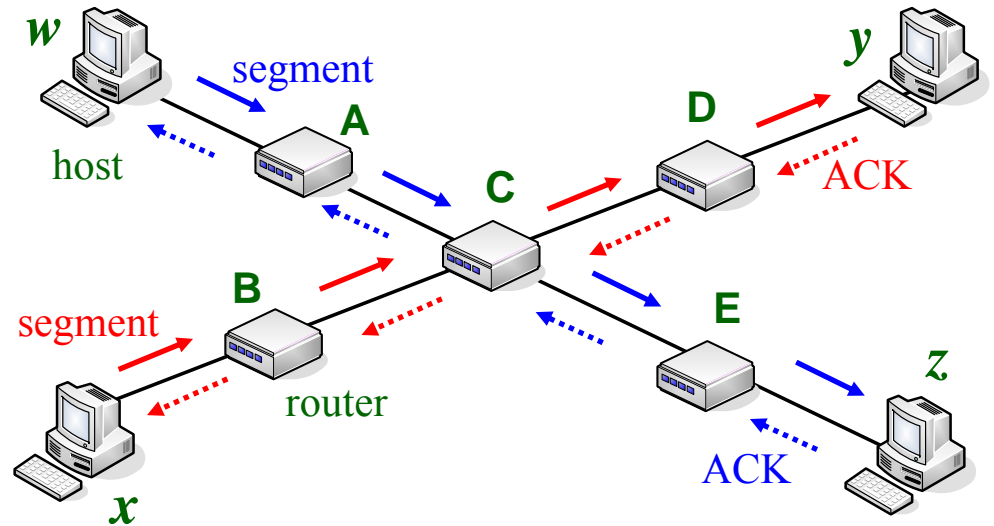
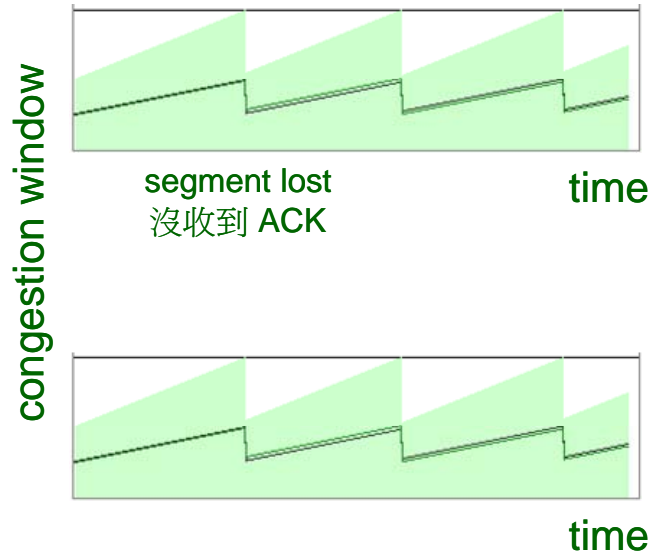
Congestion Window for Pipelined Transmission



假設 host A 持續有 segment 必須傳給 host B。假設 A 收到 B 的 ACK 之後才能送下一個 segment 給 B，那麼 end-to-end throughput（單位時間所能傳送的資料量）將會很低，因為浪費很多時間在閒置（等待 ACK）。

所以我們希望 host A 能在沒有收到 host B 的 ACK 之前，能夠傳送 $n \geq 1$ 個 segments 給 host B。我們稱 n 的值為 congestion window。

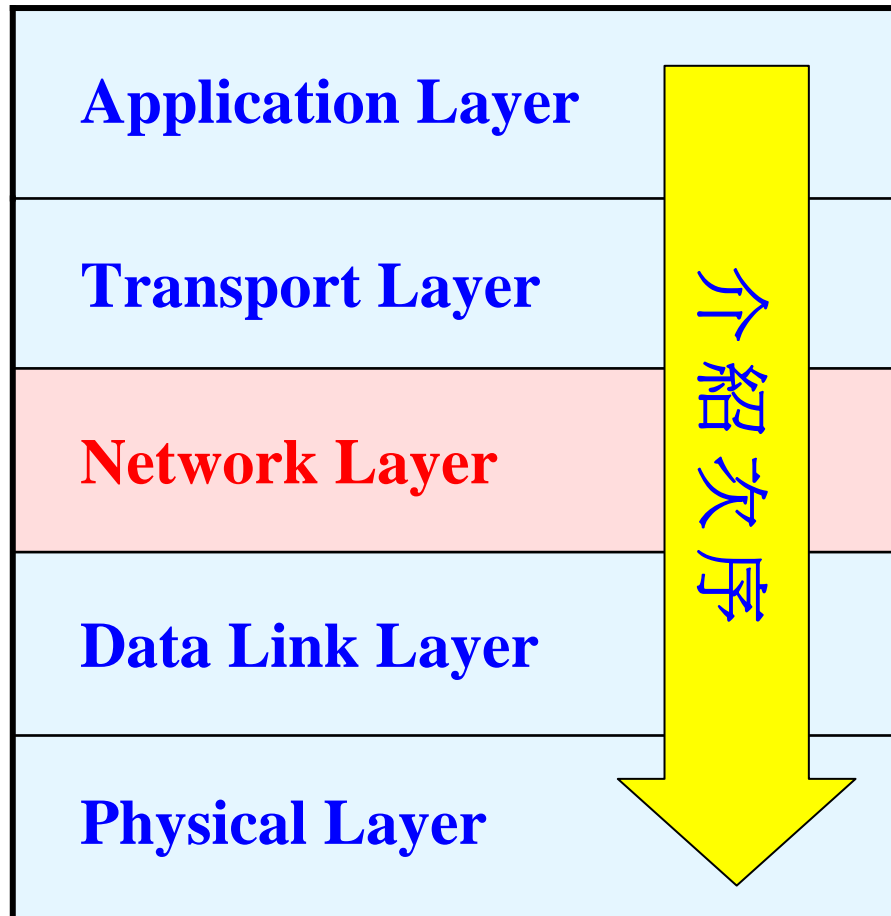
TCP Congestion Control



當 n 的值 (congestion window) 的值太小，會導致 source host 幾乎處在閒置的狀態；當 n 的值太大，可能導致 source host 到 destination host 中間某個 router 發生擁塞，使得 segments 在中途被丟棄。因此 TCP 必須動態調整 n 的值。

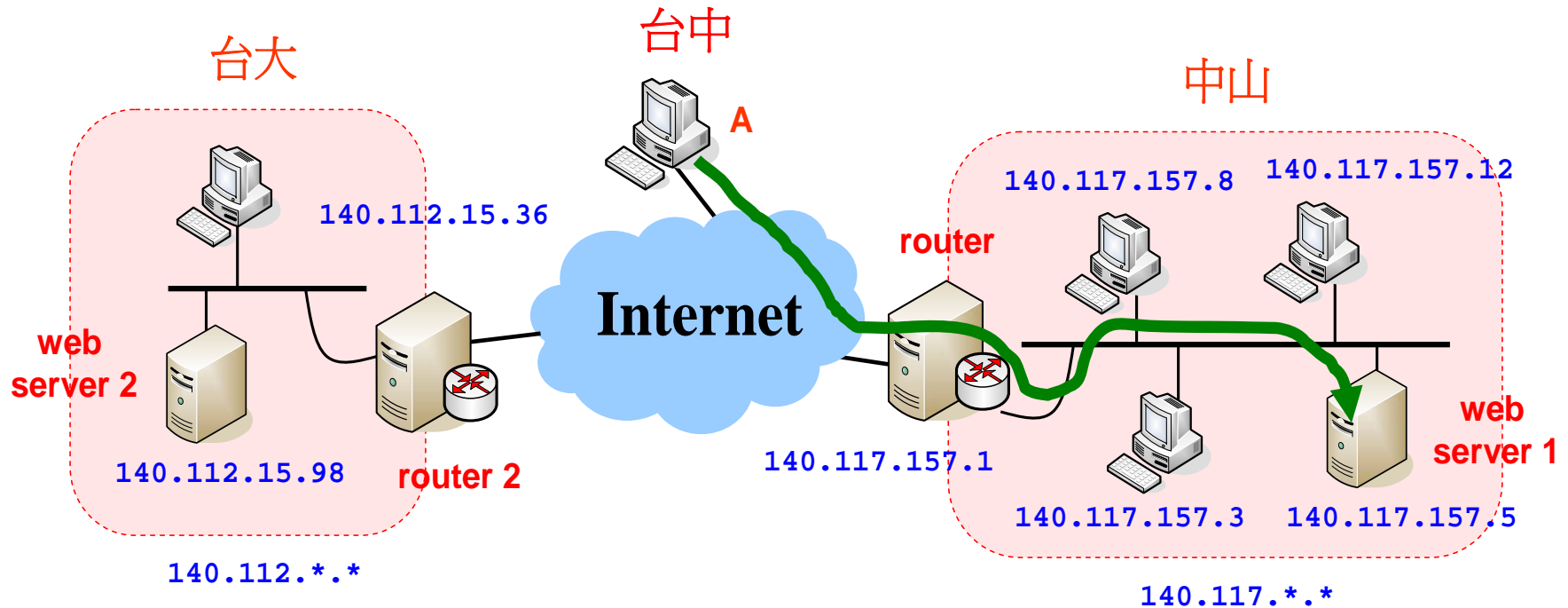
TCP 擁塞控制的原理：若 source host 順利收到來自 destination host 的 ACK， n 的值便慢慢加大；若 source host 無法順利接收 ACK， n 的值便快速下降。所以 congestion window 會呈現鋸齒狀。註：更詳細的細節，以後會再介紹

Agenda : Network Layer



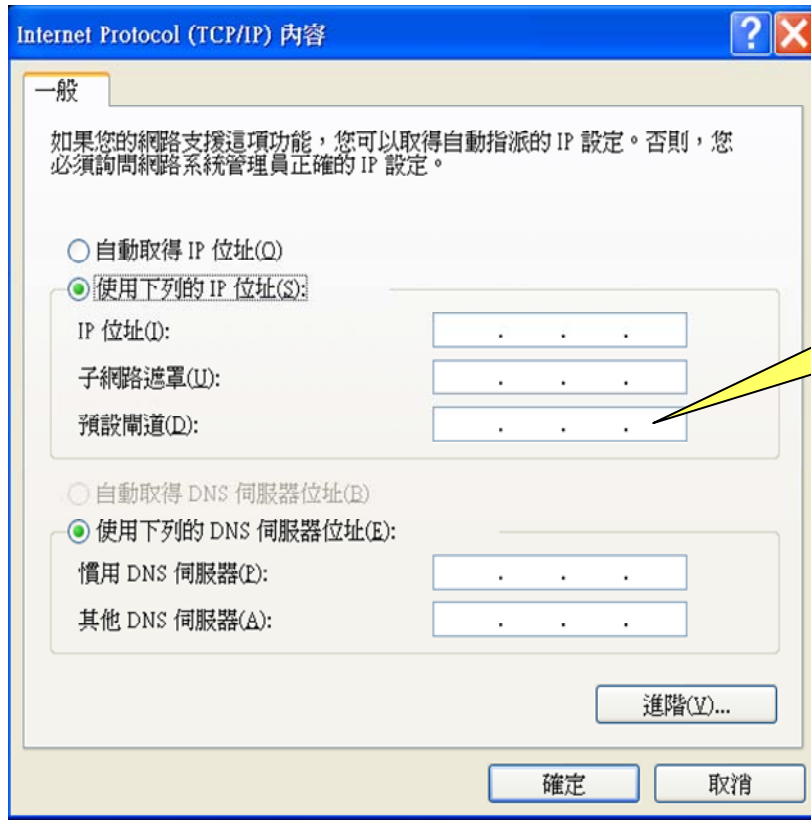
TCP/IP Reference Model

How Can Data Reach the Right Destination ?

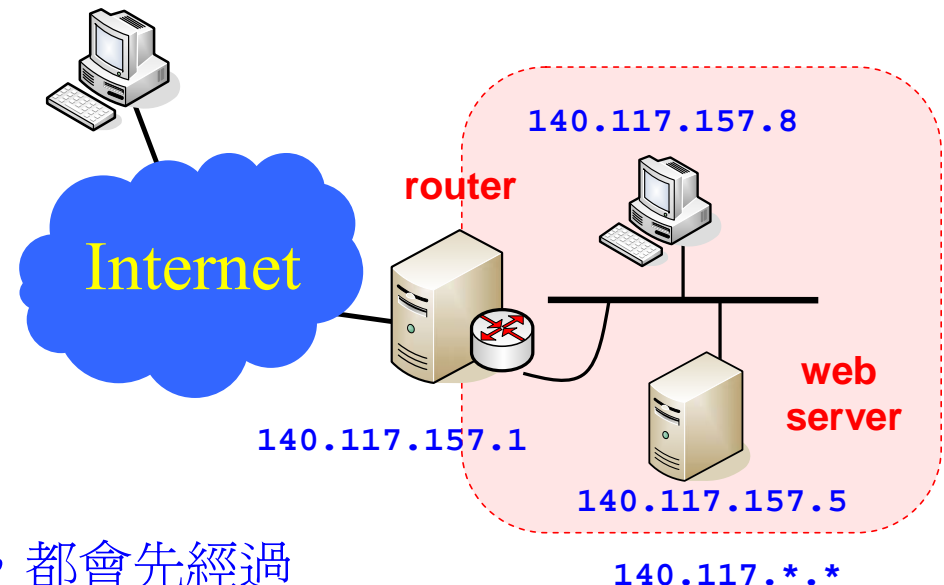


你有沒有想過，假設你在台中，何以當你的瀏覽器送出 HTTP request（假設目標是中山大學電機系）之後，這個訊息怎麼知道該往南部跑，跑到中山大學在哪兒？而不會往北部跑，跑到台大去？

The Role of Router

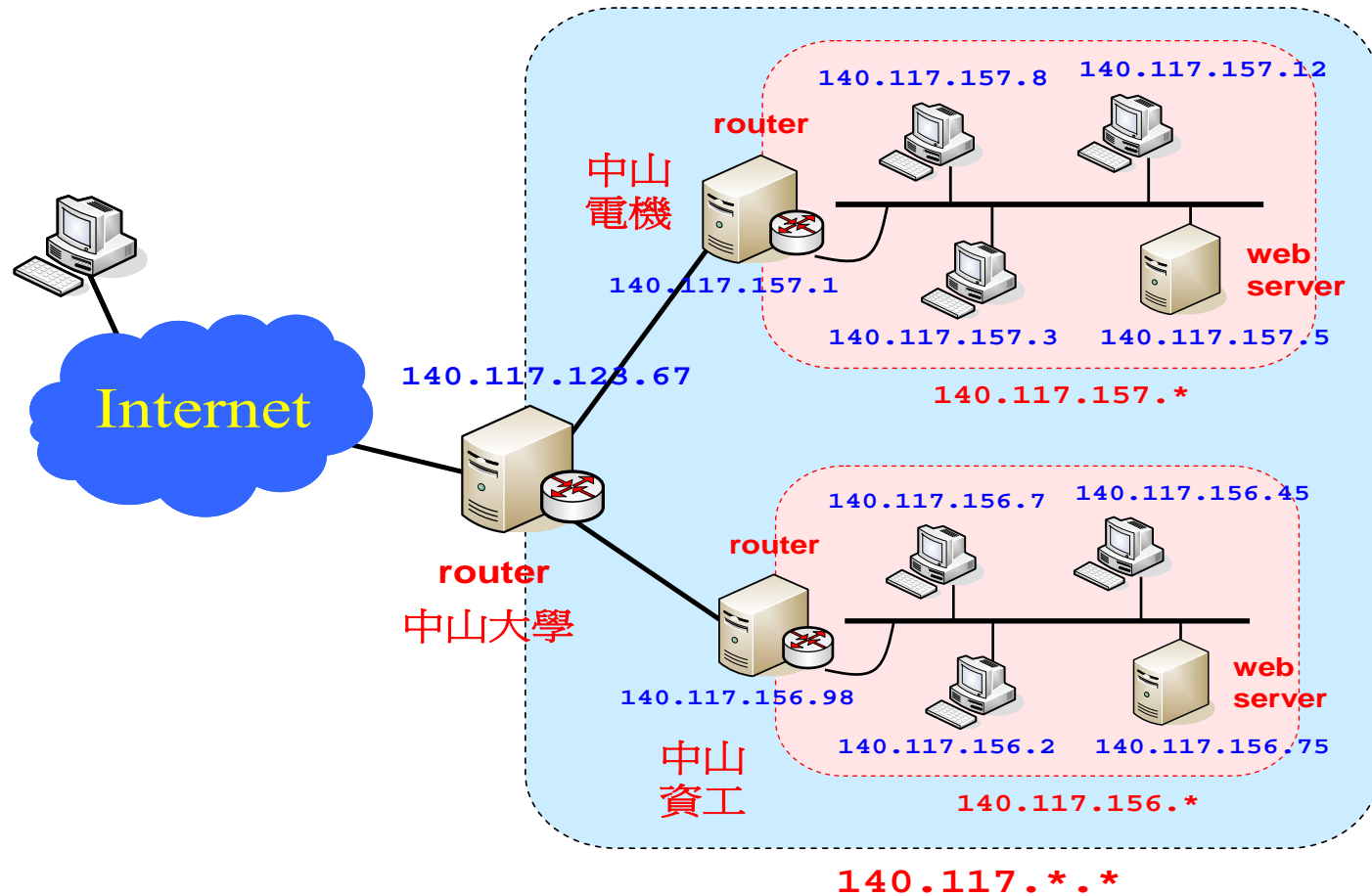


中山大學會有一台專門負責找尋路徑的機器，稱為 **router** 或 **gateway**（預設閘道）



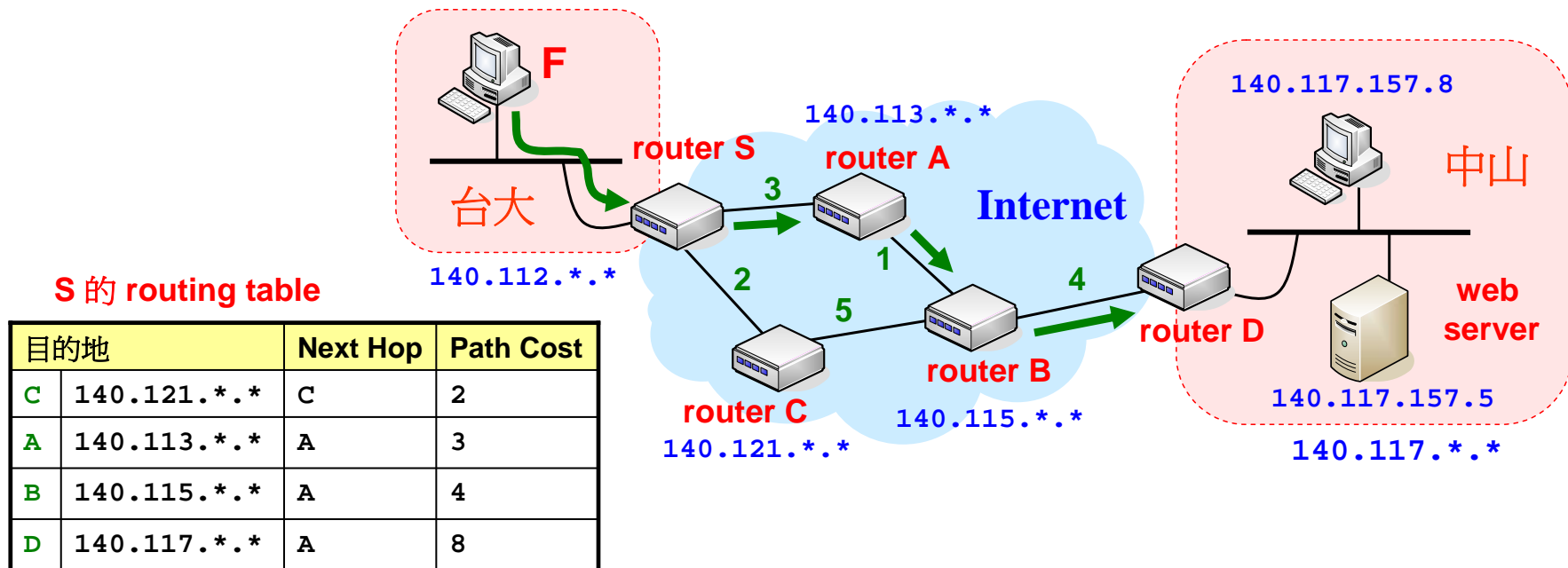
所有從 140.117.*.* 要送出去的資料，都會先經過這個 router。所有資料要送給 140.117.*.* 的機器，也都會先經過這個 router。

Hierarchical Router Architecture



或許你會在中山大學的每個科系都看到有 router，但這些 routers 最終都會連到一台專門負責中山大學所有科系的 router。

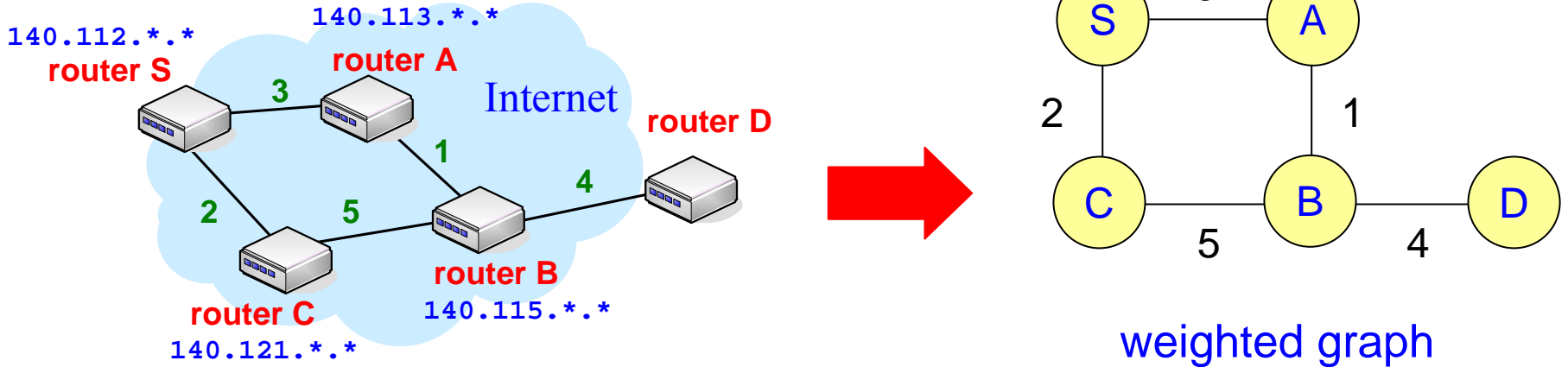
The Role of Routing Table



Router 裡頭都存有 routing table，裡頭會記載「若要到哪個目的地，該前往哪個 router」。如此，只要透過查表法，就可以將訊息送往正確的 router。但是這個 routing table 是如何建構出來的呢？

注意：為了縮小 routing table 的 size，routing table 所記錄的目的地只會是 subnet（如：140.117.*.*），而不會紀錄個別的 host（如：140.117.157.5）

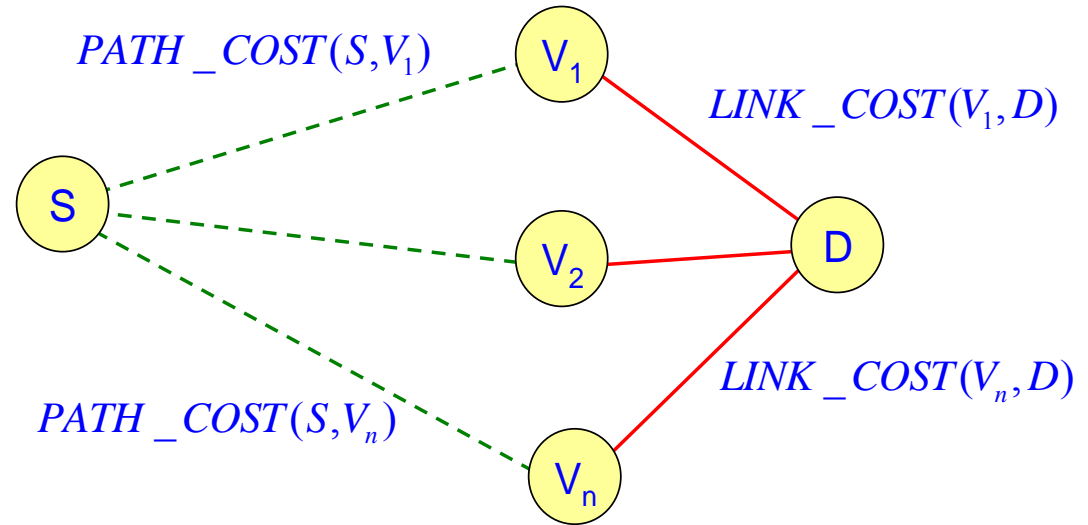
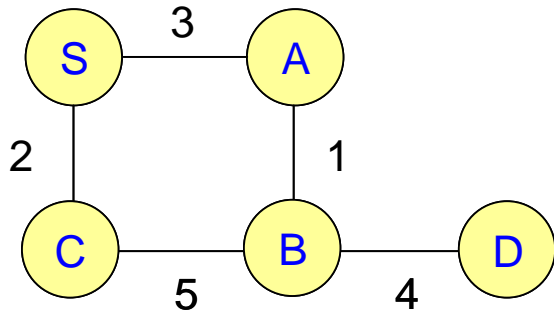
Model the Router Network as a Graph



我們可以將左圖 model 成右邊的 weighted graph。以 graph theory 術語來說，graph 是由 vertices（或稱為 nodes）和 edges（或稱為 links）所構成的集合，用符號 $G = (V, E)$ 來表示。上圖中， $V = \{S, A, B, C, D\}$ 表示 vertex set， $E = \{(S, A), (S, C), (A, B), (C, B), (B, D)\}$ 表示 edge set。Weighted graph 則包含一個 weight function $w : E \rightarrow \mathbb{R}$ 將每個 edge 對應到一個實數值的 weight。如上圖 $w(S, A) = 3$ 。這裡的 weight 可以全部設成 1，或者設定為頻寬的倒數

Principle of Minimum Cost Routing Algorithm

$$LINK_COST(S, A) = 3$$



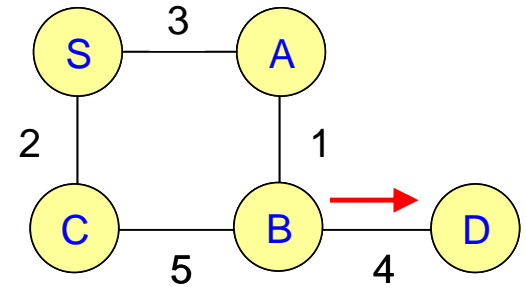
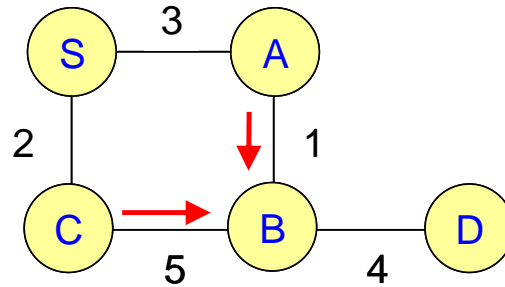
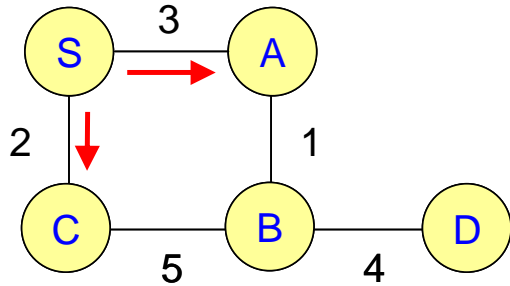
如左圖，我們可以把 $w(S, A)$ 記為 $LINK_COST(S, A)$ 。若從 S 到 B 的 path 為 $S \rightarrow A \rightarrow B$ ，那麼這條 path 的 cost 為 $LINK_COST(S, A) + LINK_COST(A, B) = 4$

令 $PATH_COST(S, D)$ 表示從 S 到 D 所有 paths 的 costs 的最小值。那麼我們有

$$PATH_COST(S, D) = \min \{ PATH_COST(S, V_i) + LINK_COST(V_i, D) : V_i \text{ 為 } D \text{ 的鄰居} \}$$

OSPF / Step 1 : Flooding

S 所送出的 link-state packet 內容包含
{ LINK_COST(S, A) = 3,
LINK_COST(S, C) = 2 }

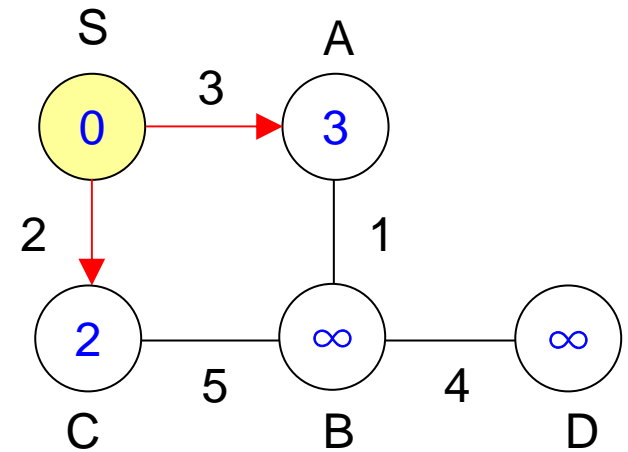


IETF 所制訂的 routing algorithm 包含 RIP (routing information protocol) 和 OSPF (open shortest path first)。由於現今幾乎所有的 router 產品都是執行 OSPF，因此我們只介紹 OSPF。註：transport layer 裡頭的資料稱為 segment，在 network layer 裡頭的資料稱為 packet。

在 OSPF 裡頭，每個 router 都必須知道任兩個 router 之間的 LINK_COST，因此 router 會週期性 (每 30 分鐘) 的全面廣播 link-state packet，裡頭記載自己和每個相鄰 router 之間的 LINK_COST。

OSPF / Step 2 : Dijkstra Algorithm (1/3)

```
Q = {s};
for each n ∈ V \ {s}
    PATH_COST(s,n) = LINK_COST(s,n);
    if (PATH_COST(s,n) ≠ ∞)
        parent(n) = s;
while (Q ≠ V) {
    m = argu∈V\Q min{PATH_COST(s,u)};
    Q = Q ∪ {m};
    for each n ∈ V \ Q
        if (PATH_COST(s,m) + LINK_COST(m,n)
            < PATH_COST(s,n)) {
            PATH_COST(s,n) = PATH_COST(s,m)
                + LINK_COST(m,n);
            parent(n) = m; }
}
```

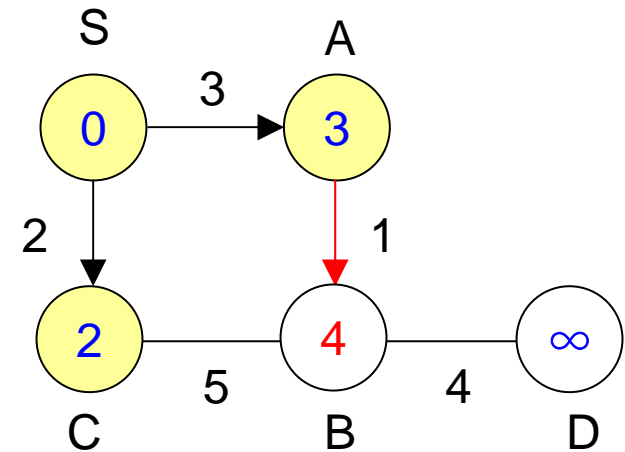
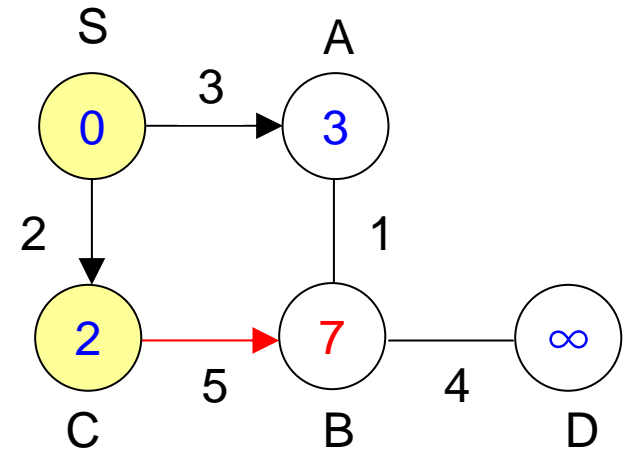


經過一段時間之後，每個 router 都可以知道任兩個相鄰 router 之間的 LINK_COST。此時可以開始執行左邊的演算法，這個演算法是由 Edsger W. Dijkstra 所發明。

註：Q 裡頭的 nodes 用黃色表示。圓圈裡頭的數字表示 PATH_COST，符號 $S \rightarrow A$ 表示 A 的 parent 為 S。

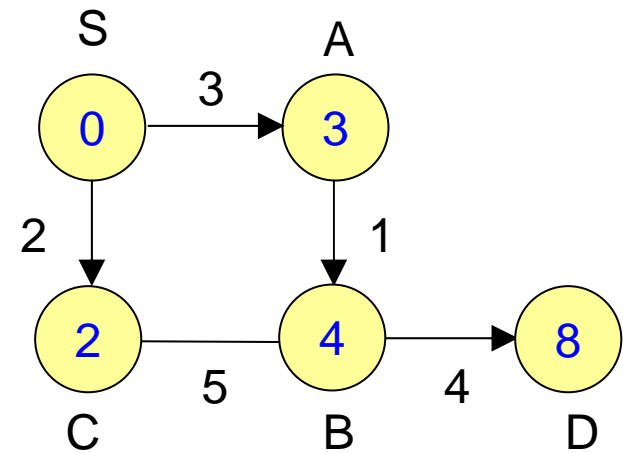
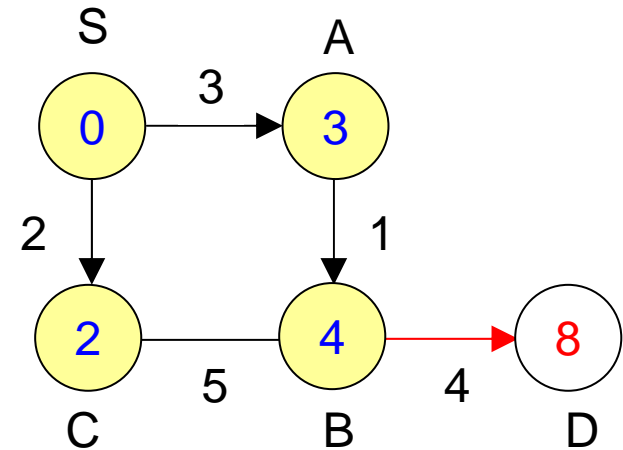
OSPF / Step 2 : Dijkstra Algorithm (2/3)

```
Q = {s};  
for each n ∈ V \ {s}  
    PATH_COST(s,n) = LINK_COST(s,n);  
    if (PATH_COST(s,n) ≠ ∞)  
        parent(n) = s;  
while (Q ≠ V) {  
    m = argu ∈ V \ Q min{PATH_COST(s,u)};  
    Q = Q ∪ {m};  
    for each n ∈ V \ Q  
        if (PATH_COST(s,m) + LINK_COST(m,n)  
            < PATH_COST(s,n)) {  
            PATH_COST(s,n) = PATH_COST(s,m)  
                + LINK_COST(m,n);  
            parent(n) = m; }  
}
```

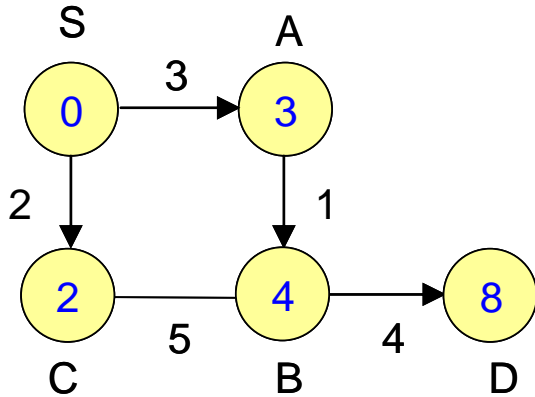


OSPF / Step 2 : Dijkstra Algorithm (3/3)

```
Q = {s};
for each n ∈ V \ {s}
  PATH_COST(s,n) = LINK_COST(s,n);
  if (PATH_COST(s,n) ≠ ∞)
    parent(n) = s;
while (Q ≠ V) {
  m = argu ∈ V \ Q min{PATH_COST(s,u)};
  Q = Q ∪ {m};
  for each n ∈ V \ Q
    if (PATH_COST(s,m) + LINK_COST(m,n)
        < PATH_COST(s,n)) {
      PATH_COST(s,n) = PATH_COST(s,m)
        + LINK_COST(m,n);
      parent(n) = m; }
}
```



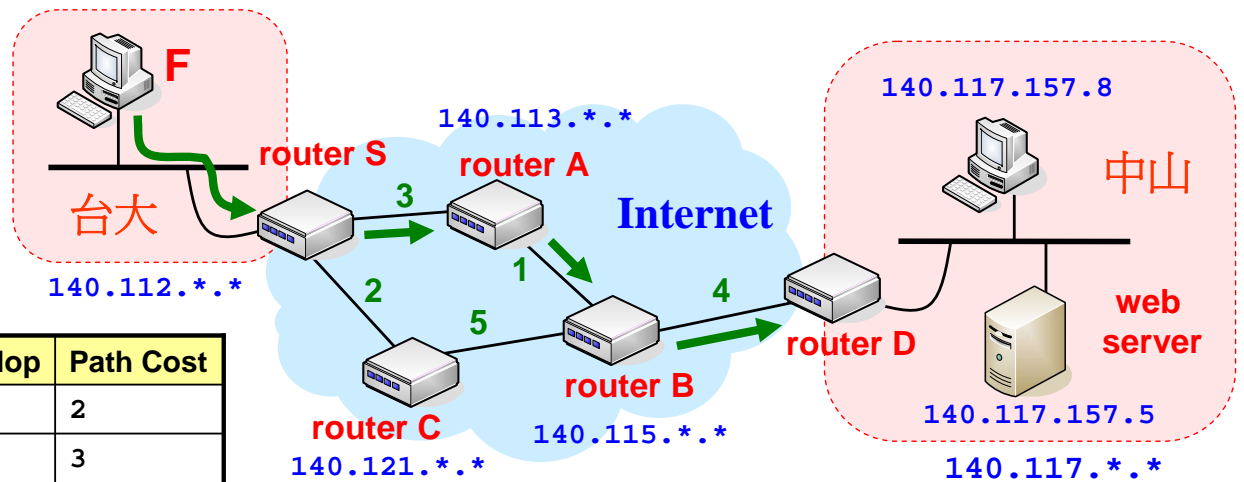
The Construction of Routing Table



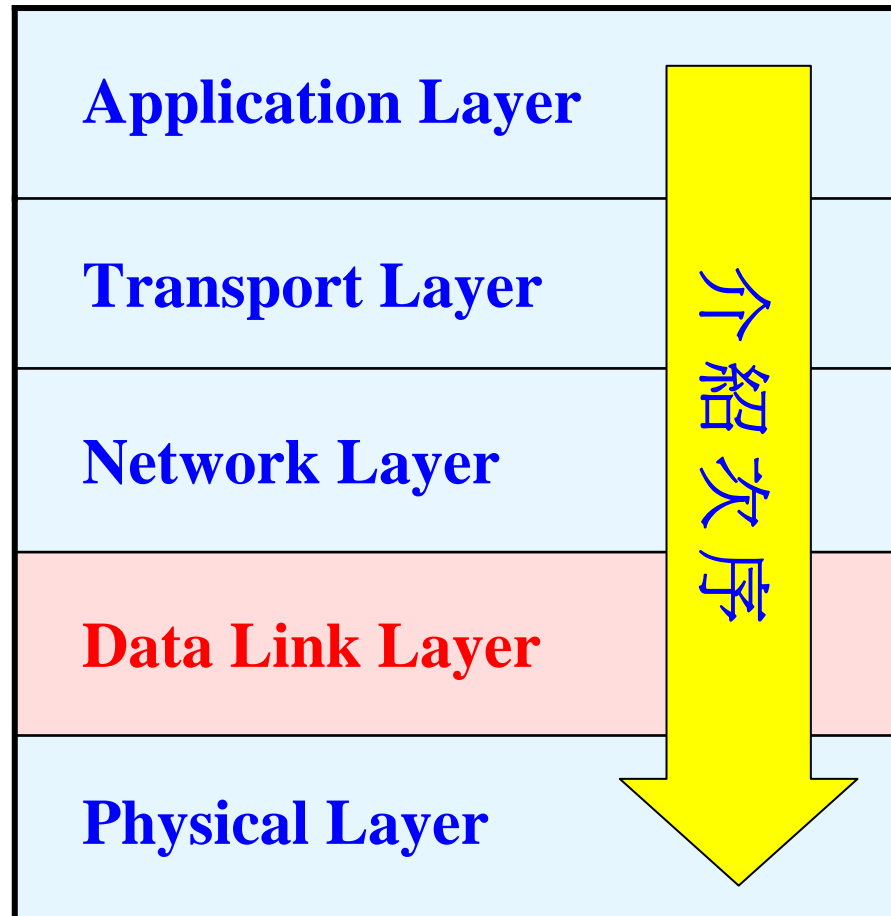
Dijkstra 演算法執行完畢之後，router 便可根據每個 node 的 parent 關係建構出 routing table，如下圖所示。

S 的 routing table

目的地	Next Hop	Path Cost
C	C	2
A	A	3
B	A	4
D	A	8

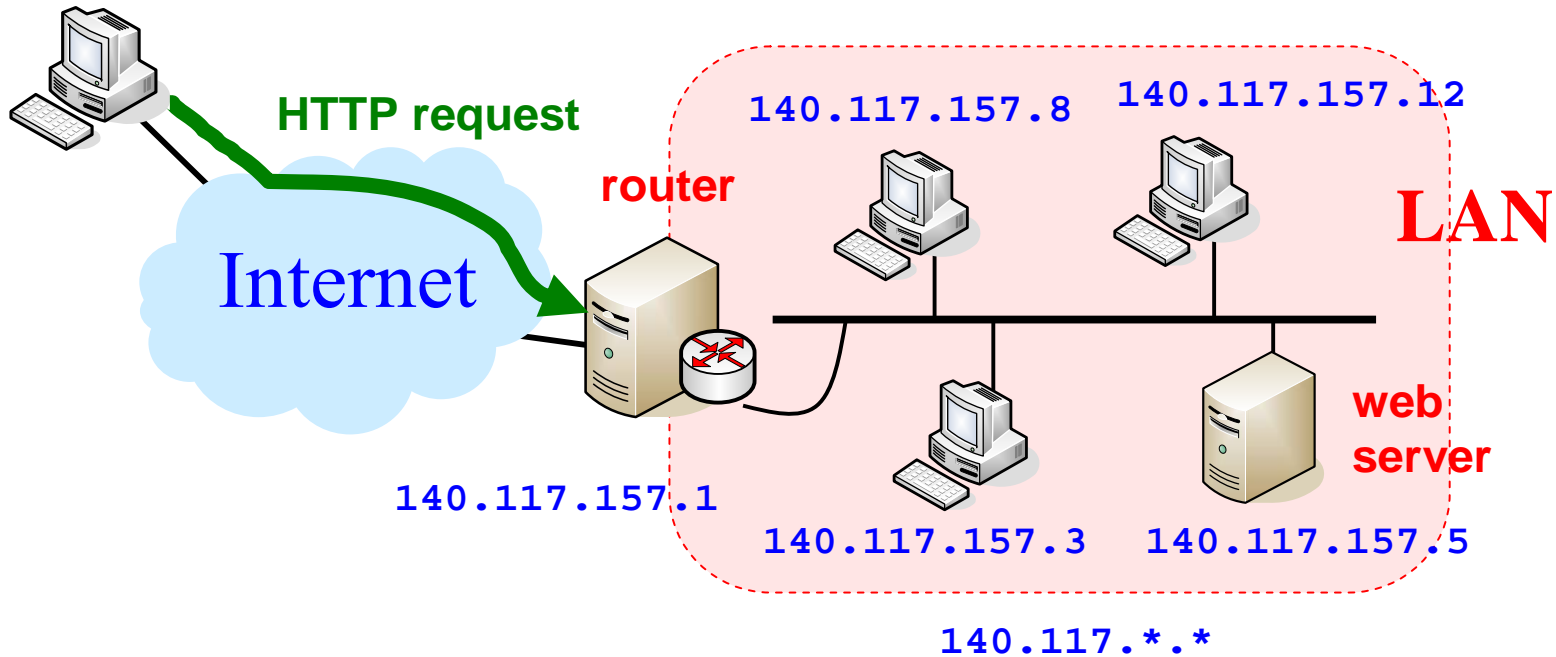


Agenda : Data Link Layer



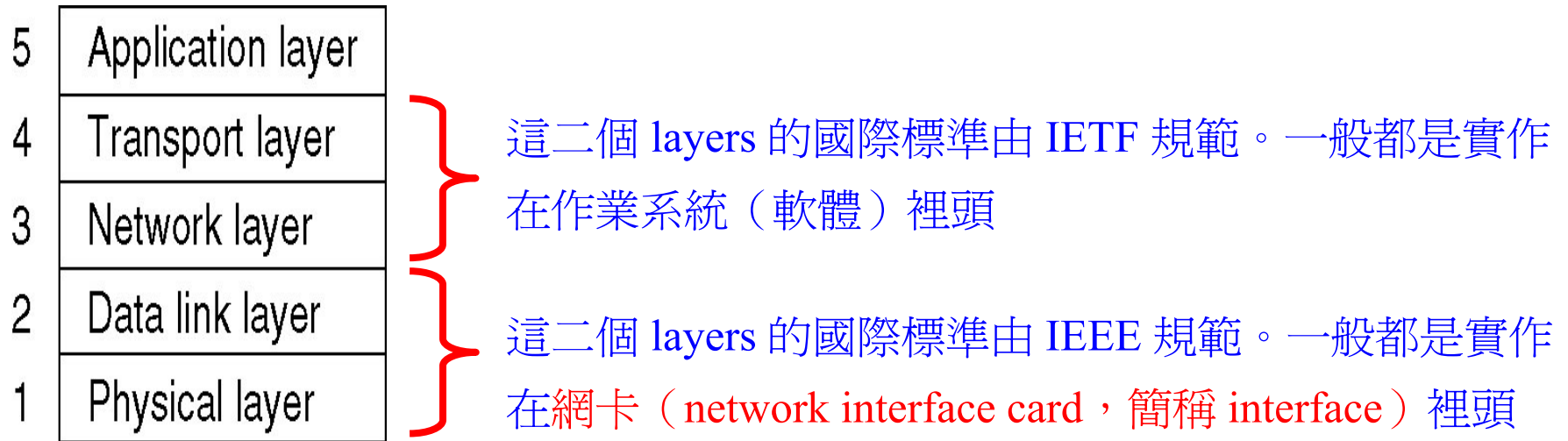
TCP/IP Reference Model

How to Send Data in a Local Area Network ?



我們剛才說過，所有要進出 140.117.*.* 的訊息必須透過 router。那麼 router 如何將資料送給屬於同一個 LAN（區域網路，local area network）裡頭的 web server 呢？首先我們來看一下什麼是 LAN？在 LAN 裡頭，任兩台電腦彼此都能夠「直接」收到對方的訊號。

MAC Address for Local Communication

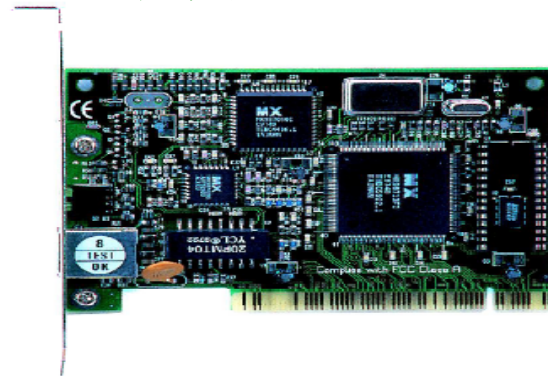


要能夠接收資料，必須要有 address。
IETF 規定的 address 稱爲 IP address。

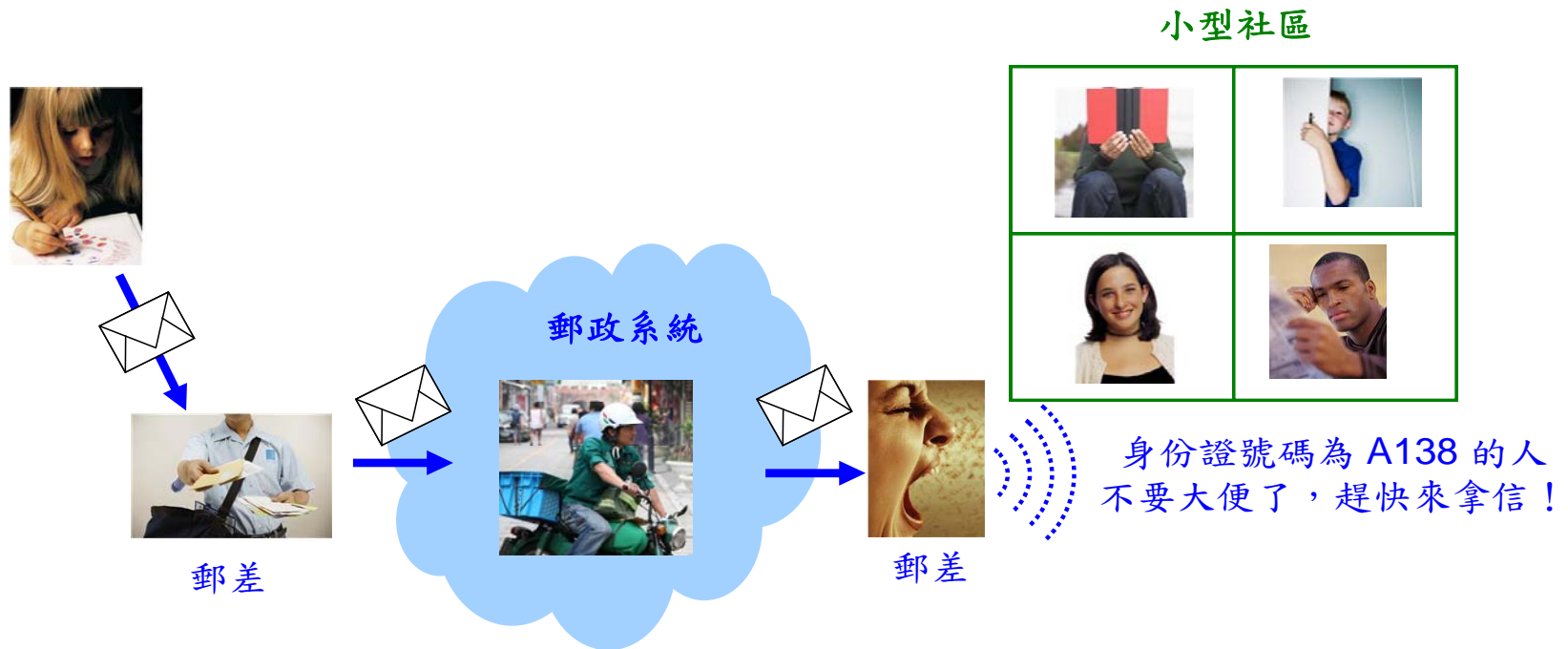
然而 IEEE 規定：在區域網路之內
收送資料，必須使要 **MAC address**。

因此一台電腦要同時有 IP address 和
MAC address 才能上網。

MAC address 燒在網卡裡頭，
共有 **48 bits**，內容由 **IEEE** 和
廠商共同規範。不會有二張網卡
擁有相同的 **MAC address**。

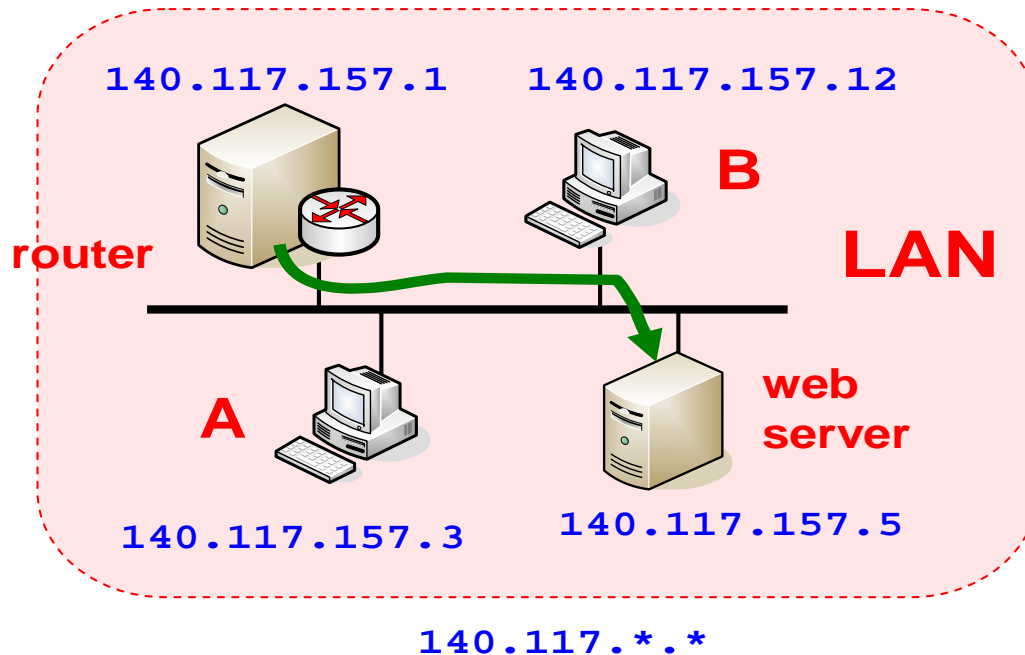


Analogical Scenario in Apartment Community



想像 IP address 就像住家地址。當郵差（相當於 router）攜帶著信件到達目的地的住家社區（相當於 LAN）時，就不會再騎機車前往其他地方，同時也不會呼喊「住家地址是高雄市蓮海路 80 號的人來拿信」，而改成呼喊「身份證號碼是 A138 的人來拿信」。畢竟身份證號碼（相當於 MAC address）比較短，溝通速度較快。

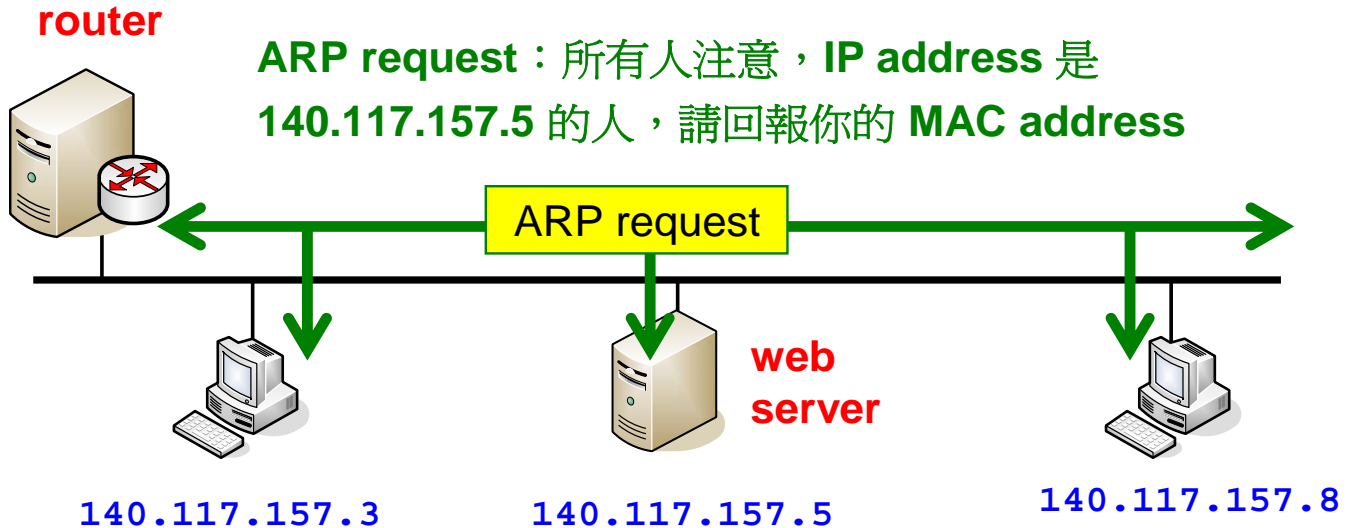
How to Get the MAC Address ?



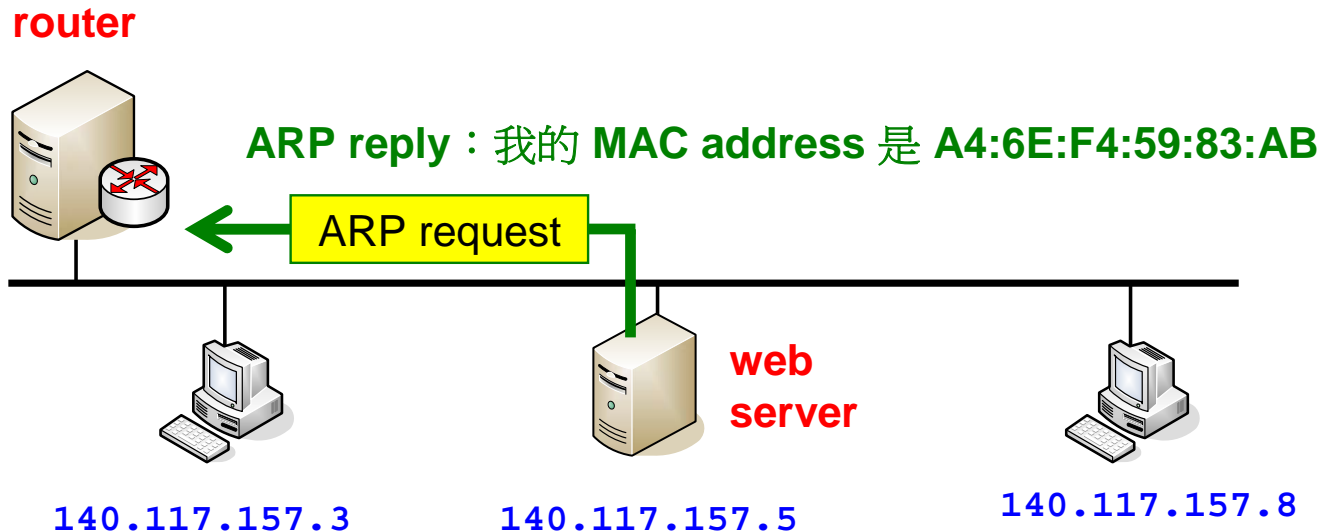
當 source 和 destination 位在相同的 LAN 時，source 送給 destination 的資料不會離開 router 所管轄的 LAN，因此不需透過作業系統查詢 routing table；此時使用 MAC address 速度較快，因為是實作在硬體裡頭。然而 router 怎麼知道 web server 的 MAC address？這時候我們需要使用 ARP。

ARP : Address Resolution Protocol

1



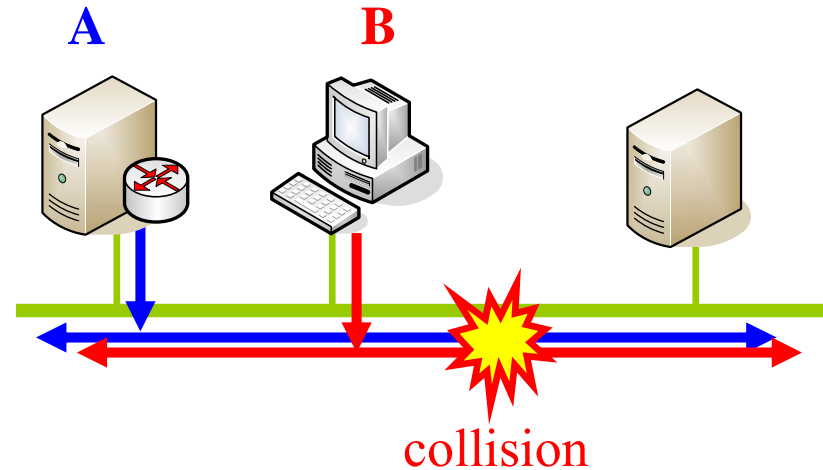
2



Why We Need MAC (Medium Access Control) ?

A 和 B 同時發言，導致大家都聽不清楚他們在講什麼

小型社區



我們之前說過：在 LAN 裡頭，任兩台電腦彼此都能夠「直接」收到對方的訊號。所以在 LAN 裡頭，若有兩台或兩台以上的電腦同時發送訊號傳送資料，那麼將會造成 **collision**。想像一下，在一個極小型的社區裡頭，如果有二個人同時發言，那麼大家都會聽不清楚。**MAC** 的目的就是要制訂「發言管理辦法」，用來決定該怎麼發言，誰可以發言，誰不可以發言

Ethernet/802.3 : Carrier Sense Multiple Access

1. 想要傳送資料之前先聽聽看是否有其他電腦正在傳送資料。
2. 若有，則持續監聽，直到沒有其他電腦在傳送資料為止。
3. 若無，則立即傳送。
4. 傳送過程中若偵測到 **collision**，則持續送出 **jamming** 訊號一段時間，之後便暫停；接著**隨機**等待一段時間再重新傳送資料。 註：當 **B** 偵測到 **collision** 時，爲了要讓 **A** 也知道發生 **collision**，所以 **B** 故意送出 **jamming** 訊號

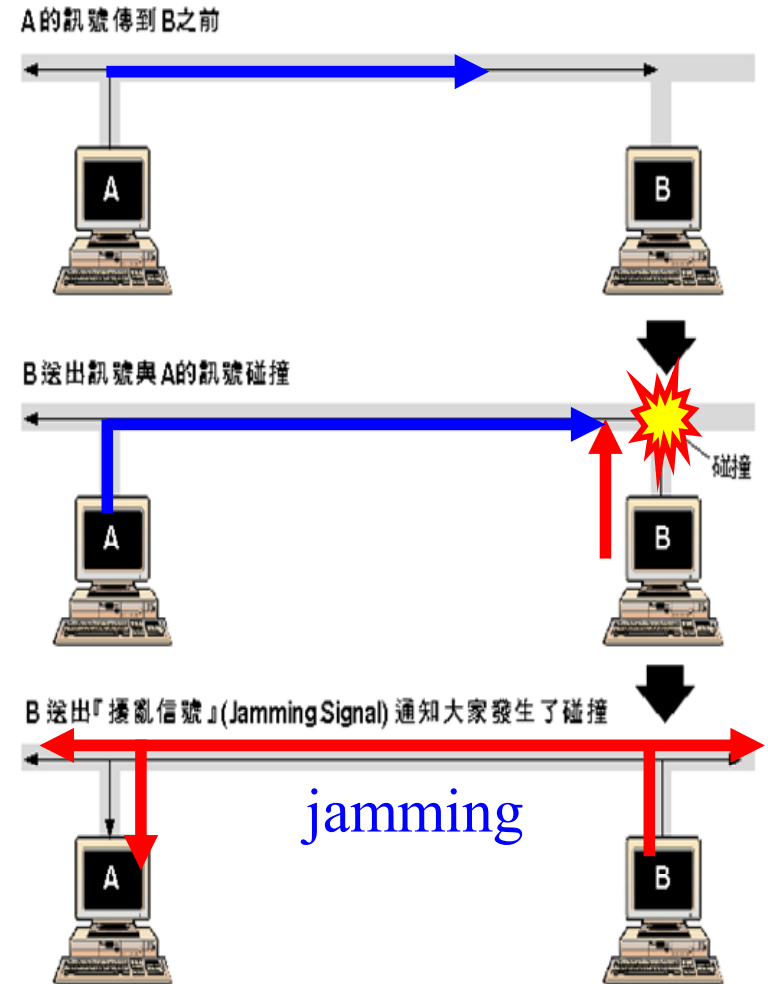
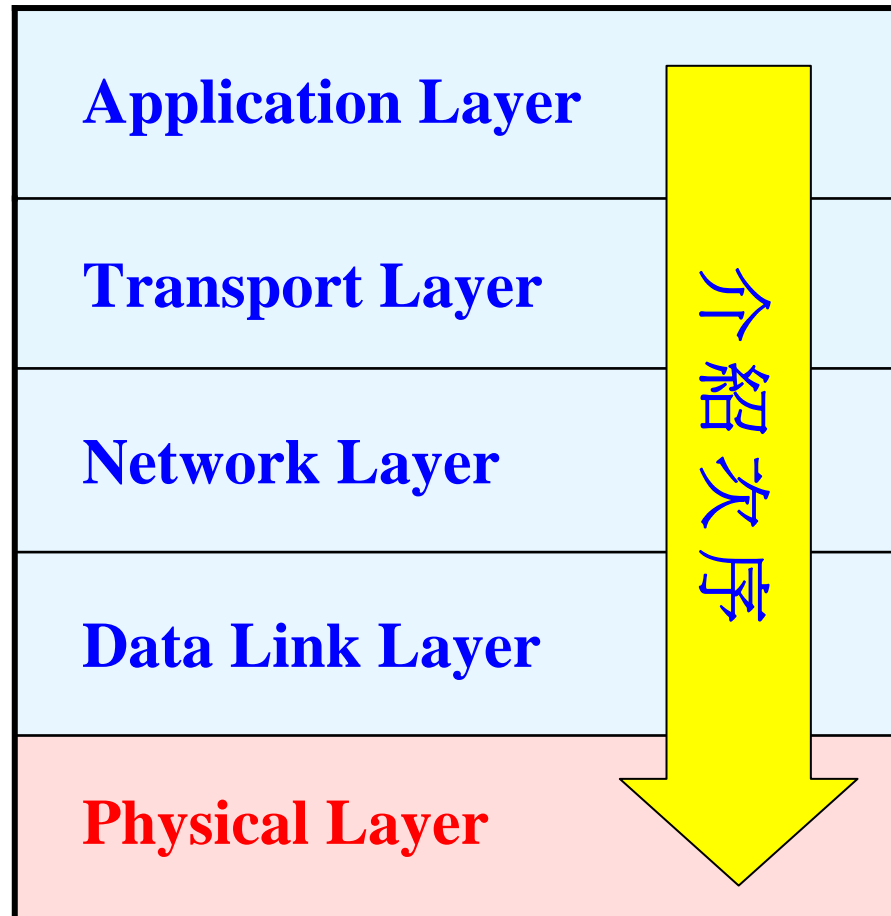


圖 5-07 碰撞偵測的持續時間

Agenda : Physical Layer



TCP/IP Reference Model

Various Transmission Media

- ◆ 以硬體實際傳輸 0、1 訊號
- ◆ 使用不同的方式來實作資料傳輸媒介

光纖



雙絞銅線

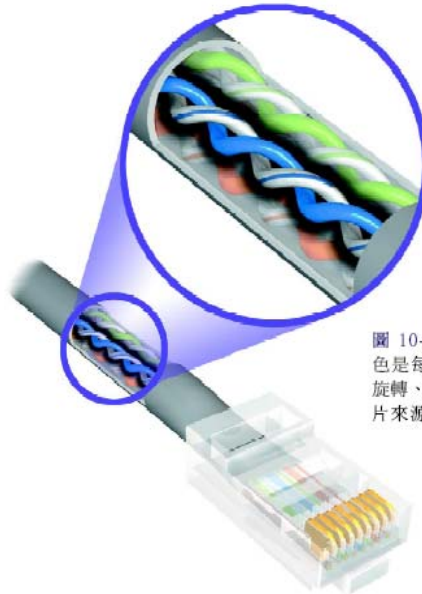


圖 10-25 雙絞線的特色是每兩條芯線相互旋轉、纏繞在一起 (圖片來源：旗立資訊)

無線電波

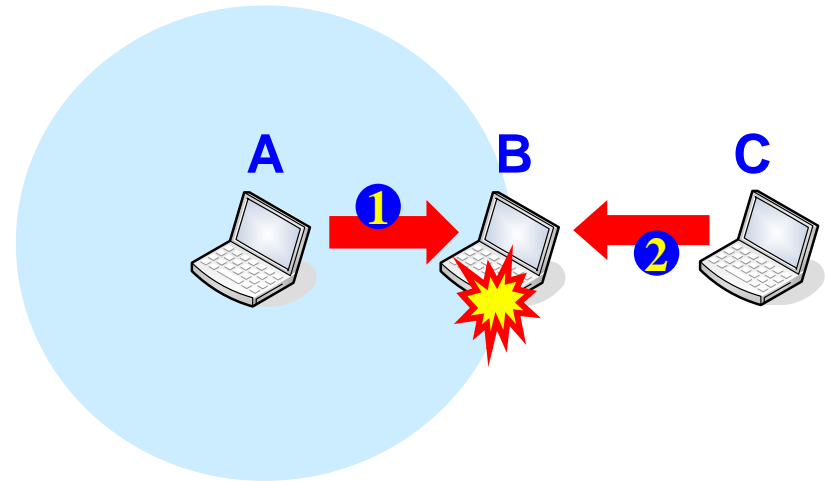
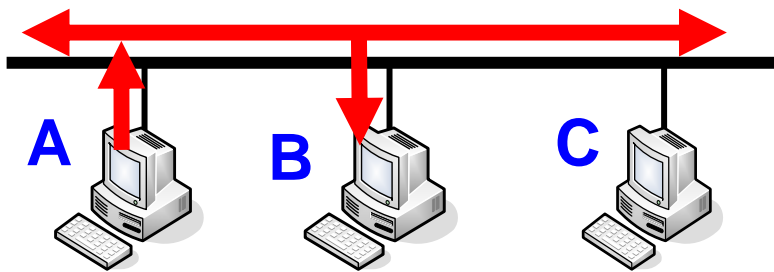


How Wireless Technologies Influence Internet ?



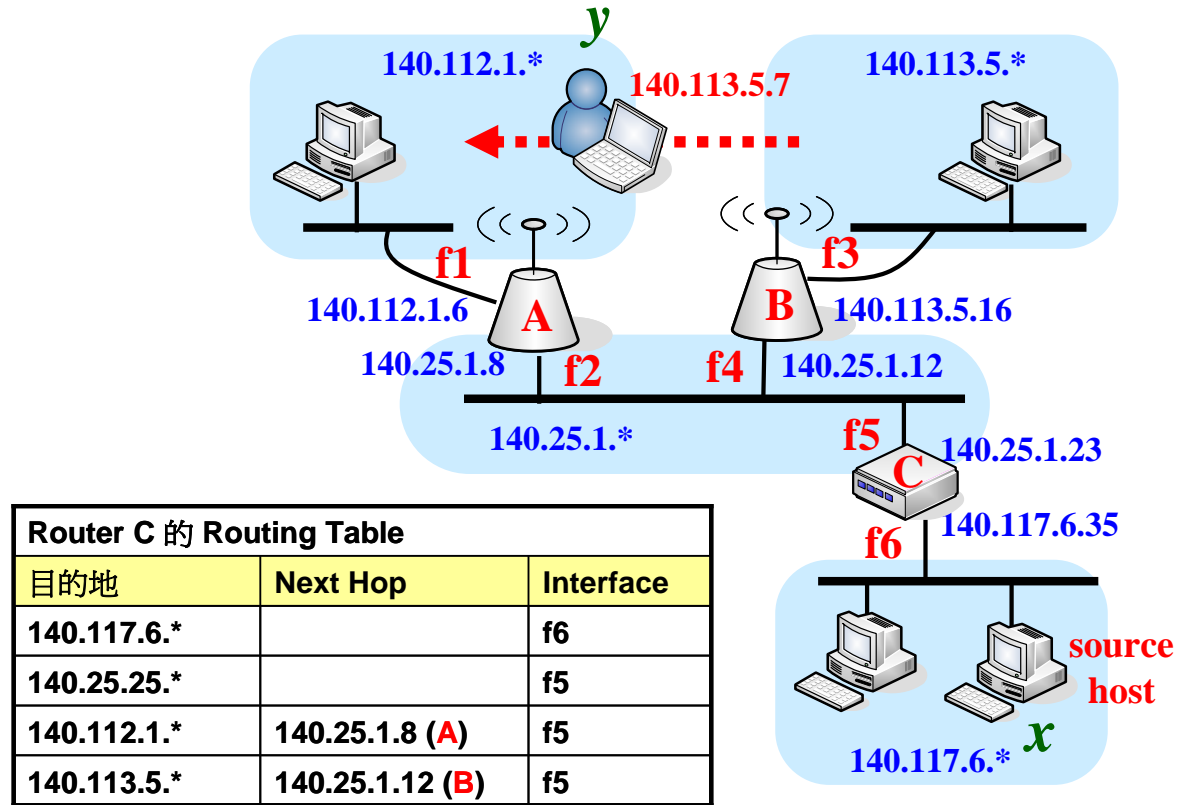
這些鳥兒必然會某種無線網路的技術

Hidden Terminal Problem in a Wireless Network



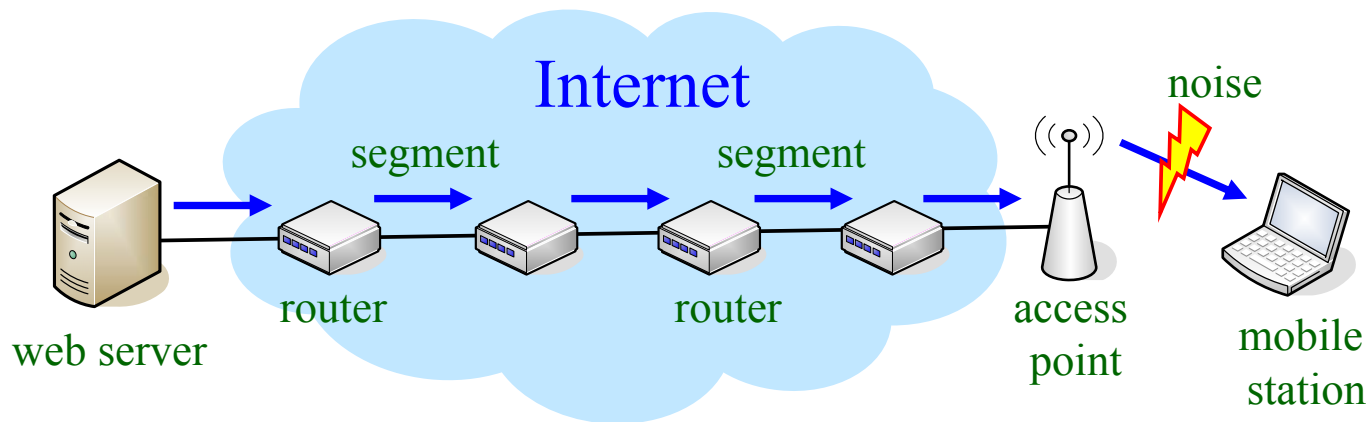
在傳統的 wired LAN 裡頭，如果 C 想送資料給 B 時，會先 carrier sense，一旦偵測到有其他 station 正在傳送資料，就會等其他 station 傳完資料之後再試圖傳送。然而，如右圖，在無線的環境裡頭，即使 C 在傳送資料給 B 之前也執行先 carrier sense，仍然無法偵測到 A 傳給 B 的訊號。此時 C 會誤以為目前 media 是 free，所以就傳送資料給 B，將會導致 collision 的發生。這就是所謂的 **hidden terminal problem**，此一問題在 wired LAN 裡頭並不會發生。目前無線區域網路的國際標準 802.11 有提出方法可以減緩 hidden terminal problem

Mobile IP



當使用者 y (140.113.5.7) 從 subnet 140.113.5.* 移動到 subnet 140.112.1.* 時，router C 的 routing table 並不會有所改變，此時所有目的地為 140.113.5.7 的 packets 仍會依照路徑 $x \rightarrow C \rightarrow B \rightarrow y$ 來走，因此這些 packets 都會 lost 掉。本課程會介紹 mobile IP (MIP) 的相關技術來解決此一問題。

TCP over Last-Hop Wireless Networks



和有線網路相比，無線網路特別容易受到雜訊干擾。如上圖，假設 mobile station 正在下載某個網頁。即使從 web server 到 access point 之間並沒有任何設備發生 congestion，然而 mobile station 仍可能因雜訊干擾而無法正確接收網頁的資料，因而不會回 ACK。按 TCP 的規定，只要 source host 沒收到 destination host 所回應的 ACK，便會縮小 congestion window 的值。如此一來，throughput 將會降低。事實上，只是發生 wireless loss 時，congestion window 不應該縮小。會發生這個問題是因為 TCP 做出一個錯誤的假設：資料發生 lost 的原因一定是因為 router 擁塞所造成的。本課程將會介紹 snoop TCP 來解決此一問題。

課程大綱：Bottom-Up Approach

Layer	國際標準或論文技術	內容大綱
Physical		簡單介紹 radio (無線電波) 的特性
MAC	IEEE 802.11	無線區域網路的國際標準
	IEEE 802.11e	802.11的增強版，增加支援服務品質保證的功能
	IEEE 802.16/802.16e	無線都會網路的國際標準
	IEEE 802.22	無線認知網路的國際標準 (目前仍在草案階段)
Network	Mobile IP	新的 IP 技術，讓使用者在移動環境下仍可順利上網
	GSM、GPRS、UMTS	手機系統，focus 在 location management 的介紹
	Routing in MANET/WSN	介紹無線隨意網路及無線感測網路的 routing 技術
Transport	Snoop TCP	在有線與無線的整合環境之下，確保 TCP 的效能
	Ad-Hoc TCP	在無線隨意網路中，確保 TCP 的效能
Application	Cross-Layer Design	介紹無線監控錄影網路的應用，結合 PHY、MAC、及 application layer 的技術

介紹次序



國際標準的部分用藍色字表示

自我評量

1. TCP 假設 segments 發生 lost 的原因是什麼？
2. 何謂 hidden terminal problem？試舉例說明
3. 參考下圖，graph 裡頭的 node 代表 router，edge 上的 weight 代表 link cost，試利用 Dijkstra algorithm 求出 router A 的 routing table（不能直接給答案，要有執行步驟）

