

電腦網路實驗手冊

預測、觀察與診斷

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

L01 觀察本機通訊	3
L02 延遲與視窗模型	5
L03 位址與最長前綴	6
L04 DNS 訊息格式	7
L05 本機 TCP 與 UDP 訊息	8
L06 ARQ 與重複交付	9
L07 鏈路狀態最短路徑	10
L08 NAT 對映與過濾	11
L09 本機 CONNECT、SOCKS5 與解析責任	12
L10 封裝開銷與路由	13
L11 效能統計與實驗設計	14
L12 故障診斷與專題工作紙	15
L13 本機 TLS 的信任與名稱驗證	16
L14 代理 / VPN 分層與路徑工作表	17
L15 單介面多 IP 與政策路由	19
L16 LAN 交付、多播成員與服務探索	21

L01 觀察本機通訊

目標與驗證範圍

辨認 TCP 的來源 / 目的端點，觀察 HTTP 方法、狀態、標頭與回應內容，區分本機 loopback 實驗與家庭網路的真實路徑。本實驗已提供可自行結束的示範，以及方便封包觀察的分開啟動方式。

環境：Python 3.10 以上，僅使用標準函式庫。程式與本文件都在 labs/，下列指令由課程根目錄執行；Windows 可用 `py -3` 替換 `python3`。服務繫結 127.0.0.1，以作業系統分配的空間埠啟動。

執行前預測

1. 客戶端與服務端的 IP 是否相同？它們的埠是否相同？
2. 本次路徑是否會經過家庭的 AP、閘道或 NAT？
3. 若收到 HTTP 200，最多可以證明哪些步驟成功？
4. 如果回應包含中文字，Content-Length 計算字元數還是 bytes？

步驟 A：一個指令完成示範

在課程目錄執行：

```
python3 labs/L01-local-http.py
```

程式會啟動本機 HTTP 服務，建立一個客戶端 TCP 連線，請求 `/lesson`，讀完回應，然後結束服務。輸出是一份 JSON 觀察紀錄。實際埠與時間每次可不同。

紀錄欄位包括 `method`、`request_path`、`response_headers`，以及 `client_endpoint`、`server_endpoint`、`status`、`content_length`、`received_bytes`、`elapsed_ms` 及 `response`。檢查服務端看到的客戶端埠，是否和客戶端紀錄一致。

步驟 B：分開啟動，方便觀察

第一個終端：

```
python3 labs/L01-local-http.py --serve
```

記下印出的埠，例如 54321。第二個終端用實際埠執行：

```
python3 labs/L01-local-http.py --client 54321
```

可以重複請求；觀察客戶端來源埠是否改變。完成後，在第一個終端按 `Ctrl-C` 結束服務。

若你已有 Wireshark，可在適當的 loopback 介面觀察，使用 `tcp.port == 54321` 作顯示篩選。介面名稱及擷取權限依平臺而異。這是延伸觀察，必修部分以程式紀錄完成；程式紀錄本身沒有擷取或證明完整的 TCP 握手封包。

紀錄表

項目	第一次	第二次	解釋
客戶端 IP / 埠	自行填寫	自行填寫	是否同一個連線？
服務端 IP / 埠	自行填寫	自行填寫	是否同一個服務？
HTTP 狀態	自行填寫	自行填寫	應用層結果
宣告 / 收到 bytes	自行填寫	自行填寫	說明 parser 依 Content-Length 讀取；不足會報 IncompleteRead，並非獨立線路完整性測試。
elapsed_ms	自行填寫	自行填寫	包含哪些工作？

受控制的故障

先正常請求 /lesson，再請求 /missing：

```
python3 labs/L01-local-http.py --client 54321 --path /missing
```

預期收到 HTTP 404。TCP 與 HTTP 交換仍可能成功，只是服務沒有該資源。結束服務後再用同一個埠請求；通常會遇到 connection refused，但若該埠被其他服務接手，結果可能不同。記下例外種類，並將「沒有連線」與「有連線但應用回覆失敗」分開。

判讀與交付

繳交兩次成功紀錄、一個 404 紀錄、服務關閉後的結果，以及約 200 字的解釋。說明本實驗為什麼沒有驗證外部 DNS、NAT、TLS 或遠端路由。不要把本機 elapsed_ms 解讀為單向網路延遲。

評分：預測 20%、紀錄完整 30%、故障區分 30%、證據範圍 20%。

L02 延遲與視窗模型

目的、環境與預測

對應單元：02、07、08。Python 3.10 以上，標準函式庫；在課程根目錄執行。資料性質：離線模型或人工資料分析。

模型參數：1500-byte 封包經三段各 100 Mb/s、各傳播 2 ms 的 store-and-forward 鏈路；路徑 100 Mb/s、RTT 40 ms；傳送端視窗 64 KiB (65,536 bytes)。先預測三段鏈路總延遲、BDP 與視窗上界。

執行與紀錄

```
python3 labs/models.py L02
```

Windows 可用 `py -3` 替換 `python3`。將輸出儲存，記錄日期、Python 版本、參數與預測。

條件	執行前預測	實際輸出	差異與解釋
原始設定	自行填寫	貼關鍵欄位	寫出依據
修改設定	自行填寫	貼關鍵欄位	一次改一個變因

完成預測與紀錄後，再到解答冊的「預期輸出」核對。

變更與判讀

使用 `python3 labs/models.py L02 --rtt 0.08 --window 65536` 重新計算；保留原始參數和修改值。

RTT 變大時 BDP 與視窗上界各如何改變？為何本模型不能證明一條真實 TCP 連線的速度？

驗證範圍與提交

輸出是明示假設下的模型或人工資料；不能當成真實封包擷取、OS 核心狀態或 Internet 量測。

提交預測、原始 / 修改輸出、兩個判讀答案與模型限制。評分：預測 20%、紀錄 30%、推理 35%、限制 15%。標準推理見解答冊；輸出中的埠與真實計時可以變動。

L03 位址與最長前綴

目的、環境與預測

對應單元：03、11。Python 3.10 以上，標準函式庫；在課程根目錄執行。資料性質：離線模型或人工資料分析。

手算 198.51.100.160/27，並用 03 章 3.2 的路由表預測 198.51.100.20 與 198.51.100.140 走哪個下一跳。

執行與紀錄

`python3 labs/models.py L03`

Windows 可用 `py -3` 替換 `python3`。將輸出儲存，記錄日期、Python 版本、參數與預測。

條件	執行前預測	實際輸出	差異與解釋
原始設定	自行填寫	貼關鍵欄位	寫出依據
修改設定	自行填寫	貼關鍵欄位	一次改一個變因

完成預測與紀錄後，再到解答冊的「預期輸出」核對。

變更與判讀

加入 198.51.100.128/26 的路由並測 .140 與 .200：`python3 labs/models.py L03 --route 198.51.100.128/26=192.0.2.67 --lookup 198.51.100.200`。另建 `::/0` 和 `2001:db8::/32`，查 `2001:db8::80` 與 `2001:db9::80`：在同一指令加上 `--route ::/0=fe80::1 --route 2001:db8::/32=fe80::2 --lookup 2001:db8::80 --lookup 2001:db9::80`。下一跳只是模型標籤。

更具體路由不可達時會自動改用短前綴嗎？需考慮哪些平臺與路由有效性條件？

驗證範圍與提交

輸出是明示假設下的模型或人工資料；不能當成真實封包擷取、OS 核心狀態或 Internet 量測。

提交預測、原始 / 修改輸出、兩個判讀答案與模型限制。評分：預測 20%、紀錄 30%、推理 35%、限制 15%。標準推理見解答冊；輸出中的埠與真實計時可以變動。

L04 DNS 訊息格式

目的、環境與預測

對應單元：05。Python 3.10 以上，標準函式庫；在課程根目錄執行。資料性質：離線模型或人工資料分析。

以下是程式產生的人工回應訊息（hex）；查詢的 ID 為 0x1234。兩行屬同一則訊息，換行只為排版：

```
1234818000010001000000000066c6573736f6e0474657374
0000010001c00c000100010000012c0004c0000250
```

先將 hex 以兩個字元一組拆開，再預測 QR、RD、RA、Question 與 Answer 數量。

執行與紀錄

```
python3 labs/models.py L04
```

Windows 可用 `py -3` 替換 `python3`。將輸出儲存，記錄日期、Python 版本、參數與預測。

條件	執行前預測	實際輸出	差異與解釋
原始設定	自行填寫	貼關鍵欄位	寫出依據
修改設定	自行填寫	貼關鍵欄位	一次改一個變因

完成預測與紀錄後，再到解答冊的「預期輸出」核對。

變更與判讀

在紙上定位 header 12 bytes、QNAME 與固定欄位。選做：python3 labs/models.py L04 --ttl 60，核對回應中 TTL 的 4 bytes 如何改變。

這份資料是否證明某臺 Internet DNS 提供此答案？NOERROR 空 Answer 與 NXDOMAIN 有何不同？

實際解析器不能只比對 ID，還需比對 Question、來源與查詢狀態，並處理 TC、RCODE 及回應完整性；此 fixture 不是完整 resolver。

驗證範圍與提交

輸出是明示假設下的模型或人工資料；不能當成真實封包擷取、OS 核心狀態或 Internet 量測。

提交預測、原始 / 修改輸出、兩個判讀答案與模型限制。評分：預測 20%、紀錄 30%、推理 35%、限制 15%。標準推理見解答冊；輸出中的埠與真實計時可以變動。

L05 本機 TCP 與 UDP 訊息

目的、環境與預測

對應單元：06、08。Python 3.10 以上，標準函式庫；在課程根目錄執行。資料性質：真實本機 Socket 通訊。

程式依序送出三份 TCP 訊息：cat、網路 (UTF-8) 與空訊息，每份前加 4-byte 大端長度；UDP 只送前兩份。先預測每份長度前綴的值、接收端的重組結果與兩份 UDP 資料報；中文用 UTF-8 算 bytes。

執行與紀錄

`python3 labs/L05-sockets.py`

Windows 可用 `py -3` 替換 `python3`。將輸出儲存，記錄日期、Python 版本、參數與預測。

條件	執行前預測	實際輸出	差異與解釋
原始設定	自行填寫	貼關鍵欄位	寫出依據
修改設定	自行填寫	貼關鍵欄位	一次改一個變因

完成預測與紀錄後，再到解答冊的「預期輸出」核對。

變更與判讀

複製 `labs/L05-sockets.py` 後修改 `demo()`：在 `messages` 加入一份 100-byte 訊息，並讓傳送端只送出該訊息的一部分就關閉寫入方向；`parser` 應報錯，而非交付半個訊息。

限制 `recv` 長度是否意味每個 TCP 封包只有 2 bytes？程式如何知道訊息結束？

驗證範圍與提交

本機 Socket 驗證應用介面，不模擬遠端路徑、網路遺失或真實代理效能。

提交預測、原始 / 修改輸出、兩個判讀答案與模型限制。評分：預測 20%、紀錄 30%、推理 35%、限制 15%。標準推理見解答冊；輸出中的埠與真實計時可以變動。

L06 ARQ 與重複交付

目的、環境與預測

對應單元：07、08。Python 3.10 以上，標準函式庫；在課程根目錄執行。資料性質：離線模型或人工資料分析。

本例採逐單位 ACK 的 Selective Repeat (SR)，不是 Go-Back-N；以離散回合、無限接收 buffer 及一回合 timer 簡化。

sender 依序傳送序號 0–4 共五個資料單位，起始序號為 0，視窗為三個單位。在序號 1 首傳遺失、ACK 0 首回遺失的條件下，先畫回合表。

執行與紀錄

```
python3 labs/models.py L06
```

Windows 可用 `py -3` 替換 `python3`。將輸出儲存，記錄日期、Python 版本、參數與預測。

條件	執行前預測	實際輸出	差異與解釋
原始設定	自行填寫	貼關鍵欄位	寫出依據
修改設定	自行填寫	貼關鍵欄位	一次改一個變因

完成預測與紀錄後，再到解答冊的「預期輸出」核對。

變更與判讀

以 `python3 labs/models.py L06 --no-ack-loss` (或 `--no-data-loss`) 去掉一項故障規則，比較重傳次數；以 `--window 1` 比較回合數，回合不是毫秒。

為何 ACK 遺失的 0 仍可在 receiver 交付？這個模型沒有包含哪幾項 TCP 功能？

驗證範圍與提交

輸出是明示假設下的模型或人工資料；不能當成真實封包擷取、OS 核心狀態或 Internet 量測。

提交預測、原始 / 修改輸出、兩個判讀答案與模型限制。評分：預測 20%、紀錄 30%、推理 35%、限制 15%。標準推理見解答冊；輸出中的埠與真實計時可以變動。

L07 鏈路狀態最短路徑

目的、環境與預測

對應單元：10。Python 3.10 以上，標準函式庫；在課程根目錄執行。資料性質：離線模型或人工資料分析。

使用 10 章的無向圖：A-B=2、A-C=5、B-C=1、B-D=4、C-D=1。從 A 手算 Dijkstra，先寫每輪已確定節點、暫時距離與前驅。

執行與紀錄

```
python3 labs/models.py L07
```

Windows 可用 `py -3` 替換 `python3`。將輸出儲存，記錄日期、Python 版本、參數與預測。

條件	執行前預測	實際輸出	差異與解釋
原始設定	自行填寫	貼關鍵欄位	寫出依據
修改設定	自行填寫	貼關鍵欄位	一次改一個變因

完成預測與紀錄後，再到解答冊的「預期輸出」核對。

變更與判讀

執行 `python3 labs/models.py L07 --bc-cost 10`，把 B-C 成本兩方向都改為 10。先預測新的最短路徑與成本，並檢查是否有等價路徑；輸出的 `paths` 只列程式 tie-breaking 選出的一條，`all_shortest_paths` 列出全部等價最短路徑。

最短路徑是否足以描述 BGP 跨組織政策？不對稱圖與無向圖差別為何？

驗證範圍與提交

輸出是明示假設下的模型或人工資料；不能當成真實封包擷取、OS 核心狀態或 Internet 量測。

提交預測、原始 / 修改輸出、兩個判讀答案與模型限制。評分：預測 20%、紀錄 30%、推理 35%、限制 15%。標準推理見解答冊；輸出中的埠與真實計時可以變動。

L08 NAT 對映與過濾

目的、環境與預測

對應單元：12。Python 3.10 以上，標準函式庫；在課程根目錄執行。資料性質：離線模型或人工資料分析。

內部 10.0.0.2:51000 經 EIM 固定對映為 203.0.113.5:40000，曾對外聯絡 198.51.100.10:9000 與 198.51.100.20:9000。在 ADF 下，預測入站探測 198.51.100.10:9000、198.51.100.10:9999 (同 IP 不同 port) 及 198.51.100.30:9000 (全新 IP) 的結果。

執行與紀錄

```
python3 labs/models.py L08
```

Windows 可用 `py -3` 替換 `python3`。將輸出儲存，記錄日期、Python 版本、參數與預測。

條件	執行前預測	實際輸出	差異與解釋
原始設定	自行填寫	貼關鍵欄位	寫出依據
修改設定	自行填寫	貼關鍵欄位	一次改一個變因

完成預測與紀錄後，再到解答冊的「預期輸出」核對。

變更與判讀

以 `python3 labs/models.py L08 --filter APDF` 改為要求 remote IP 與 port 都曾聯絡過；再以 `--elapsed 120 --lifetime 60` 模擬 state 已逾時，比較入站結果。

對映不變是否表示任何 Internet 主機都能入站？STUN 看見一個對映能否保證穿透？

驗證範圍與提交

輸出是明示假設下的模型或人工資料；不能當成真實封包擷取、OS 核心狀態或 Internet 量測。

提交預測、原始 / 修改輸出、兩個判讀答案與模型限制。評分：預測 20%、紀錄 30%、推理 35%、限制 15%。標準推理見解答冊；輸出中的埠與真實計時可以變動。

L09 本機 CONNECT、SOCKS5 與解析責任

目標與預測

對應 17、22。Python 3.10 以上，標準函式庫。使用真實本機 TCP 與 HTTP，僅轉送由程式建立的 127.0.0.1 服務。lesson.test 是代理內靜態別名，不進行外部 DNS。協定子集：HTTP/1.1 CONNECT、SOCKS5 無驗證 CONNECT 之 IPv4 / 域名；不含 BIND、UDP ASSOCIATE、IPv6、TLS 和一般目的地。

預測三種模式源站看到誰的來源埠？名稱模式是否由客戶端送出 IP？

執行與觀察

```
python3 labs/L09-proxy.py
```

依序取得 HTTP CONNECT 域名、SOCKS 域名、SOCKS IP 三次源站 HTTP 200。CONNECT 另有代理的 200 Connection Established；SOCKS 成功則是 REP=0x00，不是 HTTP 狀態碼。每筆結果記 client_source 與 origin_peer；proxy_events 記 proxy_client_peer、proxy_origin_source 與 resolution。

模式	客戶送出的目的	解析責任（本實驗）	要比對的端點
CONNECT 域名	lesson.test:port	P 查靜態別名表	C source 與 P client peer； S peer 與 P origin source
SOCKS 域名	ATYP3 + lesson.test	P 查靜態別名表	同上
SOCKS IP	ATYP1 + 127.0.0.1	已提供 IP	同上

實際埠每次變動。若兩組端點各自相符，支持兩段 TCP 的模型；這不是完整 TCP 握手擷取。程式將目的範圍鎖在本機單一服務；不是可部署的公開代理。

變更、問題與提交

手畫「本地 DNS 先解析後送 IP」與「傳域名至 P」兩圖。真實 DNS 解析不在此實驗內，不能聲稱程式量測了 DNS 速度或洩漏。選做新增一個應被拒絕的請求，記錄回覆並確認沒有新增上游事件；保留原目的限制。SOCKS 不支援命令回 REP=0x07，不支援 ATYP 回 0x08，目的政策拒絕回 0x02。

1. HTTP CONNECT 回 200 後，代理轉送的是哪些 bytes？
2. 為何 origin_peer 不是 client_source？
3. 加入內層 HTTPS 後，哪些證據仍需新增？
4. SOCKS CONNECT 成功是否證明 UDP ASSOCIATE 也支援？

提交三次輸出、兩圖與限制；評分預測 20、端點證據 30、解析責任 30、限制 20。標準答案見解答冊。

L10 封裝開銷與路由

目的、環境與預測

對應單元：19、21、22。Python 3.10 以上，標準函式庫；在課程根目錄執行。資料性質：離線模型或人工資料分析。

假設 outer MTU 1500，列出 outer IP、UDP、WG 資料標頭與 tag。

執行與紀錄

```
python3 labs/models.py L10
```

Windows 可用 `py -3` 替換 `python3`。將輸出儲存，記錄日期、Python 版本、參數與預測。

條件	執行前預測	實際輸出	差異與解釋
原始設定	自行填寫	貼關鍵欄位	寫出依據
修改設定	自行填寫	貼關鍵欄位	一次改一個變因

完成預測與紀錄後，再到解答冊的「預期輸出」核對。

變更與判讀

以 `python3 labs/models.py L10 --path-mtu 1492` 重算，再考慮實際 padding 和 inner IP/TCP；在紀錄中寫出端點免入隧道的路由例外。

為何不能把 1420 當成所有 VPN 的固定最佳 MTU？若 VPN 端點本身被匯入該 VPN 會發生什麼？

驗證範圍與提交

輸出是明示假設下的模型或人工資料；不能當成真實封包擷取、OS 核心狀態或 Internet 量測。

提交預測、原始 / 修改輸出、兩個判讀答案與模型限制。評分：預測 20%、紀錄 30%、推理 35%、限制 15%。標準推理見解答冊；輸出中的埠與真實計時可以變動。

B：普通 TLS/TCP 代理的多層預算

人工假設：outer IPv4 MTU 1500，IPv4 header 20、TCP header 20、TLS 1.3 header 5、內部 content type 1、AEAD tag 16；不計 TCP options 及 TLS padding。一份外層 TLS record 的明文包含 20 bytes 代理控制資料，以及一份 1000 bytes 的內層 TLS record（已含其 header 與 tag）。假設整份外層 record 放進一個 TCP 段，求 IP 封包大小與剩餘預算。

再問：若內層分成兩份 record，應如何重算？為何 TLS record 不能普遍當成一個 TCP 段？這是普通雙層模型，不適用 Vision 切換後的 direct-copy 階段。現實的分段、聚合、padding 及 offload 需要實作與擷取證據，不只相加定值。

L11 效能統計與實驗設計

目的、環境與預測

對應單元：15、16、18、20、23。Python 3.10 以上，標準函式庫；在課程根目錄執行。資料性質：離線模型或人工資料分析。

人工數列 (ms)：A = 10、11、9、12、10、11、10、12、9、50；B = 13、14、12、13、15、14、13、12、14、16。先算 A 與 B 的中位數；p95 使用 nearest-rank 定義，排序後取第 $\text{ceil}(0.95 \times n)$ 筆。

執行與紀錄

```
python3 labs/models.py L11
```

Windows 可用 `py -3` 替換 `python3`。將輸出儲存，記錄日期、Python 版本、參數與預測。

條件	執行前預測	實際輸出	差異與解釋
原始設定	自行填寫	貼關鍵欄位	寫出依據
修改設定	自行填寫	貼關鍵欄位	一次改一個變因

完成預測與紀錄後，再到解答冊的「預期輸出」核對。

變更與判讀

填一份控制表：版本、CPU、路徑、RTT、丟包、頻寬、負載、次數、計時起迄。

A 比較快嗎？說明互動延遲中位與尾端需求。n=10 如何限制尾端結論？本例 nearest-rank p95 等於最大值，只由一筆觀測決定。

驗證範圍與提交

輸出是明示假設下的模型或人工資料；不能當成真實封包擷取、OS 核心狀態或 Internet 量測。

提交預測、原始 / 修改輸出、兩個判讀答案與模型限制。評分：預測 20%、紀錄 30%、推理 35%、限制 15%。標準推理見解答冊；輸出中的埠與真實計時可以變動。

B：期限內交付率

以下是人工完整分布，不是測量資料：100 份互動訊息，deadline=100 ms。設計 A 最終到達 95%，其中 99% 在期限內；設計 B 最終全數到達，但只有 88% 在期限內。計算各自期限內成功率、晚到率及未到率，再說明哪個指標適合即時互動。百分比可視為大量樣本的期望，不需取整數。

C：HTTP 與代理的比較計畫

不執行外網測速，只設計控制表。固定同一服務端、網路路徑、硬體與內容，分四格：首次 / 恢復 × 小請求 / 大檔案。另記 DNS 與連線是否重用、TLS 恢復及 0-RTT 是否真的啟用、快取是否命中、H2/H3 實作版本、並行 stream 數、RTT、丟包、CPU、goodput 及延遲分布。先寫各格假設，再列需要收集的證據；「恢復」不能直接推論用了 0-RTT。代理加密比較另固定核心版、cipher、payload 與加密層數。

將 B、C 附於原提交資料。評分沿用本實驗規準；沒有真實核心與原始量測，只能交實驗計畫與人工數值分析。

L12 故障診斷與專題工作紙

資料性質與任務

對應 09、12、16、23、24。以下全為人工情境，無需額外工具。先寫事實、提出不同層次假設，再選最能區分的下一份證據。看解答前保留自己的推理。

五個案例

A：Wi-Fi 關聯成功，本機只有 169.254.x.x、沒有有效 DHCP 租約與預設路由。問：目前能證明什麼？先查 DNS 是否是最佳一步？

B：名稱解析成功；TCP 已建立；TLS 回 hostname mismatch。問：有何安全修正？只依此可否斷言遭受中間人攻擊？

C：100-byte 內容正常，20000-byte 內容停滯；只有「網站打不開」報告。問：提出 MTU 與應用各一個假設，需何證據區分？

D：瀏覽器 IP 檢查顯示代理出口；應用 UDP 和 IPv6 沒有入口日誌，實體出口可觀察直接到目標的封包。問：預期全流量政策哪裡有缺口？DNS 改成加密能直接修好嗎？

E：人工拓撲：C=198.51.100.20:51000，連路由器 R 的 203.0.113.10:443；R 只 DNAT 至 S=10.0.0.2:8443，不改來源。S 的預設閘道是另一臺 R2=10.0.0.1；R 有狀態紀錄，R2 沒有。R 已將 SYN 轉到 S，S 已回 SYN/ACK，但 C 未收到。列出回程、反向轉換及可達性的假設；為何不能只看 R 的 DNAT 規則斷言成功？提出不改公開服務名稱的兩種路徑修正設計。

案例	已知事實	假設 1/2	下一份證據	仍不能確定
A	自填	自填	自填	自填
B	自填	自填	自填	自填
C	自填	自填	自填	自填
D	自填	自填	自填	自填
E	自填	自填	自填	自填

專題報告模板

問題與可反駁假設；拓撲與每段端點；DNS/路由/加密/信任；至少兩個設計；固定版本與環境；負載和計時；原始資料；統計摘要；受控故障與恢復；取捨和未確認結論。

L09 的三模式輸出可完成端點專題；L02/06/11 可完成模型與 deadline 專題；L03/08/14 可完成政策專題。至少使用兩份不同資料，但不得把相互獨立實驗拼成已部署整套 VPN 的證據。評分依 23 單元規準。

L13 本機 TLS 的信任與名稱驗證

目標、環境與預測

對應 13、14。Python 3.10 以上、支援 TLS 1.3 的 ssl backend，另需可建立 X509 憑證的 OpenSSL 相容指令。程式建立一次性 RSA2048 自簽憑證，SAN 有 DNS:localhost 與 IP:127.0.0.1；只加入指定 Python SSLContext，不更動 OS 信任庫；暫存憑證及私鑰在結束時刪除。

預測三次結果：明確信任 + localhost；明確信任 + wrong.test；預設信任庫 + localhost。名稱與信任為獨立因素，但驗證通常只回報第一個失敗原因，不能由單一錯誤碼同時讀出全部原因。

執行與紀錄

```
python3 labs/L13-tls.py
```

每案記錄 verified、錯誤類別、verify_code / reason 與 reply；verify_code / reason 在不同 backend 可不同，不以完全相同文字評分。結果同時記錄憑證產生指令的版本與 Python ssl 版本，它們可能是不同軟體。

此實驗不是真實公開 CA 部署，不測 ECH、0-RTT 或流量可辨識性。固定自簽 leaf 為明確信任的實驗 anchor，不能推廣為忽略任何外部憑證。

沒有 OpenSSL / TLS 1.3 的離線路線

程式會在建立任何暫存檔前檢查 Python 3.10 以上、ssl backend 的 TLS 1.3 與 OpenSSL 相容指令；任一不符即結束並提示改走本路線。使用下列人工條件完成相同判讀，清楚註明未執行握手。三案的憑證都在有效期內，用途與其他驗證條件也都符合，只改變是否明確信任此憑證與預期名稱：

預期名稱	明確信任此憑證？	憑證 SAN	預測是否通過與原因
localhost	是	DNS:localhost	——
wrong.test	是	DNS:localhost	——
localhost	否，且不在信任鏈	DNS:localhost	——

問題與提交

1. TCP 可達與 TLS 驗證各支持何結論？
2. 明確信任為何不能解除名稱錯誤？
3. 為何不使用 check_hostname=false「修正」？
4. 憑證工具與 Python ssl 版本不同有何影響？

提交三案例輸出或離線表，標示路線、版本及限制。評分預測 20、紀錄 30、原因 30、信任邊界 20。OpenSSL 缺失、政策限制等不是更改系統安全設定的理由，使用離線路線即可。

本版只驗證 macOS / Python 3.12.14，另顯式測 STRICT / PARTIAL_CHAIN；不等同 3.13 以上實測。憑證含 SKI / AKI，完整版本與結果見 README 及驗證紀錄。3.13 起預設旗標改變，見 [Python ssl 文件](#)；其他環境結果不同時，記錄錯誤並保留驗證。

L14 代理 / VPN 分層與路徑工作表

任務與資料範圍

對應 15、18 至 22，也複習 13 至 17。無需第三方核心。以下是概念分析與官方資料閱讀，不是實際部署 / 效能測試。每列填明協定或核心版、查閱日期與來源，不將未知寫成支援。

A：分層

逐列填 proxy protocol、協定內 encryption、transport、security、flow、接管與 OS route：

1. 基礎 VLESS none + RAW/TLS + Vision + SOCKS 入口。
2. VLESS + WebSocket + REALITY：查指定核心相容表，判斷問題。
3. SS2022 + 系統 SOCKS 入口 + 應用 HTTPS。
4. Hysteria 2 轉送一個 TCP 下載與一個 UDP 互動流。
5. TUIC v5 兩種 UDP relay 模式。
6. WireGuard + TUN 類 IP 介面，OS 僅路由企業私網。

B：目的、DNS 與信任

欄位	要填的證據或假設
最後目的	IP、域名、port；由哪一層傳
DNS	bootstrap、目標 A/AAAA、應用自帶 DoH 各由誰處理
金鑰	PSK、身分私鑰、憑證根、會話祕密來源
PFS	對何種長期祕密洩漏，哪些資料期別
重放	哪一段由誰以序號 / session / salt / 時間檢查
可見性	C、P、出口、S、路徑觀察者各能看到什麼

C：全流量政策

預期所有外部資料經 P，LAN 直連、P endpoint 走 underlay。列出 RA 來源 / IPv6 預設路由、IPv4 TCP、IPv6 TCP、UDP、DNS、應用 DoH、直接 IP、LAN 與 P endpoint 的預期及實際路徑。若只有瀏覽器 IP 檢查結果，哪些列仍未確認？

D：交付與評分

提交六列分層表、至少兩個 key / 信任分析、完整雙協定政策表及一個無法驗證的限制。附錄 B 可查，但需寫出理由。評分角色分類 30、版本/證據 20、信任 20、路徑 20、限制 10。

最重要的反例：TUN 本身不加密；AllowedIPs 不等於 OS 所有路由；QUIC stream 不是不可靠 DATAGRAM；AEAD 不自動 PFS；外觀相似不保證匿名或無法辨識。標準分析見解答冊。

E : H2/TCP 與 H3/QUIC 的三串流遺失

人工模型：stream 1、2、3 互相獨立、沒有應用或 QPACK 依賴。H2 資料共享一條 TCP；其中較早的 TCP bytes 遺失，承載 stream 2、3 的後續 bytes 已到。H3 的 UDP 封包只帶 stream 1 資料並遺失，stream 2、3 的資料在其他已到封包。

各畫一條時序：遺失、後續到達、buffer、可交付、恢復。問 H2 為何受 TCP 連續交付限制？H3 哪些 stream 可先交付？再列兩項現實中仍可能共享的限制，如壅塞控制、CPU 或應用依賴。本題只分析傳輸交付，不推論真實 H3 一定更快。E 納入角色分類與證據兩項評分，不另加分。

L15 單介面多 IP 與政策路由

問題與驗證範圍

對應 03A、10、22。必修為 Python 3.10 以上標準函式庫離線查表，不需要管理員權限；不修改任何主機網路。延伸是 Linux 命名空間與 iproute2 / ping，需要隔離 Linux VM 與 root，尚未在本教材製作環境執行。先讀 03A 的 H / G1 / G2 拓樸。

A：預測與執行

在課程根目錄執行：

```
python3 labs/lan_models.py L15
```

執行前填下表，不先查看解答。

案例	來源	目的	預測表 / 下一跳
no_policy	198.51.100.10	203.0.113.80	_____
source_unspecified	未指定	203.0.113.80	_____
source_policy	198.51.100.10	203.0.113.80	_____
lan_correct	198.51.100.10	198.51.100.55	_____
lan_missing_route	198.51.100.10	198.51.100.55	_____
endpoint_exception	198.51.100.10	203.0.113.81	_____
local_delivery	198.51.100.10	192.0.2.10	_____
throw_fallthrough	198.51.100.10	198.51.100.55	_____
explicit_reject	198.51.100.10	198.51.100.55	_____

輸出中的 kind=model 表示人工查表；trace 列出規則是否匹配、所查表與結果。selected_source 只表示此簡化模型的固定 / 偏好來源，不模擬完整 OS 來源選擇。

B：判讀與修改

1. 比較 no_policy 與 source_policy：來源是否改變？下一跳是否改變？
2. lan_missing_route 中 main 仍有 /24。指出為何 lookup 200 的 default 先成功；在 L15 函式建立 bad[200] 的地方，將缺少的 /24 加回 bad[200] 後重做；demo_tables 原本已有直連路由。
3. endpoint_exception 用 priority 50 目的規則查 main，早於 priority 100 來源規則。移到 150 後，預測會變成哪個下一跳再驗證。
4. throw_fallthrough 與 explicit_reject 目的相同，何者繼續查 main？trace 在哪裡停止？
5. 比較來源尚未指定與明確指定，說明 from 規則的資料前提。

修改前另存自己的程式副本；提交差異與兩組輸出。模型不處理 ECMP、route cache、mark 建立時機、NAT 或完整 Linux socket；不得稱作 Linux 互通測試。

C：隔離 Linux 延伸

只在自己管理的 Linux VM 中執行原檔：

```
sudo bash labs/L15-linux-netns.sh
```

腳本建立具唯一名稱的 H、SW、G1、G2 命名空間。H 僅一個 eth0 帶兩個 IP；SW 橋接，G1 / G2 的 loopback 各放 203.0.113.80/32。這是兩臺獨立隔離的回應者共用目的 IP，用於觀察第一跳；不是同 LAN 配置重複主機位址，也不模擬 Internet 或完整雙 WAN。G1 另有回來源 2 的路由，讓 bind 來源 2、尚未加政策時也有回程。

腳本依序列出位址、規則、main / 200 / local 表、route get、ping 與鄰居表；先看未加政策，再看只放 200 default 的錯例，最後補直連路由與 endpoint 例外。預期：S / 來源 2 從 G1 改到 G2，LAN 錯例顯示 via G2，補正後顯示直接 dev eth0。實際文字、ping 成功與封包必須以自己的 Linux 版本記錄。

所有介面與路由操作在新建命名空間內，沒有連接實體網卡或改 host default。正常結束及可處理的中斷會清除；強制終止或主機崩潰仍可能殘留，請依開頭顯示的四個精確名稱在 VM 內逐一刪除，勿使用刪除所有命名空間的操作。

若要抓 ARP，需自行在對應 namespace 啟動 tcpdump 並確認權限；腳本本身未抓封包。ping 只證明該隔離拓樸的 ICMP 回應，不能推論代理、TLS 或 WAN 效能。

紀錄與評分

每案記「預測 / kind / 來源 / priority / table / prefix / next_hop / dev / 實際差異 / 限制」。離線路線滿分 10 分：查表 4、LAN 與 endpoint 2、來源 / SNAT 區別 2、驗證限制 2。Linux 延伸不影響必修完成；另交 OS、kernel、ip 版本與原始輸出。

來源：03A 章末的 iproute2 與 Linux 官方文件。解答見 answers/lan-solutions.md。

L16 LAN 交付、多播成員與服務探索

問題與驗證範圍

對應 04A、05、11、22。必修使用標準函式庫離線交換模型，沒有送出任何封包。FDB 來源學習與簡化 MDB 事件可觀察；join、leave 與 expire 是人工事件，不編碼 / 解析 IGMP、MLD、PIM 或 mDNS。Linux bridge 觀察列為未實測延伸。

A：先預測交換器出口

讀 04A 四埠加 VLAN 20 拓樸，執行：

```
python3 labs/lan_models.py L16
```

案例	預測出口	判讀依據
unknown_unicast	——	FDB / VLAN
known_unicast	——	FDB
broadcast	——	廣播域
multicast_known	——	成員 2、3 與 router port 4
multicast_after_leave	——	已確認移除埠 3
multicast_after_expiry	——	埠 2 老化，仍有 router port 4； unknown=flood
multicast_unknown_drop	——	unknown=drop 仍保留 router port 4
mdns_link_local	——	模型的 224.0.0/24 特殊政策
multicast_snooping_off	——	同 VLAN 泛洪

members_before 與 members_after_expiry 是 MDB 快照。leave 在模型中立刻移除整個埠，是「已確認無其他接收者」後的狀態；真實共享埠不能直接照此操作。expiry 以人工時鐘觸發，沒有真實 Query / Report 計時器。router port 4 不一定有本地應用接收者，仍需收到供路由的資料。

B：改一個變因

1. 將 router_ports 移除，對比 known、unknown=flood 與 unknown=drop。
2. 不呼叫 join，為何模型仍可能泛洪？這是否證明主機已加入群組？
3. 將埠 3 移到 VLAN 20 後，重新設定合法成員，預測 VLAN 10 的出口；跨 VLAN 輸出不得出現 3 或 5。
4. 用 ipv4_multicast_mac 比較 239.192.0.42 及 224.64.0.42，再解釋 IP 群組仍不同。模型 MDB 依 IP 群組鍵查找，不模擬僅按 MAC 過濾的別名副本。
5. 模型特別處理 224.0.0/24，不模擬 IPv6 / MLD。提出若要加入 IPv6，除了改位址長度還要補哪些 scope 與控制政策。

C：離線服務探索工作表

這些是人工 DNS-SD 資料，不是擷取記錄。

欄位	資料
PTR 查詢	_ipp._tcp.local
PTR 實例	Lab Printer._ipp._tcp.local
SRV	printer.local , port 631
TXT	rp=printers/lab
A	192.0.2.50
AAAA	2001:db8:10::50

1. 列出 mDNS 探索封包的群組、UDP 埠；列印連線是否也必須多播？
2. 若 C 在 VLAN 20，能手動連 192.0.2.50 卻看不見 PTR，分別寫探索與資料路徑。
3. 若 PTR 可見但 AAAA 路徑被防火牆拒絕，補什麼證據能定位連線失敗？
4. 某 gateway 反射所有服務，該如何限制可見範圍並確認服務授權？

D：Linux bridge 觀察延伸

只在自主管理的 Linux 隔離環境使用現有 bridge，先確認名稱 / VLAN。以下只讀命令中的 br0 需換成自己的實驗 bridge：

```
bridge fdb show br br0
bridge -d -s mdb show dev br0
bridge vlan show
ip maddr show
ip -d link show dev br0
```

分別記錄 FDB、MDB、router port、multicast_snooping / querier 及成員介面。ip maddr 是介面群組資訊，不能單獨證明應用 socket 或 snooping 狀態。若有自有封包紀錄，檢查 Query / Report 及 UDP 5353，而非僅看資料流是否到達。沒有 bridge 或擷取環境就完成 A 至 C，不以人工結果冒充 Linux 觀察。

紀錄與評分

每案填「預測 / 實際模型出口 / VLAN / FDB / MDB / 政策 / 模型限制」。10 分：出口與廣播域 3、成員 / router port / 未知政策 3、DNS-SD 路徑 2、證據與限制 2。來源：04A 章末 RFC 及 Linux bridge 文件。解答見 answers/lan-solutions.md。