電腦網路概論

考前準備書

第 1 章　計算機網路與網際網路

第 2 章　應用層

第 4 章　網路層：資料層

課本：Kurose & Ross, Computer Networking: A Top-Down Approach, 8th ed.

授課：徐武孝 教授

整理自課程講義（中文 1 份、英文 3 份，共 493 張投影片）

整理日期：2026 年 11 月 2 日

目錄

第 1 章　計算機網路與網際網路 4
1.1　什麼是網際網路 4
1.2　網路邊際 5
1.3　網路核心 6
1.4　延遲、遺失與產出率 8
1.5　協定分層與服務模型 10
1.6　面臨攻擊的網路 11
1.7　網際網路歷史 12
第 2 章　應用層 13
2.1　網路應用程式原理 13
2.2　Web 與 HTTP 15
2.3　DNS 網域名稱系統 19
2.4　內容傳遞網路 CDN 21
第 4 章　網路層：資料層 23
4.1　網路層概觀 23
4.2　路由器與最長前綴比對 24
4.3　IP 資料包格式 24
4.4　IP 定址 27
4.5　特殊位址、路由聚合與私有位址 29
4.6　NAT 網路位址轉換 30
4.7　DHCP 動態主機設定協定 30
附錄 A　公式總表 31
附錄 B　中英術語對照 32
附錄 C　數字速查 33
附錄 D　模擬試題 34
附錄 E　講義疑點與不一致 37
附錄 F　考前檢查清單 38

如何使用本書

本書把中文與英文講義的第 1 章合併，再接上第 2、4 章，依主題重新編排，不是逐張投影片的翻譯。專有名詞保留英文，方便對照英文講義與作答。

代號	檔案	張數	內容
中 1	中文／CH01.pptx	141	第 1 章（含資安與歷史）
英 1	英文／chap1_overview_推.pptx	124	第 1 章（連線網路與計算例子較細）
英 2	英文／Chap2_applications.pptx	133	第 2 章：原理、HTTP、DNS、CDN
英 4	英文／chap4_Network.pptx	95	第 4 章：IP、定址、NAT、DHCP

每節開頭的「重要度」與「來源」可以這樣用：

- **重要度**：★★★ 優先、★★☆ 次之、★☆☆ 瀏覽。這是依講義的篇幅、例題與計算量做的**推估**，不是老師公布的考試重點。
- **來源**：如「英 2 p.59–60」代表英文第 2 章第 59 到 60 張投影片，有疑問時可回去看原圖。
- **方框**：藍色是考點、橘色是易混淆、灰色是例題、綠色是補充。

建議讀法：

1. **還有幾天**：依序讀第 1、2、4 章，把例題自己算一遍，再寫附錄 D 模擬試題。
2. **只剩一天**：先看各章速覽與所有 ★★★ 小節，接著練計算題（延遲、產出率、HTTP RTT、快取、分割、子網路）。
3. **進考場前**：看附錄 A 公式、附錄 C 數字與附錄 F 檢查清單。

> **請注意**　講義中有幾處數字或說法前後不一致，已整理在**附錄 E**，也列出講義沒有涵蓋的範圍（例如沒有第 3 章）。建議考前先看一眼。

第 1 章　計算機網路與網際網路

> **本章速覽**　本章要帶走的六件事：① 協定的定義；② 封包交換與電路交換的差異；③ 四種節點延遲與公式；④ 流量強度 La/R；⑤ 瓶頸連結與產出率 min{...}；⑥ 五層協定堆疊、各層封包名稱與封裝。

1.1　什麼是網際網路

重要度 ★★☆　｜　來源：中 1 p.6–19；英 1 p.4–17

網際網路可以從兩個角度描述：一是**組成它的軟硬體元件**，二是**它提供給應用程式的服務**。考題常直接問這兩種角度各看到什麼。

角度	看到的內容
基本元素（nuts and bolts）	主機／終端系統、通訊連結、封包交換器、ISP、協定、標準組織
服務（service）	提供服務給**分散式應用程式**的基礎設施，並透過 **socket 介面（API）**讓應用程式取用，就像寄信要遵守郵政服務的規矩

基本元素

- **主機（host）＝終端系統（end system）**：執行網路應用程式的裝置，位於網際網路的邊際；又分成用戶端（client）與伺服器（server）。
- **通訊連結（communication link）**：光纖、銅線、無線電、衛星；**傳輸速率**以 bits/sec 計。
- **封包（packet）**：傳送端把資料切成區段，加上標頭（header）後形成。
- **封包交換器（packet switch）**：從向內連結收封包、由向外連結轉送。兩大類：**路由器**（多用於網路核心）、**連結層交換器**（多用於連線網路）。
- **路由／路徑（route / path）**：封包從傳送端到接收端經過的一連串連結與交換器。
- **ISP**：終端系統透過它存取網際網路，本身也是由封包交換器與連結組成的網路，彼此必須互連，所以網際網路是「**網路的網路**」。
- **標準**：網際網路標準由 **IETF** 制定，文件稱 **RFC**（Request for Comments）；**IEEE 802** 委員會制定乙太網路與 WiFi 標準。
- **最重要的兩個協定**：TCP 與 IP，合稱 TCP/IP。

> **考點**　**協定（protocol）**定義了兩個以上通訊實體之間交換訊息的**格式**與**順序**，以及在傳送、接收訊息或發生其他事件時要採取的**動作**。背誦時抓三個關鍵字：格式、順序、動作。

1.2　網路邊際

重要度 ★★☆　｜　來源：中 1 p.20–47；英 1 p.19–54

連線網路（access network）是把終端系統連到路徑上第一具路由器（**邊際路由器**）的實體連結。

連線網路比較

類型	基礎設施／設備	運作重點	數字與特性
DSL	既有電話線（雙絞線） DSL 數據機、分離器（splitter）、**DSLAM**	資料與電話以不同頻率共用線路，一條線看起來像三條；DSLAM 把資料送進網際網路、電話送進電話網	下行 24–52 Mbps、上行 3.5–16 Mbps；**專用**線路、**不對稱**
纜線 Cable	有線電視設施，**HFC**（光纖＋同軸） 纜線數據機、頭端 **CMTS**	光纖到鄰里光纖節點，再以同軸到戶；以 **FDM** 分頻道；**共用廣播媒介**，多人同時下載時每人速率會明顯下降	每個光纖節點 500–5000 戶；DOCSIS 2.0 下行 42.8、上行 30.7 Mbps；不對稱
FTTH	光纖到戶（光世代） **ONT**、光分離器、**OLT**	兩種架構：**AON**（本質上是交換式乙太網路）與 **PON**。PON 由分離器把多戶（通常少於 100 戶）併到一條共用光纖；OLT 收發的封包在分離器複製	可達 Gbps 等級；PON 又分 GPON、EPON
FTTB＋VDSL	光纖到大樓	機房到大樓用光纖，大樓機房到住家用乙太網路或 VDSL（銅線）	台灣常見的過渡方案
5G 固定無線	基地台到家	不必從中央機房鋪線到住家	省去昂貴又易故障的佈線
乙太網路	乙太網路交換器	企業、大學與家庭最普遍的 LAN 技術	10 Mbps–10 Gbps
WiFi	存取點（IEEE 802.11）	使用者須在存取點數十公尺內	WiFi 6／7；共用無線媒介
廣域無線	電信業者基地台	與蜂巢電話相同的基礎設施	涵蓋數十公里；4G、5G
資料中心	高速連結	連接成百上千台伺服器並接上網際網路	每條 10–100 多 Gbps

> 考點　**DSL 三個頻帶**：0–4 kHz 雙向電話；4–50 kHz 中速上行；50 kHz–1 MHz 高速下行。
> **DSL 實際速率受限因素**：業者的費率層級、住家到中央機房的距離、雙絞線線徑、電氣干擾。DSL 本來就是為短距離設計。
> **DSL 與纜線最關鍵的差別**：DSL 是專用線路；纜線是共用廣播媒介。

> 補充　家庭網路通常把寬頻數據機（纜線／DSL／FTTH）與路由器、WiFi 存取點結合。HiNet 光世代資費表（公告日期 2025/5/2）同時有對稱（如 1G/1G）與不對稱（如 100M/40M、16M/3M）方案，可當作「不對稱」的實例。

實體媒介

位元經由電磁波或光脈衝跨越實體媒介傳送。**導引式媒介**：訊號沿固體媒介前進（銅線、同軸、光纖）；**非導引式媒介**：訊號在大氣或外太空自由傳播（無線電、衛星）。

媒介	類別	特性	數字
雙絞銅線	導引式	最便宜、最常用；兩條絕緣銅線互絞以降低電磁干擾；LAN 常用 UTP	Cat 5：100 Mbps／1 Gbps Cat 6 以上：10 Gbps 第 6a 類 UTP：100 公尺內 10 Gbps
同軸電纜	導引式	兩條銅導體**同軸**排列；雙向；可當共用媒介；寬頻（一條線多個頻道）	每頻道數百 Mbps
光纖	導引式	傳光脈衝，每個脈衝一個位元；錯誤率低、中繼器間距大、**不受電磁干擾**；用於骨幹與長途	點對點 10–100 多 Gbps OC-n＝n × 51.8 Mbps
地表無線電	非導引式	不必鋪線、可穿牆、支援行動、距離長；廣播、半雙工；受路徑遺失、遮蔽衰減、多路徑衰減、干擾影響	WiFi：10–數百 Mbps、數十公尺 4G：數十 Mbps、約 10 km 藍牙：短距低速 微波：45 Mbps
衛星	非導引式	**同步衛星**固定於上空 36,000 公里；**LEO 低軌道衛星**離地近、會繞地球移動，可彼此及與地面站通訊	每頻道可達 45 Mbps 端對端延遲約 270 ms

1.3　網路核心

重要度 ★★★　｜　來源：中 1 p.48–74；英 1 p.56–78

核心的兩大功能

	轉送（Forwarding）	繞送（Routing）
範圍	**區域**動作，單一路由器內	**全域**動作，整個網路
做什麼	把封包從輸入連結移到適當的輸出連結	決定封包從來源到目的地的路徑
靠什麼	**轉送表**（目的位址對映向外連結）	**繞送演算法／協定**，用來自動設定轉送表

封包交換

- **儲存轉送（store-and-forward）**：交換器必須**收完整個封包**才能開始傳到下一條連結。把 L 位元封包推進速率 R 的連結要 L/R 秒。
- **佇列延遲與封包遺失**：每條連結都有**輸出緩衝區**。抵達速率暫時超過連結傳輸速率時封包就排隊；緩衝區滿了，新到或已在佇列中的封包會被丟棄。
- **如何決定走哪條連結**：來源把目的 IP 位址寫進標頭，路由器以該位址查轉送表。

> 例題　L＝10 Kbits，R＝100 Mbps。單一跳的傳輸延遲＝$10^4 \div 10^8$＝**0.1 ms**。
> 若路徑有 3 條相同的連結（2 具路由器），且忽略其他延遲，儲存轉送造成的端對端延遲＝3 × L/R＝**0.3 ms**。

電路交換

傳送前先建立端對端連線，路徑上的資源（緩衝區、傳輸速率）在整個通訊期間都被**預約保留**，傳送端能以受保障的固定速率傳輸。傳統電話網路是典型例子。連結中的迴路以兩種方式實作：

	FDM 分頻多工	TDM 分時多工
切什麼	切**頻譜**，每筆連線一段專用頻帶（頻帶寬度即頻寬）	切**時間**成時框，每個時框再切固定數目的時槽
每筆連線得到	全程專用一段頻帶	每個時框固定一個專用時槽

封包交換與電路交換比較

比較項目	封包交換	電路交換
資源分配	**依需求**分配，逐封包共用	**預先**分配，不管是否需要
閒置時	別人可使用連結容量	沉靜期迴路閒置，形成浪費
建立連線	不需要	需要，且須複雜的訊號軟體協調路徑上的交換器
延遲	不固定、不可預測（主因是佇列延遲）	固定速率，有保障
優點	頻寬分享較好、較簡單、較有效率、實作成本低	適合需要穩定品質的即時服務
缺點	不提供保障，可能壅塞與遺失	資源利用率低

網路的網路：ISP 結構如何一步步形成

1. **全部兩兩互連？** 數量級 $O(N^2)$，無法擴展。
2. **全部接到一家全球轉運 ISP**：連線 ISP 是**顧客**，全球 ISP 是**供應者**，依流量付費。
3. **有利可圖就會有競爭者**：出現多家全球 ISP，它們彼此也必須互連。
4. **區域 ISP 出現**：連線 ISP 接區域 ISP，區域 ISP 接第一層 ISP。
5. **IXP 與對等**：第三方建立網際網路互換點，讓多家 ISP 在此對等互連。
6. **內容供應者網路**：Google 等自建私有 TCP/IP 網路連接資料中心，盡量直接連低層 ISP，略過上層。

名詞	意義
第一層 ISP（tier-1）	類似全球轉運 ISP，但不必遍布每座城市；全球約十幾家（如 Level 3、AT&T、Sprint、NTT、Verizon）；**位於頂端，不付費給任何人**
PoP（存取點）	供應者網路中一組路由器，顧客 ISP 可在此接入，通常租用高速連結連到 PoP 的路由器
多點上層連接（multi-home）	連到**兩個或以上**的供應者 ISP（第一層以外的 ISP 都可這麼做）
對等（peer）	**同層**鄰近 ISP 直接互連，流量不必經過上游
IXP	多家 ISP 可在此對等的會合點，通常由第三方公司建立
內容供應者網路	如 Google、Microsoft、Akamai；把內容與服務帶到使用者附近，常略過第一層與區域 ISP

> **考點**　付費方向：連線 ISP → 區域 ISP → 第一層 ISP。第一層不付費給任何人。Google 仍會連到第一層 ISP，並就交換的流量付費。

1.4　延遲、遺失與產出率

重要度 ★★★　｜　來源：中 1 p.75–96；英 1 p.80–103

四種節點延遲

總節點延遲：$\mathbf{d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}}$

延遲	內容	公式	大小與取決因素
處理延遲 d_{proc}	檢查封包標頭、檢查位元錯誤、決定輸出連結	—	高速路由器約微秒級以下；通常可忽略，但會影響路由器最大產出率
佇列延遲 d_{queue}	在輸出佇列等待被傳出	統計量	微秒到毫秒；取決於壅塞程度，每個封包都可能不同
傳輸延遲 d_{trans}	把**整個封包的所有位元推進連結**的時間	$\mathbf{L / R}$	L：封包長度（bits）；R：連結速率（bps）；**與距離無關**
傳播延遲 d_{prop}	**一個位元**從連結一端傳到另一端的時間	$\mathbf{d / s}$	d：連結長度；s：傳播速率（約 2–3 × 10^8 m/s）；**與封包長度和速率無關**

> **易混淆**　**傳輸 vs 傳播**是本章最常考的觀念。傳輸延遲看「封包多長、連結多快」，傳播延遲看「距離多遠、媒介是什麼」。
> 何者主導要看情境：同校園內 d_{prop} 可忽略，同步衛星連結則達數百毫秒並成為主導項；d_{trans} 在 10 Mbps 以上通常可忽略，低速撥接傳大封包卻可達數百毫秒。

佇列延遲與流量強度

令 a＝封包平均抵達速率（封包/秒），L＝封包長度（bits），R＝傳輸速率（bps）。**流量強度（traffic intensity）＝ La / R**。

La/R	結果
接近 0	平均佇列延遲很小
接近 1	平均佇列延遲**急遽增加**，強度增加一點，延遲增加很多
大於 1	到達的工作量超過能服務的量，平均延遲趨於無限大（理論上）

- **黃金守則**：設計系統時，流量強度不可大於 1。
- 佇列延遲還取決於**流量的抵達型態**（週期性或突發）。實際抵達過程是隨機的，只看 La/R 不足以完整描述，因此常用平均值、變異數、超過某值的機率等統計量。
- 例子：10 個封包同時抵達空佇列，第一個沒有佇列延遲，最後一個延遲相對大。
- **封包遺失**：佇列容量有限，所以延遲不會真的到無限大；佇列滿時路由器丟棄封包。流量強度愈高，遺失比例愈高。遺失的封包可能由前一節點或來源重傳，也可能不重傳。

端對端延遲與 Traceroute

來源到目的地之間有 N−1 台路由器（N 條連結），網路暢通時（佇列可忽略）：**$d_{end\text{-}end} = N\,(d_{proc} + d_{trans} + d_{prop})$**。

- **Traceroute** 對路徑上每個路由器 i 送 **3 個探測封包**（TTL＝i）；路由器 i 回傳封包，來源量測往返時間。
- 輸出中的 * 代表沒有回應（探測遺失或路由器不回覆）。
- **後面一跳的延遲可能比前一跳小**，因為佇列延遲隨時間變動。
- 延遲突然大幅增加（如從 22 ms 跳到 104 ms）通常代表經過**跨洋連結**。
- 其他延遲：共用媒介的終端系統可能蓄意延後傳輸；VoIP 有**媒介封包化延遲**。

產出率（Throughput）

- **瞬間產出率**：某一時刻接收位元的速率。**平均產出率**：一段較長時間的速率。
- **瓶頸連結（bottleneck link）**：路徑上限制端對端產出率的連結。
- 兩段連結：產出率＝$\min\{R_s, R_c\}$。N 段連結：產出率＝**$\min\{R_1, R_2, \ldots, R_N\}$**。
- 今日網際網路的限制因素通常是**連線網路**；但共用核心連結不夠大時，瓶頸會移到核心。

> 例題　10 組下載共用核心連結 R，R_s＝2 Mbps，R_c＝1 Mbps，R＝5 Mbps。
> 每組分到 R/10＝0.5 Mbps，所以產出率＝min{2, 1, 0.5}＝**500 kbps**。瓶頸不在連線網路，而在核心的共用連結。
> 若 R 遠大於 R_s 與 R_c，每組產出率就是 $\min\{R_s, R_c\}$。

1.5　協定分層與服務模型

重要度 ★★★　｜　來源：中 1 p.97–118；英 1 p.105–124

每一層透過兩件事提供服務：**(1) 在該層內執行某些動作；(2) 使用下一層提供的服務**。各層協定合稱**協定堆疊（protocol stack）**。協定層可由軟體、硬體或兩者實作，並分散在終端系統、交換器等元件中。

為什麼要分層（優點）	分層的缺點
結構明確，方便討論系統各部分的關係	某一層可能重複下層功能（例如逐連結與端對端都做錯誤復原）
模組化，容易維護與更新；改某層實作不影響其他部分	某層功能可能需要只有另一層才有的資訊，破壞分層原則

網際網路五層協定堆疊

層	封包名稱	工作	協定例子	實作位置
應用層	**訊息** message	網路應用程式與應用層協定的所在	HTTP(S)、DNS、SMTP	軟體（終端系統）
傳輸層	**區段** segment	在應用程式端點之間傳送應用層訊息	TCP、UDP	軟體（終端系統）
網路層	**資料包** datagram	把資料包從一台主機移到另一台	IP（唯一）、繞送協定	軟硬體混合
連結層	**訊框** frame	把封包從一個節點移到下一個節點	乙太網路、WiFi、PPPoE	網路介面卡
實體層	位元 bit	把訊框中的個別位元移到下一個節點，依實際媒介而定	依雙絞線、光纖等	網路介面卡

傳輸層協定	服務
TCP	連線導向；保證送達；流量控制；切割長訊息；壅塞控制
UDP	無連線；最低限度服務：不提供可靠性、流量控制、壅塞控制

> 考點　**ISO/OSI 七層**多了**表現層**（讓應用程式解讀資料意義，如加密、壓縮）與**會議層**（同步、檢查點、復原）。網際網路堆疊沒有這兩層，若需要就由應用程式自行實作。
> **裝置實作哪幾層**：連結層交換器＝第 1–2 層（認得乙太網路位址）；路由器＝第 1–3 層（能執行 IP）；**主機＝全部五層**。這反映「把複雜度放在網路邊際」的架構觀點。

封裝（Encapsulation）

層	內容	名稱	標頭用途
應用層	M	訊息	—
傳輸層	H_t + M	區段	H_t：實作傳輸服務
網路層	H_n + H_t + M	資料包	H_n：含來源與目的位址
連結層	H_l + H_n + H_t + M	訊框	H_l：實作連結層服務

- 每層封包都有**標頭欄位**與**內容欄位（payload）**；內容通常是上一層的封包。
- 資料走下傳送端堆疊，經交換器（到第 2 層）、路由器（到第 3 層），再走上接收端堆疊。
- 實際更複雜：大訊息可能切成多個區段，區段也可能再切成多個資料包，接收端要重建。

1.6　面臨攻擊的網路

重要度 ★★☆　｜　來源：中 1 p.119–129（英文講義只在目錄列出）

威脅	作法	重點
惡意軟體	感染裝置後刪檔、裝間諜軟體竊取資料；大多能**自我複製**	受害主機可能被納入**殭屍網路（botnet）**，用來發垃圾信或 DDoS
DoS：弱點攻擊	送妥善編造的訊息給有弱點的應用程式或作業系統	利用程式缺陷
DoS：頻寬滿載	送洪水般的封包塞滿目標的連線連結	合法封包到不了伺服器
DoS：連線滿載	建立大量半開或全開的 TCP 連線	耗盡連線資源
DDoS	控制多個來源同時攻擊，常透過數千台殭屍主機	比單一來源 DoS **更難偵測與防範**
封包竊聽	**封包竊聽器（packet sniffer）**是記錄經過封包副本的被動接收器；有線與無線都可能	因為被動所以難偵測；最佳防禦與**密碼學**有關。Wireshark 就是這類工具
IP 詐騙	注入帶假來源位址的封包（IP spoofing）	對策：**端點認證**

網際網路會這麼脆弱，是因為它原本建立在「**一群互信的使用者連接到透明的網路**」的模型上，不需要安全性。

1.7　網際網路歷史

重要度 ★☆☆　｜　來源：中 1 p.130–137

時期	關鍵事件
1961–1972 封包交換的發展	電腦共用需求興起；**ARPAnet** 是第一個封包交換計算機網路，也是今日網際網路的直系祖先；主機間協定 NCP；1972 年 Ray Tomlinson 寫出第一個電子郵件程式
1972–1980 專屬網路與互連	TCP、UDP、IP 概念在 1970 年代末出現；**ALOHA** 是第一個多重存取協定
1980–1990 蓬勃發展	主機數由約 200 台增至數十萬台；**1983 年 1 月 1 日** TCP/IP 成為 ARPAnet 標準主機協定
1990 年代 爆炸性成長	四大殺手級應用：電子郵件、資訊網（Web）、即時訊息（ICQ）、點對點 MP3 分享（Napster）
新千禧	寬頻到府、高速無線、線上社交網路、Google 與微軟的私有網路、雲端執行應用程式

第 2 章　應用層

本章速覽　本章要帶走的七件事：① socket 與「IP＋埠號」定址；② TCP 與 UDP 服務差異及應用選擇；③ HTTP 訊息格式、方法與狀態碼；④ 三種 HTTP 連線的 RTT 計算；⑤ Cookie 與 Web 快取計算；⑥ DNS 階層、遞迴與迭代查詢、資源紀錄；⑦ CDN 如何靠 DNS 導向使用者。

補充　講義的目錄列出 FTP、電子郵件（SMTP、POP3、IMAP）、P2P、Socket 程式設計，但這份投影片**沒有這些內容**，本書因此不展開；實際講授的是 2.1 原理、Web/HTTP、DNS、CDN。考試範圍請向老師確認。

2.1　網路應用程式原理

重要度 ★★★　｜　來源：英 2 p.2–27

開發應用程式與主從式架構

網路應用程式寫在**終端系統**上，不必為網路核心裝置寫程式，因為核心裝置不執行使用者應用程式；這也是應用程式能快速開發與擴散的原因。

伺服器（server）	用戶端（client）
一直開機的主機 有固定（永久）IP 位址 常放在資料中心以利擴充	與伺服器聯絡通訊，可間歇連線 可能使用動態 IP 位址 用戶端之間**不直接通訊**

- 單一伺服器應付不了大量請求，因此以**資料中心**（可達數十萬台伺服器）組成強大的虛擬伺服器；搜尋引擎、電子商務、社群網路都如此。
- **白牌伺服器**：不掛傳統品牌，由代工廠依雲端業者需求客製（常依 OCP 開放運算計畫標準）；台灣主要廠商有廣達、緯穎、鴻海、緯創。

Socket 與行程定址

- **Socket** 是主機內**應用層與傳輸層之間的介面**，又稱應用程式與網路之間的 **API**，可比喻為「門」；每次通訊涉及兩端各一個 socket。
- 應用程式開發者控制 socket 應用層那一側；傳輸層那一側只能**選擇傳輸協定**，以及調整少數參數（如最大緩衝區、最大區段大小）。
- 識別接收行程需要兩項資訊：**IP 位址**（找主機）＋**埠號**（找行程），例如 128.119.245.12 的 80 埠。

傳輸層可提供的四類服務

服務	說明	應用
可靠資料傳輸	保證資料正確且完整送達	需要：電子郵件、檔案傳輸、Web 可容忍遺失（loss-tolerant）：多媒體
產出率	保證至少 r bits/sec	頻寬敏感：多媒體（如語音 32 kbps） 彈性（elastic）：電子郵件、檔案、Web
時間（Timing）	保證延遲上限，如 100 ms 內送達	網路電話、虛擬環境、視訊會議、多人遊戲
安全性	傳送端加密、接收端解密	—

應用	資料遺失	產出率	時間敏感
檔案傳輸／下載	不可遺失	彈性	否
電子郵件	不可遺失	彈性	否
Web 文件	不可遺失	彈性	否
即時影音	可容忍	音訊 5 Kbps–1 Mbps 視訊 10 Kbps–5 Mbps	是，數十 ms
串流影音	可容忍	同上	是，數秒
互動遊戲	可容忍	Kbps 以上	是，數十 ms
文字訊息	不可遺失	彈性	有些是、有些否

TCP 與 UDP

項目	TCP	UDP
連線	連線導向，須先建立連線	不需建立連線
可靠性	可靠傳輸	不可靠
流量控制	有（傳送端不會淹沒接收端）	無
壅塞控制	有（網路過載時抑制傳送端）	無
都不提供	時間保證、最低產出率保證、安全性	時間保證、產出率保證、安全性

應用	應用層協定	傳輸協定
檔案傳輸	FTP	TCP
電子郵件	SMTP	TCP
Web	HTTP	TCP
網路電話	SIP、RTP 或專有協定	TCP 或 UDP
串流影音	HTTP、DASH	TCP
互動遊戲	WOW、FPS（專有）	UDP 或 TCP

TLS：替 TCP 加上安全性

- 一般 TCP、UDP socket **沒有加密**，明文密碼會以明文穿越網際網路。
- **TLS** 提供加密的 TCP 連線、資料完整性、端點認證。
- TLS 實作在**應用層**：應用程式使用 TLS 函式庫，函式庫再使用 TCP。送進「socket」的明文，在網際網路上是加密的。

應用層協定的定位

- 公開協定（如 HTTP）定義於 RFC，照規格實作就能互通；也有**專有協定**（如 LINE）不公開。
- 應用層協定只是應用程式的一部分。Web 包含文件格式（HTML5）、瀏覽器、Web 伺服器、HTTP。

> 思考題　思考題（講義 p.23–25、p.65）：選兩種應用分析傳輸需求並選擇 TCP 或 UDP。分析順序：遺失容忍度 → 產出率 → 時間敏感度 → 安全性 → 協定選擇。

應用	遺失	產出率	時間	建議與理由
檔案傳輸	不可	彈性	不敏感	**TCP**：內容必須完整，可晚一點到
視訊會議	可容忍	敏感	數十 ms	**UDP**：寧可畫面小瑕疵，也不要等重傳
多人遊戲	可容忍	Kbps 以上	數十 ms	狀態更新用 **UDP**；登入、交易等關鍵資料可用 TCP
電子郵件	不可	彈性	不敏感	**TCP**：內容完整優先
即時通訊	文字不可	彈性	部分敏感	文字用 **TCP**；語音或視訊通話可用 UDP

> 補充　上表是依講義需求表推導的參考答案，不是講義原文。作答時把「需求 → 選擇」的理由寫出來，比只寫協定名稱重要。

2.2　Web 與 HTTP

重要度 ★★★　｜　來源：英 2 p.29–83（第四次小考標記在 p.84）

基本名詞

- **HTTP** 是 Web 的應用層協定，以用戶端與伺服器兩支程式實作，定義訊息結構與交換方式。
- **網頁由物件組成**：物件就是檔案（HTML、JPEG、影片等），每個物件以一個 URL 定址。基本 HTML 加 5 張圖＝**6 個物件**。
- **URL ＝主機名稱＋路徑名稱**：`www.someschool.edu/someDept/pic.gif`。
- **HTTP 使用 TCP**：用戶端先建立 TCP 連線，雙方透過 socket 收送；訊息送進 socket 後就交給 TCP 負責。
- **HTTP 是無狀態協定（stateless）**：伺服器不保存用戶端的狀態資訊。

HTTP 與 HTTPS

	HTTP	HTTPS
URL 開頭	`http://`	`https://`
加密	無	有，以 **TLS**（前身 SSL）加密請求與回應
埠號	**80**	**443**
握手	TCP 3-way 握手（1 RTT）後送請求	TCP 握手（1 RTT）＋TLS 握手後才送加密資料

> 補充　講義 p.37 的圖以 TLS 1.2 為例：TCP 握手後依序是 Client Hello → Server Hello／憑證／Server Hello Done，再來是 Client Key Exchange／Change Cipher Spec／Finished → Change Cipher Spec／Finished，約多出 2 個 RTT。握手以**非對稱加密**交換 session key，之後的資料以**對稱加密**傳送。

HTTP 訊息格式

```
GET /index.html HTTP/1.1\r\n            ← 請求列（方法 URL 版本）
Host: www-net.cs.umass.edu\r\n          ← 標頭列
User-Agent: Mozilla/5.0 ...\r\n
Accept-Language: en-us,en;q=0.5\r\n
Connection: keep-alive\r\n
\r\n                                    ← 空行代表標頭結束

HTTP/1.1 200 OK                         ← 狀態列（版本 狀態碼 片語）
Date: Tue, 08 Sep 2020 00:53:20 GMT
Last-Modified: Tue, 01 Mar 2016 18:57:50 GMT
Content-Length: 2651
Content-Type: text/html; charset=UTF-8
\r\n
data data data ...                      ← 實體內容
```

方法	狀態碼
GET：取物件；也可把使用者資料放 URL 中 ? 之後 **POST**：表單輸入放在實體內容 **HEAD**：只回傳標頭 **PUT**：上傳檔案，取代指定 URL 的內容 **DELETE**：刪除指定 URL 的檔案	**200** OK：成功 **301** Moved Permanently：已搬移，新位置在 Location **304** Not Modified：未修改（條件式 GET） **400** Bad Request：伺服器看不懂請求 **404** Not Found：找不到文件 **505** HTTP Version Not Supported

三種 HTTP 連線

RTT（round-trip time）：一個小封包從用戶端到伺服器再回來的時間，包含前面談過的所有延遲。

	非持續性	持續性、無管線化	持續性、有管線化
TCP 連線	**每個物件一條**	共用一條	共用一條
送請求時機	—	收到前一個回應才送下一個	遇到物件就送，**不等回應**
每個物件成本	2 RTT ＋ 傳輸時間	1 RTT	全部物件合計約 1 RTT
n 個參照物件（連 HTML，忽略傳輸時間）	**2(n＋1) RTT**	**(2＋n) RTT**	**3 RTT**

> 例題　網頁含 1 個 HTML 與 3 個物件（講義 p.59–60），忽略傳輸時間：
> 非持續性：(RTT＋RTT) × 4 ＝ **8 RTT**
> 持續性無管線化：(RTT＋RTT)＋3 RTT ＝ **5 RTT**
> 持續性有管線化：(RTT＋RTT)＋RTT ＝ **3 RTT**
> 同樣的公式：HTML＋10 張 JPEG，非持續性共開 **11 條 TCP 連線**，需 22 RTT。

- **非持續性的問題**：每個物件 2 RTT；兩端都要配置 TCP 緩衝區與變數，對伺服器負擔大。瀏覽器常開平行連線來緩解。
- **持續性連線**：伺服器回應後保留連線，一段時間未使用才關閉；請求標頭 `Connection: Keep-Alive` 可重用連線，降低伺服器負擔與延遲。
- **管線化**：回應必須依請求順序回傳（先進先出），會造成 **HOL Blocking**（隊頭阻塞）。POST 不應管線化，GET／HEAD 通常可以。
- **實務**：規格上 HTTP/1.1 預設支援管線化，但多數瀏覽器預設關閉。依慣例，單一使用者對同一伺服器不應維持超過 2 條連線。

Cookie：保存使用者狀態

HTTP 無狀態，但網站常需要辨識使用者以限制存取或提供個人化內容。Cookie 有**四個元件**：

1. HTTP **回應**中的 cookie 標頭列（`Set-Cookie: 1678`）
2. 下一次 HTTP **請求**中的 cookie 標頭列（`Cookie: 1678`）
3. 使用者主機上由瀏覽器管理的 **cookie 檔案**
4. 網站後端的**資料庫**

流程：首次造訪時伺服器產生 ID 並寫入資料庫，以 Set-Cookie 回傳；瀏覽器存入 cookie 檔；之後每次請求都帶上 Cookie，伺服器據此採取對應動作。使用者註冊後，網站可把 ID 連到個人資料，這就是「一鍵購物」的基礎。

Web 快取（代理伺服器）

代表原始伺服器回應 HTTP 請求的網路實體，有自己的磁碟並保存最近請求物件的副本。

1. 瀏覽器與快取建立 TCP 連線，送出請求。
2. **有副本**：快取直接回傳。

3. **沒有副本**：快取對原始伺服器開 TCP 連線、發請求，收到物件後**存一份**，再回給瀏覽器。

好處：縮短回應時間（快取較近）；減少機構對外連結流量；讓資源較少的內容供應者也能有效傳遞內容。快取同時是伺服器（對用戶端）也是用戶端（對原始伺服器）。

範例一（p.74–77）	原始情況	方案 1：加速連結	方案 2：裝快取
條件	物件 1 Mb，15 r/s，連結 15 Mbps，LAN 100 Mbps，RTT 2 s	連結升級到 100 Mbps	命中率 0.4
LAN 使用率	15 ÷ 100 ＝ 15%	15%	15%
對外連結使用率	15 ÷ 15 ＝ **100%**	15 ÷ 100 ＝ **15%**	0.6 × 100% ＝ **60%**
總延遲	2 s ＋ **數分鐘** ＋ ms	2 s ＋ ms ＋ μs	0.6 × 2 s ＋ 0.01 s ＋ ms，約 **1.2 s**
成本	—	昂貴	便宜

範例二（p.78–81）	原始情況	方案 1：加速連結	方案 2：裝快取
條件	物件 100 Kb，15 r/s（1.50 Mbps），連結 1.54 Mbps，LAN 1 Gbps，RTT 2 s	連結升級到 154 Mbps	命中率 0.4
LAN 使用率	0.0015	0.0015	—
對外連結使用率	1.50 ÷ 1.54 ＝ **0.97**	1.50 ÷ 154 ＝ **0.0097**	0.9 ÷ 1.54 ＝ **0.58**
平均延遲	2 s ＋ **數分鐘** ＋ μs	2 s ＋ ms ＋ μs	0.6 × 2.01 ＋ 0.4 × ms，約 **1.2 s**
成本	—	昂貴	便宜，延遲還比方案 1 低

> 易混淆　為什麼使用率 0.97 就會延遲數分鐘？回到第 1 章的流量強度曲線：La/R 接近 1 時佇列延遲急遽增加。解題步驟固定為：① 算流量（請求率 × 物件大小）；② 算使用率；③ 判斷佇列延遲量級；④ 加總延遲。

條件式 GET

快取的副本可能過期，**條件式 GET** 讓快取確認物件是否為最新。請求帶 `If-modified-since: <date>`：

- 該日期後**沒改過**：回 `304 Not Modified`，**不含物件**，節省傳輸。
- 該日期後**改過**：回 `200 OK` 並附上資料。

2.3　DNS 網域名稱系統

重要度 ★★★　｜　來源：英 2 p.86–114（第五次小考標記在 p.115）

DNS 是什麼

- 把**主機名稱轉成 IP 位址**：人好記名稱，IP 又難記又可能變動。
- 同時是**階層式 DNS 伺服器構成的分散式資料庫**，也是讓主機查詢該資料庫的**應用層協定**。
- **使用 UDP，埠號 53**；常見軟體為 BIND。HTTP、SMTP、FTP 都會用它。
- 瀏覽器流程：取出主機名稱 → DNS 用戶端查詢 → 收到 IP → 對該 IP 的 80 或 443 埠建立 TCP 連線。DNS 會增加額外延遲。
- 查詢前會先看**瀏覽器快取、作業系統 DNS 快取、hosts 檔案**。

為什麼不用單一伺服器

問題	說明
單點失效	伺服器掛掉，整個網際網路都受影響
流量龐大	所有 HTTP 請求與電子郵件的查詢都集中到一台
距離遙遠	無法靠近所有查詢者，延遲大
維護困難	資料庫龐大且須為每台新主機頻繁更新

網域名稱結構

網域名稱以點分隔，**由右往左讀**。`mcu.edu.tw`：tw 是 TLD、edu 是第二層、mcu 是子網域。**FQDN** 是特定主機的完整名稱，由主機名稱與網域名稱組成，如 `www.mcu.edu.tw`。

DNS 伺服器階層

類別	職責	例子
根 DNS	本地伺服器解析不了時聯絡它；回覆 TLD 伺服器位置	全球 **13 個邏輯**根伺服器（a–m），各自有大量複本
TLD DNS	負責頂級網域與國家網域	.com、.org、.net、.edu、.tw、.jp
授權（權威）DNS	組織自己的 DNS，提供該組織主機名稱到 IP 的權威對映；可自行維護或委外	dns.cs.umass.edu；大學通常自建主要與次要伺服器
本地 DNS（預設名稱伺服器）	**不嚴格屬於階層**；每個 ISP 都有；靠近主機；像代理一樣把查詢送進階層	dns.nyu.edu；公共 DNS 如 8.8.8.8

遞迴與迭代查詢

	遞迴查詢（recursive）	迭代查詢（iterative）
意義	請對方**代為取得**答案	對方回覆「我不知道，但你去問這台」
典型位置	**主機 → 本地 DNS**	**本地 DNS → 根、TLD、授權**

> 例題　engineering.nyu.edu 的主機查 gaia.cs.umass.edu（講義 p.101）：
> 1 主機 → 本地 DNS（dns.nyu.edu）　2 本地 → 根　3 根回 TLD 位址
> 4 本地 → TLD　5 TLD 回授權伺服器位址
> 6 本地 → 授權（dns.cs.umass.edu）　7 授權回 IP　8 本地回主機
> 共 **8 個訊息**。若本地 DNS 已快取 TLD 位址，就能**略過根伺服器**。

DNS 快取

- 收到回覆就把對映存入記憶體，可改善延遲並減少 DNS 訊息；一段時間後（TTL）丟棄。
- Windows 指令：`ipconfig /displaydns`（顯示快取）、`ipconfig /flushdns`（清除快取）。

資源紀錄（RR）

格式：**(name, value, type, ttl)**。TTL 決定資源紀錄何時從快取移除。

Type	name	value	例子
A	主機名稱	IP 位址	把主機名稱對映到 IP
NS	網域	該網域授權 DNS 的主機名稱	(foo.com, dns.foo.com, NS)
CNAME	別名	正規（真實）名稱	(foo.com, relay1.bar.foo.com, CNAME)
MX	別名	郵件伺服器的正規名稱	(foo.com, mail.bar.foo.com, MX)

> 考點　同一別名可同時用於郵件伺服器與其他伺服器：查郵件伺服器用 MX，查其他伺服器用 CNAME。

DNS 訊息格式

查詢與回覆格式相同：標頭加四個區段。**identification** 是 16 位元編號，回覆沿用查詢的編號；**flags** 標示查詢或回覆、是否希望遞迴、是否可遞迴、是否為授權回覆。

區段	內容
questions	查詢的 name、type
answers	回應查詢的 RR
authority	授權伺服器的紀錄
additional	其他有幫助的資訊

實作工具

```
nslookup
> server 10.255.255.255     指定要查詢的 DNS 伺服器
> set q=mx                  查 MX 紀錄
> contoso.com.
```

知名公共 DNS：Google `8.8.8.8`、`8.8.4.4`；Cloudflare `1.1.1.1`、`1.0.0.1`；OpenDNS `208.67.222.222`、`208.67.220.220`；AdGuard `94.140.14.14`、`94.140.15.15`。

2.4　內容傳遞網路 CDN

重要度 ★★☆　｜　來源：英 2 p.116–133

若只蓋一座巨型資料中心串流影片到全球，一旦資料中心或對外連結故障就無法服務，因此主要影音業者都使用 CDN：**把內容快取在比原始伺服器更靠近使用者的代理伺服器**。Web 主機保存原始檔案，CDN 快取這些檔案讓全球使用者更容易取得。

- **第三方 CDN**：代多個內容供應者傳遞內容，如 Akamai、Level-3、Cloudflare。
- **伺服器叢集（server cluster）**：在同一 IP 下共同運作以提高可用性的一組伺服器。
- **類似 Web 快取**：叢集沒有該影片時，向中央儲存庫或其他叢集取得，**邊存邊串流**；儲存滿了就移除不常被請求的影片。

CDN 運作

CDN 必須攔截請求，才能 (1) 為該用戶端選擇合適的叢集，(2) 把請求導向該叢集的伺服器。**多數 CDN 利用 DNS 攔截與導向**。

1. 瀏覽器向本地（遞迴）DNS 查詢 shop.example.com。
2. 本地 DNS 向 Cloudflare 的授權 DNS 查詢（假設沒有快取，根與 TLD 步驟省略）。
3. 授權 DNS 回覆 Cloudflare Anycast IP，本地 DNS 回傳 IP 並依 TTL 快取。
4. 瀏覽器以 HTTPS 連到 Cloudflare CDN；必要時 CDN 才回源到原始伺服器。

叢集選擇策略

策略	作法
地理最近	收到某本地 DNS 的請求時，選地理上最近的叢集
即時量測	定期量測叢集與用戶端之間的延遲與遺失，例如從各叢集對全球本地 DNS 送探測封包

案例

- **Netflix**：影片多個版本的副本事先上傳到 CDN 伺服器。觀看時：① 使用者在 Amazon 雲端上的註冊、帳務伺服器管理帳號；② 瀏覽影片；③ 取得該影片的 **manifest 檔**；④ 選定 DASH 伺服器並開始串流。
- **Canva**：Cloudflare 官方公開確認 Canva 使用其 CDN、WAF 與 DDoS 防護。
- **動手查**：講義 p.133 要求查看 www.tsmc.com.tw 的網站，可用 nslookup 觀察其 CNAME 是否指向 CDN。

面向	Cloudflare	Akamai	台灣本地業者
導入方式	DNS 代理，相對容易	多依企業需求規劃	可由本地團隊協助
入門成本	有免費方案	多以企業報價為主	依流量、頻寬與合約
適合對象	個人、中小企業到大型組織	大型企業、媒體、關鍵服務	主要市場在台灣的企業

第 4 章　網路層：資料層

> **本章速覽**　本章計算題最多。要帶走的七件事：① 轉送與繞送、資料層與控制層；② 最長前綴比對；③ IP 標頭每個欄位；④ IP 分割的長度、Offset、MF 計算；⑤ Checksum；⑥ CIDR 與子網路切割；⑦ 特殊位址、NAT、DHCP 四步驟。

> **補充**　講義沒有第 3 章（傳輸層）的檔案，因此本書從第 2 章直接接到第 4 章，章號維持與課本一致。

4.1　網路層概觀

重要度 ★★☆　｜　來源：英 4 p.2–8、p.15–17

網路層把區段從傳送主機送到接收主機：傳送端把區段封裝成資料包，接收端把區段交給傳輸層。**每台主機與路由器都有網路層**；路由器檢查所有經過的 IP 資料包標頭，並把資料包從輸入埠移到輸出埠。

	資料層（Data plane）	控制層（Control plane）
範圍	**區域**、每台路由器	**全網路**的邏輯
決定什麼	資料包如何從輸入埠轉送到輸出埠	資料包如何沿端對端路徑在路由器間繞送
對應功能	**轉送**（forwarding）	**繞送**（routing）
時間尺度	硬體，**奈秒**級	軟體，**毫秒**級

傳統逐路由器控制層	SDN 軟體定義網路控制層
每台路由器內都有繞送演算法元件，彼此互動	**遠端控制器**（伺服器）計算轉送表並安裝到路由器；路由器上有控制代理（CA）

服務模型

架構	服務模型	頻寬	遺失	順序	時間
Internet	**best effort**	無	無保證	無保證	無保證
ATM	Constant Bit Rate	固定速率	有	有	有
ATM	Available Bit Rate	保證最小	無	有	無
Internet	Intserv Guaranteed	有	有	有	有
Internet	Diffserv	可能	可能	可能	無

> **考點**　**IP 是不可靠、無連線的資料包協定，提供 best-effort 服務**：不保證送達、不保證時間與順序、不保證頻寬。封包可能損毀、遺失、亂序或延遲。每個資料包獨立處理，可走不同路徑，所以可能亂序。要可靠就搭配 TCP。

網路層的組成：**IP 協定**（資料包格式、定址、封包處理慣例）、**ICMP**（錯誤回報、路由器訊號）、**路徑選擇演算法**（繞送協定 OSPF、BGP 或 SDN 控制器）及轉送表。

4.2　路由器與最長前綴比對

重要度 ★★★　｜　來源：英 4 p.9–13

路由器由**輸入埠、高速交換結構（switching fabric）、輸出埠、繞送處理器**組成。**交換速率**是封包從輸入移到輸出的速率；N 個輸入、每個線路速率 R 時，理想交換速率為 **NR**。

> **考點**　**最長前綴比對**：查轉送表時，使用與目的位址相符的**最長**位址前綴。

目的位址範圍	連結介面
`11001000 00010111 00010*** ********`	0
`11001000 00010111 00011000 ********`	1
`11001000 00010111 00011*** ********`	2
其他	3

> **例題**　目的位址 `11001000 00010111 00010110 10100001`：只符合第 1 列 → **介面 0**。
> 目的位址 `11001000 00010111 00011000 10101010`：同時符合第 2 列（24 位元）與第 3 列（21 位元），取最長 → **介面 1**。

4.3　IP 資料包格式

重要度 ★★★　｜　來源：英 4 p.14–37、p.51–53

資料包總長 20–65,535 bytes，其中標頭 20–60 bytes。

<table>
<tr><td>VER
4 bits</td><td>HLEN
4 bits</td><td colspan="2">Service type
8 bits</td><td colspan="2">Total length
16 bits</td></tr>
<tr><td colspan="3">Identification
16 bits</td><td>Flags
3 bits</td><td colspan="2">Fragmentation offset
13 bits</td></tr>
<tr><td colspan="2">Time to live
8 bits</td><td colspan="2">Protocol
8 bits</td><td colspan="2">Header checksum
16 bits</td></tr>
<tr><td colspan="6">Source IP address（32 bits）</td></tr>
<tr><td colspan="6">Destination IP address（32 bits）</td></tr>
<tr><td colspan="6">Options ＋ padding（0–40 bytes）</td></tr>
</table>

欄位	長度	重點
Version	4 bits	目前為 4；版本不符就丟棄，避免誤解讀
Header length	4 bits	**單位是 4 bytes**。無選項：值 5（20 bytes）；最大值 15（60 bytes）
Service type	8 bits	現在定義差異化服務（前 6 bits Codepoint，後 2 bits 未使用）
Total length	16 bits	**單位是 byte**，標頭＋資料，最大 65,535
Identification	16 bits	與來源 IP 共同唯一識別一個資料包；同一資料包的分割片相同
Flags	3 bits	保留位元、**DF**（1＝不可分割）、**MF**（1＝後面還有分割片；0＝最後或唯一一片）
Fragment offset	13 bits	該片資料在原資料包的相對位置，**單位是 8 bytes**
TTL	8 bits	每經一台路由器減 1，到 0 就丟棄。Windows 預設 128，Unix 類預設 64
Protocol	8 bits	上層協定：**1 ICMP、2 IGMP、6 TCP、17 UDP、89 OSPF**
Header checksum	16 bits	偵錯用
Source／Destination	各 32 bits	來源與目的 IP

易混淆　三種單位最常被搞混：**Header length × 4 bytes**、**Total length 直接是 bytes**、**Offset × 8 bytes**。TTL 例子：TTL＝3 進 R1，出去變 2；進 R2 出去變 1；到 R3 減為 0 就丟棄。

MTU 與 IP 分割

協定	MTU
Ethernet	1500 bytes（巨型訊框最大 9000 bytes）
PPPoE	1492 bytes
WiFi	2304 bytes

- **MTU**：連結層訊框資料欄位的最大值，資料包必須不超過它。巨型訊框依廠商而異、不保證能穿越網際網路，多用於 SAN 等專用網路。
- 資料包大於下一段 MTU 時要**分割**成更小的資料包。
- **分割**可由來源主機或路徑上任何路由器進行；**重組只在目的主機**，因為每個分割片都是獨立的資料包。
- 分割依靠 **Identification**、**Flags**、**Fragmentation offset** 三個欄位；其餘欄位多半照抄（Total length 與 Checksum 會依分割片重算）。

公式　**分割計算四步驟**
① 每片可放資料 ＝ MTU − 標頭 20，且除了最後一片外，**必須是 8 的倍數**。
② 片數 ＝ 總資料量 ÷ 每片資料量，無條件進位。
③ 第 k 片 Offset ＝ 前面所有片的資料量 ÷ 8。
④ 最後一片 MF＝0，其餘 MF＝1；Total length ＝資料量＋20。

例題　IP 資料 5000 bytes（總長 5020），MTU 1500（講義 p.40–42）。每片資料 1480 bytes，Offset 每片加 185。

分割片	標頭	資料	Offset	MF	Total length	資料範圍
原始	20	5000	0	0	5020	0–4999
第 1 片	20	1480	**0**	1	1500	0–1479
第 2 片	20	1480	**185**	1	1500	1480–2959
第 3 片	20	1480	**370**	1	1500	2960–4439
第 4 片	20	560	**555**	**0**	580	4440–4999

> 例題　分割片還能再被分割（講義 p.38）。原資料 4000 bytes，ID＝14,567，Total length＝4020。

分割片	資料範圍	Total length	MF	Offset
第 1 片	0–1399	1420	1	000
第 2 片	1400–2799	1420	1	175
第 3 片	2800–3999	1220	0	350
第 2.1 片	1400–2199	820	1	175
第 2.2 片	2200–2799	620	1	275

> 易混淆　第 2.2 片雖然是第 2 片再分割的最後一塊，但**它後面還有第 3 片**，所以 MF 仍為 1。Offset 一律相對於**原始**資料包計算。

> 實作　`ping -l 1600 www.twnic.net.tw`：ICMP 標頭 8＋資料 1600＝1608 bytes。乙太網路 MTU 1500 下分成 2 片，資料量為 **1480＋128**，Offset 為 0、185。
> `ping -l 3200 www.mcu.edu.tw`：8＋3200＝3208 bytes，分成 3 片，資料量為 **1480＋1480＋248**，Offset 為 0、185、370。
> Wireshark 篩選所有分割封包：`ip.flags.mf == 1 or ip.frag_offset > 0`

Checksum 計算

1. **傳送端**：Checksum 欄先填 0，把標頭每 16 位元一組相加（溢位要繞回加到最低位），再取 **1 的補數**填入。
2. **接收端**：把包含 Checksum 的整個標頭相加再取補數，結果為 **0** 代表無誤，否則丟棄。

欄位值	16 位元字組
4, 5, 0	`01000101 00000000`
Total length＝28	`00000000 00011100`
Identification＝1	`00000000 00000001`
Flags, Offset＝0, 0	`00000000 00000000`
TTL＝4, Protocol＝17	`00000100 00010001`
Checksum 先填 0	`00000000 00000000`
來源 10.12.	`00001010 00001100`
來源 14.5	`00001110 00000101`
目的 12.6.	`00001100 00000110`
目的 7.9	`00000111 00001001`
加總	`01110100 01001110`
Checksum（取補數）	`10001011 10110001`（十進位 35761）

接收端把這個 Checksum 一起加總，得到 `11111111 11111111`，取補數為 0，表示正確。

4.4　IP 定址

重要度 ★★★　｜　來源：英 4 p.54–71

基本概念

- **IP 位址**：32 位元，對應到主機或路由器的**介面**；路由器通常有多個介面，主機通常有一或兩個（有線、無線）。
- 點分十進位：`223.1.1.1 = 11011111 00000001 00000001 00000001`。
- **遮罩**：網路位元全 1、主機位元全 0。IP **AND** 遮罩＝網路位址：`200.2.3.4 AND 255.255.255.0 = 200.2.3.0`。
- **CIDR**：`a.b.c.d/x`，x 是子網路部分位元數，可為任意長度。

例子	遮罩	前綴長	後綴長	位址數
12.23.24.78/8	255.0.0.0	8	24	2^{24}
130.11.232.156/16	255.255.0.0	16	16	65,536
167.199.170.82/27	255.255.255.224	27	5	32

前綴	位址範圍
0.0.0.0/0	0.0.0.0 – 255.255.255.255
128.0.0.0/1	128.0.0.0 – 255.255.255.255
128.0.0.0/24	128.0.0.0 – 128.0.0.255
198.128.128.192/27	198.128.128.192 – 198.128.128.223
165.195.130.107/32	只有 165.195.130.107

> 公式　**/n 區塊速算**
> 位址總數 ＝ $2^{(32-n)}$；可分配給主機的數量 ＝ 總數 − 2。
> **第一個位址**是網路位址，**最後一個位址**是直接廣播位址，都不能指派給主機。
> 遮罩最後一個非 255 位元組：/25＝128、/26＝192、/27＝224、/28＝240、/29＝248、/30＝252。
> 區塊大小 ＝ 256 − 該位元組值，網路位址一定是區塊大小的倍數。

子網路切割

把主機位元改成網路位元，就能把一個區塊切成多個子網路：**多借 k 個位元，得到 2^k 個子網路**。

> 例題　130.34.12.64/26，要切成 4 個子網路（講義 p.67–68）。
> 後綴 6 位元，共 64 個位址；要 4 個子網路就借 2 位元，前綴變 **/28**，每個子網路 16 個位址。

子網路	網路位址	範圍
1	130.34.12.64/28	130.34.12.64 – 130.34.12.79
2	130.34.12.80/28	130.34.12.80 – 130.34.12.95
3	130.34.12.96/28	130.34.12.96 – 130.34.12.111
4	130.34.12.112/28	130.34.12.112 – 130.34.12.127

> 例題　ISP 擁有 200.23.16.0/20，切成 8 個 /23 給組織：Organization 0＝200.23.16.0/23、1＝200.23.18.0/23、2＝200.23.20.0/23……7＝200.23.30.0/23。每個區塊 512 個位址，第三個位元組每次加 2。

交換器、路由器與預設閘道

	Layer 2 交換器	路由器
層級	連結層	網路層
依據	**MAC 位址**轉送訊框	**IP 位址**繞送資料包
網段	連接同一子網路的裝置，裝置間不需路由器	**每個介面屬於不同 IP 子網路**

預設閘道（default gateway）是讓主機能連到其他網路的路由器。

4.5　特殊位址、路由聚合與私有位址

重要度 ★★★　｜　來源：英 4 p.74–83、p.89

位址	名稱	用途
0.0.0.0/32	全 0 位址	主機還不知道自己的 IP 時使用，通常在開機階段，如 DHCP discover 的來源
255.255.255.255/32	受限廣播	傳給同網路所有主機，如 DHCP discover 的目的地
區塊最後一個位址	直接廣播	不可指派給個別電腦
區塊第一個位址	網路位址	通常不指派給電腦
127.0.0.0/8	迴路位址	測試本機軟體，封包**不會離開機器**

路由聚合（階層式定址）

CIDR 的優點之一：多個位址區塊可聚合成一個並配給一家 ISP，只要對外通告一條路由。

- Fly-By-Night-ISP 對外說：「把開頭為 200.23.16.0/20 的都給我」，涵蓋 Organization 0–7。
- ISPs-R-Us 對外說：「把開頭為 199.31.0.0/16 的都給我」。
- **Organization 1 換到 ISPs-R-Us 後**，ISPs-R-Us 再多通告 200.23.18.0/23。這條路由**更具體**，依最長前綴比對，流量會走 ISPs-R-Us。

私有位址

區塊	位址數
10.0.0.0/8	16,777,216
172.16.0.0/12	1,048,576（講義誤植為 1,047,584，見附錄 E）
192.168.0.0/16	65,536
169.254.0.0/16	65,536

私有位址不被全球路由辨識，只能單獨使用或搭配 NAT。

ISP 的位址從哪裡來

- **ICANN** 透過 **5 個區域註冊機構（RR）**分配 IP 位址，RR 可再分配給地區註冊機構；ICANN 也管理 DNS 根區域。
- ICANN 在 **2011 年**把最後一批 IPv4 位址配給 RR。**NAT** 緩解位址耗盡，**IPv6** 則有 **128 位元**位址空間。

4.6　NAT 網路位址轉換

重要度 ★★★　｜　來源：英 4 p.84–88

NAT 在私有與全球位址之間提供對映，讓內部使用私有位址、對外使用少數全球位址。站點必須透過一台具 NAT 能力的路由器連到網際網路。

- **外送**：路由器把來源位址（與埠號）換成全球 NAT 位址。
- **內送**：路由器把目的位址（與埠號）換回對應的私有位址。
- **轉換表**記錄兩邊的對映，以埠號區分內部不同的連線，所以**一個 IP 可供多台主機共用**。

步驟	封包（LAN 主機 192.168.1.1，NAT 對外 IP 200.1.1.2，伺服器 202.1.1.1）	說明
1	S: 192.168.1.1:50000　D: 202.1.1.1:80	主機送出
2	S: 200.1.1.2:49000　D: 202.1.1.1:80	NAT 改來源並寫入表
3	S: 202.1.1.1:80　D: 200.1.1.2:49000	回應抵達 NAT
4	S: 202.1.1.1:80　D: 192.168.1.1:50000	NAT 查表改目的後轉給主機

LAN 側	WAN 側
192.168.1.1 : 50000	200.1.1.2 : 49000

4.7　DHCP 動態主機設定協定

重要度 ★★★　｜　來源：英 4 p.90–95

主機加入網路時從伺服器動態取得 IP 位址：可續租；只在連線時持有，所以位址能重用；適合行動使用者。DHCP 伺服器通常與路由器放在一起，服務路由器連接的所有子網路。

訊息	來源 IP：埠	目的 IP：埠	內容
1 Discover	0.0.0.0:68	255.255.255.255:67	有 DHCP 伺服器嗎？yiaddr＝0.0.0.0，ID＝654
2 Offer	223.1.2.5:67	255.255.255.255:68	可用這個位址：yiaddr＝223.1.2.4，ID＝654，lifetime 3600 秒
3 Request	0.0.0.0:68	255.255.255.255:67	我要用這個位址，ID＝655
4 ACK	223.1.2.5:67	255.255.255.255:68	位址給你了，ID＝655

> 考點　**四步驟口訣 DORA**：Discover → Offer → Request → ACK，四個訊息都是廣播；前兩步在用戶端要重用先前位址時可省略。
> **DHCP 除了 IP 還給什麼**：第一跳路由器位址、DNS 伺服器名稱與 IP、（子網路遮罩）。
> **封裝**：DHCP 在 UDP 內，UDP 在 IP 內，IP 在乙太網路內；訊框廣播目的為 FFFFFFFFFFFF。伺服器埠 67、用戶端埠 68。

附錄 A　公式總表

名稱	公式	備註
傳輸延遲	$d_{trans} = L / R$	L 位元、R bps；注意 byte 要乘 8
傳播延遲	$d_{prop} = d / s$	s 約 2–3 × 10^8 m/s
節點延遲	$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$	
端對端延遲（無壅塞）	$d_{end\text{-}end} = N (d_{proc} + d_{trans} + d_{prop})$	N 條連結、N−1 台路由器
流量強度	La / R	不可大於 1
端對端產出率	$\min\{R_1, R_2, \ldots, R_N\}$	共用連結要先除以連線數
OC-n 速率	n × 51.8 Mbps	
非持續性 HTTP	2(n＋1) RTT	n 為參照物件數，忽略傳輸時間
持續性、無管線化	(2＋n) RTT	
持續性、有管線化	3 RTT	
單一物件回應時間	2 RTT ＋ 檔案傳輸時間	非持續性
連結使用率	請求率 × 物件大小 ÷ 連結速率	有快取時再乘（1 − 命中率）
快取平均延遲	(1 − h) × 原始伺服器延遲 ＋ h × 快取延遲	h 為命中率
IP 標頭長度	HLEN × 4 bytes	HLEN 5–15
IP 資料長度	Total length − HLEN × 4	
資料起始位置	Offset × 8 bytes	
區塊位址數	$2^{(32-n)}$	可用主機數再減 2
切割子網路	借 k 位元得 2^k 個	新前綴 ＝ n ＋ k

附錄 B　中英術語對照

英文	中文	英文	中文
host / end system	主機／終端系統	packet switch	封包交換器
access network	連線網路	edge router	邊際路由器
transmission rate	傳輸速率	physical medium	實體媒介
guided media	導引式媒介	unguided media	非導引式媒介
store-and-forward	儲存轉送	output buffer	輸出緩衝區
forwarding	轉送	routing	繞送
circuit switching	電路交換	packet switching	封包交換
processing delay	處理延遲	queuing delay	佇列延遲
transmission delay	傳輸延遲	propagation delay	傳播延遲
traffic intensity	流量強度	throughput	產出率
bottleneck link	瓶頸連結	protocol stack	協定堆疊
message	訊息	segment	區段
datagram	資料包	frame	訊框
encapsulation	封裝	payload	內容（酬載）
multi-home	多點上層連接	peering	對等
botnet	殭屍網路	packet sniffer	封包竊聽器
IP spoofing	IP 詐騙	end-point authentication	端點認證
reliable data transfer	可靠資料傳輸	loss-tolerant	可容忍遺失
bandwidth-sensitive	頻寬敏感	elastic	彈性
flow control	流量控制	congestion control	壅塞控制
stateless	無狀態	persistent connection	持續性連線
pipelining	管線化	HOL blocking	隊頭阻塞
Web cache / proxy	Web 快取／代理	origin server	原始伺服器
conditional GET	條件式 GET	resource record	資源紀錄
recursive query	遞迴查詢	iterative query	迭代查詢
authoritative DNS	授權（權威）DNS	server cluster	伺服器叢集
data plane	資料層	control plane	控制層
switching fabric	交換結構	longest prefix match	最長前綴比對
fragmentation	分割	reassembly	重組
subnet mask	子網路遮罩	route aggregation	路由聚合
default gateway	預設閘道	loopback address	迴路位址

附錄 C　數字速查

埠號	用途	Protocol 值	協定
80	HTTP	1	ICMP
443	HTTPS	2	IGMP
53	DNS（UDP）	6	TCP
67	DHCP 伺服器	17	UDP
68	DHCP 用戶端	89	OSPF

項目	數字
TCP/IP 層數／OSI 層數	5 層／7 層
IPv4／IPv6 位址長度	32 位元／128 位元
IP 標頭	20–60 bytes；資料包 20–65,535 bytes
欄位長度	Version 4、HLEN 4、ToS 8、Total length 16、ID 16、Flags 3、Offset 13、TTL 8、Protocol 8、Checksum 16
MTU	Ethernet 1500、PPPoE 1492、WiFi 2304、Jumbo 9000
TTL 預設值	Windows 128、Unix 類 64
根 DNS	13 個邏輯伺服器
同步衛星	36,000 公里；延遲約 270 ms
DSL 頻帶	0–4 kHz 電話、4–50 kHz 上行、50 kHz–1 MHz 下行
DHCP	4 個訊息：Discover、Offer、Request、ACK
DNS 查詢範例	8 個訊息
Cookie	4 個元件

附錄 D　模擬試題

以下題目依講義範圍**自編**，用來檢查理解，不代表實際考題。建議先遮住解答，一次寫完再對答案。

一、選擇題

1. 下列何者與兩路由器之間的距離有關？（A）傳輸延遲（B）傳播延遲（C）處理延遲（D）以上皆是
2. 連結層交換器實作協定堆疊的第幾層？（A）1（B）1–2（C）1–3（D）1–5
3. 網路層的封包稱為？（A）訊息（B）區段（C）資料包（D）訊框
4. 哪一種連線技術是共用廣播媒介？（A）DSL（B）纜線（C）專線（D）以上皆非
5. TCP 不提供下列何者？（A）流量控制（B）壅塞控制（C）可靠傳輸（D）最低產出率保證
6. DNS 使用的傳輸協定與埠號預設為？（A）TCP 80（B）UDP 53（C）TCP 443（D）UDP 67
7. 主機對本地 DNS 伺服器的查詢通常是？（A）遞迴（B）迭代（C）廣播（D）匿名
8. 快取副本未過期時，伺服器對條件式 GET 回覆？（A）200（B）301（C）304（D）404
9. 負責存放「郵件伺服器正規名稱」的 RR 類型是？（A）A（B）NS（C）CNAME（D）MX
10. IP 分割後的重組在哪裡進行？（A）第一台路由器（B）最後一台路由器（C）任何路由器（D）目的主機
11. Header length 欄位值為 5，代表標頭長度？（A）5 bytes（B）20 bytes（C）40 bytes（D）60 bytes
12. DHCP Discover 的來源與目的 IP 為？（A）0.0.0.0 → 255.255.255.255（B）伺服器 → 廣播（C）127.0.0.1 → 伺服器（D）0.0.0.0 → 伺服器

二、計算題

1. 封包 1,500 bytes，連結 10 Mbps、長 2,000 km，$s = 2 \times 10^8$ m/s。求傳輸延遲、傳播延遲及兩者總和。
2. 封包平均每秒抵達 1,200 個，每個 1,000 bits，連結 1.5 Mbps。求流量強度，並判斷系統是否合理。
3. 10 組下載共用核心連結 $R = 10$ Mbps，$R_s = 4$ Mbps，$R_c = 2$ Mbps。求每組產出率與瓶頸所在。
4. 網頁含 1 個 HTML 與 5 張圖，求三種 HTTP 連線方式所需 RTT（忽略傳輸時間）。
5. 機構每秒 20 個請求、平均物件 1 Mb，對外連結 25 Mbps。求無快取時的使用率，以及命中率 0.5 時的使用率。
6. IP 資料 4000 bytes，MTU 1500。列出每片資料量、Total length、Offset、MF。
7. IP 資料 1900 bytes，MTU 1000。列出每片資料量、Offset、MF。（注意 8 的倍數）
8. `ping -l 2000` 在乙太網路上會分成幾片？各片 IP 資料量與 Offset 為何？
9. 標頭欄位 HLEN＝7，Total length＝1000。求標頭長度、選項長度與資料長度。
10. 192.168.10.37/27 的遮罩、網路位址、廣播位址、可用主機範圍與數量為何？
11. 把 200.1.2.0/24 切成 8 個子網路，求新前綴、每個子網路位址數、第 3 個子網路的範圍。

12. 把 200.23.16.0/24、200.23.17.0/24、200.23.18.0/24、200.23.19.0/24 聚合成一條路由。
13. 用 4.2 節的轉送表，判斷目的位址 `11001000 00010111 00011001 00000001`、`11001000 00010111 00010101 00000001`、`11001000 00010110 00000000 00000001` 各走哪個介面。

三、問答題

1. 說明封包交換相較於電路交換的優缺點。
2. 為什麼 Web 快取能同時降低回應時間與機構對外流量？它帶來什麼新問題、如何解決？
3. 為什麼 DNS 不採集中式設計？
4. 說明 NAT 如何讓多台主機共用一個 IP。
5. 連結層交換器與路由器有何不同？

解答

選擇題

1	2	3	4	5	6	7	8	9	10	11	12
B	B	C	B	D	B	A	C	D	D	B	A

計算題

1. $d_{trans} = 12{,}000 \div 10^7 = \mathbf{1.2\ ms}$；$d_{prop} = 2 \times 10^6 \div (2 \times 10^8) = \mathbf{10\ ms}$；總和 **11.2 ms**。
2. $La/R = (1{,}000 \times 1{,}200) \div (1.5 \times 10^6) = \mathbf{0.8}$。小於 1，系統穩定，但已開始有明顯佇列延遲。
3. 每組分到 R/10＝1 Mbps，min{4, 2, 1}＝**1 Mbps**；瓶頸在**核心共用連結**。
4. 非持續性 2 × 6＝**12 RTT**；持續性無管線化 2＋5＝**7 RTT**；持續性有管線化 **3 RTT**。
5. 流量 20 × 1 Mb＝20 Mbps。無快取：20 ÷ 25＝**0.8**；有快取：0.5 × 20 ÷ 25＝**0.4**。
6. 每片最多 1480。三片資料 1480、1480、1040；Total length 1500、1500、1060；Offset 0、185、370；MF 1、1、0。
7. 1000−20＝980，不是 8 的倍數，取 **976**。資料 976、924；Offset 0、122；MF 1、0。
8. ICMP 8＋2000＝2008 bytes。分 **2 片**：1480、528；Offset 0、185。
9. 標頭 7 × 4＝**28 bytes**；選項 28−20＝**8 bytes**；資料 1000−28＝**972 bytes**。
10. 遮罩 **255.255.255.224**；區塊大小 32，37 落在 32–63；網路位址 **192.168.10.32**；廣播 **192.168.10.63**；可用 **.33–.62，共 30 個**。
11. 借 3 位元得 **/27**；每個 **32** 個位址；第 3 個子網路 **200.1.2.64 – 200.1.2.95**。
12. 第三位元組 16–19 為 000100xx，前 22 位元相同，聚合為 **200.23.16.0/22**。
13. 第一個符合 `00011***` → **介面 2**；第二個符合 `00010***` → **介面 0**；第三個第二位元組不符 → **介面 3**。

問答題重點

1. 優點：依需求共用、頻寬分享好、簡單有效率、成本低、不必預約。缺點：延遲不固定且無保障，壅塞時會遺失，較不適合嚴格即時服務。
2. 快取離用戶端近，命中時不必經過對外連結。新問題是副本可能過期，以條件式 GET（`If-modified-since`，未修改回 304）解決。
3. 單點失效、流量龐大、距離遙遠造成延遲、維護困難。所以採階層式分散資料庫。
4. 路由器把內部（私有 IP、埠）換成（全球 IP、新埠）並記入轉換表；回應抵達時依埠號查表換回。靠埠號區分不同主機的連線。
5. 交換器在第 2 層，依 MAC 位址轉送訊框，連接同一子網路；路由器在第 3 層，依 IP 位址繞送，每個介面屬於不同子網路，並以最長前綴比對查轉送表。

附錄 E　講義疑點與不一致

整理時發現下列地方與計算結果或其他頁面不一致。本書正文採右欄的處理方式；若考試依講義原文出題，請以老師說法為準。

位置	講義內容	本書處理
英 4 p.83	172.16.0.0/12 位址數為 1,047,584	2^{20}＝**1,048,576**，判斷為誤植
英 4 p.83	169.254.0.0/16 列為私有位址	依講義列出。一般而言它是 link-local（自動設定）位址，與 RFC 1918 三段私有位址不同
英 2 p.36 與 p.20	表格把 HTTPS 的 OSI 層寫成 Transport，p.20 則說 TLS 實作在應用層	依 p.20，HTTPS 屬應用層、TLS 位於應用程式與 TCP 之間。**建議向老師確認**
英 4 p.45、p.50	ping 分割圖把 8 bytes ICMP 標頭畫在**最後一片**	各片資料量總和正確（128、248）。實際上 ICMP 標頭在**第一片**資料的開頭；考試若問數字，答 1480＋128、1480＋1480＋248
英 1 p.122	連結層「使用網路層服務」傳送資料包	應是使用**實體層**服務（下一層），判斷為誤植
英 2 p.42	PUT 的內容放在 POST 請求訊息	應為 PUT 請求訊息，判斷為誤植
中 1 p.41	銅線直徑「約 1 厘米」	課本原文一般是約 1 mm；厘米（cm）顯然過粗，判斷為翻譯問題
英 2 p.22 與 p.29	HTTP 1.1 標 RFC 7320；另處標 RFC 1945、2616	RFC 編號版本不同。不建議花時間背
英 2 p.53、55 與 p.57、64	管線化是 HTTP/1.1 預設；多數瀏覽器卻預設關閉	不算矛盾：前者是規格，後者是實作
中 1 p.38、英 1 p.49、54	廣域無線速率分別為 60 Mbps、1–10 Mbps、數十 Mbps	技術世代不同。計算題以題目給定為準
英 1 p.82 與 p.86	傳播速率約 2×10^8，或 $2–3 \times 10^8$ m/s	媒介不同所致。計算題以題目給定為準
英 2 p.74 與 p.75–76	LAN 延遲有時寫 msecs、有時 usecs；p.76 仍顯示未裝快取的數字	量級都很小不影響結論；裝快取的實際計算在 p.77
多處	dprog、tire-1、global transmit ISP、DCP server 等拼字	分別應為 dprop、tier-1、transit、DHCP

範圍提醒

- **沒有第 3 章**：提供的檔案只有第 1、2、4 章，傳輸層（TCP 細節）不在本書內。
- **第 2 章**：目錄列出 FTP、電子郵件、P2P、Socket 程式設計，但投影片無內容。
- **第 1 章的 1.6、1.7**：英文講義只列在目錄，內容僅見於中文講義。
- **第 4 章**：IPv6、ICMP、繞送演算法、SDN 只有概略提及；p.95 的 Wireshark 輸出只有標題，本書未能取得細節。
- **小考標記**：英 2 p.84 有「第四次小考」、p.115 有「第五次小考」，表示這兩段的內容在課堂上被單獨測驗過，可找回小考卷對照題型。

附錄 F　考前檢查清單

每一項都試著**不看書說出來或算出來**。做不到的，回到括號中的章節。

第 1 章

- ☐ 協定定義的三個關鍵字（1.1）
- ☐ DSL、纜線、FTTH 的設備名稱與專用／共用差異（1.2）
- ☐ 導引式與非導引式媒介的例子（1.2）
- ☐ 轉送與繞送的差別（1.3）
- ☐ FDM 與 TDM 各切什麼（1.3）
- ☐ 封包交換與電路交換比較（1.3）
- ☐ 四種延遲名稱、公式與傳輸／傳播的差別（1.4）
- ☐ La/R 三種情況（1.4）
- ☐ 產出率 min 公式與共用連結算法（1.4）
- ☐ 五層、各層封包名稱、裝置實作層數（1.5）
- ☐ 三種 DoS 與 DDoS（1.6）

第 2 章

- ☐ Socket 定義與 IP＋埠號（2.1）
- ☐ 四類傳輸服務與常見應用需求表（2.1）
- ☐ TCP、UDP 服務與都不提供的項目（2.1）
- ☐ HTTP 方法、狀態碼與 HTTP／HTTPS 埠號（2.2）
- ☐ 三種連線的 RTT 公式（2.2）
- ☐ Cookie 四元件（2.2）
- ☐ 快取範例的使用率與延遲（2.2）
- ☐ 條件式 GET 與 304（2.2）
- ☐ DNS 三類伺服器與本地 DNS（2.3）
- ☐ 遞迴與迭代、畫出 8 個步驟（2.3）
- ☐ A、NS、CNAME、MX（2.3）
- ☐ CDN 如何用 DNS 導向與兩種叢集選擇策略（2.4）

第 4 章

- ☐ 資料層與控制層、SDN（4.1）

- ☐ best-effort 不保證什麼（4.1）
- ☐ 最長前綴比對（4.2）
- ☐ 畫出 IP 標頭與各欄位長度（4.3）
- ☐ HLEN、Total length、Offset 的單位（4.3）
- ☐ 分割計算與 MF，含 8 的倍數陷阱（4.3）
- ☐ Checksum 流程（4.3）
- ☐ 由 /n 算遮罩、網路位址、廣播位址、主機數（4.4）
- ☐ 子網路切割與路由聚合（4.4、4.5）
- ☐ 特殊位址與私有位址區塊（4.5）
- ☐ NAT 轉換表四步驟（4.6）
- ☐ DHCP 四訊息、來源與目的 IP、埠號（4.7）