

電腦網路與 Internet 通訊

主講義

核心查閱：2026-10-03；LAN 補充：2026-10-06

修訂版 v1.2 | 2026-10-08 勘誤與一致性修訂

MD 原稿與本閱讀版共用正文。

目錄

課程大綱	3
單元 01 網路全貌：從一臺電腦到遠端服務	8
單元 02 分層、封裝與效能	12
單元 03 IPv4、子網與下一跳	15
單元 03A 單介面多 IP、來源選擇與政策路由	17
單元 04 Ethernet、ARP 與區域網路	21
單元 04A 單播、多播、廣播與 LAN 服務探索	23
單元 05 DHCP、DNS 與設定的生命週期	28
單元 06 Socket、UDP 與應用訊息邊界	30
單元 07 從遺失到可靠傳輸	32
單元 08 TCP 的狀態、視窗與壅塞	34
單元 09 期中整合與路徑追蹤	36
單元 10 路由演算法與跨網域政策	38
單元 11 IPv6 與雙協定路徑	40
單元 12 NAT、防火牆與穿透	42
單元 13 通訊安全的密碼學基礎	44
單元 14 TLS、HTTPS 與可見資訊	46
單元 15 HTTP、QUIC、承載與快取	48
單元 16 無線、量測與分層排錯	50
單元 17 代理的端點、目的位址與解析位置	52
單元 18 Shadowsocks 與 VMess：加密代理的責任	54
單元 19 VLESS、Trojan 與分層組合	56
單元 20 Hysteria 2 與 TUIC：QUIC 代理的取捨	59
單元 21 WireGuard、OpenVPN 與 IPsec	61
單元 22 流量接管、分流、DNS 與軟路由整合	63
單元 23 專題：設計、故障與證據	65
單元 24 綜合評量與設計答辯	67
附錄 A 入門工具、單位與學習方法	69
附錄 B 代理與 VPN 評估總表	71
附錄 C 隔離 Linux 多網段延伸	73
術語表	74
來源與版本註記	76

課程大綱

課程定位與成果

本課程從一臺電腦連上遠端服務的完整過程出發，依序建立區域網路、位址與路由、名稱解析、可靠傳輸、安全通訊及代理 / VPN 的知識。每個模型都連到可以觀察或量測的現象。

完成課程後，學習者應能：

1. 畫出一次通訊的拓樸，逐跳追蹤 IP 封包、鏈路訊框與代理連線。
2. 根據位址、子網與路由表判斷下一跳，並說明 DNS 與資料流的關係。
3. 用延遲、視窗、頻寬延遲乘積與遺失模型分析傳輸行為。
4. 用 Socket 建立可重現的小型網路應用，分析 TCP 與 UDP 的差異。
5. 區分驗證、加密、前向保密、重放防護及流量可觀察性。
6. 比較代理協定、承載方式、安全機制、接管方式與 IP 層 VPN。
7. 以封包、日誌及受控制的量測作為故障診斷與設計選擇的依據。

先備知識與入門補充

不要求先修網路課。附錄補充二進位、單位換算、Python 程式啟動、命令列路徑與 Socket 的基本概念。必修程式以標準函式庫為主。路由演算法用小型圖例與虛擬碼說明；實作密碼學不是本課必修工作。

閱讀採兩層深度：正文提供完整的問題、模型與例題；「延伸」提供規範細節、演算法推導及版本差異。先完成正文再進入延伸。

評量與自學

自學者可用相同規準檢核。解答獨立成冊，說明推理過程與常見錯法。基本題、分析題及延伸題分開標示；延伸題不作為完成零基礎路線的必要條件。

項目	比例	核心證據
習題	15%	正確假設、推導與解釋
實驗	25%	預測、紀錄、差異分析與可重現性
期中	20%	概念、計算與路徑追蹤
期末題組	15%	10 至 22 單元的概念、計算與故障定位
期末專題	25%	設計、量測、故障定位與取捨

24 週核心進度與兩個 LAN 補充單元

週次 / 單元	問題與內容	可觀察成果
01 網路全貌	主機、交換器、路由器、AP、ISP；端點與路徑	家庭拓模圖及服務路徑假設
02 分層與效能	封裝、封包交換、傳送 / 傳播 / 排隊延遲、吞吐量、BDP	計算與受控延遲模型
03 IPv4	CIDR、子網、最長前綴匹配、預設路由、ICMP、MTU	位址設定與下一跳判斷
04 區域網路	Ethernet、MAC、ARP、交換表、VLAN、迴路與 STP 概念	逐跳訊框表與交換演練
05 DHCP 與 DNS	租約、遞迴 / 權威解析、快取、TTL、名稱與位址	DHCP 時序圖及 DNS 回應分析
06 Socket 與 UDP	埠、端點、多工、資料報與位元組串流的應用介面	本機 UDP 與 TCP 小型服務
07 可靠傳輸	校驗、序號、ACK、逾時、重傳、滑動視窗	ARQ 模擬與視窗效率圖
08 TCP	建連、關閉、序號、流量控制、壅塞控制、RTT	TCP 串流與離線遺失模型分析
09 期中整合	網路設定到取得內容；概念、計算與證據	期中題組與路徑診斷報告
10 路由演算法	轉送 / 路由、距離向量、鏈路狀態、OSPF、BGP 政策	最短路徑與路由選擇演練
11 IPv6	位址、NDP、SLAAC、DHCPv6、雙協定、ICMPv6	IPv6 位址與路徑分析
12 NAT 與防火牆	對映 / 過濾、連線狀態、CGNAT、STUN、TURN、ICE	NAT 表與穿透可行性分析
13 密碼學概念	對稱 / 非對稱、雜湊、MAC、簽章、DH、AEAD、PFS、重放	安全性需求與信任分析
14 TLS 與 HTTPS	信任鏈、名稱驗證、握手、SNI、0-RTT、可見資訊	本機 TLS 驗證與憑證觀察
15 HTTP 與 QUIC	HTTP/1.1 / 2 / 3、串流、DATAGRAM、快取、CDN、WebSocket / gRPC 承載	隊首阻塞與快取決策分析
16 無線、量測與排錯	Wi-Fi 媒介競爭、AP、家庭分段、分層排錯、量測誤差	最低路線做排錯；無線細節列延伸

週次 / 單元	問題與內容	可觀察成果
17 代理基礎	HTTP CONNECT、SOCKS5、目的位址、名稱解析位置、正向 / 反向代理	本機代理與連線路徑
18 加密代理	Shadowsocks AEAD 2017 / 2022、VMess AEAD、驗證、時間與重放	封裝與安全機制對照
19 VLESS / Trojan 與分層組合	VLESS、Trojan、TLS / REALITY、Vision、WebSocket / gRPC / XHTTP、VLESS Encryption ; AnyTLS / NaïveProxy 延伸	分欄辨認 protocol、encryption、transport、security、flow
20 QUIC 代理	Hysteria 2、TUIC v5、串流 / 資料報、UDP 轉送、壅塞與填充	同條件量測設計
21 IP 層 VPN	WireGuard、AllowedIPs、路由、MTU、OpenVPN、IKEv2 / IPsec	IP 隧道與代理的路徑對照
22 接管與整合	系統代理、TUN / TAP、透明代理、分流、Fake IP、DoH / DoT、鏈式代理；軟路由整合	DNS / IPv6 / UDP 路徑檢查
23 專題設計	拓樸、服務、量測、受控故障、報告	專題初稿與證據
24 綜合評量	設計取捨、故障定位、協定組合與反思	完整專題與期末題組

專題於第 17 週選題，第 19 週交拓樸，第 21 週交量測計畫，第 23 週完成初稿，第 24 週整合。這些里程碑與每週實驗共用成果。

LAN 補充的正式位置

LAN 補充包含必讀 03A「單介面多 IP、來源選擇與政策路由」及 04A「單播、多播、廣播與 LAN 服務探索」，以及 L15、L16 與獨立解答。完整自學可用 26 次學習，03A 接 03、04A 接 04；4A.5 的 DNS-SD 在 05 後回讀，政策路由細節可在 10 後回讀。

若維持 24 週，將 03A 分在第 03 與 10 週，04A 分在第 04 與 05 週，每個補充單元約另需 3 小時，完整 Linux 延伸另計。這兩單元是完整課程的必讀內容；需在 09 期中前完成基礎，在 22 整合前完成來源政策與探索分析。LAN 評量題組可替換原期中 / 期末部分題目，維持總比例。

新增成果：能追蹤同介面兩個來源的規則、表與下一跳；能預測 FDB / MDB / VLAN 的資料出口；能分開畫探索、單播服務及代理上游路徑。

每章結構

每章依序包含：學習目標與先備關係、具體問題、原理與模型、完整例題、圖解或時序表、常見誤解、實驗連結、習題、章末檢核及來源。抽象模型會明示假設與限制；實作特性會標示版本與平臺。

必修環境與實驗

必修環境：macOS、Windows 或 Linux 的 Python 3.10 以上；僅使用標準函式庫。L13 的本機憑證實驗另需 OpenSSL 相容指令，提供憑證資料的離線條件表替代路線。Wireshark 為延伸工具，非必修前提。每週最低路線為正文、基本題與指定實驗步驟，約 5 至 6 小時；完整分析與延伸約 7 至 8 小時，代理單元可拆成兩次學習。

必修實驗提供本機可執行版本或離線資料分析：

- L01 (01、06) 觀察端點、資料與連線；建立本機 HTTP 服務。
- L02 (02、07、08) 延遲、單位、視窗及頻寬延遲乘積。
- L03 (03、11) 位址與最長前綴匹配。
- L04 (05) DNS 訊息與名稱解析。
- L05 (06、08) TCP / UDP Socket 與訊息邊界。
- L06 (07、08) ARQ 與滑動視窗模擬。
- L07 (10) 路由演算法。
- L08 (12) NAT 對映、過濾與連線狀態模型。
- L09 (17、22) 本機 CONNECT / SOCKS 代理與 DNS 路徑。
- L10 (19、21、22) 隧道開銷、MTU 與分流決策。
- L11 (15、16、18、20、23) 同條件效能比較與統計摘要。
- L12 (09、12、16、23、24) 綜合故障診斷與專題。
- L13 (13、14) 本機 TLS、名稱驗證、信任與憑證。
- L14 (15、18、19、20、21、22) 代理 / VPN 的協定欄位與路徑分析工作表。
- L15 (03A、10、22) 單介面多 IP、來源選擇、RPDB 與 LAN / endpoint 例外。
- L16 (04A、05、11、22) 單播 / 廣播 / 多播交付、成員老化與 DNS-SD 路徑。

每份實驗包含預測、步驟、預期現象、紀錄表、判讀題與驗證範圍。模擬結果只能支持模型內的結論；真實代理 / VPN 的效能與可辨識性需要指定實作及實際網路資料。

Linux 虛擬環境延伸提供封包擷取、多網段、NAT / 防火牆與 WireGuard / 代理設定的指引。它們不列為零基礎讀者完成必修課程的環境前提。將清楚區分已在本機驗證的程式、可離線分析的範例與尚未執行的跨平臺延伸。

代理與 VPN 的評估方法

對每個協定使用相同欄位：設計目標、轉送單位、目的位址、驗證方式、加密位置、金鑰與前向保密、重放防護、TCP / UDP 承載、多工、DNS 解析位置、可觀察資訊、實作相容性、效能條件與失敗案例。

協定名稱、承載方式、安全機制、流量接管及軟體核心分開描述。TLS 與 REALITY 不當作和 VLESS 等價的代理協定；TUN 不被當作加密保證。各種組合的有效性以指定實作及版本核對，不能將工具中的一個標籤當作普遍標準。

不以一次測速排列「最快協定」。比較要控制硬體、實作版本、路徑、延遲、遺失、頻寬、工作負載與並行連線數；同時紀錄延遲分布、吞吐量、CPU 與額外流量。安全、匿名性、偽裝與可用性分別分析。

單元 01 網路全貌：從一臺電腦到遠端服務

學習目標

讀完本章，你應能畫出主機、交換器、無線基地臺、路由器與服務端的位置；區分 MAC 位址、IP 位址、埠與網域名稱；把一次網站存取拆成可驗證的步驟。本章不需要程式先備。

1.1 先問「誰要與誰通訊」

假設筆電中的瀏覽器要取得一個網站的頁面。真正交換資料的是瀏覽器與伺服器上的程式。它們所在的主機需要網路介面、位址與可用路徑，程式需要知道遠端服務的端點，雙方還需要對訊息格式與處理方式有共同約定。

「網路通了」不是單一狀態。無線介面已連上基地臺、取得位址、找得到閘道、能解析名稱、能建立連線、能完成 TLS 驗證、能取得正確 HTTP 回應，分別是不同的檢核點。某一步成功，不能替其他步驟作證。

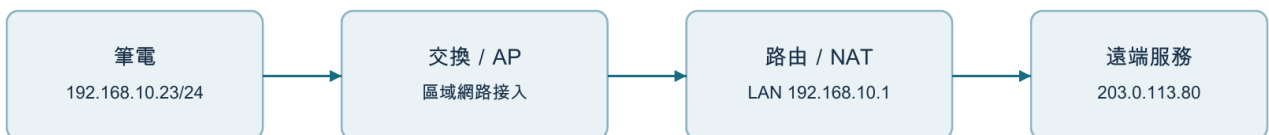
服務也不一定對應一臺固定機器。一個名稱可以解析到多個位址；反向代理可以代表後端服務接收請求；內容可能由 CDN 快取回覆。初學時先採用「一個客戶端、一個服務端」的模型，之後再逐步放寬假設。

1.2 圖上的裝置各做什麼

主機產生及消費資料。交換器通常根據鏈路層資訊轉送訊框，讓同一個區域網路的介面互通。路由器根據目的 IP 與路由表選擇下一跳，連線不同網段。無線基地臺 (AP) 提供無線接入，常把無線流量橋接到有線網路。ISP 提供對外連線及更大範圍的路由。

家庭中俗稱的「路由器」常同時包含 AP、交換器、IP 路由、NAT、防火牆、DHCP 伺服器與 DNS 轉送等功能。一個外殼包含多個角色，所以分析時要寫角色，而不只寫產品名稱。獨立 AP 不一定做 NAT；能交換訊框也不代表正在做 IP 路由。

先追蹤本地下一跳，再分析對外路徑



外側示例：198.51.100.7

外側為文件示例位址；本機實驗使用 loopback。

主機到服務的教學拓模

圖中的對外位址使用文件示例網段，僅供推理，不作為實驗連線目標。本機實驗使用 127.0.0.1。

1.3 四種容易混淆的識別資訊

資訊	主要用途	例子	分析時要問
網域名稱	在名稱系統中識別服務或其他資源	course.example	誰解析？回應是否來自快取？
IP 位址	在 IP 網路中描述介面 / 端點；私有位址可在管理的私有網路路由，不直接在 Internet 全域路由	192.168.10.23	目的地是否在同一網段？
MAC 位址	在 Ethernet 等鏈路中識別介面	02:00:00:00:00:23	本次訊框要交給誰？
埠號	供 TCP / UDP 等傳輸協定區分端點	TCP 443	哪個程式在接收？協定是什麼？

一個 TCP 連線通常以來源 IP、來源埠、目的 IP、目的埠區分，並處在 TCP 的協定脈絡中。分析 TCP 與 UDP 混合流量時，需要把傳輸協定也列出。相同數字的 TCP 埠與 UDP 埠不是同一個端點。

IP 位址可以改變；主機可以有多個介面與多個位址；MAC 也可能隨機化或由軟體設定。因此，不能把其中任何一個識別資訊當作永遠固定的「人的身分」。埠的慣例也不保證實際服務，例如 TCP 443 上不必然是你想像的那個網站。

1.4 一次網站存取的依賴關係

採用下列明確假設：筆電已有 IPv4 位址 192.168.10.23/24；預設閘道及 DNS 轉送器是 192.168.10.1；目標名稱與鄰居位址尚未快取；使用 HTTP/1.1、TCP 與 TLS；沒有應用代理；僅走 IPv4，瀏覽器使用系統解析器，不啟用 DoH / DoT；閘道執行 NAPT。目標的示例位址為 203.0.113.80。

1. 瀏覽器或作業系統需要把名稱解析成可用位址。
2. DNS 查詢要送到 192.168.10.1。因它在同一網段，而筆電還不知道其 MAC，會先需要 ARP 鄰居解析。
3. 得到名稱解析結果後，筆電查路由表。遠端位址不在本地 /24，資料交給預設閘道；步驟 2 已取得其 MAC，快取有效時不再做第二次 ARP。若 DNS 位於遠端網段，送 DNS 查詢時的 ARP 對象也通常是本地閘道。
4. TCP 與遠端建立連線。承載 TCP 段的 IP 封包先放進訊框交給閘道；閘道取出封包後，由後續路由器逐跳轉送，每一跳都重新作鏈路封裝。
5. TLS 協商與驗證建立受保護的通訊；客戶端驗證伺服器憑證及名稱。
6. 瀏覽器送出 HTTP 請求，接收狀態、標頭及內容。

這是一個受假設約束的流程，不是所有網站存取的固定時間表。DNS 快取可以略去網路查詢；IPv6 使用不同的鄰居解析；HTTP/3 使用 QUIC 而不是 TCP。ARP 是鏈路送出訊框時的依賴，可能由 DNS 流量先觸發，不能一律排在 DNS 解析之後。

若筆電連網時連位址都沒有，還要先處理鏈路接入與位址設定，可能包含 Wi-Fi 關聯及 DHCP。後續章節會逐項建立這些模型。

1.5 完整例題：下一個訊框送給誰

筆電向遠端服務送資料，來源 IP 是 192.168.10.23，目的 IP 是 203.0.113.80。本地訊框的來源 MAC 是筆電介面，目的 MAC 是閘道的本地介面，而不是遠端伺服器的 MAC。

原因是 Ethernet 訊框只負責這一段鏈路交付。閘道收到訊框後取出 IP 封包，選擇新的出口，再建立適合下一段鏈路的訊框。普通路由轉送通常保留來源 / 目的 IP，更新 TTL 等欄位；若經過 NAT，位址或埠還可能被改寫。因此「MAC 逐跳改變」與「IP 絕對不變」不能被綁成一條無條件口訣。

觀察位置	鏈路目的地	IP 目的地	能直接推論的事
筆電的區域網路出口	閘道的本地 MAC	遠端服務 IP	此訊框先交給閘道
路由器下一段出口	下一跳鏈路位址	遠端服務 IP	路由器已作下一跳決策
NAT 外側	依外側鏈路而定	遠端服務 IP	來源可能已換成 NAT 對映

在 point-to-point 或 loopback 等介面，鏈路格式與可見欄位可能不同。不能把 Ethernet 的圖直接套在每個介面。

1.6 用證據逐步排錯

先描述症狀：是所有名稱失敗、只有一個服務失敗，還是內容讀取很慢？再提出最小假設：例如「名稱解析失敗」，然後選擇能支持或反駁該假設的觀察。

能 ping 一個位址只表示有某種 ICMP 回應，不保證 TCP 服務存在。ping 沒回應也不一定代表主機不可達，因為防火牆或裝置政策可能不回覆。HTTP 狀態碼表示已與某個 HTTP 端點交換訊息，但仍需檢查內容及是否由中間服務回覆。

這門課反覆採用同一個方法：先畫路徑，寫下假設，取得證據，確認證據的範圍，再修正模型。

常見誤解

- 「同一臺裝置只有一個 IP」：多介面、多位址與虛擬介面都很常見。
- 「交換器和路由器只是速度不同」：它們的轉送依據及角色不同。
- 「知道遠端 MAC 才能上網」：通常只需知道本地下一跳的鏈路位址。
- 「連上 Wi-Fi 就能取得網站內容」：接入只是完整流程的一個環節。

實驗與習題

完成 L01。本機實驗會觀察 TCP 端點、HTTP 請求與內容；它不經過家庭閘道、外部 DNS、NAT 或 TLS。

1. 畫出你自己的網路拓樸，將家庭裝置中的角色分開標示。

2. TCP 443 與 UDP 443 可以同時由不同服務使用嗎？說明理由。
3. 在上述示例中，第一個 ARP 可能由哪個應用流程觸發？
4. 為什麼封包的遠端 IP 與本地訊框的目的 MAC 可以指向不同裝置？
5. HTTP 200、ping 成功、Wi-Fi 已連線，各能支持什麼結論？

章末檢核

不用背流程清單，試著逐項回答：誰產生資料？名稱如何解析？下一跳是誰？哪個服務端點接收？每種觀察最多能支持什麼結論？

來源

RFC 826 (ARP)、RFC 791 (IPv4)、RFC 9293 (TCP)、RFC 9110 / 9112 (HTTP)、RFC 1034 / 1035 (DNS)、RFC 9846 (TLS 1.3)、RFC 5737 (文件位址)、RFC 2606 (保留域名)。規範連結及版本註記見 [references/source-notes.md](#)。圖與例題為本教材自行建立的教學模型。

單元 02 分層、封裝與效能

學習目標

本章以單元 01 的端點與路徑為基礎。讀完後，你應能說明分層介面的用途，區分傳送、傳播、處理及排隊延遲，計算多跳 store-and-forward 路徑的理想延遲，並用頻寬延遲乘積分析視窗限制。

2.1 分層是把責任和介面說清楚

HTTP 描述應用如何請求及回覆資源；TCP 提供可靠、有序的位元組串流；IP 提供跨網路的封包交付；鏈路層將資料送到相鄰節點。每一層利用下層的服務，並向上層提供介面。分層讓應用不必處理每一種網線的電氣細節，也讓鏈路技術可以替換。

OSI 七層是理解責任的參考模型，Internet 常以應用、傳輸、網路、鏈路等層次討論。模型中的層數不比實際責任重要。TLS、VPN、代理與多重封裝尤其需要畫出完整堆疊，而不能僅用一個層號下結論。



此圖只示意普通 IP 轉送；代理、NAT 與隧道另作模型。

封裝與逐跳交付

在一個例子中，應用資料成為 TLS 記錄的內容，再由 TCP 傳送；TCP 段放入 IP 封包；IP 封包放入鏈路訊框。資料單位可能被分段、合併或拆成多個記錄，所以一個 HTTP 訊息不等於一個 TCP 段，也不等於一次 Socket 讀取。Socket 是程式使用作業系統通訊服務的介面；recv 讀取目前可交付、且不超過指定上限的 bytes，不保證對齊應用訊息。

2.2 標頭、有效資料與開銷

假設每個 IP 封包有 20 bytes IPv4 標頭和 20 bytes TCP 標頭，沒有 options；其中承載 1000 bytes 應用資料，暫不計 TLS 與鏈路開銷。應用資料在 IP 層的比例為：

$$1000 / (1000 + 20 + 20) = 0.9615, \text{ 約 } 96.15\%.$$

這個比例不是整條網路的實際效率。ACK、重傳、TLS、鏈路標頭及媒介時間都會影響實際佔用。若計入更多層，需要明確指定量測位置，不能把 IP bytes 與實體線速混在同一個分母。

吞吐量描述某個觀察位置每秒傳過多少資料；goodput (有效吞吐量) 常用來描述應用得到的有效資料率。報告量測時應寫清楚：計入哪一層、方向、時間區間、有效內容及重傳是否算入。

2.3 四種延遲

類型	模型	直覺與限制
傳送延遲	L / R	把 L bits 推上速率 R bit/s 的鏈路需要時間
傳播延遲	d / v	訊號沿距離 d 以速率 v 傳播
處理延遲	依裝置與工作而定	檢查、轉送、排程或協定處理
排隊延遲	依到達、服務與排程而定	封包等待可用出口資源

頻寬提高可以降低傳送延遲，但不能直接消除遠距離的傳播延遲。排隊延遲會隨時間改變，因此同一路徑的測量不會總是相同。RTT 是往返時間，不保證正反方向的單向延遲各佔一半。

單位換算要先做：1 byte = 8 bits；本教材網路速率用十進位，例如 100 Mb/s = 100,000,000 bit/s。MiB 等二進位容量另行標示，避免把 MB 與 MiB 混用。

2.4 完整例題：封包穿過三段鏈路

一個 1500-byte 封包穿過三段各 100 Mb/s 的鏈路。每段傳播延遲是 2 ms，各中間節點使用 store-and-forward，忽略處理及排隊。問題是第一個 bit 出發到最後一個 bit 到達的總時間。

封包長度 $L = 1500 \times 8 = 12,000$ bits。每段傳送延遲：

$$L/R = 12,000 / 100,000,000 \text{ s} = 0.00012 \text{ s} = 0.12 \text{ ms}。$$

三段 store-and-forward 各要等完整封包到齊後才開始下一段傳送，所以總時間：

$$3 \times (0.12 + 2) \text{ ms} = 6.36 \text{ ms}。$$

這是單一封包的理想模型。兩個相同封包連續送出時，第二個只多一個傳送間隔 0.12 ms，理想總時間為 6.48 ms。連續多個封包可以在不同鏈路同時前進，形成流水線；不能把每個封包的單一延遲相加，當作整個檔案傳輸的時間。裝置採 cut-through、鏈路速率不同或有排隊時，也要重新建模。

2.5 瓶頸、流水線與檔案傳輸

若三段鏈路的速率分別為 100、20、100 Mb/s，理想的穩態資料率受到 20 Mb/s 的瓶頸限制。這是忽略協定開銷、競爭、流量控制及壅塞的上界。實際應用吞吐量可能更低。

檔案時間可以粗略估計為「連線及請求的啟動時間 + 內容大小 / 穩態 goodput」。這個近似在大檔案時較有用；小型資源可能主要花在名稱解析、連線建立、驗證及請求往返。改善吞吐量與改善頁面啟動延遲，可能需要不同方法。

2.6 頻寬延遲乘積與在途資料

若可用速率為 R ，往返時間為 RTT ， $BDP = R \times RTT$ 。它描述為了持續利用這個速率，需要維持的在途資料量尺度。其單位取決於 R ：若 R 是 bit/s，乘積是 bits。

例：100 Mb/s、RTT 40 ms， $BDP = 100,000,000 \times 0.04 = 4,000,000 \text{ bits} = 500,000 \text{ bytes}$ 。

64 KiB 是本題假設值，不是未縮放 TCP 視窗欄位的 65,535-byte 上限；TCP 視窗縮放見 RFC 7323。假設傳送端最多允許 64 KiB 在途，且沒有其他限制，視窗模型的上界約為：

$$W / RTT = 65,536 / 0.04 = 1,638,400 \text{ byte/s} = 13.11 \text{ Mb/s}。$$

即使鏈路能傳 100 Mb/s，這個視窗模型仍只允許約 13.11 Mb/s。後續 TCP 章會再區分接收視窗、壅塞視窗、實際 in-flight bytes、RTT 變動及應用供給。本章的公式是分析工具，不是對所有 TCP 連線的精確預測。

2.7 排隊、遺失與 bufferbloat

一個出口的服務速率有限。若短時間到達的封包超過服務能力，緩衝區會累積資料；緩衝區滿時，裝置可能丟棄封包或採用其他管理策略。平均利用率相同，也可以有很不同的尖峰與延遲分布。

增大緩衝區可以吸收短暫尖峰，卻也可能讓封包等待很久。大量上傳時互動程式突然變慢，可能與排隊有關。要測試這個假設，可以比較負載前後的 RTT 分布與排隊行為；僅看下載速率不足以定位原因。

M/M/1 等排隊模型需要特定的到達及服務假設。除非這些假設適用，不應把其公式直接當作真實家庭網路的精確延遲。延伸閱讀可以討論利用率、服務變異及不同排程的影響。

2.8 可重現的量測

報告至少記錄工作負載、資料大小、連線數、觀察層、計時起迄、暖身、重複次數、硬體與版本。同時檢視中位數、尾端分位數與異常值，而不是只交一個平均。

分清楚三種資料：實際量測、依模型計算、人工示例。L02 會提供模型計算與參數對照；L01 的本機時間則是實測，但包含應用與排程等開銷，不能直接當作單向鏈路傳播延遲。

習題

- 1200-byte 封包在 10 Mb/s 鏈路上的傳送延遲是多少？
- 兩段 10 Mb/s 鏈路，各傳播延遲 1 ms；依 store-and-forward 模型求上一題封包的總時間。
- 20 Mb/s、RTT 100 ms 的 BDP 是多少 bytes？
- 為什麼一次 recv 不必對應一個應用訊息？
- 如何分辨大檔案吞吐量瓶頸與小請求啟動延遲問題？
- 一段路徑利用率是 80%，為什麼不能僅憑此數字推算其排隊延遲？

章末檢核與來源

你應能在計算前寫出資料單位、量測位置與模型假設，並說明什麼現象不在模型內。延伸閱讀：RFC 9293 (TCP)、RFC 5681 (壅塞控制)、RFC 9000 / 9002 (QUIC 與遺失 / 壅塞處理)。例題、延遲模型與教學圖為本教材自行設計。

單元 03 IPv4、子網與下一跳

學習目標與問題

先讀單元 01、02。你應能將十進位位址轉成前綴判斷，計算一般子網範圍，使用最長前綴匹配找出口，並解釋 ICMP 與 MTU。問題是：目的主機不在同一個網段時，電腦究竟把訊框交給誰？

補充 03A 進一步處理單介面多 IP、來源偏好與政策路由；完成 L03 後接 L15。

3.1 位址與前綴

IPv4 位址有 32 bits，通常寫成四個 0 至 255 的十進位數。CIDR 的 /n 表示前 n bits 是網路前綴；它不是可用主機數。192.0.2.64/26 的遮罩為 255.255.255.192，每塊有 64 個位址。一般廣播子網中網路位址為 .64、廣播為 .127，主機為 .65 至 .126。/31 點對點鏈路和 /32 主機路由有不同用途，不能機械套用「減二」。

判斷是否同網段：將位址與遮罩逐 bit 做 AND，再比較網路前綴。最後一個 octet 的 .70 是 01000110；與 11000000 做 AND 得 01000000，即 .64。位址長得相似不代表同一子網。

私有位址 10/8、172.16/12、192.168/16 供內部使用；它們不是自動加密或安全區。192.0.2/24、198.51.100/24、203.0.113/24 是本文範例使用的文件用位址。不要把例題目的主機當成 Internet 上的實際服務。

3.2 路由表選的是下一跳

主機和路由器都要查轉送資訊。查表以目的 IP 比較所有可用路由，優先選最長匹配前綴；同前綴的優先序與成本再依平臺、路由來源及設定決定。預設路由 0.0.0.0/0 只在沒有更具體匹配時使用。

前綴	下一跳 / 出口	用途
192.0.2.64/26	on-link，介面 A	直接連線的子網
198.51.100.0/24	192.0.2.65，介面 A	特定遠端網段
198.51.100.128/25	192.0.2.66，介面 A	更具體的網段
0.0.0.0/0	192.0.2.65，介面 A	其餘目的地

目的 .140 符合 /24 和 /25，選 /25；目的 198.51.100.20 選 /24；192.0.2.80 選 on-link。遠端目的 IP 仍寫在 IP 封包內，鏈路目的 MAC 則是下一跳路由器。一般情況下 ARP 查的是下一跳的 IPv4 位址，不會一路廣播找遠端主機。

下一跳本身也必須可達。上表假設 .65、.66 在介面 A 的直連網段，且鏈路正常。路由存在不等於服務存活；正向路徑可達也不保證回程可達。

3.3 IP 標頭與逐跳處理

來源與目的位址識別 IP 端點；Protocol 欄位指向 TCP、UDP 或 ICMP 等上層；TTL 限制轉送次數。每經路由器轉送會降低 TTL，降到零時丟棄，通常可回 ICMP Time Exceeded。TTL 現在主要表示跳數限制，不能當作精確的存活秒數。

IP 本身不保證到達、順序或無重複。應用是否需要重傳、排序、驗證，要看上層。IPv4 標頭檢查和只涵蓋 IP 標頭；它不能保證整份應用資料不被修改。

3.4 MTU、分片與 ICMP

鏈路 MTU 限制可承載的 IP 封包大小；路徑 MTU 在此模型取各段鏈路 MTU 的最小值。IPv4 可在允許時分片，但分片遺失會讓整個資料報無法重組，且中間裝置可能丟棄分片。設定 DF 的過大封包不能由中間路由器分片，通常需要 ICMP 回報限制。PMTUD 利用這類訊息；封鎖必要 ICMP 可能造成「小資料能通、大資料卡住」。

例：Ethernet MTU 1500，無 options 的 IPv4 20 bytes、TCP 20 bytes，理想 TCP 資料上限為 1460 bytes。加上 VPN、TCP options 或其他封裝後要重新計算；MSS 不是所有封包永久固定為 1460。

ICMP 是控制與診斷訊息，不是 TCP / UDP 埠上的服務。ping 的 Echo Reply 只支持特定路徑與回覆的結論；拒絕 ping 的網站仍可能接受 HTTPS。traceroute 的各跳延遲也不是每段鏈路延遲，ICMP 限速、不同回程和負載平衡會影響結果。

實驗、習題與檢核

做 L03：先手算三個目的位址，再用 ipaddress 和最長前綴查表確認。

1. 求 198.51.100.160/27 的網路、廣播、一般可用範圍與遮罩。
2. 上表目的 198.51.100.200、203.0.113.9 各選哪條路由？
3. 解釋遠端封包的目的 IP 和第一跳目的 MAC 為何不同。
4. 連線可傳 100-byte 內容卻卡在大型回應，提出兩個可測試假設。

檢核：能分開回答「去哪個 IP」「走哪個介面」「交給哪個下一跳」。來源補充 RFC 1918（私有位址）、RFC 5737（文件位址）。來源：[RFC 4632](#)、[RFC 791](#)、[RFC 1191](#)、[RFC 3021](#)。

單元 03A 單介面多 IP、來源選擇與政策路由

學習目標與先備關係

先讀 03 的 CIDR 與下一跳；MAC 與 ARP 可配合 04 閱讀。本單元先用兩個闡道說明位址不等於出口，再學 Linux 政策路由。尚不熟悉規則者，可先完成 3A.1 至 3A.3，讀 10 後回到 3A.4。完成後應能預測指定來源的下一跳、寫出 LAN 例外與回程條件，並區分路由、來源偏好、SNAT 及代理分流。

問題：一張網卡同時持有兩個 IP，應用指定第二個 IP 後，是否一定走第二個闡道？

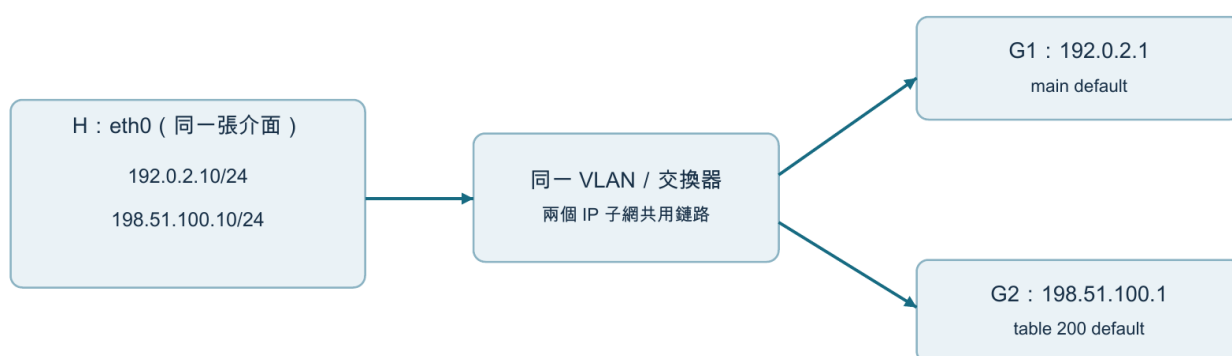
3A.1 一個介面不只一個位址

介面是收送封包的邊界；IP 位址是三層識別。Linux 可對同一介面加入多個位址，無須建立另一張網卡；舊式「IP alias」名稱或 label 不會自動產生獨立鏈路。兩個位址可屬同一子網或不同子網，但新增位址本身不會建立 VLAN 隔離。

以下文件用位址只在隔離實驗中使用。H 的 eth0、G1、G2 都接同一個交換器 / VLAN，G1 與 G2 各有自己的遠端網路。

節點	位址	角色
H eth0	192.0.2.10/24、198.51.100.10/24	一個介面、兩個來源候選
G1	192.0.2.1/24	預設出口 1
G2	198.51.100.1/24	希望來源 2 使用的出口
S	203.0.113.80	範例遠端目的；兩條路徑均可達的假設

單介面多 IP：來源與下一跳分開決定



來源 2 命中 priority 100 → table 200；未指定來源時，本例走 main。

LAN 直連前綴須在所選表可用；綁定來源不會自動選闡道或做 SNAT。

一張介面上的兩個來源與兩個下一跳

兩個不同 IP 子網共用同一個廣播域是可行拓撲；它們仍共用二層廣播，不能達成訪客隔離。若 eth0.10 與 eth0.20 是 802.1Q 子介面，則另有 VLAN 標記與交換器設定，不能與「同 eth0 加兩個位址」混為一談。

新增一般前綴位址時，Linux 通常會建立相應直連前綴路由，並在 local 表記錄本機位址；noprefixroute 等選項及網路管理程式會影響結果。要看實際表，不靠位址清單猜測。

3A.2 四個決策分開回答

決策	例子	不代表什麼
應用來源	bind 198.51.100.10	不自動指定 G2
規則選表	from 來源 2 lookup 200	不先比較所有表的前綴長度
表內路由	default via G2 dev eth0	不會改寫來源位址
鏈路交付	ARP 取得 G2 的 MAC	不保證 G2 有回程或服務可用

Linux 路由的 src 欄位是偏好來源，適用於尚未固定來源的本機發送；它不是 SNAT，不能把已綁定來源或轉送封包一律改成該位址。SNAT 需另讀 12，並記錄改寫前後四元組。

若只有 main 表預設經 G1，應用即使 bind 來源 2，也可能以來源 2、下一跳 G1 送出；G1 及遠端必須知道如何回到來源 2。來源位址與下一跳不是依序排列的一對一配對。

未綁定來源的新連線，初始查表時可能尚未有具體來源可匹配。單靠 from 來源 2 規則，不能期望系統先猜「我要 G2」再選來源 2。本例普通未指定來源的查表走 main / G1，再取該路由偏好的來源 1。需要穩定選出口時，可顯式 bind，或使用目的 / mark 等先選表的政策；真實 socket 與系統可能有其他限制，須實際查驗。

3A.3 main 表的最長前綴匹配

H 的 main 表如下，所有路由皆走 eth0。

前綴	下一跳	偏好來源
192.0.2.0/24	on-link	192.0.2.10
198.51.100.0/24	on-link	198.51.100.10
0.0.0.0/0	192.0.2.1	192.0.2.10

目的 198.51.100.55 匹配 /24，直接 ARP .55；目的 S 匹配 /0，ARP G1。指定來源 2 並不改變此表的目的查找。若下一跳不在可達鏈路上，不能任意加 onlink 來掩蓋拓樸錯誤；須確認實際鄰居解析可行。

3A.4 RPDB 規則先選表，LPM 在表內

Linux Routing Policy Database 按 priority 由小至大檢查規則；規則可以依來源、目的、入口或 fwmark 等查表。預設常見順序為 0 查 local、32766 查 main、32767 查 default。下面在其間加入 priority 100；數字 200 是自訂表號，與 priority 不同。

```
ip rule add priority 100 from 198.51.100.10/32 lookup 200
ip route add table 200 192.0.2.0/24 dev eth0 src 192.0.2.10
ip route add table 200 198.51.100.0/24 dev eth0 src 198.51.100.10
ip route add table 200 default via 198.51.100.1 dev eth0 src 198.51.100.10
```

這是隔離 H 的設定摘錄；完整步驟與清除見 L15，不在日常主機直接照貼。查表規則通常在該表找到可用路由後停止；沒有匹配路由或命中 throw 可繼續下一規則，unreachable 等拒絕路由則可產生終止錯誤。main 內更長的前綴不會跨表推翻已成功的 table 200 查找。

完整例題

目的 / 來源	命中規則與表	下一跳 / 理由
S / 未指定	main	G1，路由偏好來源 1
S / 來源 1	main	G1，來源 1 不匹配 priority 100
S / 來源 2	priority 100→200	G2，表 200 的預設路由
198.51.100.55 / 來源 2	priority 100→200	on-link，200 內的 /24
本機 192.0.2.10 / 來源 2	priority 0→local	本機交付，不送 G2

若表 200 只放 default，來源 2 送同 LAN 的 .55 也會命中該 default、交給 G2；main 已有 /24 不能補救。G2 可能再轉回 LAN，也可能拒絕，不能保證一定斷線。可在 200 放直連路由，或讓較早的目的例外查 main，但要檢查 main 的 default 是否使例外涵蓋過廣。

同表相同前綴可再比較 metric 等；跨表依規則順序，不能把「metric 最低」當全系統唯一排序。ECMP、VRF、mark、IPv6 來源選擇另有機制，本例不模擬。

3A.5 回程、ARP 與反向路徑檢查

正向正確只是一半。G2 須能到 S，S 須能回來源 2，沿路防火牆 / NAT 須允許同一連線的回覆。非對稱路由由本身不必然錯誤，但可能繞過有狀態防火牆或 NAT。TCP 服務回應常以收到連線的本機位址作來源；UDP 服務若綁萬用位址，回覆來源可能受 socket 與查表影響，需紀錄實際端點。

Linux 的 rp_filter 嚴格模式檢查反向最佳路徑的介面，寬鬆模式檢查來源是否可達；有效值還與 all 及介面設定有關。同一 eth0 上的 G1 / G2 通常不會因「閘道不同」而被嚴格模式單獨區分。新增第二張介面時，非對稱路徑才更容易與此檢查衝突。不要將所有回程問題都歸因於 rp_filter。

ARP 行為也受 arp_ignore、arp_announce、arp_filter 等設定影響；多介面同子網尤其要注意主機對其他介面位址的回應。先核對路由、鄰居與封包，再決定是否調整。改 ARP 或反向檢查不是完成隔離的替代品。

3A.6 與代理 / VPN 分流整合

OS 政策路由看到的是該查表階段的 IP 與標記等資訊；域名分流需代理請求、DNS 關聯或其他資料。軟路由還要區分「LAN 客戶端轉送流量」與「代理程式新建的上游 socket」：前者的來源可能是客戶端 IP，後者是代理主機 IP。只寫 from 客戶端規則，不保證代理上游仍走同一路徑。

完整順序例：保留 local 交付；目的 LAN 直連；VPN / 代理 endpoint 走 underlay；剩餘流量依來源或 mark 入隧道。若 endpoint 例外把某 IP 留在 underlay，對同 IP 其他服務也可能受影響，應說明取捨。IPv6、DNS 及 UDP 另做覆蓋表；只加兩個 IPv4 位址不能驗證全流量政策。

Linux 的 ip 命令多為執行中設定；重開機、NetworkManager、systemd-networkd 或其他管理程式可覆寫，持久化需配合實際管理工具。macOS / Windows 有不同路由與來源選擇介面，不能直接套用 Linux RPDB 命令。

實驗、習題與檢核

做 L15。必修離線路線不改主機設定；延伸在 Linux 命名空間逐次看 route get、neigh 與 ping。本例未作 Linux 實測，預期輸出須與實測紀錄分開。

1. 兩個 IP 同 eth0、同 VLAN，為何不等於兩個隔離 LAN？
2. 僅有 main 表時，bind 來源 2 到 S 會用哪個閘道？還需哪項回程條件？
3. 200 只有 default 時，main 內 /24 為何不救回 LAN 目的？提出兩種修正。
4. 表 200 的 src 來源 2 能否把來源 1 改寫成來源 2？
5. 若希望 203.0.113.81 作 VPN endpoint 始終走 G1，在哪個 priority 放目的例外？如何測試來源 2？
6. 延伸：代理上游 socket 的來源與客戶端來源不同，如何用擷取證明兩次查表的差異？

檢核：能在同一筆紀錄分別寫來源、規則、表、匹配前綴、下一跳、介面及回程；能說明哪些只是模型。解答見 answers/lan-solutions.md。

來源（查閱 2026-10-06）：[ip-address](#)、[ip-rule](#)、[ip-route](#)、[Linux IP sysctl](#)。命令以 Linux / iproute2 語義編寫；本文拓樸與題目為獨立設計。

單元 04 Ethernet、ARP 與區域網路

學習目標與問題

以單元 03 的子網及下一跳為基礎，追蹤每段鏈路的來源 / 目的 MAC，演練交換表學習，並區分 VLAN、路由與安全邊界。同一 Wi-Fi 名稱下的兩臺裝置一定可以互通嗎？需進一步知道 VLAN、隔離、子網與政策。

補充 04A 提供單播 / 多播 / 廣播、IGMP / MLD、snooping 及 LAN 服務探索的完整模型，配 L16。

4.1 訊框與交換表

Ethernet 訊框包含目的 MAC、來源 MAC、型別、內容及錯誤檢測等欄位。MAC 位址主要供鏈路交付，IP 位址供跨網路交付。網路卡可採隨機 MAC，軟體也能修改 MAC；它不是可信的永久身分。

交換器從收到訊框的「來源 MAC、入口埠、VLAN」學習轉送表。已知單播目的送到對應埠；未知單播會在同一 VLAN 可轉送埠泛洪，排除入口埠；廣播也限制在該廣播域。表項會老化。這個學習是鏈路轉送，不能理解為交換器已驗證使用者。

例：A 在埠 1、B 在埠 2，表原本空白。A 傳給 B，交換器學得 A→1，泛洪到其他可轉送埠。B 回 A，學得 B→2，查 A 後只送埠 1。第三次 A 傳 B 即可單播。若所有埠抓到一份封包，也要排除埠鏡像、未知單播泛洪與多播等原因。

4.2 ARP 的時序

A=192.0.2.70/26，閘道=192.0.2.65，要送遠端 198.51.100.20。

步驟	動作	得到的資訊
1	查路由，選下一跳 .65	應解析的是閘道 MAC
2	沒有有效鄰居快取時，ARP 廣播詢問 .65	請求只在本鏈路範圍
3	閘道回覆其 MAC，A 更新快取	位址到鏈路識別的對應
4	A 送訊框到閘道 MAC	內層目的 IP 仍是遠端 .20
5	閘道移除入口鏈路封裝，查路由再封裝	下一段有新的鏈路位址

ARP 回覆通常是單播；快取有效時不必每個封包都先廣播。ARP 不會取代路由判斷，也不提供密碼學驗證。ARP 欺騙可能改變本鏈路流向；安全通訊仍需獨立的端點驗證。

4.3 VLAN 與路由

VLAN 將一組實體交換裝置分成不同鏈路廣播域。802.1Q 標籤讓 trunk 承載多個 VLAN；終端的 access 埠常收送未標記訊框，由交換器賦予 VLAN。封包跨 VLAN 一般需三層轉送。VLAN ID 與 IP 子網沒有必然的一對一數學關係，但一個 VLAN 配一個子網通常較易管理。

例：訪客 VLAN 20 可經防火牆上網，但不能到管理 VLAN 10。真正限制來自轉送與存取政策；僅分開 VLAN 卻允許所有跨 VLAN 流量，並不達成該隔離目標。AP 可以橋接到 VLAN，也可能有路由 / NAT 功能，需看設定。

4.4 迴路、STP 與無線

一般 Ethernet 訊框沒有像 IP TTL 的逐跳上限。二層迴路可能使廣播反覆轉送、來源學習震盪。STP 系列透過建立無迴路的轉送拓樸、阻擋部分路徑處理問題；實際收斂和保護功能依版本及裝置。不要為了「提高速度」任意取消迴路保護。

延伸：Wi-Fi 使用共享無線媒介，802.11 訊框與 Ethernet 不完全相同。AP 可在無線與有線鏈路間橋接；「有四個 MAC 欄位」等細節需以具體 802.11 模式說明，不能套用簡化 Ethernet 圖。

演練、習題與檢核

在紙上建立三埠交換表，依 A→B、B→A、C→A 逐步列出「新增表項 / 出口」。可選用 Wireshark 觀察自己的 ARP；沒有擷取資料時提交模型表即可，不聲稱已抓到封包。

1. A、B 同 VLAN 不同子網，為何不能只憑 VLAN 推斷 IP 可達？
2. A 找本子網 B 和找遠端 C，ARP 目標分別是誰？請明示路由假設。
3. 為何 IP 封包可以保持遠端目的位址，MAC 卻逐段改變？
4. 為何普通交換器不能僅靠來源 MAC 判定身分可信？

檢核：能逐跳寫來源 MAC、目的 MAC、來源 IP、目的 IP，並將 NAT 另行標示。來源：[RFC 826](#)、[IEEE 802.1Q](#) 工作組。

單元 04A 單播、多播、廣播與 LAN 服務探索

學習目標與先備關係

先讀 04 的交換表及 VLAN。4A.1 至 4A.4 處理鏈路與多播；4A.5 在讀 05 的 DNS 後回讀，IPv6 細節與 11 互相參照。完成後應能追蹤不同交付方式、區分 FDB 與 MDB、解釋 IGMP / MLD 與 snooping，並定位「能以 IP 連線卻找不到印表機」的問題。

本文以「多播」稱 multicast，其他資料也常稱群播或組播。問題：交換器看到多播訊框，是否一定送到所有主機？

4A.1 先分交付語義，再看實際出口

類型	目的與範圍	交換器簡化行為
已知單播	一個鏈路目的 MAC	FDB 命中時送目的埠
未知單播	目的仍是單播 MAC	FDB 未命中時，同 VLAN 泛洪
廣播	ff:ff:ff:ff:ff:ff	同 VLAN 可轉送埠，排除入口
多播	一個群組，可能零 / 一 / 多個接收者	依成員、router port 及裝置政策決定

上表假設無埠隔離、ACL、鏡像及特殊控制訊框規則。泛洪是交換器的轉送動作，不會把單播目的改成廣播。接收端收到鏈路副本，也不表示 IP / socket 或應用一定接受。

IPv4 有限廣播 255.255.255.255 限制在本鏈路；子網導向廣播如 192.0.2.255/24 的語義不同，路由器預設不應轉送這類導向廣播。不要為了跨網段探索直接開啟廣播轉送。IPv6 沒有 broadcast 位址，以不同 scope 的 multicast 等機制完成相關功能。

ARP 請求通常用 Ethernet 廣播；但 ARP 本身不是 IP 資料報 (EtherType 0x0806)，既不是 IPv4 廣播也不是 IPv4 多播。NDP 是 ICMPv6，鄰居解析常送 solicited-node multicast；IGMP / MLD 管理接收成員，不能替 ARP / NDP 找單播鄰居。

4A.2 IP 群組與鏈路 MAC 不是一對一

IPv4 多播範圍 224.0.0.0/4；224.0.0.0/24 供本地網路控制，239.0.0.0/8 供管理範圍，實際邊界仍需路由與管理政策。IPv6 群組位址由 ff00::/8 開始，含 scope 欄位；ff02 是 link-local scope，不能只看它是 IPv6 就當作全球可路由。

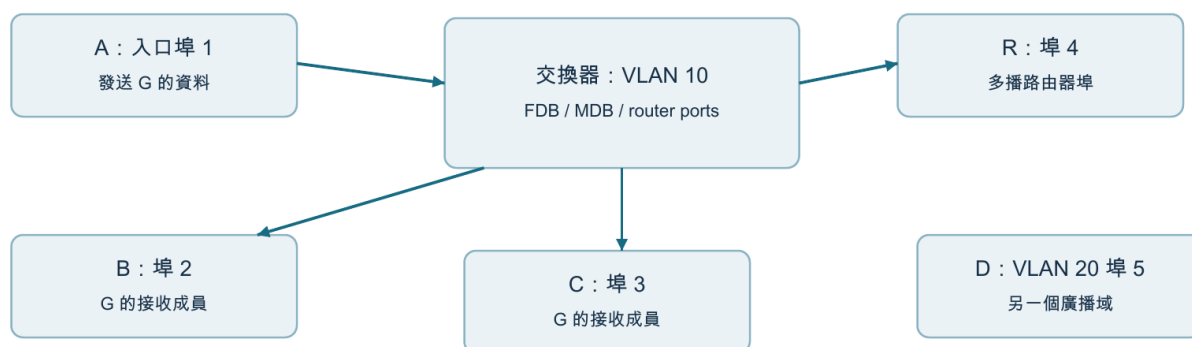
Ethernet 上的 IPv4 多播 MAC 使用 01:00:5e 加上群組 IP 的低 23 bits，所以 32 個 IPv4 群組可以對映到同一 MAC。239.192.0.42 與 224.64.0.42 皆對映到 01:00:5e:40:00:2a；若設備只按 MAC 過濾，可能多送副本，IP 層仍需按群組過濾。IPv6 通常對映到 33:33 加目的位址低 32 bits，也可能別名。

多播發送者不必先加入接收群組；接收者透過 socket 加入指定介面的群組。群組不是可直接建立普通 TCP 連線的單一伺服器。多播資料常用 UDP，但管理封包 IGMP 或 ICMPv6 本身不使用 UDP 埠；不能將「多播只有 UDP」當成分層定義。

4A.3 成員管理、snooping 與路由三種工作

LAN 交付：FDB 與多播成員表分開

G=239.192.0.42：成員埠 2、3 + router port 4



此圖為同 VLAN 模型；跨 VLAN 路由與服務探索需另外設計。

同 VLAN 中的 FDB、多播成員與 router port

機制	知道什麼	負責什麼
FDB 來源學習	VLAN、來源 MAC、入口埠	單播鏈路轉送
IGMP / MLD	某鏈路有接收者對哪些群組 / 來源有興趣	主機向相鄰多播路由器報告
IGMP / MLD snooping、MDB	群組對應成員埠，另記 router port	交換器限制 LAN 內多播副本
多播路由 / PIM 等	跨鏈路的樹、上游與下游	路由器跨子網轉送多播

IGMPv3 用於 IPv4，MLDv2 用於 IPv6 且屬 ICMPv6，兩者都支援來源篩選。ASM (any-source multicast) 以群組 G 識別服務，接收送往 G 的任意來源，可再排除部分來源；SSM (source-specific multicast) 以 (S,G) 識別通道，只接收指定來源 S 送往 G 的流量。來源篩選是協定能力，不等同 ASM / SSM 的定義本身。group membership 不提供密碼學身分驗證，也不代替防火牆。

現行規範 RFC 9776 / 9777 於 2025 年分別取代 3376 / 3810，協定仍是 IGMPv3 / MLDv2。部署可能使用較早版本，snooping 支援也需看設備；不要因 RFC 號更新就宣稱所有交換器已支援新版功能。

一次成員生命週期

1. B 的應用在指定介面加入群組 G；OS 更新接收狀態，依協定發送成員報告。
2. snooping 交換器從控制訊息更新 B 所在埠的 MDB 狀態；router port 另由協定觀察或靜態設定取得。
3. A 送 G 的資料；已知群組通常送成員埠與相關 router port，排除入口，不查普通單播 FDB 來找全部成員。
4. Querier 發 Query，接收者回報仍需哪些流量；表項會依時間與規則更新。

5. 應用 leave 或停止回報，不一定瞬間刪掉整個埠：埠後方可能還有其他接收者，需查詢 / 逾時或明確追蹤。Fast leave 只適合符合拓樸假設的部署。

沒有 querier 時，某些 snooping 環境會在老化後丟失成員狀態，呈現「剛開始正常，幾分鐘後不穩」。不同設備可能退回泛洪或丟棄；不能把症狀當唯一診斷。查 Query、Report、MDB、router port 與計時器才有證據。

未知多播是否泛洪取決於設備與政策；router port、控制群組及版本相容性也會影響轉送。IPv4 的 224.0.0.x 非 IGMP 資料在 RFC 4541 建議下應送所有相關埠，不能一律按普通群組的 MDB 裁切。因此「開 snooping 後 mDNS 只到已加入者」不是通用保證。

4A.4 多播跨 VLAN 需要另一層設計

同 VLAN 的橋接不減少 IP TTL / Hop Limit；路由轉送則要處理跳數與 scope。snooping 不建立跨 VLAN 多播路徑。若目標是跨子網送同一資料流，需合適的多播路由、接收成員、來源路由、RPF 檢查及安全政策；PIM 是常見控制協定家族，不等於所有多播部署都必須用 PIM。

許多服務探索採本鏈路 scope；即使網路支援一般多播路由，也不代表能直接轉送這些探索訊息。AP 的多播轉單播、用戶端隔離、無線省電 / 速率策略會改變鏈路現象。先描述自己的 VLAN 與 AP 模式，再解讀擷取。

完整轉送例題

A、B、C、R 在 VLAN 10 的埠 1、2、3、4；D 在 VLAN 20 的埠 5。G=239.192.0.42，MDB 已知 B 與 C 是成員，R 是 router port；無其他過濾。

A 傳入埠 1 的訊框	出口	判斷
已知單播 B	2	FDB 命中
未知單播 X	2、3、4	同 VLAN 泛洪，目的仍單播
廣播	2、3、4	不送 VLAN 20
已知 G 多播	2、3、4	成員埠加 router port
C 離開後的 G，假設成員移除已確認	2、4	不再送 3，仍送 router port
mDNS 224.0.0.251	本例 2、3、4	link-local 控制範圍採泛洪政策

這是明示政策的紙上模型，並非所有交換器的逐條保證。IGMP 控制訊息有自己的轉送規則，不以此資料訊框表替代規範。

4A.5 mDNS 與 DNS-SD：發現不等於連得上

mDNS 將本鏈路名稱查詢 / 回覆送到 224.0.0.251 或 ff02::fb 的 UDP 5353；.local 是重要的本地名稱空間。協定也包含可要求單播回覆等情況，所以看到單播 UDP 5353 不必立即判定異常。RFC 6762 §11 建議 mDNS 回應（含單播回應）以 IP TTL / Hop Limit 255 發送，以相容早期實作。對送往本鏈路目的位址的查詢，查詢端須只接受本鏈路回應：送至 224.0.0.251 / ff02::fb 的回應視為來自本鏈路；單播回應另檢查來源是否屬於本地子網 / on-link 前綴。本鏈路限制不靠把 TTL 設成 1。IP TTL / Hop Limit 與 DNS record TTL 是不同欄位，前者管封包跳數 / 協定檢查，後者管記錄快取。

DNS-SD 描述如何用 DNS 記錄尋找服務：PTR 列出服務實例，SRV 給主機與埠，TXT 提供屬性，A / AAAA 得到位址。它可使用 mDNS，也可用普通單播 DNS；不能把 DNS-SD 與 mDNS 當同義詞。

例：查詢 _ipp._tcp，PTR 得到「Lab Printer」實例，SRV 指向 printer.local:631，TXT 描述能力，再查 A / AAAA 並連線。探索訊息使用多播，不表示列印資料也用多播；後續連線可能是單播 TCP，且仍需路由、ACL 與服務端接受。

現象	優先蒐集的證據
手動 IP 可連但列表沒有印表機	VLAN、5353 收送、AP 隔離、服務廣告與快取
列表有實例但連線失敗	SRV 埠、A/AAAA 可達、服務、防火牆與名稱驗證
跨 VLAN 突然能發現大量裝置	reflector / gateway 的範圍、允許服務與信任邊界
VPN 開啟後本地探索消失	群組出口介面、5353 政策、scope、路由與應用支援

跨 VLAN 可評估 mDNS reflector 或具政策的 discovery gateway，或改用單播 DNS-SD；它們有不同快取、可見範圍與實作限制。只把 LAN 單播路由排除出 VPN，不能保證探索也正常。HTTP CONNECT / 普通 SOCKS CONNECT 不會自動搬運二層廣播或多播；TAP、TUN、WireGuard 等也要依其轉送單位、路由與實作能力另外分析，不能僅靠名稱保證。

不要將看見服務廣告視為端點可信；完成連線後的名稱 / 憑證驗證與存取控制仍是獨立工作。

實驗、習題與檢核

做 L16，改變 MDB、成員老化、router port、VLAN 與未知多播政策，先預測再執行。讀 05 後完成 DNS-SD 資料表；Linux bridge 觀察是延伸，未列為本機實測。

1. 未知單播送到多埠，為何仍不是廣播？
2. 239.192.0.42 與 224.64.0.42 為何可能共用 MAC？它們是否因此是同一群組？
3. snooping、querier 與 PIM 各解決哪個問題？
4. B 離開群組時，為何不能立即刪除整個共享埠？
5. 印表機跨 VLAN 可 ping 卻未顯示列表，提出兩個可區分假設及證據。
6. 已發現 printer.local:631 但 HTTPS 名稱驗證失敗，發現成功能否取代身分驗證？

檢核：能分開記錄目的 IP / MAC、VLAN、FDB、MDB、router port、應用實例與實際連線；知道本模型未實作 IGMP 封包。解答見 [answers/lan-solutions.md](#)。

來源 (查閱 2026-10-06) : [RFC 1112](#)、[RFC 2464](#)、[RFC 4291](#)、[RFC 2365](#)、[RFC 2644](#)、[RFC 9776](#)、[IGMPv3](#)、[RFC 9777 MLDv2](#)、[RFC 4541 snooping](#)、[RFC 4607 SSM](#)、[RFC 6762 mDNS](#)、[RFC 6763 DNS-SD](#)、[Linux bridge](#)。RFC 4541 為 Informational 建議，不代替每款交換器的規格。

單元 05 DHCP、DNS 與設定的生命週期

學習目標與問題

接續 IP 與 ARP。你應能說明 DHCP 租約、DNS 遞迴與權威角色、TTL 快取，以及名稱解析失敗與服務失敗的差異。連上 AP 表示鏈路成立，並不代表已拿到正確 IP、開道和解析器。

普通 DNS 解析與本鏈路 mDNS 需分開；讀完本章接 04A 的 4A.5 與 L16 DNS-SD 工作表。

5.1 設定從哪裡來

IPv4 DHCP 常透過 Discover、Offer、Request、ACK 的流程提供租約與選項，包括位址、遮罩、路由器和 DNS。初始客戶端可使用 0.0.0.0，訊息可能廣播；relay 可把請求轉送到其他網段的伺服器。後續續租不一定重演四個廣播訊息。

租約到期前會嘗試續租及重新繫結；超過租約有效期不能繼續假設位址合法。手動靜態設定、作業系統快取與其他設定機制也可能參與。若 Wi-Fi 正常但只有 link-local 位址，應先查 DHCP 與本地設定，不要立即歸因 DNS。

5.2 DNS 是分散式名稱資料庫

stub resolver 通常把查詢交給遞迴解析器。解析器可回快取，或從根、頂級域、權威伺服器逐步找委派。權威伺服器對特定 zone 提供資料，不等於替使用者遞迴解析所有名稱。實際流程受快取影響，不是每次都詢問根。

記錄	功能	注意
A / AAAA	名稱到 IPv4 / IPv6	可有多筆，不保證只用第一筆
CNAME	別名指向另一名稱	仍需解析後續名稱
NS	zone 的權威伺服器	委派可能需要 glue 位址
MX	郵件交換器	不是網頁服務位址
TXT	應用定義文字資料	本身不提供安全保證

DNS 不解析完整 URL。https://example.test:8443/a 的主機名是 example.test，埠與路徑由其他協定處理。.test 是示例用途；本教材不要求連到外部該網域。

5.3 查詢、回應與快取

DNS header 含 transaction ID、旗標、各區段數量。Question 有名稱、type、class；Answer 可以含資料與 TTL。名稱用長度加 label 表示，回應可用壓縮指標引用訊息中先前出現的名稱。parser 必須限制長度及指標迴圈，不能把網路輸入當成可信字串。

TTL 是允許快取儲存的時間尺度。假設權威 TTL=300 秒，解析器在 t=0 收到資料，t=120 回覆時正常剩餘 TTL 約 180；新增查詢不會任意把原資料再延長 300。負面回覆也可能快取。NXDOMAIN 表示名稱不

存在；NOERROR 且無指定型別資料並不等於 NXDOMAIN。timeout 則根本沒有拿到有效回應。

DNS 支援 UDP 與 TCP。UDP 回覆截斷時可能改用 TCP；不能把「DNS 通常用 UDP」簡化為「封鎖 TCP 53 不影響 DNS」。DoH / DoT 改變傳送的保密方式，DNSSEC 提供資料驗證；加密傳送不等於 DNS 資料本身已驗證，也不使解析器無法看到查詢。

5.4 完整故障例題

使用者說「網站不能開」。已知以正確 IP 連線可到服務、原名稱解析 timeout，且解析器位址不可達。先檢查解析器路由、DNS 設定及政策，再測查詢。若解析得到位址但 HTTPS 名稱驗證失敗，不能改成停用憑證檢查；應核對名稱與憑證。直接以 IP 開 HTTPS 可能因 SNI / Host 與名稱驗證失敗，不能用它取代完整的域名測試。

實驗、習題與檢核

L04 提供人工作成的 DNS 二進位訊息，解析 header、QNAME 與 A 回應。它驗證格式理解，不代表已向真正權威伺服器查詢。

1. 快取在 $t=40$ 收到 $TTL=600$ 的資料， $t=190$ 時的正常剩餘 TTL 為何？
2. 將遞迴解析器、權威伺服器和 DHCP 伺服器的責任分開。
3. NXDOMAIN、沒有 AAAA、逾時各支持什麼結論？
4. 請畫出「已快取」與「未快取」兩種解析時序。

檢核：能寫出 DNS 查詢實際送往哪個端點，並區分名稱、位址、服務可達與身分驗證。來源：[RFC 2131](#)、[RFC 1034](#)、[RFC 1035](#)、[RFC 2308](#)。

單元 06 Socket、UDP 與應用訊息邊界

學習目標與問題

理解單元 03 的 IP 與單元 05 的名稱解析後，本章建立應用與作業系統傳輸服務的介面。你應能寫出端點，解釋 bind、listen、accept、connect，並設計可靠的訊息切分。為何 send 兩次，對方卻不能保證 recv 兩次？

6.1 介面與端點

Socket 是程式使用網路服務的介面，不是某種實體線路。TCP 連線可用協定、來源 IP / 埠、目的 IP / 埠識別。伺服器的 listening socket 等待新連線，accept 取得每條連線自己的 socket；接收新客戶端不必為服務換一個固定埠。

bind 設定本地端點；connect 發起 TCP 連線或為 UDP 設定預設對端。UDP 的 connect 不建立 TCP 式可靠握手。127.0.0.1 僅本機 IPv4；:::1 僅本機 IPv6；綁 0.0.0.0 可能接受其他介面流量。本課程程式固定 loopback，埠由系統挑選，避免佔用其他服務。

6.2 UDP 資料報

UDP 保留資料報邊界，通常每次 recvfrom 取得一份資料報及來源端點；buffer 太小時可能截斷或依平臺報錯，不能用第二次讀取補回同一份資料報。UDP 本身不提供可靠性、排序、流量控制或壅塞控制服務；應用若需要就要設計，例如 QUIC。在公共網路大量傳 UDP 也必須考慮壅塞與公平性。

資料報適合以可接受遺失與時效性為條件的工作。舊畫面晚到沒有價值時，重傳可能適得其反；但遊戲登入、狀態同步與檔案傳輸仍有不同可靠性需求。「UDP 都快、TCP 都慢」沒有指明工作負載。

6.3 TCP 位元組串流

TCP 提供一條有序位元組串流。send 可能只接受部分 bytes；Python sendall 幫忙持續傳送，仍不表示遠端應用已處理。recv 最多回指定長度，可能更少；回 b" 表示對方傳送方向已正常結束。timeout 是等候期限耗盡，不等於收到 EOF。

應用需自己切訊息：固定長度、分隔符號、長度前綴或自描述格式。二進位可採「4-byte 大端長度 + payload」。讀取程式要一直累積到需要的長度，限制最大訊息，遇中途 EOF 就報錯。不得對每次 recv 直接做 JSON 解析，因為它可能含半個或多個 JSON。

例：先傳 00000003 (十六進位) 和 cat，再傳第二訊息。底層可切成 [前兩個長度 bytes]、[剩餘長度 + c]、[at + 下一訊息]。parser 的結果仍應是完整 cat，與如何切段無關。

6.4 HTTP 與計時

HTTP/1.1 有自己的標頭 / 內容界線；Content-Length 是 bytes 數，不是中文字數。使用現成 HTTP parser 可避免自行處理 chunked、關閉與持續連線的許多細節。L01 以標準函式庫觀察端點及 UTF-8 bytes。

記錄「開始 connect 到收到完整回應」與「既有連線送 request 到收到第一個 byte」是不同指標。OS 排程、應用處理和緩衝都會參與，不應宣稱一次 loopback 計時就是 TCP 傳播時間。

實驗、習題與檢核

做 L05：TCP 伺服器以每次最多兩個 bytes 讀取長度前綴與內容，正確解析多份訊息；另觀察 UDP 保留資料報邊界。兩者使用真實本機 socket，但無人為網路遺失。

1. 為何不能假設 `recv(1024)` 一定收到 1024 bytes？
2. 4-byte 長度說明內容 100 bytes，讀到 60 就 EOF，應如何處理？
3. TCP 已 ACK 的 bytes 是否代表應用已寫入硬碟？
4. 你會如何避免不可信長度讓 parser 設定過多記憶體？

檢核：能區分「應用訊息」「API 呼叫」「TCP 段」「IP 封包」。來源：[Python socket 文件](#)、[RFC 768](#)、[RFC 9293](#)。

單元 07 從遺失到可靠傳輸

學習目標與問題

先讀單元 02、06。你應能用狀態機解釋序號、ACK、逾時與重傳，計算 stop-and-wait 的效率，並比較 Go-Back-N 與 Selective Repeat。如何在可能遺失或重複的通道交付一次正確訊息？

7.1 偵測與恢復是不同責任

校驗可偵測部分資料損壞，卻不能恢復遺失內容；普通 checksum 也不抵抗惡意修改。ACK 回報接收狀態，序號讓接收端辨認重複與順序，計時器讓傳送端在沒有進展時再傳。可靠性需求必須說明「對傳輸接收端交付」還是「對業務執行一次」。後者往往還要交易 ID、持久化與去重。

7.2 Stop-and-wait 狀態機

傳送端先傳一個資料單位，再等待對應 ACK。接收端收到新序號就交付並 ACK；重複序號只重 ACK，不再次交付。資料遺失會使傳送端逾時；ACK 遺失也會逾時，重複資料由接收端去除。收到任何 ACK 就前進是不正確的：舊 ACK 可能延遲到達。

事件	傳送端	接收端
資料 0 到達	等 ACK 0	交付 0，回 ACK 0
ACK 0 遺失	逾時重傳 0	收到重複 0，不重複交付，重 ACK
有效 ACK 0 到達	前進到資料 1	準備接收下一序號

交替 bit 只用 0、1，需假設通道 FIFO、不重排，且舊資料不會存留任意久，否則重用序號會產生歧義。實際協定使用更大序號空間與生命週期規則。

7.3 完整效率例題

無遺失，資料長 1000 bytes、速率 10 Mb/s，資料傳送需 0.8 ms；假設從資料最後一個 bit 送出到 ACK 收齊需 40 ms，且 ACK 傳送開銷已包含在這 40 ms。每次週期 40.8 ms， $\text{goodput} = 8000 / 0.0408 = \text{約 } 196 \text{ kb/s}$ ，鏈路利用率約 1.96%。

視窗允許多個單位在途，就能讓等待時間和後續傳送重疊。若理想視窗 W 個同長單位，粗略上界為 $\min(R, W * L / (RTT + L/R))$ ， L 以 bits 計量，此處 RTT 採上述等待定義。實際 TCP 的 RTT 取樣及視窗以 bytes 計量，本例不能直接替代 TCP 模型。

7.4 滑動視窗與兩種恢復方法

Go-Back-N 教學模型以累積 ACK，接收端只按序接收；缺口後的資料可丟棄，逾時重傳缺口起仍未確認的一段。Selective Repeat 可快取失序資料，只重傳缺少的單位，代價是更多狀態與序號管理。對於環狀序號空間大小 M ，典型 Selective Repeat 模型要求視窗不超過 $M/2$ ，避免新舊視窗重疊。

例：送 0、1、2、3，1 遺失。GBN 接收端先交付 0，後續 2、3 不在預期位置；SR 可留住 2、3，重傳 1 到齊後按序交付。不要把教學 GBN / SR 完全等同於現代 TCP；TCP 累積 ACK 可搭配 SACK 和不同遺失恢復機制。

7.5 逾時與證據

逾時太短產生不必要重傳，太長使遺失恢復慢。延遲變動、重新排序、ACK 延遲和真正遺失可能產生相似觀察。重傳次數不是直接的線路丟包率。L06 是確定性的離散回合模型，故障規則由我們指定，這讓因果可重現，卻不能量測真實 TCP 的 RTO 或壅塞演算法。

實驗、習題與檢核

L02 比較不同視窗的 W / RTT 近似上界；它不含本節額外 L / R 的週期模型；L06 在資料和 ACK 各遺失一次時追蹤交付與重傳。

1. ACK 遺失為何可能產生重複資料？接收端如何避免重複交付？
2. 將例題等待時間改為 80 ms，求理想 goodput。
3. 序號空間 16，典型 SR 的最大視窗是多少？理由為何？
4. 「可靠串流」與「扣款剛好一次」需要的保障有何差別？

檢核：能把 sender、receiver 的狀態分開，指出 ACK 與 timer 如何改變狀態。延伸來源：[RFC 2018](#) (TCP SACK)、[RFC 6298](#) (RTO)。本章 ARQ 例題為教學模型。

單元 08 TCP 的狀態、視窗與壅塞

學習目標與問題

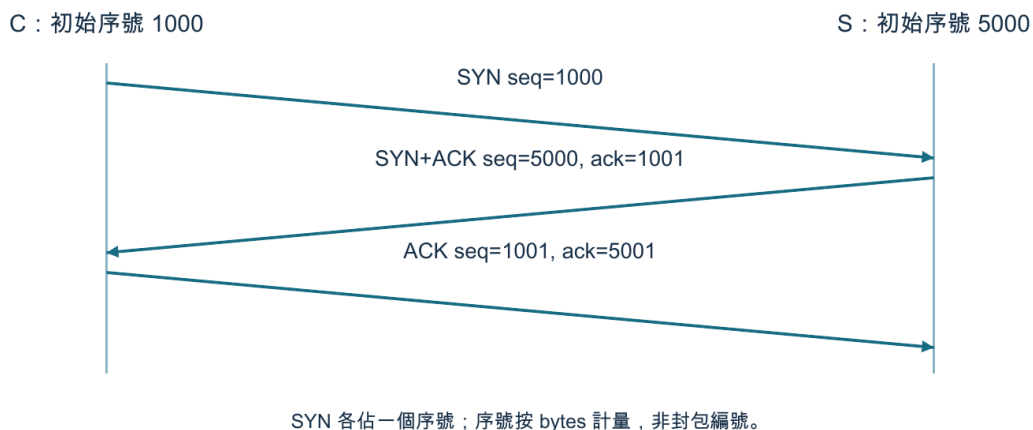
以單元 06、07 為先修。你應能追蹤三向握手、序號與 ACK，區分流量控制與壅塞控制，並分析吞吐量限制。遠端讀取太慢與路徑太擁擠是兩種不同問題。

8.1 握手與序號

假設客戶端初始序號 1000，伺服器 5000。SYN(seq1000)、SYN+ACK(seq5000,ack1001)、ACK(seq1001,ack5001) 建立雙向序號狀態。SYN 各佔一個序號；接著客戶端 100 bytes 的下一個待傳序號 1101。TCP 序號按 bytes 計量，並以模數規則處理回繞。

累積 ACK=N 表示 ACK 發出時希望取得的下一個 byte；不能理解成「第 N 個封包已到」。失序資料可能留在緩衝區，SACK 可描述額外區塊。ACK 到達不代表應用已消費，更不代表完成業務交易。

TCP 握手：兩端建立序號狀態



TCP 序號握手模型

8.2 關閉與半關閉

兩個方向可各自用 FIN 結束傳送；FIN 佔一個序號。關閉可能四段完成，也可合併 ACK/FIN；「一定四次」不成立。半關閉允許一邊已不再傳送、仍接收另一邊資料。RST 代表中止，不能和正常 EOF 混為一談。

TIME_WAIT 協助處理重送的末尾封包及避免舊連線殘留干擾。大量連線時的資源問題需要量測與合理重用，不宜把關閉等待設定為零當成通用最佳化。

8.3 兩種視窗

rwnd 是接收端通告的可接收尺度，保護接收緩衝區；cwnd 是傳送端壅塞控制允許的在途尺度，保護網路。未確認 bytes 在簡化模型下受 $\min(rwnd, cwnd)$ 限制，實際可新增傳送量還要扣掉已在途資料，並考慮應用供給與其他機制。

例：已協商視窗縮放， $rwnd=256\text{ KiB}$ 、 $cwnd=64\text{ KiB}$ 、 $RTT=50\text{ ms}$ 、應用持續供給，視窗上界約 $65536/0.05*8=10.49\text{ Mb/s}$ 。即使把 $rwnd$ 加到 1 MiB ， $cwnd$ 不變時仍不能單靠此提高上界。接收程式

停讀可能令 `rwnd` 降到零；此時不是一定發生網路壅塞。

8.4 遺失恢復與 RTO

RTO 使用 RTT 估計與變動尺度；重傳後難以確定 ACK 對應哪次傳送，經典 Karn 規則避免用模糊樣本更新。逾時通常有退避。重複 ACK 與 SACK 等可協助較早偵測缺口，但重排也可能產生類似訊號。

經典 RFC 6298 更新模型： $RTTVAR \leftarrow (1-1/4)*RTTVAR + 1/4*|舊SRTT-R|$ ，再 $SRTT \leftarrow (1-1/8)*SRTT + 1/8*R$ ； $RTO = SRTT + \max(G, 4*RTTVAR)$ ， G 是時鐘粒度；首次取樣設 $SRTT=R$ 、 $RTTVAR=R/2$ 。RFC 6298 的一秒下限是 SHOULD，具體平臺可能採更短值。作業系統可能採不同允許的演進策略；本教材不以離線表格聲稱實測核心 RTO。

8.5 壅塞控制與公平性

經典 Reno 模型以 slow start 快速探索可用量，再以 congestion avoidance 較緩增加，偵測壅塞後降低視窗。slow start 雖名為「慢啟動」，在持續有資料可送、仍處慢啟動且 ACK 理想時，`cwnd` 約每個 RTT 倍增；增長仍受 `ssthresh`、ACK 時序與實作影響。CUBIC、BBR 等採不同模型，不能把一張鋸齒圖當成所有 TCP。

遺失型演算法與延遲 / 模型導向演算法可能對排隊和競爭有不同反應。公平性受 RTT、演算法、並行流數、瓶頸和排程影響；開更多連線可能取得更多資源，不代表協定更有效率。AQM / ECN 可提供壅塞訊號，深緩衝則可能造成 `bufferbloat`。

8.6 從資料到診斷

下載低速可以由小視窗、丟包、應用停頓、CPU、磁碟或瓶頸造成。看到重傳不能立即指控 ISP；看到低 CPU 也不能排除單執行緒瓶頸。先記錄工作負載、RTT、bytes、接收視窗和時間範圍，再針對假設增補資料。

實驗、習題與檢核

L05 驗證串流邊界，L06 驗證教學重傳狀態；選做 Wireshark 才討論真實 TCP 序號與視窗，擷取位置要記錄；Wireshark 預設常顯示相對序號，比對時要分清相對序號與原始序號。

1. 上述握手後客戶端連續送 100 與 200 bytes，無遺失時累積 ACK 是多少？
2. `rwnd` 很大、`cwnd` 很小時，增加接收緩衝可以保證增速嗎？
3. 接收端零視窗與遺失造成逾時，所需調查有何不同？
4. 為何不可只用下載平均速度判定演算法公平性？

檢核：能區分 bytes 序號、`rwnd`、`cwnd`、in-flight、goodput。來源：RFC 9293、RFC 5681、RFC 6298、RFC 7323、RFC 9438 (CUBIC)。

單元 09 期中整合與路徑追蹤

學習目標與案例

本章整合 01 至 08 及 03A / 04A 的基礎，不增加新的協定。你應能將鏈路、設定、解析、路由、TCP 與 HTTP 連成一份有假設及證據的報告。以下為人工題組，文中的位址與日誌均是教學資料。

A=192.0.2.70/26，閘道 .65，DNS=.66；三者在同一 Ethernet。DNS 將 lesson.test 解析為 198.51.100.20。假設 ARP 快取空、伺服器支援 TCP/HTTP，沒有 TLS 與 NAT，路由及回程有效。應用取得 1000000 bytes，瓶頸 20 Mb/s，RTT 50 ms；視窗 65536 bytes。

9.1 先畫依賴，再畫訊息

查詢本地 DNS 前先判斷 .66 的下一跳，必要時 ARP 找 .66。DNS 回覆之後，目的 .20 不在 on-link 前綴，ARP 找 .65，送 SYN 到閘道 MAC。TCP 成功後才能傳 HTTP 請求與回應。這是本題的順序，不能說所有網路一律先 DNS 再 ARP：快取、IPv6、加密 DNS、代理與既有連線會改變依賴。

每個路由器重新做鏈路封裝，轉送時降低 TTL。沒有 NAT 的假設下，IP 來源 / 目的保持端點；TCP 可靠性在兩個 TCP 端點之間。若加入終止 TCP 的代理，應重畫兩段連線。

9.2 100 分期中題組

1. 位址與路由 (20)：列網路 / 廣播 / 範圍，解析 DNS 與遠端時各找誰的 MAC。
2. 依賴與責任 (20)：畫上述時序，列出 DNS 回答能證明和不能證明的事情。
3. 效能 (20)：求 BDP、視窗上界、忽略啟動和開銷的內容時間下界。
4. 串流 (20)：HTTP 內容含中文時 Content-Length 如何計算？回應中途 EOF 如何處理？
5. 診斷 (20)：已收到 DNS 回答，TCP connect 卻 timeout。提出兩個不同層次假設，以及能區分它們的下一份證據。

第 1 題的 20 分可改為 LAN 題：H 同 eth0 持有兩個子網位址，給出 main 與 200 路由表及 priority 100 來源規則，預測遠端與 LAN 下一跳 (12 分)；用 FDB / MDB 解釋未知單播與已知多播的出口 (8 分)。資料採 03A / 04A 例題，解答見 LAN 補充解答。總分仍 100 分。

評分看假設、步驟與證據，不只數值。數值誤差可給方法分；將 ping 成功等同 HTTP 成功不得給該推論分。

9.3 完整分析例題

$BDP = 20 \times 10^6 \times 0.05 / 8 = 125000$ bytes。視窗上界 $= 65536 \times 8 / 0.05 \approx 10.49$ Mb/s，比瓶頸 20 Mb/s 低。理想有效資料時間至少 $1 \times 10^6 / 1310720 \approx 0.763$ 秒；實際還有建立、請求、慢啟動、標頭與停頓。因此單用檔案大小除 20 Mb/s 的 0.4 秒低估了本視窗模型。

timeout 可由正向封包被丟棄、回程不可達、服務未回覆或狀態裝置策略造成。查本地路由與送出 SYN 的擷取，再查服務端是否收到及回 SYN/ACK，可以逐步縮小；沒有端點證據時不要指定唯一原因。

實驗與章末檢核

用 L12 的「已知事實、假設 1/2、下一份證據、仍不能確定」欄位完成 9.2 題組，再附時序與計算。至少指出一個目前證據不能確定的原因。解答冊提供標準推理及評分。來源沿用 01 至 08；題組與人工案例由本課程編寫。

單元 10 路由演算法與跨網域政策

學習目標與問題

以單元 03 查表為基礎，區分轉送與路由控制，演練 Dijkstra 及距離向量，理解 OSPF 和 BGP 解決不同問題。最短路徑不一定是組織願意使用的路徑。

10.1 資料平面與控制平面

轉送為每個封包查表選出口；路由控制交換拓樸或可達資訊，產生可供轉送的路由。靜態路由由人設定，動態路由可對變動重新計算，仍要等待偵測與收斂。封包到達的當下，裝置使用的是當時有效的表，不是全網瞬間的完美檢視。

10.2 鏈路狀態與 Dijkstra

鏈路狀態模型讓節點取得拓樸及成本，再算到各目的地最短路徑。A-B=2、A-C=5、B-C=1、B-D=4、C-D=1，假設無向、成本非負且同步一致。

已確定節點	暫時成本 B	C	D
A	2	5	無限
A、B	2	3 (經 B)	6 (經 B)
A、B、C	2	3	4 (經 C)
A、B、C、D	2	3	4

演算法可用下列虛擬碼重現；本例假設非負邊權：

```

dist[A] = 0; 其他 dist = 無限; 未確定 = 所有節點
while 未確定不為空:
    u = 未確定中 dist 最小者
    若 dist[u] 為無限: 結束 ( 剩餘節點不可達 )
    從未確定移除 u
    for u 的每個仍未確定鄰居 v:
        若 dist[u] + cost(u,v) < dist[v]:
            dist[v] = dist[u] + cost(u,v); prev[v] = u
  
```

到 D 路徑 A-B-C-D 成本 4。若成本相同，可有多個正確最短路徑；實作的 tie-breaking 或 ECMP 影響結果。OSPF 使用鏈路狀態思想，實際包含 area、鄰接、泛洪與不同 LSA，不能把小圖當完整 OSPF 封包流程。

10.3 距離向量

距離向量以鄰居通告的距離更新： $D_x(y) = \min_v [c(x, v) + D_v(y)]$ 。節點不必知道完整拓樸，但資訊可能過時。鏈路失效時兩個鄰居可能互相相信對方有路，逐步增加成本，形成 count-to-infinity。split horizon、poison reverse 等可減少部分問題，不能宣稱消除所有任意拓樸的暫態迴路。

例：原 B 可直接到 D，C 經 B。B-D 斷時，假設未啟用 poison reverse，若 C 還公告舊路徑，B 可能改經 C，而 C 仍經 B。需要觀察訊息順序、失效通告和收斂，不只最終表。

10.4 BGP 與政策

自治系統 AS 是路由政策管理邊界。BGP 交換前綴可達資訊與屬性；AS_PATH 協助表達經過的 AS 及迴路避免，選路還受 LOCAL_PREF 等政策影響。更短 AS_PATH 不保證先於本地政策，也不等於物理距離短或延遲低。

企業可能優先付費較低的鏈路，或禁止把某兩個上游之間的流量當免費 transit。可達性宣告與資料封包實際轉送都是需要驗證的狀態。RPKI origin validation 可檢查起源 AS 與前綴授權，不能驗證整條 AS 路徑或保證沒有劫持。

實驗、習題與檢核

L07 重算成本變動，列出同成本路徑。延伸畫三節點距離向量失效時序。

1. 手算上表 A 到各節點最短路徑。
2. B-C 成本改 10，A 到 D 有哪些最短路徑？
3. 路由協定收斂前為何可能有黑洞或暫態迴路？
4. 為何 BGP 不能只使用 Dijkstra 的最短距離決定路由？

檢核：能區分前綴匹配、路徑成本、政策優先和實際量測。來源：RFC 2328 (OSPFv2)、RFC 2453 (RIP)、RFC 4271 (BGP)、RFC 6811 (Origin Validation)。

單元 11 IPv6 與雙協定路徑

學習目標與問題

先讀 03 至 05。你應能解析 128-bit 位址、說明 NDP 及 SLAAC、判斷 IPv6 預設路由來源，並追蹤雙協定的不同路徑。IPv4 代理正常不代表 IPv6 一定同樣被接管。

多播成員管理 MLDv2 與 NDP 用途不同；詳 04A。03A 以 IPv4 來源政策示範，不替代 IPv6 完整來源選擇。

11.1 位址與標頭

IPv6 常以八組 16-bit 十六進位表示；每組可省略前導零，連續零組可用 `::` 壓縮一次。RFC 5952 正規化用小寫、壓縮最長連續零組，同長選左側，單一零組不用 `::`。2001:db8::1 是文件用範例位址；`::1` 是 loopback；fe80::/10 是 link-local 範圍，跨介面使用時常需 scope / zone 識別；fc00::/7 是 ULA 範圍。

IPv6 基本標頭固定 40 bytes，含 Next Header 和 Hop Limit，部分資訊放 extension headers。路由器不替轉送的過大 IPv6 封包分片；來源需處理 PMTU、必要時來源分片。因此不能把「沒有中間分片」寫成「IPv6 從不分片」。ICMPv6 Packet Too Big 是重要路徑訊號。

11.2 NDP 不是 ARP 改成更長位址

NDP 使用 ICMPv6，包含鄰居解析、路由器探索、重複位址偵測等。Neighbor Solicitation/Advertisement 常配合 solicited-node multicast 做鄰居解析；Router Solicitation/Advertisement 提供路由器及前綴相關資訊。IPv6 不用 IPv4 式 ARP 廣播。

link-local 位址可用於相鄰路由器下一跳。預設路由通常來自 RA；DHCPv6（現行 RFC 9915）可提供位址及其他參數，但沒有標準的預設路由選項，因此不能把 IPv4 DHCP 提供開道的模式直接套到 DHCPv6（手動設定另計）。各系統取得 DNS 的方法可能包括 RA 的 RDNSS 或 DHCPv6，需查平臺支援。

11.3 SLAAC 與位址生命週期

SLAAC 利用 RA 的前綴資訊形成位址，還需遵守旗標、有效期和 DAD。Ethernet 等常見鏈路的 SLAAC 需要 64-bit 介面識別碼，通常使用 /64 前綴；不能說所有 IPv6 路由都必須 /64。介面識別碼不必由 MAC 永久生成，穩定位址與臨時隱私位址降低不同形式的追蹤。主機可以同時持有 link-local、global 及臨時位址。

例：路由器公告 2001:db8:10::/64 及預設路由。主機可形成 2001:db8:10::abcd，送另一遠端前綴時選路由器的 link-local 下一跳，再用 NDP 解析鏈路位址。全球位址存在不保證防火牆允許入站，也不保證 ISP 回程正常。

11.4 雙協定與 Happy Eyeballs

名稱可有 A 及 AAAA。應用可競逐或調整 IPv6 / IPv4 連線嘗試，改善其中一條路徑不佳的等待。兩條路徑可能不同，DNS 回覆、路由、代理接管、防火牆和 VPN 設定都需兩套檢查。

IPv6 不依賴 IPv4 式 NAT 才能對外通訊，但仍需防火牆、端點安全和合理位址管理。若全隧道政策只涵蓋 0.0.0.0/0，::/0 可能走原出口；這是政策覆蓋問題，並不是 IPv6 天然不安全。

實驗、習題與檢核

L03 練 IPv6 正規化與前綴；L14 追蹤 A / AAAA 兩種路徑。只做離線表不需要改動自己的 IPv6 設定。

1. 將 2001:0db8:0000:0000:0000:0000:0010:0001 縮寫。
2. IPv6 預設路由通常從哪類訊息取得？
3. 為何「IPv6 沒有 NAT」不等於所有服務都對外開放？
4. 預期全部走 VPN 時，應如何檢查 IPv4 及 IPv6 政策？

檢核：能把位址形成、路由、鄰居解析和存取政策分開。版本註記：RFC 9915 於 2026 年取代 RFC 8415；Happy Eyeballs v3 此次查閱仍為 Internet-Draft，正文採 RFC 8305 的已發布模型。位址表示另查 RFC 4291/5952，RA 的 DNS 選項查 RFC 8106。

來源：[RFC 8200](#)、[RFC 4861](#)、[RFC 4862](#)、[RFC 9915](#)、[RFC 8305](#)。

單元 12 NAT、防火牆與穿透

學習目標與問題

先讀 IPv4、Socket 與 TCP。你應能畫出對映表、區分對映和入站過濾、解釋 CGNAT 與 ICE。兩個內網主機各自可以上網，仍不保證能直接互連。

12.1 位址 / 埠轉換

NAPT 把內部端點改為外部端點，例如 10.0.0.2:51000→203.0.113.5:40000。目的服務看到外部來源；回覆要透過狀態反向轉換。兩個內部端點可以共用一個外部 IP，靠埠與狀態區分。狀態有生命週期，UDP 流量停太久後的回包可能沒有有效對映。

NAT 是轉換功能，防火牆是允許 / 拒絕政策；常在同裝置組合，責任仍不同。stateful 防火牆追蹤流狀態，回程流量符合狀態才可能被允許，但具體 TCP 狀態與規則依實作。入站 port forwarding 既需要轉換設定，也需要政策與回程。

12.2 兩個獨立維度

Endpoint-independent mapping (EIM) 表示同一內部 IP/port 對不同外部目的可重用外部對映。Address-dependent 或 address-and-port-dependent mapping 對 remote IP 或 tuple 更敏感。Filtering 則決定哪些 remote 端點可向該對映入站，可分 EIF、ADF、APDF。名字相似，但不能從對映直接推斷過濾。

L08 例：EIM 把 10.0.0.2:51000 固定為 203.0.113.5:40000；ADF 只准已聯絡過的 remote IP。聯絡過 .10:9000 後，.10:9999 可能可入站，.30:9000 被拒絕。改 APDF 則要求曾聯絡的 IP 與 port 皆符合。

傳統 full cone、restricted cone、port-restricted cone、symmetric NAT 是歷史分類，不足以描述所有現代裝置、不同協定和逾時。報告優先寫實測的對映及過濾條件，不只貼一個 cone 標籤。

12.3 STUN、TURN 與 ICE

STUN 讓客戶端得知特定伺服器看到的外部端點，並可用於連通性檢查；它不自動建立可供所有 peer 使用的轉送服務。TURN 在需要時提供 relay，代價是中繼頻寬與額外路徑。ICE 收集候選、交換候選並檢查可行組合，選可用路徑。信令交換、認證和應用協定仍另行處理。

穿透需要雙方的對映、過濾、state 時效與路徑匹配。某次 STUN 顯示公網 IP，不能證明其他目的下對映一樣，也不保證 P2P 一定成功。UDP 被封、雙層 NAT 或政策限制可令 relay 成為必要選項。

12.4 CGNAT 與回程例題

家庭路由器 WAN 可能是私有位址或共享位址 100.64.0.0/10，而 ISP 再做 NAT。你設定家用 port forwarding，外部封包仍要先經 ISP，家庭設定不能單獨解除上游 NAT。IPv6 可能提供另一條可達路徑，但仍須授權的防火牆政策和服務繫結。

例：服務 10.0.0.3:8080 對映到外部埠 8443。到達 DNAT 後，服務回覆若走另一閘道，反向轉換路徑可能失配。診斷需看入口、轉換後端點、服務日誌和回程表，不只看一個 port test。

實驗、習題與檢核

L08 比較 EIM+ADF/APDF ; L12 分析回程與狀態案例。

1. EIM 是否表示任意來源可入站？
2. STUN 與 TURN 各提供什麼服務？
3. 家用埠轉送已設定，外部卻仍無法連入，CGNAT 如何解釋？
4. 為何從單次 STUN 觀察不能決定所有目的下的 NAT 行為？

檢核：能逐段寫出轉換前後 tuple、狀態與政策。共享位址見 RFC 6598；TCP NAT 狀態見 RFC 5382。

來源：RFC 4787、RFC 6888、RFC 8489 (STUN)、RFC 8656 (TURN)、RFC 8445 (ICE)。

單元 13 通訊安全的密碼學基礎

學習目標與問題

以單元 06 的訊息介面為基礎。你應能區分保密、完整性、身分驗證、前向保密與重放防護，畫出金鑰的來源與生命週期。本章理解服務與威脅，不自行發明或實作密碼協定。

13.1 先問威脅與信任

在路徑上的被動觀察者可看未加密內容；主動攻擊者可改、丟或注入訊息。伺服器本身可能看到解密後資料，端點遭控制時傳輸加密也無法保護應用明文。匿名性關注能否連結使用者、目的與行為，和保密不是同一保證。

13.2 原語與用途

原語	主要用途	不能獨自保證
雜湊	固定長度摘要	無金鑰雜湊不能證明傳送者
對稱加密	用共享祕密保護內容	單獨加密不一定防修改
MAC / HMAC	共享金鑰的訊息驗證	不是網路卡 MAC，也不提供公開驗證
AEAD	保密、完整性及關聯資料驗證	不自動記住以前收過的訊息
公開金鑰簽章	用私鑰簽、公開金鑰驗	仍需知道公開金鑰屬於誰
DH 類金鑰協商	產生共享祕密	無驗證的協商可能被中間人取代

AEAD 接收 key、nonce、plaintext 和 associated data，輸出 ciphertext 與 tag。nonce 唯一性 / 使用限制依演算法及協定；重用 nonce 可能有嚴重後果，不可為教學方便固定到生產設定。associated data 被驗證但不由 AEAD 加密；需要保密的內容應放入 plaintext。

13.3 完整例題：安全的對話需要什麼

A 與 B 各產生臨時 DH 金鑰，交換公開值後推導共享祕密。若 A 沒有驗證 B 的公開值，攻擊者可分別與兩端協商。加入由可信身分金鑰簽署的握手 transcript，可把協商繫結到身分與參數；再用 key derivation 分離不同方向 / 用途金鑰。握手結束後採 AEAD 保護記錄。

這是概念堆疊，不是可直接部署的協定。真實協定還需版本降級防護、transcript 定義、錯誤處理、隨機來源及嚴格訊息格式。

13.4 前向保密與重放

前向保密 PFS 關注長期身分 / 認證金鑰日後洩漏時，已錄下的舊會話能否被解密。臨時 DH 祕密確實銷毀、協定使用正確時，可提供這種保護；僅依固定 PSK 加密的資料通常沒有同樣保證。PFS 不保證端點記憶體或會話祕密當下洩漏仍安全。

一份舊 AEAD 密文可能仍有有效 tag。防重放還需序號視窗、唯一 session 識別、時間或已見 nonce / salt 狀態，按協定組合。時間驗證也會受時鐘偏差及視窗設定影響。TLS 0-RTT 的重放風險需應用處理，不能只看「使用 AEAD」就宣稱安全。

13.5 密碼、憑證與實務邊界

人類密碼通常不是高熵 key，需要合適的密碼衍生；例如 Shadowsocks 2022 要求合適長度高熵 PSK，不接受舊式由人類密碼衍生的方式。憑證把公開金鑰與名稱 / 身分繫結，驗證時還需要信任來源、名稱、有效期及政策。不熟悉的自簽憑證不能僅以「有鎖頭」判斷可信。

實驗、習題與檢核

L13 使用現成 TLS 函式庫，比較信任正確、名稱錯誤與不信任三種情境；不自行實作 AEAD。L14 為每種代理寫金鑰及信任表。

1. 為何普通 hash 不是 MAC？
2. 有效 AEAD tag 為何不一定防重放？
3. 固定 PSK 與臨時 DH 在長期祕密洩漏時可能有何差別？
4. 加密到代理伺服器是否使代理無法知道最後目的地？

檢核：能說出「保護哪一段、對抗誰、信任誰」。來源：[RFC 5116 \(AEAD \)](#)、[RFC 2104 \(HMAC \)](#)、[RFC 5869 \(HKDF \)](#)、[RFC 9846 \(TLS 1.3 \)](#)。

TLS 版本註記：2026 年修訂的 RFC 9846 取代 RFC 8446，協定版本仍為 TLS 1.3；ECH 參考 RFC 9849。

單元 14 TLS、HTTPS 與可見資訊

學習目標與問題

先讀 13。你應能追蹤經驗證的 TLS 連線，區分憑證鏈與名稱驗證，說明 SNI / ECH 及 0-RTT 限制。連線能加密，不代表連線的是預期對象。

14.1 TLS 保護的端點

HTTPS 以 HTTP 承載在 TLS 等安全傳輸上。TLS 通常提供伺服器身分驗證、保密及完整性；客戶端憑證驗證屬於另行設定。網頁帳號密碼是應用登入，不等同 TLS 客戶端認證。TLS 終止在反向代理 / CDN 時，該裝置可讀 HTTP 明文，往後一段需獨立分析。

14.2 TLS 1.3 握手概念

以非 PSK、臨時金鑰協商的一般握手為例，ClientHello 提供版本、key share 等；ServerHello 選參數並建立握手 key；後續包含憑證、簽章和 Finished 等，確認身分與 transcript。TLS 1.3 大部分後續握手已加密。重試、恢復、PSK 及客戶端認證使流程變化，圖解要先寫所採情境。

TLS 用記錄層保護應用 bytes，記錄邊界不等於 HTTP 訊息或 TCP recv 邊界。QUIC 使用 TLS 的握手與金鑰協商，但不是把完整 TLS 記錄層再套在 UDP 上；其封包保護由 QUIC 完成。

14.3 憑證鏈與名稱

驗證至少需要可信根 / 設定、鏈、有效期、用途與預期名稱。連線 localhost 時，即使伺服器憑證由可信來源簽署，SAN 不符合 localhost 仍應失敗。SNI 讓服務端知道希望連到的名稱；SNI 提供名稱不表示客戶端已驗證回傳憑證。IP 位址也有對應的 SAN 型別，不能以任意 DNS 名稱憑證自動通過 IP 驗證。

L13 只把實驗憑證加入該 Python SSLContext，不修改 OS 信任庫。對錯誤名稱和預設不信任分別做負面測試。停用驗證可讓錯誤消失，卻同時放棄預期的安全性，不能當成修復。

14.4 SNI、ECH 與中間節點可見性

沒有 ECH 時 ClientHello 通常可暴露 SNI 及其他協商資訊；TLS 1.3 本身不自動隱藏 SNI。ECH 已於 RFC 9849 標準化，但實際成功依賴客戶端、服務端、設定發現與網路支援。即使 ECH 成功，目的 IP、封包大小、時間、外層 ClientHello 的 public_name 及其他管道仍可能透露資訊。若 DNS 未加密，名稱查詢也可能暴露。

觀察位置	通常能看到	是否能看到 HTTP 內容
本地應用	URL、名稱、內容	能
TLS 前的本地 proxy 入口	若 CONNECT，含目標 host:port	不終止內層 TLS 則不能直接讀取內容
客戶到 proxy 路徑	proxy IP、流量後設資料 (metadata) ； 未保護協定可能暴露更多	看保護範圍
proxy 出口	最終目的 IP 及相關協商	HTTPS 正常端到端時仍不讀 HTTP 明文
終止 TLS 的伺服器 / CDN	解密後請求與回覆	能

14.5 恢復與 0-RTT

會話恢復可降低後續握手成本；0-RTT early data 在完成新握手前送資料，有重放風險，並不具有與正常 1-RTT 資料相同的前向保密條件。應用需決定哪些操作可安全重試 / 重放。支付、修改狀態等不能只因它「比較快」就允許重複執行。效能報告還需分開首次連線、恢復、early-data 被接受或拒絕。

實驗、習題與檢核

做 L13 的正確名稱 / 錯名稱 / 未信任三項。記錄錯誤類別，不要求各系統完全相同文字。

1. 可信憑證為何仍可能名稱驗證失敗？
2. TLS 1.3 是否一律隱藏 SNI？ECH 保護後仍可見什麼？
3. 代理若不終止目標 TLS，可以看到哪些目的資訊？
4. 為什麼應用要另行處理 0-RTT 重放？

檢核：能在拓樸上標示 TLS 起點、終點與信任根。QUIC/TLS 整合見 RFC 9001；HTTP early data 處理見 RFC 8470。來源：[RFC 9846](#)、[RFC 9525 \(服務身分 \)](#)、[RFC 9849 \(ECH \)](#)、[Python ssl 文件](#)。

TLS 版本註記：2026 年修訂的 RFC 9846 取代 RFC 8446，協定版本仍為 TLS 1.3；ECH 參考 RFC 9849。

單元 15 HTTP、QUIC、承載與快取

學習目標與問題

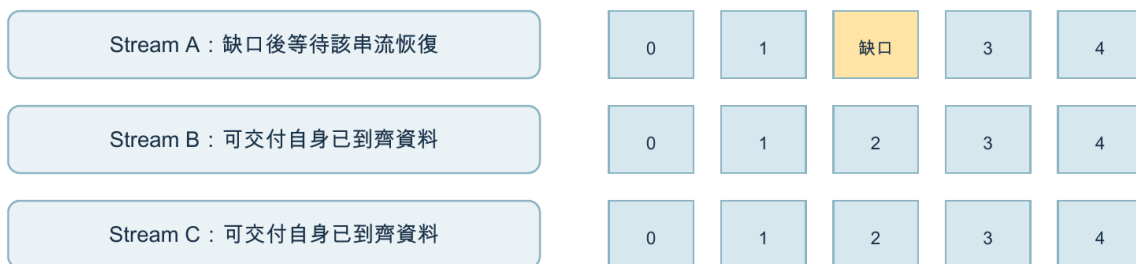
先讀 TCP 及 TLS。你應能區分 HTTP 版本、多工與隊首阻塞，說明 QUIC stream / DATAGRAM，以及 WebSocket / gRPC 在代理中的承載角色。同時發出十個請求不代表底層有十條獨立 TCP 連線。

15.1 HTTP 語義與傳送格式

HTTP 定義 method、status、headers 和資源語義。HTTP/1.1 在 TCP 串流內使用自己的訊息格式；persistent connection 可重用連線，流水化仍有回應順序限制。HTTP/2 把訊息分成 frames 並在 streams 多工，減少應用層的一些排隊，但它仍在 TCP 有序串流上，某個遺失段可阻止後續 bytes 交付到 HTTP/2 層，影響多個 stream。

HTTP/3 把 HTTP 對映到 QUIC。QUIC 使用 UDP 承載，卻為 stream 提供可靠、有序交付。QUIC stream 內遺失資料仍需恢復才能跨缺口按序交付，但一個 stream 的缺口通常不會要求其他 stream 等待同一傳輸序號。不同 streams 仍共享連線級流控、壅塞、路徑及資源；應用依賴和 QPACK 也可能造成等待。因此不能寫成「完全沒有任何 HOL 或延遲互相影響」。

QUIC：串流交付與共用資源



仍共用壅塞控制、連線流控、CPU 與路徑；DATAGRAM 是另一種服務。

QUIC 串流與共用資源

15.2 QUIC 基本機制

QUIC 整合 TLS 握手、安全封包保護、連線識別、streams、流量與壅塞控制。Connection ID 可支援位址變化下的連線識別，遷移仍需協定、實作及路徑條件，不保證行動網路切換毫無中斷。UDP 被阻擋時，應用可能退回 TCP 的 HTTP 版本，需記錄實際協商版本。

QUIC stream 可靠且有序；RFC 9221 DATAGRAM 擴充提供不可靠的訊息，遺失時不由 QUIC 重傳該 DATAGRAM，仍受壅塞和封包大小限制。「QUIC 用 UDP」不能推出所有 QUIC 資料都不可靠，也不能推出所有應用都支援 DATAGRAM。

15.3 WebSocket 與 gRPC

WebSocket 在 HTTP 握手後提供訊息 / 雙向通訊，在常見 RFC 6455 實作中承載於 TCP。代理可將自己的協定 bytes 包進 WebSocket，再經 TLS / TCP 傳輸。gRPC 常以 HTTP/2 傳輸 RPC 與 stream；代理工具借其 stream 作為承載仍需雙方相容的服務名稱與實作。多套多工並不必然更快，可能增加佇列、複製與 HOL。

協定層疊應寫成「應用 TLS / 代理 payload / WebSocket / 外層 TLS / TCP / IP」等具體圖，不可把 WebSocket、TLS、VLESS 排成同一層的可替代協定。

15.4 快取與 CDN

快取複用可省傳輸與往返；Cache-Control、Age、驗證器 ETag / If-None-Match 影響是否直接用快取、重新驗證或下載。CDN 就近提供資源，但選擇受 DNS、Anycast、負載、內容位置和政策影響，地理最近不保證最少 RTT。

例：快取回應 max-age=120，Age=90，假設其他條件不限制，freshness 約剩 30 秒；過期後若 ETag 重新驗證成功，可回 304 複用原內容。no-cache 一般要求使用前驗證，不等同禁止儲存；no-store 才是禁止儲存的指令語義。登入和敏感內容的快取需遵守適用指令。

實驗、習題與檢核

用 L14 畫 H2/TCP 和 H3/QUIC 的三 stream 遺失情境，不製造未經實作驗證的真實封包。L11 設計同路徑首次 / 恢復小請求與大檔案對照。

1. HTTP/2 多工為何仍受 TCP 交付缺口影響？
2. stream 與 DATAGRAM 的可靠性各是什麼？
3. 哪些共享資源可讓不同 QUIC streams 仍互相影響？
4. no-cache 與 no-store 為何不能當同義詞？

檢核：能同時說出應用語義、stream 語義、封包傳輸和保護方式。QUIC/TLS 與 QPACK 另查 RFC 9001、RFC 9204。WebSocket 經 H2/H3 的擴充另查 RFC 8441/9220。來源：RFC 9110、RFC 9111 (快取)、RFC 9112、RFC 9113、RFC 9114、RFC 9000、RFC 9221、RFC 6455、gRPC 核心概念。

單元 16 無線、量測與分層排錯

學習目標與問題

整合鏈路到應用。你應能選擇下一項可區分假設的證據，設計受控比較，並說明無線標稱速率為何不等於應用 goodput。最低路線聚焦排錯與量測；無線訊框結構是延伸。

16.1 Wi-Fi 的共享媒介

Wi-Fi 以共享無線媒介傳輸，通道競爭、干擾、訊號、調變、重傳和客戶端能力共同影響吞吐。標稱 PHY rate 不是獨佔應用速率；協定開銷、競爭及雙向流量都佔媒介時間。擴大通道寬度可增加潛在容量，卻也會擴大受干擾及共享範圍，不能無條件保證更快。

AP 常橋接無線到有線，但家用整合裝置可同時做路由、NAT、防火牆和 DHCP。Mesh / repeater 的無線回程也消耗資源；有線回程可以改變瓶頸，仍需同工作負載比較。僅看到滿格訊號不能排除壅塞和回程問題。

16.2 先分失敗的階段

已知現象	下一份證據	能縮小的範圍
無合法 IP 設定	租約 / 介面 / DHCP 日誌	鏈路或設定
解析 timeout	解析器端點與路由、query/reply	DNS 路徑，不等於名稱不存在
TCP timeout	SYN 出口和服務端是否收到	正向、回程、策略或端點
TLS 驗證失敗	錯誤類別、名稱、鏈、時間	身分 / 信任，不宜停用驗證
HTTP 錯誤 status	應用日誌與回應	已有 HTTP 對端回覆，需確認是源站或中繼，再查業務 / 資源
小資料正常大資料卡	packet 大小、PMTU、服務 timeout	MTU 假設與應用假設

ping、traceroute、埠探測與 HTTP 各回答不同問題。不要用它們互相代替。抓包須記錄介面、時間和方向；硬體 offload 可能使主機擷取顯示 checksum 異常或大段，不能立即判定線上封包壞。

16.3 比較的控制條件

先寫「想改變什麼、希望測什麼、保持什麼」。比較有線 / 無線時固定服務、資料、客戶端、時間和並行連線；同時記錄路徑與負載。比較代理協定還要固定軟體版本、加密方案、傳輸、伺服器、CPU 及網路條件。

延遲報告至少含樣本數、中位和適當尾端統計；吞吐報告說明有效 bytes、計時起迄和暖身。L11 使用 nearest-rank 定義 p95，不混用不同軟體的插值。只有十筆時 nearest-rank p95 等於最大值，不能強烈推論罕見尾端。

16.4 完整診斷例題

人工案例：DNS 成功；TCP 連線成功；TLS 失敗顯示 hostname mismatch；改用正確 hostname 後成功。最直接的原因是預期名稱與憑證不匹配。仍需核對是否連錯伺服器或憑證部署錯；證據不足時不可指名某個中間攻擊者。安全修復是對齊名稱 / 憑證 / 路由，保留驗證。

對受控故障，一次只改變一個條件，記錄恢復後是否重新成功。不能在同時換伺服器、網路和協定後把所有變化歸因其中一個參數。

實驗、習題與檢核

L11 提交比較計畫，L12 按證據表定位故障。

1. 為何標稱 1200 Mb/s 無線不保證下載 1200 Mb/s？
2. DNS 已成功、HTTPS 仍失敗，可從哪些階段區分？
3. 載入時 RTT 升高、無載入恢復，應設計何種排隊驗證？
4. 報告「平均更快 10%」還缺哪些資訊？

檢核：能將事實、推論、未排除假設分欄，並提供可重複步驟。來源：[Wireshark 使用手冊](#)、[RFC 6349 \(TCP 量測框架\)](#)、[RFC 7679 \(單向延遲\)](#)。無線段落為通用模型，不指定某裝置的當前效能。

單元 17 代理的端點、目的位址與解析位置

學習目標與問題

先讀 Socket、DNS、TLS。你應能畫出代理前後的連線，解釋 HTTP CONNECT、SOCKS5 的目的位址編碼，以及誰實際解析名稱。代理是代替應用連往某個目的地的服務，不自動接管電腦全部流量。

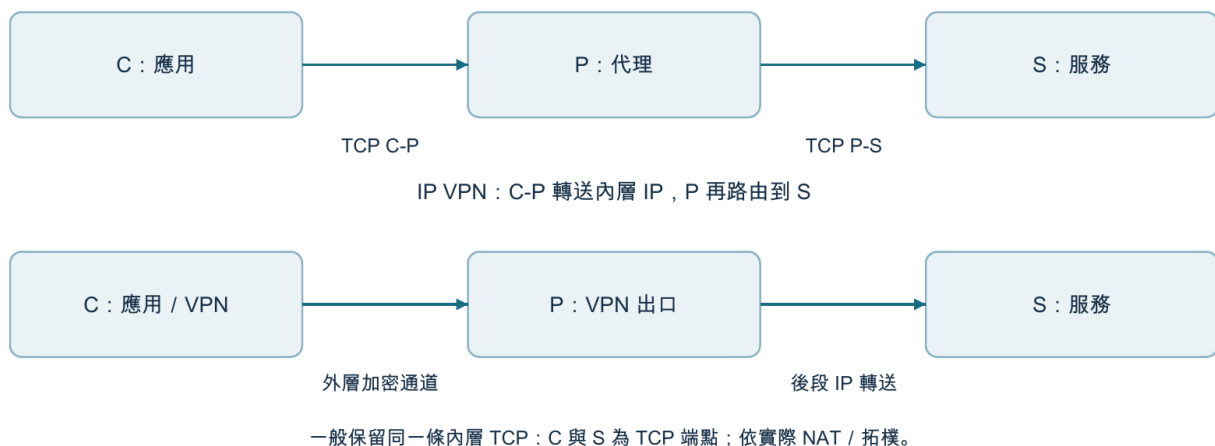
17.1 先畫三個角色

客戶 C、代理 P、目標 S。常見 TCP 代理有 C↔P 及 P↔S 兩段 TCP 連線；代理收到 bytes 後在另一條連線送出。傳輸層 ACK 的端點改變了，但內層 HTTPS 若仍在 C 與 S 間，P 不會因此自動取得 HTTP 明文。TCP 終止和 TLS 終止是不同決定。

正向代理替客戶向外連線；反向代理代表服務接收外部請求，再送後端；兩者方向是服務角色，不是有沒有加密。鏈式代理還有額外中繼與解析選擇。記錄拓樸時至少列每段來源 / 目的端點、轉送單位、TLS 終點與 DNS 發起者。

代理與 IP VPN：TCP 端點的差異

常見 TCP 代理（加密端點另行分析）



TCP 代理與 IP VPN 端點比較

17.2 HTTP CONNECT

客戶向 HTTP 代理發送 `CONNECT lesson.test:443 HTTP/1.1`，代理建立目的 TCP 連線，成功回 2xx 後進入隧道。後續 bytes 可以是 TLS，不是每份都由代理解析為 HTTP。一般 CONNECT 建立 TCP 隧道；UDP 代理需要其他擴充，例如 CONNECT-UDP，不能把普通 CONNECT 當任意 IP VPN。

若 C↔P 本身是明文 HTTP，旁觀者可讀 CONNECT 目標，即使內層 HTTPS 仍加密。若代理入口支援 HTTPS，外層 TLS 保護 C↔P 這段，仍需驗證 P 的憑證，內層 TLS 則驗 S。Proxy-Authorization 驗證的是代理使用權，不是目標網站身分。

經正向 HTTP 代理存取未加密的 HTTP 資源時，HTTP/1.1 客戶端須以 absolute-form 送出請求目標（CONNECT 與伺服器整體 OPTIONS 除外，RFC 9112 §3.2.2），例如 `GET http://lesson.test/a`；P 可讀 HTTP 內容與標頭，這與 CONNECT 隧道不同。明文 C↔P 也會暴露 Proxy-Authorization 等資訊。

17.3 SOCKS5

SOCKS5 先協商驗證方式，再送命令。CONNECT 要求建立 TCP 轉送；BIND 和 UDP ASSOCIATE 是另外功能，實作不一定全支援。目的 ATYP 可表 IPv4、域名、IPv6。域名形式可把解析交給代理；本地先解析再送 IP 則把解析留在客戶。客戶軟體的選項決定實際行為。

UDP ASSOCIATE 通常有 TCP 控制連線與 UDP relay 端點；資料報還有 SOCKS UDP 標頭，不是把應用 UDP 原樣送到 TCP 埠。SOCKS5 本身不提供全面的傳輸加密；帳密驗證若走未保護路徑也不是安全傳送。不要把 SOCKS5 與加密代理協定混為一談。

17.4 完整 DNS 路徑例題

本地先解析 lesson.test 得到 192.0.2.80，再把 IP 交給代理：本地 DNS 看見名稱，代理只收到該 IP，地理選址可能依本地解析器。傳域名給代理：本地不必為該目標查 A，代理用自己的解析器取得位址，當地可能選到不同 CDN。代理本身名稱的 bootstrap 解析仍可能在本地發生。

資料	本地解析模式	遠端解析模式
目標名稱由誰查	C / 其解析器	P / 其解析器
代理請求目的欄位	IP+port	域名+port
目標 TLS 預期名稱	仍可為 lesson.test	同樣可為 lesson.test
要查的失敗點	本地解析與代理連線	代理解析與出口連線

「DNS 經代理」需要具體描述 DNS 請求或解析責任走哪條路，不能只看一個 remote-DNS 開關。DNS 結果、目標名稱驗證與資料出口是三件事。

實驗、習題與檢核

L09 用真實 loopback CONNECT 及 SOCKS5 子集轉送本機 HTTP。它刻意不加密，只演示代理語義；不作公開服務或加密協定的相容性實測。

1. 代理終止 TCP 時，是否一定終止內層 HTTPS？
2. 普通 CONNECT 可以直接轉送任意 UDP 嗎？
3. SOCKS5 域名與 IP ATYP 如何影響 DNS 位置？
4. C→P 有 TLS 時，P 是否還知道目的位址？

檢核：能逐段標出 TCP/TLS 端點及誰解析。來源：RFC 9110 CONNECT、RFC 9112 請求目標形式、RFC 1928 SOCKS5、RFC 1929 帳密驗證、RFC 9298 CONNECT-UDP。

單元 18 Shadowsocks 與 VMess：加密代理的責任

學習目標與問題

先讀代理、密碼學與可靠傳輸。你應能區分代理請求、加密記錄及外層承載，說明 AEAD 不自動提供前向保密和重放防護，並比較版本相容性。使用者常寫的「shadow socket」通常指 Shadowsocks，正文採其正式名稱。

18.1 Shadowsocks 的核心路徑

本地代理收到應用的目的位址與資料，向 Shadowsocks 伺服器送受保護的請求；伺服器解出目的地，再開 TCP 或 UDP 出口。目的網站不需要支援 Shadowsocks。應用 TCP 在本地入口終止，資料經代理通道轉送；UDP 則以指定格式封裝及轉送。

SIP004 / SIP007 的 AEAD 模式（常稱 2017）使用 salt 推導每方向會話子金鑰，再以 AEAD 保護 chunk。TCP 的長度與內容都有自己的保護結構；UDP 通常以資料報方式保護。這些是協定記錄，不等於 Socket recv、TCP 段或 IP 封包邊界。SIP022 的 2022 版修改金鑰與訊息結構，不能只換名稱卻假設兩端互通。

18.2 金鑰、版本與重放

項目	AEAD 2017 基礎模型	AEAD 2022 (SIP022)
共同祕密	可由密碼或 key 得到主祕密	規範要求合適長度高熵 PSK
每會話材料	salt 與衍生子金鑰	依新版格式的 session / salt / packet 狀態
重放	早期基礎 AEAD 格式未強制完整重放保護；需另查實作的 salt / 狀態過濾	明確要求重放防護機制
時間檢查	基礎 AEAD 格式與實作另查	TCP / UDP header 的 timestamp 需符合規範時間窗
前向保密	不因 AEAD 自動取得	PSK 設計本身不提供 PFS
相容性	選定 cipher 與版本皆需相容	不是 2017 封包的通用替代名稱

SIP022 的 TCP / UDP header 含時間戳；偏差超過規範的 30 秒須視為重放，TCP salt 另有 60 秒保存要求。範例 method 可見 2022-blake3-aes-128-gcm，不能只比對名稱中的 AEAD。

不要由「salt 每次隨機」推導 PFS：若舊錄製流量中的非祕密 salt 與日後取得的長期 PSK 足以重建金鑰，舊會話仍可能被解密。PFS 需要分析整個 key establishment。弱人類密碼也不因使用 AEAD 而自動變成高熵祕密。

本課不提供自行實作 cipher 的實驗。比較安全性時，記錄 ciphers、key 來源、是否外加 TLS、重放快取及版本，再畫保護邊界。外掛（plugin）是額外的承載 / 流量機制，不能當成所有 Shadowsocks 的固有特性。

18.3 VMess 與 AEAD

VMess 是 V2Ray 生態中的代理協定，不是 V2Ray 軟體的同義詞。請求包括認證與目的資訊，資料可有自身保護；VMess AEAD 的單條連線驗證格式不同於舊版；伺服器是否接受兩種格式依核心版本而定。認證含時間相關條件，時鐘偏差可能令連線失敗；這不是 DNS 或 TCP 可達性的直接結論。

歷史教程的 alterId 及舊驗證選項常與現代 AEAD 模式不同。本教材把它們列為版本閱讀題，不提供無版本的「最佳 alterId 值」。V2Fly 與 Xray 的欄位及支援範圍要各自核對；「同名 VMess」不能取代實際核心版本、驗證方式和設定。

外層可能是 TCP、WebSocket 等，另有 TLS 保護。VMess 自身保護、外層 TLS 和內層 HTTPS 各有端點。外加 TLS 可改變 C $\leftrightarrow$ P 的金鑰協商與可見資訊，但不能因此宣稱所有 VMess 資料原生具 PFS、所有觀察者無法辨識流量，或代理不知道目的地。

18.4 完整安全分析例題

條件：C $\leftrightarrow$ P 只使用 SS2022 PSK，C $\leftrightarrow$ S 另有經驗證的 HTTPS。路徑觀察者看 P 端點與密文 / 時間；P 知道最後目的位址，卻不能直接解開 C $\leftrightarrow$ S 的 HTTP。若 P 的 PSK 日後洩漏，應分別分析外層舊錄製流量與內層 TLS：外層缺 PFS，不等於內層臨時金鑰 TLS 也同時失去 PFS。

這個例子顯示「雙層加密」只描述層數，不提供對所有威脅的統一答案。加入額外加密也會增加 key 管理、CPU、記錄開銷與相容性要求，是否值得需配合威脅及實測。

實驗、習題與檢核

L14 填版本及 key 表；L11 設計 CPU 和 goodput 受控比較。沒有部署真實 SS / VMess 時，提交概念表，不能稱已測加密安全或協定互通。

1. 為何 salt 與 nonce 不等於 PFS？
2. 2017 與 2022 有哪些必須分開核對的欄位？
3. VMess / SS2022 時鐘錯誤與 TLS 憑證錯誤各在哪一層？
4. 代理外層長期祕密洩漏是否必然揭露內層 HTTPS 內容？

檢核：能分開協定版本、cipher、認證、transport 與 TLS。來源：[Shadowsocks AEAD](#)、[SIP022](#)、[V2Fly VMess 設定](#)、[VMess AEAD 設計](#)。讀取日期 2026-10-03；未在本教材執行第三方核心。

單元 19 VLESS、Trojan 與分層組合

學習目標與問題

先讀 13 至 18。你應能把一組常見代理名稱拆成 protocol、encryption、transport、security、flow 及接管方式，辨認不相容組合，並分開協定保證與實作最佳化。這是評估代理最重要的習慣。

19.1 六個欄位分開看

欄位	回答的問題	例子
代理 protocol	請求與認證如何表達目的地	VLESS、Trojan
協定內 encryption	協定層是否提供額外資料保護	基礎 VLESS none、新 VLESS Encryption
transport	同儕間怎樣承載 bytes	RAW/TCP、WebSocket、gRPC、XHTTP
transport security	承載通道怎樣保護 / 驗證	TLS、REALITY
flow	特定實作如何處理資料流程	XTLS Vision
流量接管	應用資料怎樣進入代理	系統 proxy、SOCKS 入口、TUN、透明代理

這些詞不全屬 IETF 統一的設定模型。表格以 Xray 文件的概念拆分為案例；其他核心有自己的 schema。截至本版查閱日期，Xray 現行文件使用 method 與 raw；目前核心仍相容舊 network 鍵及 tcp 值，兩鍵並存時以 method 為準，舊教程則多用 network/tcp。設定須按核心版本比對，不可直接拼接。

協定層次：普通外層 TLS 承載的例子



此為普通封裝模型；Vision direct-copy 後的保護邊界需重新畫。

普通外層 TLS 代理封裝模型

19.2 VLESS 的基礎與新擴充

傳統 VLESS 以輕量請求與 UUID 等識別資訊完成代理認證，`encryption:none` 的基礎模式不在該層加密 payload，常由外層 TLS / REALITY 保護。UUID 是授權識別 / 祕密資料，不是拿來做公開身分簽章的完整 PKI。

此次查閱的 Xray 文件已列出可選 VLESS Encryption，因此「VLESS 永遠完全沒有協定內加密」已不適合作為無條件敘述。新擴充的握手、恢復、填充與相容性要按指定實作閱讀，不能把外層 TLS 欄位和協定內 Encryption 當同一開關。本課不以未經獨立安全分析的新功能名稱推導所有威脅皆已解決。

目的名稱可交給伺服器解析，或由客戶先送 IP；仍需記錄實際目的欄位和 DNS 策略。VLESS 不依 VMess 式時間認證，並不表示任何外層憑證、票證、時效政策都完全不受時鐘影響。

19.3 TLS、REALITY 與 Vision

TLS 提供承載通道的保護與驗證。REALITY 是 Xray 生態的 transport-security 機制，以特定握手 / 驗證與目標站點外觀為設計重點，不是單獨代理目的請求協定；其 client/server 參數、信任方式和 fallback 需獨立核對。正常 REALITY 驗證依預先分發的伺服器 key 材料與 shortId 等設定，不是單靠一般 WebPKI 信任鏈；設定分發管道是重要信任邊界。填充、外觀或 fingerprint 只改變部分可觀察特徵，不構成絕對不可辨識的保證。

Vision 是特定流程處理模式，不是 HTTP 代理協定，也不等同 TCP 的 `rwnd/cwnd`。特定平臺及條件下可減少重複封裝 / 資料複製，Linux splice 等最佳化需要實作條件；不能宣稱在所有系統、所有 payload 和所有 transport 都同樣有效。

相容性有三層：protocol↔transport、transport↔security、flow↔其餘條件。此次文件列 REALITY 可配 RAW、XHTTP、gRPC，不配 WebSocket；Vision 的 flow 是否適用仍須按指定核心、承載及 payload 條件另查。因此「每個欄位各自存在」不等於「任意組合可用」。具體執行前需固定核心版並用其 validator，不在本課假裝已部署驗證。

Vision 的 direct-copy 模式在符合內層 TLS 條件、完成切換後，可以繞過外層 TLS / REALITY 再加密，直接轉送已由內層 TLS 保護的記錄。此階段不能再畫成每個 byte 都有兩層加密；機密性依靠內層 TLS。非 TLS 或不符合條件的資料不能任意套用該結論。這是對此次原始碼與設計的分析，未在本課部署量測。

文件列 `xtls-rprx-vision` 可攔截 UDP/443 以促使應用改走 TCP HTTPS，另有 `-udp443` 變體。分流與 QUIC 可用性需核對實際 flow 及應用 fallback，不能只看入口支援 UDP。

19.4 Trojan

Trojan 在 TLS 通道內傳代理認證與目的請求，服務端再轉送 TCP / UDP；TLS 提供資料保護，Trojan 請求格式提供代理語義。原 Trojan 的 UDP 在 TLS/TCP 串流承載，遺失時可受外層串流 HOL 影響，過期 UDP 仍可能晚到。名稱相同的原始 Trojan 與其他核心 transport 擴充可能不同，需查具體實作。密碼正確不能代替伺服器憑證 / 名稱驗證；fallback 網站也不是對所有主動探測的普遍防護。

19.5 完整分類例題

候選 A：VLESS 基礎 none + RAW + TLS + Vision，應用經 SOCKS 入口。分類為 protocol=VLESS，協定內 encryption=none，承載=RAW/TCP，security=TLS，flow=Vision，接管=SOCKS。需要再查 TLS 名稱 / 信任、Vision 平臺及 payload 條件。

候選 B：VLESS + WebSocket + REALITY。各名字都存在，但此次 Xray transport 相容表不支援此組合，不能當有效設定。候選 C：TUN + WireGuard。TUN 只是 IP 介面，真正的 IP 隧道保護來自 WireGuard；它不屬候選 A 的同層替代欄位。

例題是分類模型，不是可直接執行的 JSON 模板。若讀者安裝其他核心，先用該版官方 schema 重建，再於隔離環境驗證；不要把教學概念表貼進生產服務。

延伸：AnyTLS 與 NaïveProxy

AnyTLS 參考專案處理 TLS 通道中的代理資料、填充及連線重用；NaïveProxy 使用 Chromium 網路堆疊及 HTTP/2 / 3 CONNECT 等機制。前者的 padding 與 session 策略、後者的瀏覽器堆疊及 HTTP 承載都屬實作 / 協定設計，不能排在 TUN / 系統 proxy 的「接管方式」欄。

兩者都需分析 TLS 驗證、出口目的、重用造成的共享狀態和資源。參考程式的預設 / 示例不一定是理想部署設定。瀏覽器外觀相近也不能由此推出匿名或無法封鎖；效能結論需同條件測量。

實驗、習題與檢核

L14 逐欄分類，L10 計算多層開銷預算。

1. VLESS、TLS、Vision、TUN 分別屬什麼角色？
2. 新 VLESS Encryption 為何需和 encryption:none 分開教？
3. 若某 transport 能用 REALITY，是否一定能用所有 Vision 設定？
4. 完整 TLS 驗證與 padding 各解決什麼問題？

檢核：任何效能或安全主張都能指出版本、組合、端點與條件。來源：VLESS 設定、Xray transport 相容表、Trojan 協定、AnyTLS、NaïveProxy。另見 Vision 設計、VisionReader/Writer 與 UnwrapRawConn、REALITY 設定。文件查閱 2026-10-03，第三方核心未在本課實測。

單元 20 Hysteria 2 與 TUIC : QUIC 代理的取捨

學習目標與問題

先讀 QUIC 與代理分層。你應能追蹤應用 TCP / UDP 如何經 QUIC 代理，分析丟包下的語義，設計控制公平性的效能比較。不要把「基於 UDP」當成跳過可靠性與壅塞控制的捷徑。

20.1 Hysteria 2 的路徑

Hysteria 2 以 QUIC 建立安全通道，認證後把 TCP 轉送放到雙向 stream，UDP 轉送放到 QUIC 不可靠 DATAGRAM。名稱含 2 與歷史 Hysteria 1 不能當同版設定；官方文件把 2.0.0 起的規範另行定義。

C 的 TCP 在本地代理入口終止，P 向目標再開 TCP。兩者間以 QUIC stream 保序 / 可靠交付；應用 UDP 則有 session、目的位址及可能的 fragment 封裝。DATAGRAM 是 ack-eliciting 且受壅塞控制，但不重傳該資料報。如果 DATAGRAM 的分片之一遺失，重組資料報可能整份丟棄，QUIC 不會替該 DATAGRAM 重傳。必須看伺服器是否允許 UDP 及最大資料報條件。

認證與一般 HTTP/3 外觀是設計的一部分，但不代表所有代理資料都是普通 HTTP 請求，也不保證任何觀察者永遠無法區分。通道 TLS 的驗證仍重要；設定忽略憑證不是正常安全前提。

20.2 TUIC v5 的路徑

此處版本指規範 VER=0x05，不是任一軟體的發行版 5。TUIC 使用認證、Connect、Packet 等命令；常見 QUIC 對映以雙向 stream 承載 TCP，UDP 可經 native DATAGRAM 或可靠的單向 stream 模式。可靠承載可以恢復代理段的遺失，卻也可能令過期 UDP 內容晚到；這是不同語義，不是免費效能提升。

TUIC 的認證 token 綁定 TLS 會話材料，與只傳一個固定明文密碼不同。協定層的免等待 / 並行處理和 TLS 早期資料 0-RTT 又是不同問題，需分開記錄是否首次、恢復、驗證完成及重放風險。官方協定 repo 與第三方核心實作分開，不能僅以 repo 名稱判斷所有實作的維護狀態。

20.3 共用 QUIC 的收益與代價

多個獨立 stream 可以減少 TCP 串流缺口對所有代理工作造成的交付 HOL，但仍共享 cwnd、連線級 flow control、CPU 和路徑。單一 QUIC 連線斷線可影響多個轉送；並行工作過多可令互動延遲變差。datagram 模式和 stream 模式還要分開量測遺失、重排、延遲及有效內容。

UDP 在某些網路被限制，QUIC 可能遇中間設備、MTU 或 CPU 問題；TCP 模式可能更易通過，但不一定較快。埠切換 (port hopping)、混淆和填充是額外設定，不改變所有傳輸服務的本質，也不能作為安全性的唯一證據。

20.4 完整比較例題

目標是低尾端延遲的互動訊息，每 50 ms 一份，100 ms 後過期。人工分布：模式 A 用 DATAGRAM，有 95% 訊息最終到達，其中 99% 在 100 ms 內；模式 B 用可靠 stream，全數最終到達，但只有 88% 在 100 ms 內。A 的 deadline 成功率 = $0.95 \times 0.99 = 94.05\%$ ，B = 88%。這是人工情境，若過期訊息不能使用，B 的「零最終遺失」不等於更高有效成功率。

相反，下載檔案需要全部 bytes，可靠 stream 適合其需求。比較時分別計算「deadline 內完整交付率」「有效 bytes/s」「尾端延遲」，不把不同需求壓成一個最快排名。

20.5 壅塞與公平性

Hysteria 的速率設定與協商可影響發送策略。填高 tx/rx 不創造額外實體頻寬；過高可能造成排隊、遺失與競爭流受損。此次官方 client 文件列按方向用 bandwidth 選擇 Brutal；未選 Brutal 時用 congestion 選定控制器，預設 BBR，也可 Reno。Brutal 的 loss compensation 可增加發送量，設定與伺服器忽略客戶頻寬的策略會改變行為。固定目標速率不同遺失時退讓的公平模型；應專門做競爭流實驗。設定值、控制器、補償、限速及雙向負載要明列。比較單協定獨佔和多協定競爭，可能得到不同結論。

實驗至少控制：client/server 硬體和版本、UDP 可用、RTT、遺失、path MTU、總瓶頸、工作負載、並行度、首次/恢復、warm-up；記 CPU、有效吞吐、尾端延遲、額外 bytes。L11 人工數列只練統計，不提供真實 HY2 與 TUIC 排名。

實驗、習題與檢核

L14 畫 stream 與 datagram 轉送；L11 設計 deadline 比較；Linux 延伸可用隔離多網段做受控遺失，本機必修不注入真實網路故障。

1. QUIC stream 可靠，為何 HY2 的 UDP 仍可能遺失？
2. TUIC 以 stream 承載 UDP 如何改變時效性？
3. 免等待的代理命令是否等於 TLS 0-RTT？
4. 提高設定速率為何可能損害互動延遲？

檢核：每個「快」都能寫出工作負載、指標與條件。來源：[Hysteria 2 規範](#)、[TUIC v5 規範](#)、[RFC 9000](#)、[RFC 9221](#)。[Hysteria client 速率與控制器設定](#)。查閱 2026-10-03。

單元 21 WireGuard、OpenVPN 與 IPsec

學習目標與問題

先讀 IP、路由、密碼學、NAT 和代理。你應能畫出 IP 層隧道、區分 OS 路由與 WireGuard AllowedIPs，預算 MTU，並比較 OpenVPN 及 IKEv2/IPsec。VPN 轉送 IP 不代表使用者所有流量都已進入 VPN。

LAN 單播直連與本鏈路服務探索分開驗證，參照 03A 的路由例外與 04A / L16 的探索路徑。

21.1 WireGuard 的內外兩層

WireGuard 把完整 IP 封包放進經認證加密的 UDP 傳輸。內層來源 / 目的是虛擬或被路由網段的 IP；外層 UDP/IP 負責到 peer endpoint 的 underlay 路徑。普通 TCP 應用仍在原始應用端點間保持 TCP，與逐段終止 TCP 的代理不同。

peer 以公開金鑰識別，握手使用 Noise 設計及臨時金鑰材料，建立並輪換資料金鑰；協定處理重放序號與視窗。私鑰、peer 公鑰和設定可信性仍需管理。通道保護在 peer 間終止，peer 若把 IP 轉送到明文服務，後段不因 VPN 自動加密。

21.2 AllowedIPs 不是整份 OS 路由表

傳出時，WireGuard 按內層目的前綴選 peer；傳入解密後，檢查來源 IP 是否被該 peer 允許。OS 先決定是否把封包送到 WireGuard 介面，兩者不同查表。wg-quick 等整合工具可能依 AllowedIPs 建立 OS 路由；協定自身不替所有平臺自動設定全部流量。

例：peer P 的 AllowedIPs=10.10.0.0/24，而 OS 只有 10.10.0.0/24→wg0。目的 10.10.0.8 進 P，普通網站走原出口。若要全隧道，IPv4/IPv6 路由與 peer 前綴都需覆蓋；同時保留 peer 外層 endpoint 從實體出口可達，避免遞迴入自己的隧道。DNS 也需另行指定及驗證。

共享一個 peer 的多個前綴通常表示路由到該 peer，不代表每個 IP 都是獨立驗證的使用者身分。重疊前綴、轉送服務與多 peer 政策需仔細設計，不能把 AllowedIPs 當任意 ACL 語言。

21.3 MTU 與 NAT

內層封包加 outer IP、UDP、WireGuard 資料 header 與 tag 後要符合路徑限制，還可能有 padding。L10 以 1500-byte outer MTU 計算：外層 IPv4 預算 $1500 - 20 - 8 - 16 - 16 = 1440$ ；外層 IPv6 預算 $1500 - 40 - 8 - 16 - 16 = 1420$ ，這裡的 IP 版本指外層，這是 budget 而非所有平臺的最佳設定。PPPoE、額外隧道或更小 PMTU 會降低可用大小。若承載 IPv6，需滿足 IPv6 鏈路最小 MTU 1280 等要求；更小 underlay 應由適用封裝 / 分片機制處理，不能把內層介面任意降到不足。MSS 調整只影響 TCP，不解決所有 UDP / IPv6 問題。

NAT 後 peer 可因已驗證封包更新 endpoint；必要時 keepalive 能維持某些 NAT 狀態，但增加流量、耗電。UDP 封鎖不由 keepalive 自動解除。僅看到握手成功也不證明內層路由、轉送、防火牆和 DNS 皆正確。

21.4 OpenVPN 與 TCP-over-TCP

OpenVPN 可使用 UDP 或 TCP，控制通道通常用 TLS 管理認證與金鑰，資料通道另有資料加密格式。TUN 承載 IP，TAP 可承載 Ethernet；實作與模式支援需查平臺。TAP / 橋接會擴展廣播與二層問題，不能只憑「能傳 MAC」判定更適合。

當內層 TCP 在外層可靠 TCP 隧道中，兩層重傳、隊首阻塞與壅塞行為可能耦合，遺失下產生額外延遲，常稱 TCP-over-TCP 問題。這並不適用於所有雙層 TLS 或所有 TCP 代理：終止內層 TCP 後只轉送 bytes 的代理，沒有把原始 TCP 段保留在外層 TCP 中，不屬於同一模型。UDP outer 避免該種雙重 TCP，但不保證任何路徑總是更快。

21.5 IKEv2 與 IPsec

IPsec 提供 IP 層保護與 policy/SA 等架構；ESP 可保護內容，tunnel 模式常封裝內層 IP，transport 模式保護上層 payload。IKEv2 負責認證、協商及建立金鑰/SA，不是直接替代 ESP 資料封包的另一個名稱。NAT traversal 常涉及 UDP 封裝；演算法套件、憑證 / PSK 與平臺版本都要核對。

項目	TCP 代理	IP VPN
轉送單位	bytes / 目的請求，UDP 另行	IP 封包
應用 TCP 端點	常被分成兩段或多段	常保留原端點的 TCP
非 TCP/UDP	協定 / 實作特定	可承載 IP 協定，仍受 policy 限制
名稱分流	需目的域名或可信關聯	通常 IP 路由，需要另加 DNS / 策略整合
全流量	入口及 app 支援決定	OS 路由及雙協定 policy 決定

實驗、習題與檢核

L10 預算 MTU，L14 分開 OS route、peer 選擇、出口 NAT 與 DNS。必修為離線推演；Linux 部署延伸需在隔離環境固定軟體版。

1. AllowedIPs 含 0.0.0.0/0 是否保證所有 IPv4 已送進 wg 介面？
2. 加密資料從 peer 出口到目標後，還有哪些信任邊界？
3. 為何 IPv4 全隧道不涵蓋 IPv6？
4. TCP-over-TCP 為何不同於一般 TLS 套 TLS 的 bytes 代理？

檢核：能畫兩層 IP、兩次查表與每一段加密終點。來源：[WireGuard 協定](#)、[WireGuard 概念](#)、[WireGuard 白皮書](#)、[OpenVPN 協定](#)、[RFC 4301](#)、[RFC 4303](#)、[RFC 7296](#)。

單元 22 流量接管、分流、DNS 與軟路由整合

學習目標與問題

整合 17 至 21。你應能解釋系統代理、TUN/TAP 和透明代理的差異，畫出 DNS / IPv4 / IPv6 / UDP 路徑，並找出策略缺口。協定只解決通道的一部分，「代理已連線」不能替完整路徑作證。

來源政策與代理分流整合見 03A / L15；LAN 探索、跨 VLAN 與印表機資料路徑見 04A / L16。代理重新建立上游 socket 後，來源不一定仍是客戶端來源。

22.1 應用怎樣進入通道

系統代理是 OS / 應用約定，配合 HTTP / SOCKS 端點；不遵守的應用可以直接建立其他 socket。PAC 可依請求選擇，但支援與規則依實作。TUN 讓程式收發三層 IP 封包，TAP 收發二層訊框；介面本身不提供加密、遠端 peer 或可靠傳輸。使用者空間 stack 或路由程式還需處理封包和通道。

透明代理以 redirect / TPROXY / policy routing 等機制讓未顯式設定的流量進代理；必須保留 / 取得原始目的資訊，處理回程、loop 及 UDP 狀態。具體實作和系統權限差異很大，不能把 Linux 命令套進 macOS 或 Windows。軟路由是一種設備 / 部署角色，可組合路由、防火牆、DHCP、DNS、代理/VPN，不是新的加密協定。

22.2 分流先定義意圖

目標可以是「企業私網經 VPN」「某類域名經代理」「所有應用資料經指定出口」。規則要寫輸入、優先序、預設值及不可獲得域名時怎麼處理。域名規則常需要顯式代理請求、DNS 關聯、Fake IP 或 sniffing 提供資訊；SNI 加密、快取、直接 IP 和連線重用會限制觀察。

Fake IP 用保留的合成位址與對映表把名稱關聯到流量，代理再還原真實目的。它不是公開 DNS 位址，也不是加密機制；位址池衝突、對映壽命、繞過 DNS 的應用和快取可能影響正確性。實作選擇的位址池和排除規則須查具體版本。

22.3 DNS 是獨立路徑

應用可能用系統 DNS，也可能自己做 DoH/DoT 或直接 IP。DoT 把 DNS 放 TLS，DoH 把 DNS 放 HTTPS，DoQ 用 QUIC；加密 DNS 不自動通過 VPN/代理，也不讓 resolver 看不見查詢。代理伺服器的名稱需要 bootstrap，若所有解析都依賴尚未建立的代理，可能產生依賴循環。

先畫：應用→本地 stub→遞迴 resolver→權威；再標每個請求的出口。比較實際與政策，才能判斷是否有不預期的 DNS 路徑。DNSSEC 驗證與通道加密另行標示，不將「防洩漏」等同解析資料真實。

22.4 完整整合例題

政策：本機網站流量都走 P，LAN 印表機直連，P 的外層 endpoint 走實體出口。實際規則只有 IPv4 TCP 被接管；瀏覽器使用 AAAA 與 QUIC 時，IPv6 UDP 可能走原出口。修正要看接管覆蓋、IPv6 路由、UDP 轉送及 fallback 政策；僅把 DNS 改成 P 提供的 resolver 無法自動修補資料路由。

流量	預期	必須核對
目標 A/TCP	P	入口、目的規則、出口
目標 AAAA/TCP	P	IPv6 接管與 P 的 IPv6 出口能力
目標 UDP/443	P 或明確拒絕並 fallback	UDP 入口、relay 能力、應用行為
DNS / DoH	指定 resolver 路徑	應用繞過、bootstrap 與出站路由
LAN 印表機	本地	直連例外優先、對應網段
P endpoint	underlay	免進入自身隧道、防 loop

本題位址 / 行為是模型。實際驗證需本機入口、實體出口和 P 端日誌等證據；不能僅靠網頁顯示一個公網 IP 證明所有協定全部覆蓋。

軟路由還要檢查 RA 來源、預設路由及 SLAAC。若 LAN 由另一 IPv6 路由器取得預設路由，僅攔截 IPv4 的設備可能被繞過；需在自主管理的拓樸控制 RA 與雙協定接管，而非任意停用所有 ICMPv6。Fake IP 對映重啟後遺失、客戶端仍有舊合成位址快取，是另一可檢查故障。

22.5 鏈式代理與信任

C→P1→P2→S 的每段通道、解析責任和認證都需獨立畫。P1 可能知道 P2，P2 知道 S；端到端 TLS 仍可保護內容，但時間相關、多方觀察、帳號關聯和日誌仍可暴露身分。增加跳數不能保證匿名，且增加故障點、延遲與管理負擔。

出口失敗應區分：規則未匹配、proxy 域名解析失敗、握手認證失敗、目標解析失敗、目標連線失敗、MTU 和資料階段。每層日誌用相同時間與連線 ID 關聯，注意時鐘誤差。不要同時修改所有規則來「試試看」。

實驗、習題與檢核

L09 比較域名/IP 請求，L10 檢查 endpoint 例外，L14 填寫完整策略與證據表；L12 故障案例可離線完成。

1. TUN 是否保證加密與全流量覆蓋？
2. 系統 proxy 為何可能漏掉應用 UDP？
3. 為什麼加密 DNS 不自動表示經 VPN 傳送？
4. 單次公網 IP 測試不能驗證哪些流量？

檢核：能分別檢查接管、規則、DNS、通道及出口，而且同時考慮 IPv4/IPv6 與 TCP/UDP。來源：[Linux TUN/TAP 文件](#)、[Linux TPROXY 文件](#)、[RFC 7858 \(DoT \)](#)、[RFC 8484 \(DoH \)](#)、[RFC 9250 \(DoQ \)](#)、[Xray FakeDNS 設定](#)。

單元 23 專題：設計、故障與證據

學習目標與任務

整合前 22 單元。專題不是把設定截圖排在一起，而是回答一個可驗證的問題，提出至少兩個設計，說明選擇、量測及限制。最低路線可完全在 loopback 與離線模型完成；真實 VPN / 多網段為延伸。

23.1 三個可選題目

A「代理的信任與端點」：用 L09 證明兩段 TCP，L13 證明名稱驗證；設計 HTTP CONNECT、SOCKS 及 IP VPN 的路徑對照，分清實測和推演。

B「視窗、時效與量測」：用 L02 / L06 / L11 分析視窗、重傳及 deadline；回答高 goodput 是否能取代低尾端延遲。不得把人工數列當作真實協定排名。

C「雙協定接管與排錯」：用 L03 / L08 / L12 / L14 畫 IPv4/IPv6、DNS、TCP/UDP 路徑，設計一個策略缺口及修復；提交哪些結論仍需要外部擷取確認。

延伸題可在隔離 Linux 環境實作多網段、WireGuard 或指定代理；需保存軟體版本、設定、拓樸、測量和復原方式。不能用第三方共享服務的不明變化作唯一對照。

23.2 分階段提交

第 17 週：選題，寫一句可反駁的假設與成功指標。第 19 週：交拓樸、端點、分層及威脅 / 信任表。第 21 週：交控制條件、負載、重複次數、計時定義及故障計畫。第 23 週：交初稿與原始資料。第 24 週：交修訂和口頭 / 書面答辯。

資料分類：measurements 是真實程式或儀器輸出；models 是演算法 / 假設計算；fixtures 是人工情境；interpretation 是你的推論。只要來源與限制清楚，離線路線也可完成專題。

23.3 完整示例：C↔P↔S

問題：「經代理後，源站觀察到的是哪個來源端點？」假設源站看到 P 的出口 socket，而不是 C 到 P 的來源埠。使用 L09 兩端日誌比對：C 的 local endpoint 等於 P 的 client peer；S 的 origin peer 等於 P 的 proxy_origin_source。若兩組各自相符，支持逐段 TCP 模型。

受控變因為 CONNECT / SOCKS 域名 / SOCKS IP；源站和工作負載固定。domain 模式只用本地別名表，不作真實 DNS 測量。接著提出 TLS 終止在哪裡的設計圖；L09 沒加 TLS，因此不能用它實測「P 讀不到 HTTPS」。L13 另驗 TLS 信任，二者不可拼成已部署的加密代理證據。

23.4 報告格式與規準

節	必須包含
問題與假設	預期現象、反例、成功條件
設計	拓樸、端點、路由、DNS、信任與版本
方法	負載、控制變因、計時、重複、資料來源
結果	原始資料、摘要、差異、錯誤與缺漏
分析	哪些假設獲支持，哪些未排除
取捨	安全、時效、維護、資源、可用性

專題 100 分：問題與設計 20、可重現方法 20、證據 25、推理 25、表達與限制 10。沒有對照、偽裝資料來源或宣稱未完成實測時，相關證據分不得給；不以使用「最熱門」協定加分。

習題與章末檢核

1. 選題並寫一個可用觀察推翻的假設。
2. 為該題提出兩個不同設計及各自失敗條件。
3. 指出目前兩份資料能證明、不能證明的結論。
4. 寫一項控制變因及一項仍無法控制的外部因素。

完成 L12 專題報告模板。題目與示例由本課程編寫，參考來源回連所用單元，不新增未經實測的協定能力。

單元 24 綜合評量與設計答辯

學習目標

以完整課程為基礎，從需求選擇模型、通道與接管方式，追蹤故障並提出可檢驗的修正。期末題組 100 分，課程總成績佔 15%；專題另佔 25%。開書答題，規範可查閱，來源與推論仍需分清。

24.1 期末題組

1. 路由與政策 (15) : OS 將 10.10.0.0/24 送 wg0 , peer AllowedIPs 為 10.10.0.0/24。普通 IPv4 網站與 2001:db8::80 各走哪裡？若需要雙協定全隧道，還要確認什麼？
2. 分層分類 (15) : 分類 VLESS、TLS、REALITY、Vision、WebSocket、TUN、WireGuard；找出不能由「標籤都存在」保證相容的組合。
3. 密碼與信任 (15) : 比較 PSK 代理與經臨時金鑰協商的 TLS / WireGuard；指出 PFS、防重放和代理可見目的的差異。
4. QUIC 語義 (15) : TCP 代理 stream 與 UDP DATAGRAM 有何不同？對 100 ms 過期訊息，可靠傳送為何可能降低 deadline 內成功率？
5. 開銷計算 (10) : outer path MTU 1492，外層 IPv6 40、UDP 8、資料 header 16、tag 16，求未額外計 padding 的 inner IP 預算；說明為何它不是普遍最佳 MTU。
6. 排錯 (15) : 人工日誌為 DNS 有效、TCP 成功、TLS hostname mismatch。給出最低必要的修正與後續驗證，不能停用名稱檢查。
7. 接管證據 (15) : 瀏覽器 IP 檢查顯示 P 出口，應用 UDP 卻仍由原出口送出。說明 IP 檢查的證據邊界，並設計 DNS、IPv6、UDP 與 endpoint 例外檢查表。

第 1 題 (15 分) 可改為 LAN 題：LAN 客戶端經代理主機，說明客戶端來源政策與代理上游 socket 查表的差異 (8 分)，並列 LAN 與 endpoint 例外 (7 分)。第 7 題也可改為跨 VLAN 印表機：分開畫 mDNS 探索、SRV / A / AAAA 及實際服務連線，提出可區分故障的證據。總分與課程比例不變，解答見 LAN 補充解答。

數值及標準推理見解答冊。可以有多個合格設計，需以具體 policy、workload 和版本支援。

24.2 設計答辯

專題說明每段誰建立連線、誰驗證誰、誰解析、誰終止加密，以及故障時如何定位。回答：若 UDP 被限制？若 peer 不可達？若 DNS bootstrap 失敗？若路徑 MTU 降低？若長期祕密洩漏？若應用繞過系統 proxy？至少選兩項分析。

例如「低延遲互動 + 檔案下載」可使用不同工作負載指標，不必追求單一協定在所有條件佔優。現有通道的維護、版本相容與可靠回復也屬設計成本。測得高速度不能替代安全驗證；設定有加密也不能替代可用性量測。

答辯計入單元 23 專題規準的「推理」(25) 及「表達與限制」(10)，不額外加一份分數。

24.3 自學結業檢核

不用看解答，重畫一次「連網到取得受保護內容」：鏈路→設定→下一跳→解析→傳輸→身分驗證→應用。再加入代理或 IP VPN，重畫端點和內外封裝。你應能指出模型變化，而不是只增加一個方塊。

最後提交一頁「已確認 / 仍未確認 / 下一份證據」。若能區分各層、算出關鍵預算、保留正確信任驗證、並提出可重現診斷，已達本課核心成果。延伸可進入 socket 系統實作、路由政策、無線、現代壅塞控制或協定研究；先以官方文件與版本為基礎。

來源：題組由本課程編寫，具體規範沿用 10 至 22 各單元。

附錄 A 入門工具、單位與學習方法

二進位與單位

一個 bit 是 0 或 1；byte 為 8 bits。十六進位每個數字表示 4 bits， $0x10=16$ 。十進位 Mb/s 用 10 的次方，MiB 用 2 的次方。先把速度和大小轉為同一單位，再計時間。例：10 MiB 經 20 Mb/s 的理想內容時間為 $10 \times 1048576 \times 8 / 20000000 = 4.194304$ 秒，未計啟動與開銷。

& 是 bitwise AND；IP 前綴比較不是字串開頭比較。虛擬碼描述演算法，不一定可直接執行。表格中的 bytes、bits、ms、s 必須保留單位；推導過程不可把 ms 當 s。

本機程式啟動

先進入 course 根目錄，再用 `python3 labs/L01-local-http.py`；Windows 可用 `py -3`。查看版本用 `python3 --version`。必修最低 Python 3.10，程式僅標準函式庫；TLS 實驗另外需 OpenSSL，沒有時走離線判讀。檔案路徑不同於網路端點；127.0.0.1 是 loopback，不是家庭閘道。

程式自動挑埠，不須開路由器 port forwarding 或安裝公共伺服器。多終端模式以實際輸出埠替換示例。Ctrl-C 可停止前景實驗，完成後確認沒有持續服務。失敗時記錯誤類型、版本、指令與當前路徑，不貼任何帳密或私鑰。

常見環境問題

現象	檢查
找不到 python3	已安裝 Python？Windows 可用 <code>py -3</code> ；路徑是否正確
找不到實驗檔	是否在 course 根目錄，labs 目錄是否存在
PermissionError	是否 loopback socket 被系統或管理政策限制
ConnectionRefusedError	服務是否仍開啟、埠是否用實際值
TLS 名稱錯誤	預期 hostname 與 SAN 是否匹配，保留驗證
沒有 OpenSSL	使用 L13 離線路線，不修改系統信任庫

每週最低路線

先讀學習目標，畫一個路徑或時序；手算例題；預測後才執行實驗；做基本題；最後查看解答並寫出錯因。延伸需要額外工具時，先完成本機 / 離線必修。複習時優先追蹤責任與證據，不只背縮寫。

Socket 是什麼？

Socket 是程式與作業系統網路功能互動的介面。一個 TCP 服務以 bind 指定本地 IP / 埠，以 listen 等待，再用 accept 取得代表某條連線的 Socket；客戶端以 connect 連到服務。send / recv 操作位元組串流，應用需自己定義訊息邊界。UDP 以 sendto / recvfrom 操作資料報，不建立相同的 TCP 連線狀態。Socket 是程式介面，埠是傳輸層識別的一部分，兩者不是實體插孔。

Python 版本範圍

程式採 Python 3.10 以上的語法 / API；本版只實際驗證 3.12.14，建議新環境使用仍受維護的 3.12 或更新版本。3.10 / 3.11、3.13 以上及 Windows / Linux 未執行。TLS 除 Python 版本還依賴 SSL backend 與憑證產生工具。遇到不同結果，保留版本與錯誤，不關閉驗證；可先完成離線路線。

附錄 B 代理與 VPN 評估總表

能力與層次

此表是協定 / 官方設計資料的比較，不是此次部署的互通或效能測試。查閱 2026-10-03；各種實作擴充、舊版和分流設定另行核對。表中的 TCP / UDP 表示代理語義，不表示任意應用皆已被接管。

類型	轉送單位與入口	通道保護的基本來源	主要版本 / 相容性核對
HTTP CONNECT	TCP 隧道；HTTP proxy	普通明文 CONNECT 無；HTTPS proxy 另有 TLS	普通 CONNECT 不同 CONNECT-UDP
SOCKS5	TCP CONNECT；UDP ASSOCIATE 另行	基礎 SOCKS 無全面加密	ATYP、驗證方法、UDP relay 支援
SS AEAD 2017	目的請求 + TCP chunks / UDP 報	PSK 衍生 + AEAD	cipher、SIP004/007、plugin
SS AEAD 2022	新版 TCP / UDP 格式	高熵 PSK + AEAD	SIP022、key 長度、重放機制
VMess AEAD	代理請求、payload	協定認證 / 資料保護，可加外層	AEAD 認證、時鐘、核心、transport
VLESS 基礎 none	目的請求 / 代理資料	通常外層 TLS/REALITY；可信私鏈情境另列	protocol、transport、security、flow
新 VLESS Encryption	擴充的協定層保護	按 Xray 所列握手與保護機制	支援核心版、雙方欄位與外層組合
Trojan	TLS 內的代理請求	TLS，另有代理認證	原協定 / 核心擴充、TLS 驗證、UDP
Hysteria 2	TCP↔QUIC stream；UDP↔DATAGRAM	QUIC/TLS	Hysteria 2、UDP 能力、速率、padding
TUIC v5	TCP stream；UDP 可 stream 或 DATAGRAM	常見 QUIC/TLS 對映	VER0x05、UDP 模式、核心實作
AnyTLS	TLS 通道內代理 / 填充 / 重用	TLS 與具體協定機制	實作版、session、padding、驗證
NaïveProxy	Chromium 堆疊的 CONNECT 代理	HTTPS / HTTP/2 或 HTTP/3 承載保護	客戶/服務實作、HTTP 版本、padding
WireGuard	IP 封包；OS 路由送入介面	Noise 握手 + 資料 AEAD	peers、AllowedIPs、OS 路由、MTU
OpenVPN	TUN/IP 或 TAP/鏈路，視平臺	TLS 控制 + 獨立資料通道保護	TCP/UDP、平臺、cipher、模式

類型	轉送單位與入口	通道保護的基本來源	主要版本 / 相容性核對
IKEv2/IPsec	IKE 協商 + IPsec 資料 SA	選定 IKE/ESP 演算法與認證	policy、模式、NAT-T、憑證/PSK

安全與運作的共同問題

對每一列都填：誰知最後目的地？誰解析？哪個長期祕密？會話金鑰如何建立？有沒有 PFS、針對什麼洩漏？誰檢查重放？TLS / 加密終點在哪？IPv6 / UDP 是否覆蓋？未匹配流量怎麼處理？協定失敗如何顯示？

普通 HTTP/SOCKS 無加密不妨礙內層 HTTPS 仍端到端保護；有加密代理也不讓代理失去最後目的資訊。SS2022 的 PSK 設計沒有 PFS；TLS/WG 的 PFS 需按實際握手及 key 處理條件，不由協定名字一律推論。匿名性、可辨識性、安全和可用性分別評估。

選擇工作表

需求	至少比較	所需證據
單一 HTTP/TCP 應用	CONNECT/SOCKS 與加密外層	app 支援、名稱、TLS 驗證、出口
不修改 app 的 IP 互連	WG、OpenVPN、IPsec 與路由	雙協定、peer、MTU、回程、防火牆
低 deadline 的 UDP	不可靠 relay vs 可靠承載	deadline 成功率、遺失、尾端延遲
大檔案	可行通道的相同工作負載	goodput、CPU、RTT、丟包、資源
域名分流	目的欄位 + DNS/接管整合	關聯來源、快取、ECH、繞過路徑

來源：17 至 22 的第一手資料及 references/source-notes.md。本表不給無條件最快、安全或不可封鎖排名。

附錄 C 隔離 Linux 多網段延伸

目的與狀態

此延伸將必修模型放到多介面與真實核心轉送上。本文是設計與驗證指引，尚未在 Linux、Windows 或 macOS 跨平臺執行。已本機驗證的範圍見 README 與驗證紀錄。所有必修可用 loopback / 離線完成。

建立隔離拓樸

在你控制的 Linux 虛擬機或獨立測試設備，以 network namespaces 或多臺 VM 建立 C—R—S；C 與 S 各在不同私有子網，R 各接一介面。先做靜態路由，確認雙向可達，才加 DHCP/DNS、NAT、防火牆或 VPN。固定 distribution、kernel、iproute2、packet-capture 及代理核心版本，保留設定快照。

不要以宿主或家用既有 default route 替代測試 R，不用真實個人流量當對照。選定隔離介面後才抓包或施加延遲 / 遺失。先寫取消故障和刪除測試環境的方法，完成後恢復快照。

四組延伸工作

一、逐跳 Ethernet/IP：同時擷取 C-R 與 R-S，對同一個請求記錄兩段 MAC、IP、TTL；先無 NAT 再有 NAT，比較來源 tuple。

二、狀態與過濾：只允許必要測試服務，分開記錄 NAT 對映及 filter 決策；改回程路由，觀察正向到達但回程失配。不得把單次 STUN 直接當完整 NAT 分類。

三、可靠性與效能：在指定測試鏈路施加受控 RTT、遺失與速率，一次改一個變因。先實測 baseline，再測各 TCP / QUIC 工作負載。Linux tc/netem 的參數及執行位置需依所用版文件核對；它不模擬所有真實無線條件。

四、VPN / 代理：用官方工具在測試 C 與 P 設定指定版 WG 或代理核心；先驗證 peer / 憑證，再驗 route、DNS、MTU、IPv6 與 UDP。保存 validator 輸出及抓包。代理不得開成公網任意目的 relay；只允許測試子網與服務。

提交與來源

提交拓樸、固定版、完整測試設定、原始擷取、計時定義、對照與清理紀錄。標示每個功能已執行或僅設計。核心文件：[network namespaces 概念](#)、[Linux TUN/TAP](#)、[Linux TPROXY](#)、[WireGuard 快速開始](#)。部署指令以該環境的官方手冊核對。

術語表

本教材用語	英文	常見其他中文譯法
交換器	switch	交換機
訊框	frame	幀、框架
封包	packet	資料包
資料報	datagram	資料包
多工	multiplexing	多路複用
埠	port	端口
壅塞控制	congestion control	擁塞控制
流量控制	flow control	流控
前向保密	forward secrecy	前向安全
虛擬碼	pseudocode	偽程式碼
多播	multicast	群播、組播
對映	mapping (如 NAT 對映、多播 MAC 對映)	映射

MAC address 中的 MAC 是媒介存取控制；密碼學 MAC 是訊息驗證碼，應依上下文展開。訊框、封包、TCP 段及位元組串流不互相替代。

證據與結論的關係用「支持」或「佐證」；功能或能力用「支援」，例如「DNS 支援 UDP 與 TCP」。

LAN 補充詞彙

詞彙	本教材用法	易混淆之處
RPDB / policy routing	按規則查不同路由表	不等於所有表共同作 LPM
preferred source / src	本機發送時的偏好來源	不是 SNAT
FDB	VLAN / MAC 到轉送埠	不是 IP 多播成員表
MDB	多播群組與成員埠資訊	實作鍵與來源篩選能力可不同
IGMP / MLD	IPv4 / IPv6 接收成員管理	不是跨網段路由協定
querier	發送成員查詢的角色	不等同 snooping 開關
router port	通往多播路由器的交換埠	不表示有本地應用接收者
mDNS / DNS-SD	本鏈路名稱解析 / 以 DNS 記錄探索服務	DNS-SD 也可用單播 DNS

未知單播泛洪仍保留單播目的 MAC。

來源與版本註記

查閱基準：2026-10-03。下列是作者查閱的第一手資料索引；正文獨立撰寫，並在相應章節列出具體規範。版本相關欄位將與實驗指定版本一起維護。

網路與標準

- TCP：RFC 9293 (協定本體)；壅塞控制另查 RFC 5681；重傳計時器：RFC 6298。
- IPv4：RFC 791；CIDR：RFC 4632；IPv6：RFC 8200。
- ARP：RFC 826；DHCP：RFC 2131；DNS：RFC 1034、1035。
- IPv6 NDP：RFC 4861；SLAAC：RFC 4862；DHCPv6：RFC 9915 (取代 8415)。
- TLS 1.3：RFC 9846；HTTP 語意：RFC 9110；HTTP/1.1：RFC 9112；HTTP/2：RFC 9113；HTTP/3：RFC 9114。
- QUIC 傳輸：RFC 9000；遺失偵測與壅塞控制：RFC 9002；DATAGRAM：RFC 9221；WebSocket：RFC 6455。
- UDP NAT 行為：RFC 4787；STUN：RFC 8489；TURN：RFC 8656；ICE：RFC 8445。
- SOCKS5：RFC 1928；CONNECT-UDP：RFC 9298；DoT：RFC 7858；DoH：RFC 8484。

RFC 原文：<https://www.rfc-editor.org/rfc/rfc<編號>.html>。RFC 的歷史內容、標準狀態、更新與被取代關係需要同時核對；不能把所有 RFC 當作現行規範。

協定官方資料

協定 / 元件	第一手來源	需要保留的區分
Shadowsocks	https://shadowsocks.org/doc/aead.html ; https://shadowsocks.org/doc/sip022.html	AEAD 2017 與 2022 是不同版次；2022 的預共享金鑰模式本身不提供前向保密
VMess	https://www.v2fly.org/en_US/v5/config/proxy/vmess.html	VMess 與 V2Ray 軟體分開；驗證含時間相關條件
VLESS	https://xtls.github.io/en/config/outbounds/vless.html	傳統 encryption:none 與新 VLESS Encryption；外層 TLS / REALITY；Vision 是另一個設定維度
Xray 傳輸	https://xtls.github.io/en/config/transport.html	protocol、傳輸方式與 transport security 分開；可用組合依實作版本
Trojan	https://trojan-gfw.github.io/trojan/protocol	TLS 承載與代理請求 / 驗證分工
Hysteria 2	https://v2.hysteria.network/docs/developers/Protocol/	QUIC stream 轉送 TCP；不可靠 DATAGRAM 承載 UDP；伺服器能力與設定會影響可用性
TUIC	https://github.com/tuic-protocol/tuic ; SPEC.md	此次查閱列出的協定版為 0x05；不同協定版本不保證相容
AnyTLS	https://github.com/anytls/anytls-go	填充、分包與連線重用；參考實作的示例設定不能當作一般安全部署設定
NaïveProxy	https://github.com/klzgrad/naiveproxy	Chromium 網路堆疊、HTTP/2 / 3 CONNECT、填充；專案及其承載機制分開描述
WireGuard	https://www.wireguard.com/ ; https://www.wireguard.com/protocol/	IP 層加密隧道、金鑰與 peer、UDP 外層；不是 HTTP 專用代理
OpenVPN	https://openvpn.net/community-docs/openvpn-protocol.html	控制與資料通道、TCP / UDP、不同版本的設定差異
IKEv2 / IPsec	RFC 7296、RFC 4301、RFC 4303	IKEv2 協商與 IPsec 資料保護的分工

已核對的關鍵版本資訊

Xray 目前的 VLESS 設定文件列出可選 VLESS Encryption。教材不能將「VLESS 永遠不提供協定內加密」寫成無條件事實。傳統 encryption: none 的基礎模型與新擴充分開解釋，並將設定能力標示為 Xray 的具體實作資訊。

Shadowsocks 2022 官方規範指出其預共享金鑰設計不提供前向保密。這是對該版次的具體敘述；不能由「使用 AEAD」推導出有金鑰交換或前向保密。

QUIC 的 STREAM 提供可靠且有序的資料交付；DATAGRAM 擴充提供不可靠資料報。跨串流的資料交付隊首阻塞、串流內等待、共用壅塞控制及連線流量控制是不同現象。

課程設計與實務主題索引

- Stanford CS144 : <https://bulletin.stanford.edu/courses/2075241> 。參考課程對封包交換、分層、路由、可靠傳輸、流量控制、壅塞控制及程式實作的涵蓋。
- 不良林的作者附文 : <https://www.bulianglin.com/archives/guide.html> 、
<https://bulianglin.com/archives/proxymode.html> 、<https://bulianglin.com/archives/p2p.html> 。用於確認實務主題範圍，正文不以其影片順序或文字為教材模板。
- Wireshark 官方使用手冊 : https://www.wireshark.org/docs/wsug_html_chunked/ 。封包擷取與檢視操作以適用平臺版本核對。

2026 版狀態與實作界線

- TLS 1.3 : RFC 9846 取代 8446，版本仍 1.3；ECH 為 RFC 9849。
 - DHCPv6 : RFC 9915 (2026 年 1 月) 取代 8415。
 - Happy Eyeballs v3 : 此次查閱為 draft-ietf-happy-happyeyeballs-v3-04，仍是工作草案；正文以已發布 RFC 8305 為基礎。
 - Xray 欄位與相容表按本次所讀官方文件。新 method/raw 和舊 network/tcp 不能任意混拼；沒有在本課固定 / 部署某發行版，故不提供可宣稱已驗證的完整設定 JSON。
 - Vision 的 direct-copy 原始碼切換到 raw conn，不能把該階段當全部 payload 仍有外層再加密；參考官方設計討論 1295 與 proxy/proxy.go。
 - SS2022 有 timestamp 時間窗，不能把時鐘條件只歸給 VMess；早期 AEAD 格式沒有相同強制完整重放要求。
- 版本敏感的新 VLESS Encryption、REALITY/Vision、AnyTLS、NaïveProxy、HY2/TUIC 在本版採設計與文件分析。未部署第三方核心、不提供實測排名；實際設定需固定核心版、schema 並自行在隔離環境 validator/互通測試。

v1.1 LAN 補充：查閱 2026-10-06

Linux 介面位址、RPDB 與路由語義採 iproute2 原專案手冊的 man7 發布頁；ARP 及 rp_filter 採 Linux kernel 文件，bridge 觀察採 kernel 與 bridge 手冊。命令為 Linux 延伸，尚未在本機 macOS 實測；Python 離線模型不代表完整 kernel。來源連結集中於 03A / 04A 章末。命名空間參照 [ip-netns](#)，觀察命令參照 [bridge](#)。

IPv4 多播 MAC 對映採 RFC 1112，IPv6 對映採 RFC 2464；IPv6 位址類型採 RFC 4291，管理範圍採 RFC 2365，導向廣播預設政策採 RFC 2644。IGMPv3 RFC 9776、MLDv2 RFC 9777 皆於 2025 年發布並分別取代 3376 / 3810；是既有協定版本的修訂，部署能力須另查。Snooping 依 RFC 4541 的 Informational 建議與實際設備政策分開描述。

mDNS 依 RFC 6762，DNS-SD 依 RFC 6763；探索可見、主機位址可達、服務埠可用與端點身分驗證是四項不同證據。本版的探索資料表與拓樸是人工設計，不引用影片逐段解說或冒充抓包。