【2025 進階電腦網路 作業5】

子網路 IP 掃描器 (Subnet IP Scanner)

規則

1. 請在 **Ubuntu 24.04** 作業系統下完成本次作業。
2. 你必須使用 **C 語言** 完成本次作業，並提供 `Makefile` 。
3. **禁止抄襲他人的作業，否則該課程將不及格。**
4. 將作業壓縮成 tar 檔案，命名為 `TCPIP_HW5.zip` 或 `TCPIP_HW5.rar` ，並於期限前上傳至 Tronclass 系統 `https://elearn.nsysu.edu.tw/`
5. 如有任何問題，請發送郵件至助教信箱 `net_ta@net.nsysu.edu.tw` 或於 11:00~17:00 前往 EC5018 室。**但助教不會幫你除錯程式。**
6. **你必須深入了解你的程式在做什麼，因為助教會在 demo 時詢問你關於程式的問題。**
7. **截止日期：2025/12/17 09:00。不接受遲交。**如有任何困難，請事先透過電子郵件通知我們。

⚠️ 重要警告

- **不接受遲交！**
- **Demo 時會被問問題，必須充分理解程式！**
- 抄襲將導致課程不及格

作業5：子網路 IP 掃描器

需求說明

掃描器向所有其他子網路 IP 位址發送 **ICMP echo request**。

範例情境：

假設你的主機 IP 位址是 `140.117.171.148`，子網路遮罩是 `255.255.255.0`。

ICMP echo request（類型 8）會被分別發送到 `140.117.171.1` ~ `140.117.171.254`（除了自己），如果子網路上的主機是活躍的，就會捕獲 ICMP echo reply（類型 0）。

這是假設 ICMP 已啟用（Ubuntu 22.04 預設啟用）。然後主機會在收到 ICMP echo request 時自動發送 ICMP echo reply。

重要：ICMP echo request 的 IP 標頭 TTL（Time-to-Live）欄位 **必須設定為 1**。因此，ICMP echo request 將被限制在其子網路內。

發送 ICMP Echo Request 封包

填寫 IP 標頭（IP Header）

根據以下格式填寫 IP 標頭：

欄位	值/說明
Header length（標頭長度）	自行計算
Total length（總長度）	自行計算
Id（識別碼）	0
Flag（旗標）	don't fragment（不分段）

TTL	1
Protocol（協定）	ICMP
Checksum（檢查碼）	可以讓作業系統處理

填寫 ICMP 封包

根據以下格式填寫 ICMP 封包：

欄位	值/說明
Checksum（檢查碼）	可以讓作業系統處理
ID	Process ID（程序 ID）
Sequence number（序列號）	從 1 開始，逐一遞增
Data（資料）	**你的學號**。注意：資料大小必須與 IP 標頭相符。

接收 ICMP Echo Reply 封包

實作方式

接收 ICMP echo reply 時，你可以使用：

- 標準 socket（如 ARP 作業中使用的那個）
- 或使用 `libpcap`（`sudo apt-get install libpcap-dev`）來實作

> **關於 libpcap**
>
> `libpcap` 用於 Wireshark 和 tcpdump 中。你可以設定過濾器來從所有傳入的封包中捕獲預期的封包。
>
> 你可以使用 `tcpdump` 來測試你的過濾規則。你在 tcpdump 上使用的規則基本上也可以用在 libpcap 上。可能會有一點差異，但看起來很相似。
>
> **你應該使用過濾器來最小化接收到的封包數量。**

ICMP Echo Reply 檢查項目

> 對於 ICMP echo reply，請檢查以下欄位：
>
> 1. IP 標頭中的來源位址（source）
> 2. ICMP 類型（type）
> 3. ICMP 訊息中的 ID 是否與你設定的相符
> 4. ICMP 訊息中的序列號是否與 echo request 相同

> libpcap 初始化程序的一部分已經在我們提供的檔案中。你只需要設定合適的過濾規則。
>
> 發送的封包必須使用 socket，如下圖所示。

提供的檔案

- `fill_packet.c` / `fill_packet.h`
- `pcap.c` / `pcap.h`
- `main.c`

- `ping.c`（Ubuntu 上的原始 Ping 程式，可以參考。完整套件：`http://www.skbuff.net/iputils`）

輸入使用方式

```
# sudo ./ipscanner -i [網路介面名稱] -t [超時時間(ms)]
```

> 你的程式**必須自動獲取**使用者提供的網路介面上的主機 IP 位址和子網路遮罩。

範例情境

Ubuntu 主機網路介面 `enp4s0f2` 的 IP 位址為 `140.117.171.148`，子網路遮罩為 `255.255.255.0`。

因此程式會掃描 `140.117.171.1` ~ `140.117.171.254`

例如，有一台 IP 為 `140.117.171.1` 的主機會發送 ICMP echo reply。

封包說明

No.63-64：ARP 請求與回覆

> Socket 會自動尋找 `140.117.171.1` 的 MAC 位址。所以它會發送 ARP request。
>
> `140.117.171.1` 發送 ARP reply，讓主機知道它的 MAC 位址。

ICMP 封包內容

欄位	值
Type	8
Id	0x256
Seq	1
Data	你的學號（十六進位），**你必須改成你的學號**

無回應情況

> Socket 會尋找 `140.117.171.1` 的 MAC 位址，所以它發送 ARP request。
>
> 但如果沒有任何回覆，它會收到一個 ICMP 訊息，表示找不到回應或其他原因。

實作重點提醒

> ### ✓ 必須實作的功能
>
> - 自動取得網路介面的 IP 和子網路遮罩
> - 計算子網路範圍並掃描所有 IP（排除自己）
> - 正確設定 IP 標頭（特別是 TTL = 1）
> - 正確設定 ICMP 封包（類型、ID、序列號、資料）
> - 使用 libpcap 或 socket 接收 ICMP echo reply
> - 驗證接收到的封包欄位
> - 處理超時機制

聯絡資訊

助教 Email： net_ta@net.nsysu.edu.tw

實驗室： EC5018

時間： 11:00~17:00

Tronclass 系統： https://elearn.nsysu.edu.tw/

截止日期

2025年12月17日 上午 9:00

❌ **不接受遲交！**

如有困難請事先透過郵件通知