**電腦網路實驗實驗報告 < Packet Analysis >**

**姓名: 翁佳煌　　　學號: 409430030**

# 實驗名稱

網路封包擷取操作與分析

# 實驗目的

這次實驗有三個LAB，LAB1主要是要理解TCP和UDP協議的差異，並學習如何找出TCP三向交握和UDP的封包，以及如何使用Wireshark等工具進行協議分析，以加深對網絡協議的理解和熟練掌握分析技能。LAB2是分析HTTP協議，學習HTTP協議的基本知識和特點，深入了解HTTP協議的結構和原理，學會使用Wireshark等工具進行HTTP協議分析，以提高對網絡協議的認識和能力。LAB3是DNS分析，了解DNS協議的基本知識和作用，用Wireshark工具進行DNS協議分析，深入了解DNS協議的運作原理和相關技術，以加強對網絡協議的掌握和分析能力。

# 實驗設備

Linux作業系統之電腦。

Wireshark。

# 實驗步驟

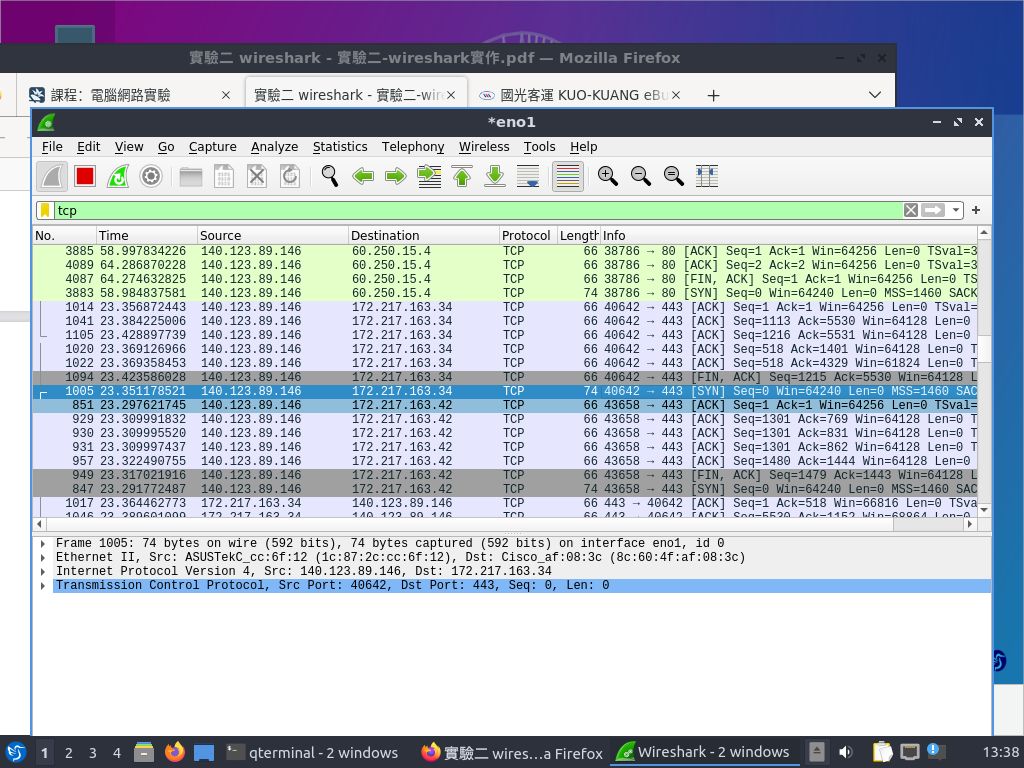
**LAB1-TCP、UDP:**

1.開啟Wireshark。

2.開啟國光客運的網站。



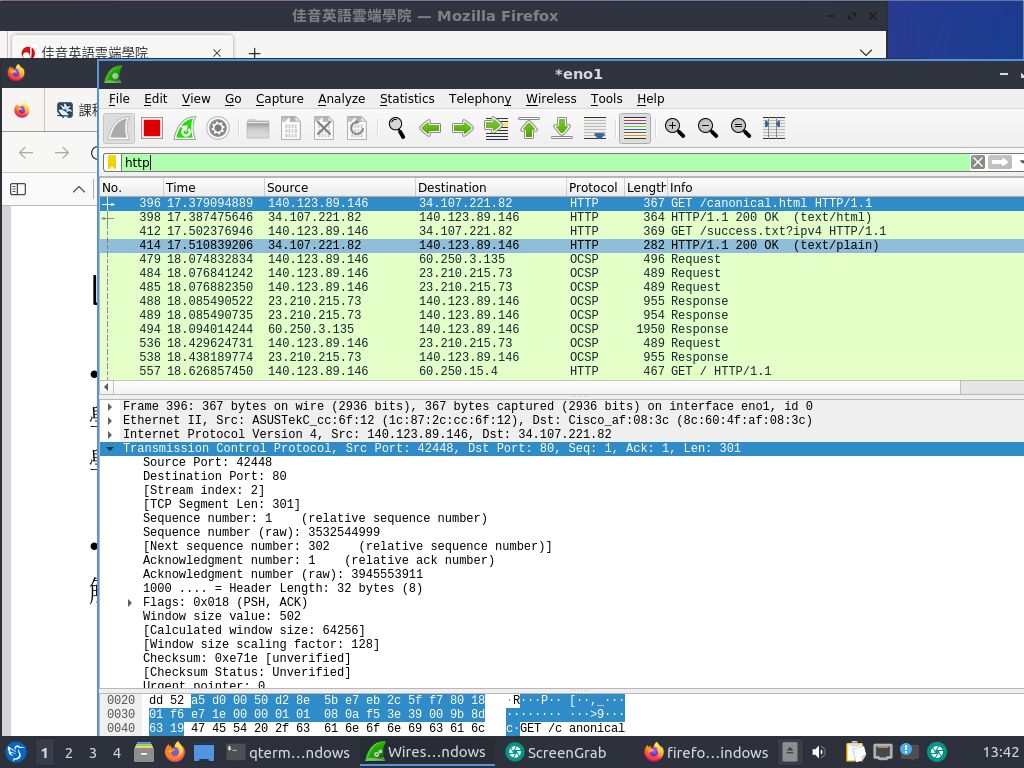
3. 擷取完畢後，經由Filter顯示TCP的封包，隱藏其他的封包。



4.觀察步驟3顯示的封包，可觀察到SYN、SYN-ACK、Seq等訊息，我將會在下面的問體與討論更深入探討這些名詞。

**LAB2- 分析HTTP協議:**

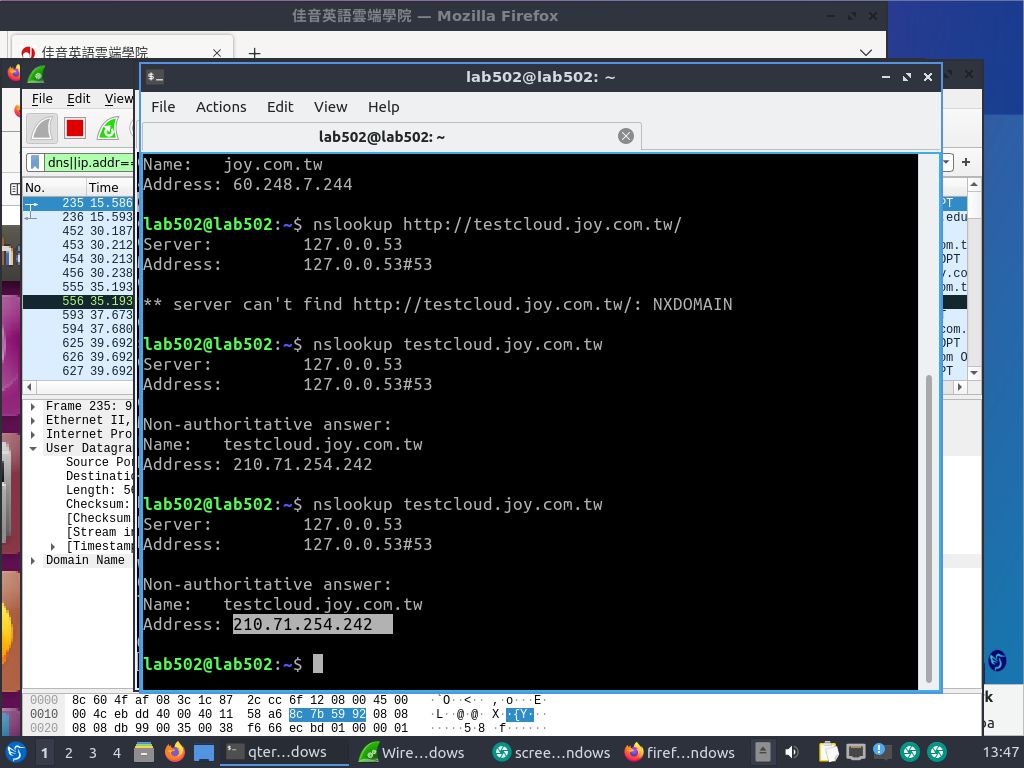
1. 開啟Wireshark。
2. 打開使用http的網站，這裡是開啟佳音英語的網站。
3. 返回Wireshark，在filter中過濾出http的封包。

****

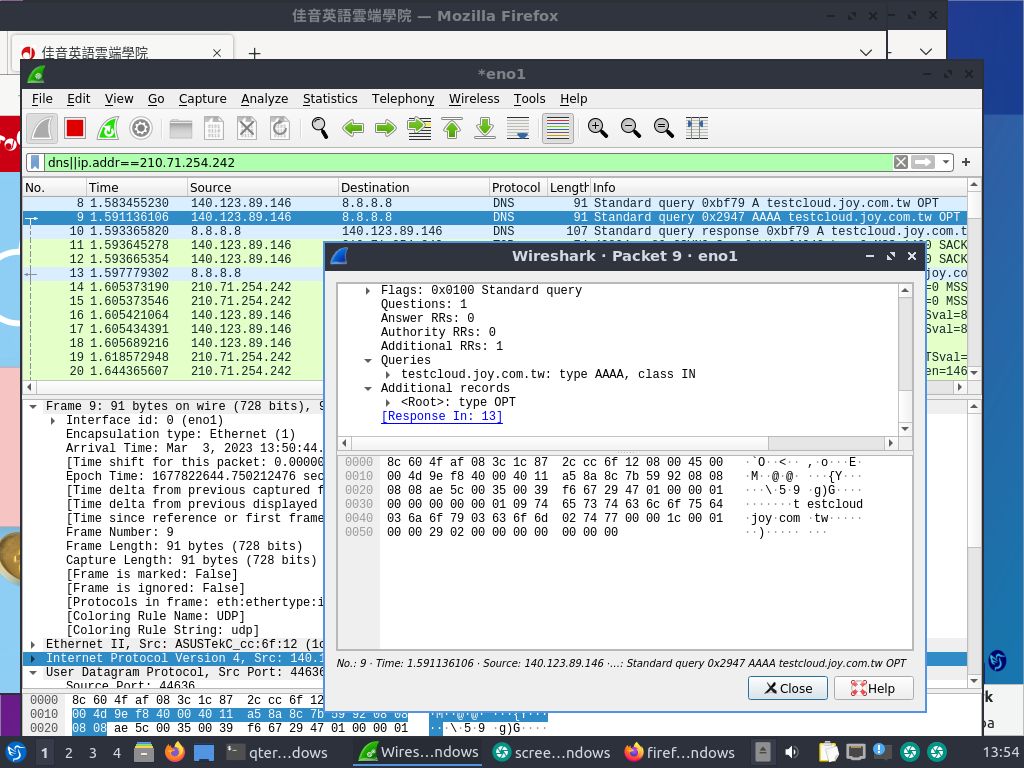
1. 觀察http協議中的內容，我將會在下面的問體與討論更深入討論這部分。

**LAB3- DNS分析:**

1.開啟terminal，使用nslookup查詢佳音英語網站的ip地址。

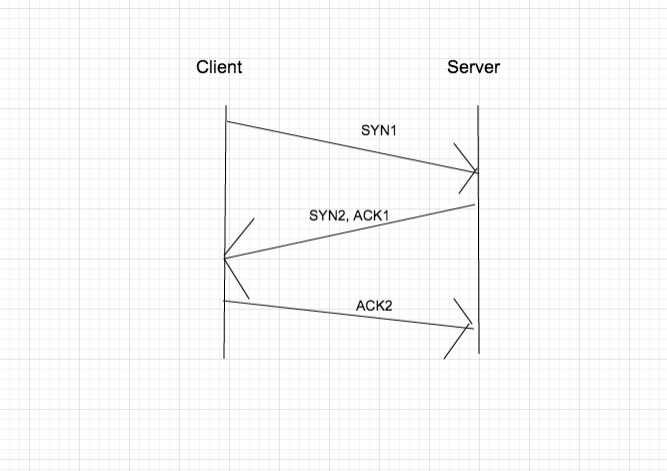
****

2.回到Wireshark，利用指令dns||ip.addr==210.71.254.24過濾後，觀察連線是使用UDP還是TCP，由紅色框框可知，DNS是使用UDP。



# 問題與討論

1. **配合封包內容解釋TCP三向交握的原理。 (解釋SYN、ACK..等) 比較TCP和UDP:**



TCP是一種面向連接的協議，它使用三向交握（Three-way Handshake）建立可靠的連接。TCP 通過確認封包是否已被接收，以確保數據的可靠性。三向交握過程如下：

客戶端向服務器端發送 SYN 封包（SYN = Synchronize Sequence Numbers），表示客戶端想要建立一個連接，並傳遞自己的初始序號 ISN（Initial Sequence Number）。

服務器端收到客戶端的 SYN 封包後，回傳一個 SYN-ACK 封包，表示服務器已經收到客戶端的請求，同時也傳遞自己的初始序號 ISN。

客戶端收到服務器端的 SYN-ACK 封包後，向服務器端回傳一個 ACK 封包（ACK = Acknowledgment），表示客戶端已經收到服務器端的回應，此時雙方建立了可靠的連接，可以開始傳輸數據。

在 TCP 連接中，每一個封包都有一個序號（Sequence Number）和一個確認號（Acknowledgment Number）。序號表示傳輸的字節流中的第一個字節的序號，確認號表示接收端期望接收的下一個序號。通過使用序號和確認號，TCP 可以確保數據的可靠性和有序性。

相較於 TCP，UDP (User Datagram Protocol) 是一種無連接的協議，它不使用三向交握或其他機制來建立可靠的連接。UDP 更加簡單，不需要像 TCP 那樣建立連接和確認數據傳輸，因此 UDP 的傳輸速度更快，但不可靠性也更高。因此，UDP 通常用於那些較需要傳輸速度較快，而不需要保證每一個封包都被傳輸成功的應用，例如視頻和音頻傳輸等。

**2. 配合LAB2封包內容解釋其中Host、accept、cookie…等:**

在LAB2的Request中，我們可以看到許多的header，以下是其中幾個的解釋：Host為請求的網站主機名稱，這個是必須的。

Accept是指定客戶端支援的MIME類型，例如text/html、application/xhtml+xml等。Cookie為將之前由該服務器發送的cookie發送回服務器，可以用於會話跟蹤等。User-Agent: 請求的客戶端類型，例如是瀏覽器還是爬蟲等。

在Response中，Server回應的服務器類型和版本號。這個header可以讓客戶端知道該回應是由哪個服務器發送的，以及服務器的版本號等信息。

Content-Type為回應的內容類型，例如text/html、application/json等。這個header可以讓客戶端知道該回應的內容類型，進而根據這個信息來進行相應的處理。

Content-Length是回應內容的字節長度。這個header可以讓客戶端知道回應的內容長度，以便客戶端可以知道何時結束接收回應。

Set-Cookie為服務器返回的cookie信息。當服務器向客戶端發送回應時，可以通過Set-Cookie header返回一個新的cookie，這樣客戶端就可以在後續的請求中將該cookie發送回服務器，以維護會話等機制。

**3.解釋為什麼DNS適合用UDP protocol 而不是TCP protocol:**

DNS是一種用於將域名解析為IP地址的協議，它使用UDP協議進行通信。以下是列出幾個原因解釋為什麼DNS適合使用UDP協議而不是TCP協議：

效率：UDP協議的優勢之一是它非常輕量級，它不需要像TCP協議那樣建立一個可靠的、雙向的連接。由於DNS是一個較為簡單的協議，通常請求和回應的大小都比較小，因此使用UDP協議可以提高效率，減少開銷。

時間：DNS通常用於網絡連接的一開始，例如瀏覽器輸入URL時，需要向DNS服務器發送請求以解析該URL的IP地址。由於DNS的請求和回應通常比較短小，且需要快速返回，因此使用UDP協議可以避免TCP協議的握手過程，減少延遲時間。

可靠性：雖然UDP協議不像TCP協議那樣可靠，但在DNS這種情況下，使用UDP協議仍然足夠可靠。由於DNS的請求和回應都很短小，且在互聯網上的DNS服務器很多，因此即使出現一些丟包或延遲的情況，客戶端仍然可以重新發送請求或向其他DNS服務器發送請求，不會對整個應用造成太大影響。

總而言之，DNS使用UDP協議是出於效率、時間敏感和可靠性等因素的綜合考慮，使用UDP協議可以更好地滿足DNS協議的特點和需求。

# 心得與感想

這次的三個LAB實在是讓人學到不少東西，首先LAB1，透過Wireshark工具觀察到了TCP三向握手和UDP封包的傳輸過程，進一步理解了TCP和UDP的區別和適用場景。對於TCP三向握手，我理解到SYN、ACK、FIN這些標誌位的含義和作用，以及三向握手的重要性和原理。在實際的網絡開發中，我發現TCP協議的穩定性和可靠性更高，但傳輸速度相對較慢；而UDP協議則更快，但可靠性相對較差。因此，在實際的應用場景中，需要根據不同的需求和情況選擇適合的協議。

再來LAB2，通過本次實驗，我深入了解了HTTP協議和網絡通信的基本原理。我通過Chrome開發者工具觀察到了HTTP協議中request和response的內容和結構，了解了其中Host、accept、cookie等字段的含義和作用。

最後LAB3，我學會了使用nslookup和wireshark工具來分析DNS的運作過程。我了解到DNS的作用和重要性，以及域名解析的流程和原理。同時，我也學習到了為什麼DNS適合使用UDP協議而不是TCP協議。

# 參考文獻

1. LAB1:

<https://notfalse.net/7/three-way-handshake>

<https://ithelp.ithome.com.tw/articles/10155938>

1. LAB2:

<https://ithelp.ithome.com.tw/articles/10158471>

<http://skenyeh.blogspot.com/2011/01/http-status-codes.html>

<https://zh.wikipedia.org/wiki/HTTP>

3. LAB3 :

<https://blog.csdn.net/jason_cuijiahui/article/details/86712107>

<https://kknews.cc/zh-tw/code/qvap9yg.html>