資訊通訊概論

實驗：Network monitoring with tools

組員：1103309 王聖允、1103318 詹承義

1103309 王聖允、1103318 詹承義

Experiment 1:

一、名詞解釋

實體位址:也稱為 MAC address，是每一張網路卡的識別碼(可在實驗三的片中看見)，由 6 組 16 進位的數字所組成的物理位置，前三組數字為廠商 ID；後三組數字為網路卡的卡號，每張網卡的 MAC address 都會是唯一的（如果沒有特別設定的話）。

網路遮罩:也稱為 Masking，可以用來取得某 IP 地址的子網域標示碼，只要將 IP 位址和網路遮罩做邏輯 AND 運算即可。

DHCP server:中文叫做「動態主機組態協定」，主要用於動態分配 IP 位址，因為 IP 位址的數量不夠，所以當有電腦要連線時他才會分配一個 IP 位址給該電腦，該電腦使用完畢後再把 IP 位址歸還給 DHCP server，這樣就能節省 IP 使用量了。

DNS server: 全名為 Domain Name Server，Domain Name 就是每個網站的網址，例如:元智的網址是 www.yzu.edu.tw，而 DNS server 就是把網址轉為 IP 位址的工具,因為在 Internet 上是使用 IP 來辨別而不是 Domain Name。

IPv4:是一種網路協定，用於建立及分配 IP 位址，使用 32 位元長度的數字位址，總共有 2 的 32 次方個獨一無二的位址，然後使用 10 進位來表示 IP 位址。

IPv6:IPv4 的升級版，也是用於建立和分配 IP 位址，使用 128 位元長度的數字位址，總共有 2 的 128 次方個唯一位址，使用 16 進位來表示 IP 位址。

IPv4 VS IPv6:IPv4 使用點區隔的十進位表示法，例如:192.168.100.2，總共有 2 的 32 次方個組合；IPv6 使用冒號區隔的 16 進位表示法，例如:fe80::a143:abbf:af78:f541%8，總共有 2 的 128 次方個組合。

Gateway:協助不同子網域之間的溝通，如果與不同子網域的裝置連線則必定需經過 Gateway 轉送。(子網域計算方法見**網路遮罩**)

二、實際操作及比較

組合一：「宿舍網路」及「宿舍網路加上個人 Router」

```
乙太網路卡 乙太網路 3:

   連線特定 DNS 尾碼 . . . . . . . . :
   描述 . . . . . . . . . . . . . . : ASIX AX88179 USB 3.0 to Gigabit Ethernet Adapter
   實體位址 . . . . . . . . . . . . : 00-0E-C6-49-30-56
   DHCP 已啟用 . . . . . . . . . . . : 是
   自動設定啟用 . . . . . . . . . . . : 是
   連結-本機 IPv6 位址 . . . . . . . : fe80::3141:9dc4:f083:e51a%12(偏好選項)
   IPv4 位址 . . . . . . . . . . . . : 140.138.242.143(偏好選項)
   子網路遮罩 . . . . . . . . . . . : 255.255.255.0
   租用取得 . . . . . . . . . . . . : 2022年10月6日 下午 04:27:13
   租用到期 . . . . . . . . . . . . : 2022年10月7日 下午 04:27:12
   預設閘道 . . . . . . . . . . . . : 140.138.242.254
   DHCP 伺服器 . . . . . . . . . . . : 140.138.200.3
   DHCPv6 IAID . . . . . . . . . . . : 369102534
   DHCPv6 用戶端 DUID. . . . . . . . : 00-01-00-01-29-09-4D-6E-00-E0-4C-68-0A-2C
   DNS 伺服器 . . . . . . . . . . . : 140.138.200.1
                                       140.138.200.3
   主要 WINS 伺服器 . . . . . . . . . : 140.138.2.100
   次要 WINS 伺服器 . . . . . . . . . : 140.138.4.100
   NetBIOS over Tcpip . . . . . . . . : 啟用
```

▲宿舍網路(上圖)　　▼宿舍網路加上個人 Router(下圖)

```
乙太網路卡 乙太網路 3:

   連線特定 DNS 尾碼 . . . . . . . . :
   描述 . . . . . . . . . . . . . . : ASIX AX88179 USB 3.0 to Gigabit Ethernet Adapter
   實體位址 . . . . . . . . . . . . : 00-0E-C6-49-30-56
   DHCP 已啟用 . . . . . . . . . . . : 是
   自動設定啟用 . . . . . . . . . . . : 是
   連結-本機 IPv6 位址 . . . . . . . : fe80::3141:9dc4:f083:e51a%12(偏好選項)
   IPv4 位址 . . . . . . . . . . . . : 192.168.0.174(偏好選項)
   子網路遮罩 . . . . . . . . . . . : 255.255.255.0
   租用取得 . . . . . . . . . . . . : 2022年10月7日 上午 01:22:47
   租用到期 . . . . . . . . . . . . : 2022年10月14日 上午 01:22:47
   預設閘道 . . . . . . . . . . . . : 192.168.0.1
   DHCP 伺服器 . . . . . . . . . . . : 192.168.0.1
   DHCPv6 IAID . . . . . . . . . . . : 369102534
   DHCPv6 用戶端 DUID. . . . . . . . : 00-01-00-01-29-09-4D-6E-00-E0-4C-68-0A-2C
   DNS 伺服器 . . . . . . . . . . . : 192.168.0.1
   NetBIOS over Tcpip . . . . . . . . : 啟用
```

解釋：

1、因為兩者使用同一張網路卡，所以 MAC 位址是一樣的

2、由資服處網站可知，元智大學 IP 皆為 140.138 開頭，所以第一張圖中 DHCP 伺服器及 DNS 伺服器為元智大學網路主機的位址，而經由元智 DHCP 伺服器分發下來的 IP 也是 140.138 開頭。但接上 Router 後，DHCP 伺服器和 DNS 伺服器位址皆變成 192.168.0.1(192.168.0.1 為路由器之 IP)，由此可知，接上 Router 後的 IP 改為由 Router 扮演 DHCP Server 分發內網 IP。

3、Mask 為 255.255.255.0，轉成二進位並分別與 IP 做運算後，可得 140.138.242.0 以及 192.168.0.0，而他們的 Gateway 分別是 140.138.242.254 和 192.168.0.1； 而 gateway 又會與裝置位於相同子網域，所以可推測 gateway 之 mask 也是 255.255.255.0。

組合二：「住家網路」及「手機熱點(WIFI 分享 vs.USB tethering)」

```
乙太網路卡 乙太網路:

   連線特定 DNS 尾碼 . . . . . . . . :
   描述 . . . . . . . . . . . . . . : Realtek PCIe GbE Family Controller
   實體位址 . . . . . . . . . . . . : 3C-7C-3F-59-11-0E
   DHCP 已啟用 . . . . . . . . . . . : 是
   自動設定啟用 . . . . . . . . . . : 是
   連結-本機 IPv6 位址 . . . . . . . : fe80::a143:abbf:af78:f541%8(偏好選項)
   IPv4 位址 . . . . . . . . . . . . : 192.168.100.21(偏好選項)
   子網路遮罩 . . . . . . . . . . . : 255.255.255.0
   租用取得 . . . . . . . . . . . . : 2022年10月6日 下午 10:04:45
   租用到期 . . . . . . . . . . . . : 2022年10月8日 下午 09:39:00
   預設閘道 . . . . . . . . . . . . : 192.168.100.1
   DHCP 伺服器 . . . . . . . . . . . : 192.168.100.1
   DHCPv6 IAID . . . . . . . . . . . : 322731071
   DHCPv6 用戶端 DUID. . . . . . . . : 00-01-00-01-28-E6-EC-24-3C-7C-3F-59-11-0E
   DNS 伺服器 . . . . . . . . . . . : 192.168.100.1
   NetBIOS over Tcpip . . . . . . . : 啟用
```

▲住家網路

```
無線區域網路介面卡 Wi-Fi:

   連線特定 DNS 尾碼 . . . . . . . . :
   描述 . . . . . . . . . . . . . . : Intel(R) Wi-Fi 6 AX201 160MHz
   實體位址 . . . . . . . . . . . . : AC-12-03-E2-A9-4B
   DHCP 已啟用 . . . . . . . . . . . : 是
   自動設定啟用 . . . . . . . . . . : 是
   IPv6 位址. . . . . . . . . . . . : 2001:b400:e387:7a5f:c560:21d9:d54:23af(偏好選項)
   臨時 IPv6 位址. . . . . . . . . . : 2001:b400:e387:7a5f:d1fa:1b5a:69f2:8f65(偏好選項)
   連結-本機 IPv6 位址 . . . . . . . : fe80::c560:21d9:d54:23af%12(偏好選項)
   IPv4 位址 . . . . . . . . . . . . : 172.20.10.3(偏好選項)
   子網路遮罩 . . . . . . . . . . . : 255.255.255.240
   租用取得 . . . . . . . . . . . . : 2022年10月7日 下午 09:54:56
   租用到期 . . . . . . . . . . . . : 2022年10月8日 下午 09:54:54
   預設閘道 . . . . . . . . . . . . : fe80::499:bbff:febd:6b64%12
                                       172.20.10.1
   DHCP 伺服器 . . . . . . . . . . . : 172.20.10.1
   DHCPv6 IAID . . . . . . . . . . . : 229380611
   DHCPv6 用戶端 DUID. . . . . . . . : 00-01-00-01-28-E6-EC-24-3C-7C-3F-59-11-0E
   DNS 伺服器 . . . . . . . . . . . : fe80::499:bbff:febd:6b64%12
                                       172.20.10.1
   NetBIOS over Tcpip . . . . . . . : 啟用
```

▲手機熱點 WIFI（上）　　▼手機熱點 USB tethering（下）

```
乙太網路卡 乙太網路 2:

   連線特定 DNS 尾碼 . . . . . . . . :
   描述 . . . . . . . . . . . . . . : Apple Mobile Device Ethernet
   實體位址 . . . . . . . . . . . . : FA-38-80-86-BD-96
   DHCP 已啟用 . . . . . . . . . . . : 是
   自動設定啟用 . . . . . . . . . . : 是
   連結-本機 IPv6 位址 . . . . . . . : fe80::3d3c:ea55:9a1f:d035%2(偏好選項)
   IPv4 位址 . . . . . . . . . . . . : 172.20.10.7(偏好選項)
   子網路遮罩 . . . . . . . . . . . : 255.255.255.240
   租用取得 . . . . . . . . . . . . : 2022年10月7日 上午 01:38:57
   租用到期 . . . . . . . . . . . . : 2022年10月8日 上午 01:38:57
   預設閘道 . . . . . . . . . . . . : 172.20.10.1
   DHCP 伺服器 . . . . . . . . . . . : 172.20.10.1
   DHCPv6 IAID . . . . . . . . . . . : 637155456
   DHCPv6 用戶端 DUID. . . . . . . . : 00-01-00-01-29-09-4D-6E-00-E0-4C-68-0A-2C
   DNS 伺服器 . . . . . . . . . . . : 172.20.10.1
   NetBIOS over Tcpip . . . . . . . : 啟用
```

解釋:

1、三者皆使用不同網路介面連線，故 MAC 位址都是不同的。

2、由下面兩張圖可知，由手機分享熱點 DHCP 伺服器、 DNS 伺服器、gateway 都會是 172.20.10.1（手機之 IP），而由手機 DHCP 分發下來的 IP 為 172.20.10 開頭。

Experiment 2:

(A)這邊我們嘗試使用 Ping 連接知名網路服務 cloudflare 的 1.1.1.1：

```
Ping one.one.one.one [1.1.1.1] (使用 32 位元組的資料):
回覆自 1.1.1.1: 位元組=32 時間=34ms TTL=54
回覆自 1.1.1.1: 位元組=32 時間=43ms TTL=54
回覆自 1.1.1.1: 位元組=32 時間=66ms TTL=54
回覆自 1.1.1.1: 位元組=32 時間=54ms TTL=54

1.1.1.1 的 Ping 統計資料:
    封包: 已傳送 = 4，已收到 = 4, 已遺失 = 0 (0% 遺失)，
大約的來回時間 (毫秒):
    最小值 = 34ms，最大值 = 66ms，平均 = 49ms

C:\Users\wilso>ping 1.1.1.1

Ping 1.1.1.1 (使用 32 位元組的資料):
回覆自 1.1.1.1: 位元組=32 時間=54ms TTL=54
回覆自 1.1.1.1: 位元組=32 時間=58ms TTL=54
回覆自 1.1.1.1: 位元組=32 時間=60ms TTL=54
回覆自 1.1.1.1: 位元組=32 時間=59ms TTL=54
```

在此圖片中可以看到，如果我們輸入 Domain name「one.one.one.one」時，Domain name 會先被轉換成 IP 才開始動作，這是因為經過 DNS Server 的轉換。以下我們將網路斷連再試一次：

```
C:\Users\wilso>ping one.one.one.one
Ping 要求找不到主機 one.one.one.one。請檢查名稱，然後再試一次。
```

在此變為找不到主機，這是因為無法連線 DNS Server 查詢並轉換地址。

(B)在這邊我們先嘗試 ping 我架在元智宿舍的 Server：

```
C:\Users\wilso>ping 140.138.242.143

Ping 140.138.242.143 (使用 32 位元組的資料):
回覆自 140.138.242.143: 位元組=32 時間<1ms TTL=64
回覆自 140.138.242.143: 位元組=32 時間=1ms TTL=64
回覆自 140.138.242.143: 位元組=32 時間<1ms TTL=64
回覆自 140.138.242.143: 位元組=32 時間<1ms TTL=64

140.138.242.143 的 Ping 統計資料:
    封包: 已傳送 = 4，已收到 = 4, 已遺失 = 0 (0% 遺失)，
大約的來回時間 (毫秒):
    最小值 = 0ms，最大值 = 1ms，平均 = 0ms
```

接著打開元智的 VPN 後，請求失敗：

```
C:\Users\wilso>ping 140.138.242.143

Ping 140.138.242.143 (使用 32 位元組的資料):
要求等候逾時。
要求等候逾時。
要求等候逾時。
要求等候逾時。

140.138.242.143 的 Ping 統計資料:
    封包: 已傳送 = 4，已收到 = 0, 已遺失 = 4 (100% 遺失)，
```

於是我們使用 tracert 指令，得到下列結果：

```
C:\Users\wilso>tracert 140.138.242.143

在上限 30 個躍點上追蹤 140.138.242.143 的路由

  1     *        *        *     要求等候逾時。
  2     2 ms     1 ms     1 ms  140.138.83.254
  3     3 ms     2 ms     4 ms  140.138.191.126
  4     3 ms     2 ms     2 ms  140.138.191.134
  5     *        *        *     要求等候逾時。
  6     *        *        *     要求等候逾時。
  7     *        *        *     要求等候逾時。
  8     *        *        *     要求等候逾時。
  9     *        *        *     要求等候逾時。
 10     *        *        *     要求等候逾時。
 11     *        *        *     要求等候逾時。
 12     *        *        *     要求等候逾時。
 13     *        *        *     要求等候逾時。
 14     *        *        *     要求等候逾時。
 15     *        *        *     要求等候逾時。
```

比對一些其他的結果：

```
C:\Users\wilso>ping sslvpn.yzu.edu.tw

Ping sslvpn.yzu.edu.tw [140.138.83.253] (使用 32 位元組的資料):
回覆自 140.138.83.253: 位元組=32 時間=1ms TTL=61
回覆自 140.138.83.253: 位元組=32 時間=2ms TTL=61
回覆自 140.138.83.253: 位元組=32 時間=1ms TTL=61
回覆自 140.138.83.253: 位元組=32 時間=1ms TTL=61

140.138.83.253 的 Ping 統計資料:
    封包: 已傳送 = 4，已收到 = 4, 已遺失 = 0 (0% 遺失),
大約的來回時間 (毫秒):
    最小值 = 1ms，最大值 = 2ms，平均 = 1ms

C:\Users\wilso>tracert 140.138.83.253

在上限 30 個躍點上追蹤 140.138.83.253 的路由

  1     1 ms    <1 ms    <1 ms  10.138.242.254
  2     1 ms     1 ms     1 ms  140.138.191.133
  3     1 ms     1 ms     1 ms  140.138.191.121
  4     1 ms     1 ms     1 ms  140.138.83.253

追蹤完成。
```

我們發現 Ping VPN 主機可以成功，但是無法經由 Ping 連接宿舍中的 Server；從上面圖片我們也可以看到 VPN 主機 IP 為 140.138.83.253，與上面圖中的一個節點 140.138.83.254 只差一個數字，因此我們猜測兩者位於同樣網段，而 VPN 主機將封包從 140.138.83 網段成功轉送至 140.138.191 網段之後就失敗了，無法成功抵達宿舍區。

以下我們做了一些其他實驗：

```
C:\Users\wilso>ping 140.138.242.143

Ping 140.138.242.143 (使用 32 位元組的資料):
回覆自 140.138.242.143: 位元組=32 時間<1ms TTL=64
回覆自 140.138.242.143: 位元組=32 時間=1ms TTL=64
回覆自 140.138.242.143: 位元組=32 時間<1ms TTL=64
回覆自 140.138.242.143: 位元組=32 時間<1ms TTL=64

140.138.242.143 的 Ping 統計資料:
    封包: 已傳送 = 4，已收到 = 4, 已遺失 = 0 (0% 遺失)，
大約的來回時間 (毫秒):
    最小值 = 0ms，最大值 = 1ms，平均 = 0ms
```

在這裡我用元智校內網路 Ping 自己架在元智宿舍內的 Server，這裡可以看到時間非常短暫。

而我平常也會用 ngrok 服務來讓非校內網路也可連線(透過 ngrok 伺服器當作中繼站，可跨越學校內網架站限制)，於是我也嘗試 ping ngrok 網址：

```
C:\Users\wilso>ping 75d9-61-218-122-10.jp.ngrok.io

Ping 75d9-61-218-122-10.jp.ngrok.io [18.177.0.235] (使用 32 位元組的資料):
回覆自 18.177.0.235: 位元組=32 時間=40ms TTL=224
回覆自 18.177.0.235: 位元組=32 時間=37ms TTL=224
回覆自 18.177.0.235: 位元組=32 時間=37ms TTL=224
回覆自 18.177.0.235: 位元組=32 時間=37ms TTL=224

18.177.0.235 的 Ping 統計資料:
    封包: 已傳送 = 4，已收到 = 4, 已遺失 = 0 (0% 遺失)，
大約的來回時間 (毫秒):
    最小值 = 37ms，最大值 = 40ms，平均 = 37ms
```

可以發現經過時間變長許多，於是我們用 tracert 檢視經過路徑：

```
C:\Users\wilso>tracert 75d9-61-218-122-10.jp.ngrok.io

在上限 30 個躍點上
追蹤 75d9-61-218-122-10.jp.ngrok.io [18.177.53.48] 的路由:

  1     1 ms     1 ms     1 ms  10.138.242.254
  2     2 ms     1 ms     1 ms  140.138.191.133
  3     1 ms     1 ms     1 ms  140.138.191.121
  4     1 ms     1 ms     1 ms  140.138.7.20
  5    12 ms     8 ms     7 ms  61-216-81-254.hinet-ip.hinet.net [61.216.81.254]
  6     4 ms     3 ms     2 ms  tyfo-3301.hinet.net [168.95.209.114]
  7     4 ms     4 ms     4 ms  tylc-3032.hinet.net [220.128.8.154]
  8     *        *        *     要求等候逾時。
  9   418 ms   267 ms   109 ms  220-128-14-149.hinet-ip.hinet.net [220.128.14.149]
 10     3 ms     3 ms     3 ms  211-22-229-105.hinet-ip.hinet.net [211.22.229.105]
 11     7 ms    12 ms     4 ms  ac6.edge2.Banqiao.idc.hinet.net [203.69.110.125]
 12     6 ms     6 ms     6 ms  52.93.136.90
 13     4 ms     3 ms     5 ms  52.93.136.97
 14     *        *        *     要求等候逾時。
 15    36 ms    40 ms    41 ms  52.93.72.139
 16    46 ms    36 ms    55 ms  52.93.72.234
 17    60 ms    59 ms    45 ms  52.93.72.225
 18    39 ms    43 ms    40 ms  15.230.129.185
 19    40 ms    40 ms    43 ms  15.230.154.135
 20     *        *        *     要求等候逾時。
 21     *        *        *     要求等候逾時。
 22     *        *        *     要求等候逾時。
 23     *        *        *     要求等候逾時。
 24     *        *        *     要求等候逾時。
 25     *        *        *     要求等候逾時。
 26     *        *        *     要求等候逾時。
 27     *        *        *     要求等候逾時。
 28    46 ms    36 ms    38 ms  ec2-18-177-53-48.ap-northeast-1.compute.amazonaws.com [18.177.53.48]

追蹤完成。
```

可以看到經過長途的轉送，由此可知使用 ngrok 服務比起校內連線會有一定程度的延遲。

Experiment 3:

（A）以下我們在 Linux 的 Terminal 下指令 ping 1.1.1.1：

```
wilson@wilson:~$ ping 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=53 time=4.21 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=53 time=3.80 ms
64 bytes from 1.1.1.1: icmp_seq=3 ttl=53 time=3.80 ms
64 bytes from 1.1.1.1: icmp_seq=4 ttl=53 time=3.51 ms
64 bytes from 1.1.1.1: icmp_seq=5 ttl=53 time=3.54 ms
64 bytes from 1.1.1.1: icmp_seq=6 ttl=53 time=3.61 ms
64 bytes from 1.1.1.1: icmp_seq=7 ttl=53 time=3.58 ms
64 bytes from 1.1.1.1: icmp_seq=8 ttl=53 time=4.20 ms
64 bytes from 1.1.1.1: icmp_seq=9 ttl=53 time=3.94 ms
64 bytes from 1.1.1.1: icmp_seq=10 ttl=53 time=3.87 ms
```

在 wireshark 中可以追蹤到以下一些封包：

```
192.168.0.174     1.1.1.1           ICMP     98 Echo (ping) request
1.1.1.1           192.168.0.174     ICMP     98 Echo (ping) reply
192.168.0.174     1.1.1.1           ICMP     98 Echo (ping) request
1.1.1.1           192.168.0.174     ICMP     98 Echo (ping) reply
192.168.0.174     1.1.1.1           ICMP     98 Echo (ping) request
1.1.1.1           192.168.0.174     ICMP     98 Echo (ping) reply
192.168.0.174     1.1.1.1           ICMP     98 Echo (ping) request
1.1.1.1           192.168.0.174     ICMP     98 Echo (ping) reply
192.168.0.174     1.1.1.1           ICMP     98 Echo (ping) request
1.1.1.1           192.168.0.174     ICMP     98 Echo (ping) reply
```

這裡我們可以看到 Ping 是使用 ICMP 協定，並且當我們送出一個 request，就會收到相對應的 reply，而 request 和 reply 的 source 和 destination 是相對應（相反）的。以下我們詳細查看封包內容：

```
▸ Frame 356: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on
▾ Ethernet II, Src: AsixElec_49:30:56 (00:0e:c6:49:30:56), Dst: D-LinkIn_
  ▸ Destination: D-LinkIn_d8:f5:98 (10:be:f5:d8:f5:98)
  ▸ Source: AsixElec_49:30:56 (00:0e:c6:49:30:56)
    Type: IPv4 (0x0800)

0000  10 be f5 d8 f5 98 00 0e  c6 49 30 56 08 00 45 00   ········ ·I0V··E·
0010  00 54 29 22 40 00 40 01  4e 2f c0 a8 00 ae 01 01   ·T)"@·@· N/······
0020  01 01 08 00 ab 70 00 02  00 02 84 ff 42 63 00 00   ·····p·· ····Bc··
0030  00 00 b8 55 0e 00 00 00  00 00 10 11 12 13 14 15   ···U···· ········
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ········ ·· !"#$%
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
0060  36 37                                              67
```

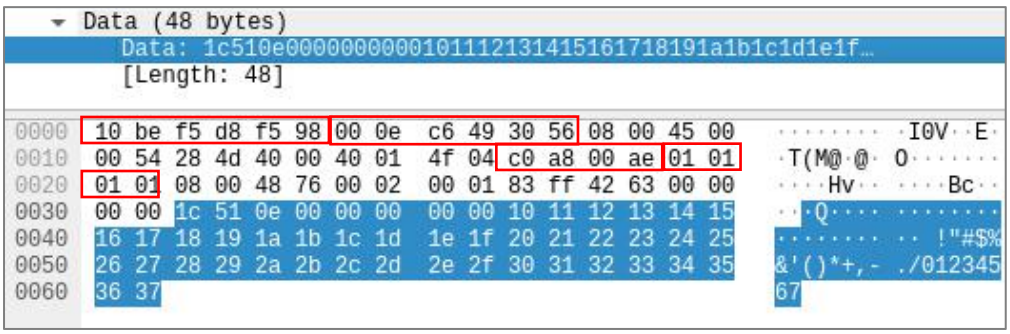
這邊可以看到最前面 12 碼的資料紀錄了目標的 MAC 位址，而接下來的 12 碼（紅框區）是來源的 MAC 位址（因為這是 request，這裡顯示的就是自己電腦的 MAC）。

封包內也有紀錄來源與目標的 IP 位址 （192.168.0.174）：

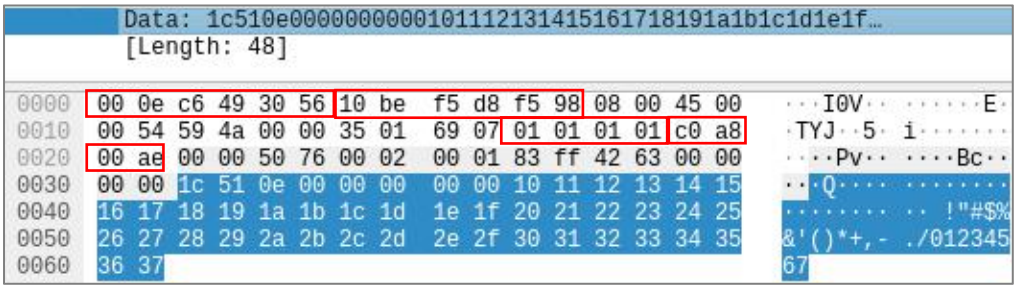
```
    Source: 192.168.0.174
    Destination: 1.1.1.1

0000  10 be f5 d8 f5 98 00 0e  c6 49 30 56 08 00 45 00   ········ ·I0V··E·
0010  00 54 29 22 40 00 40 01  4e 2f c0 a8 00 ae 01 01   ·T)"@·@· N/······
0020  01 01 08 00 ab 70 00 02  00 02 84 ff 42 63 00 00   ·····p·· ····Bc··
```

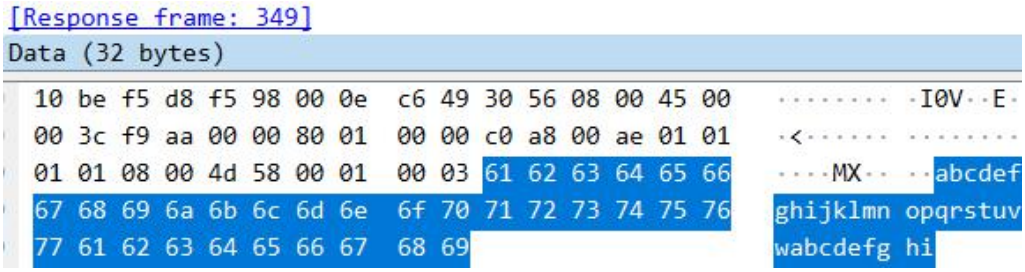
查看 Data 的部份，我們可以發現內容是照著 16 進位數字的順序，如果對照 ascii table，16 進位 21 以後為「!"#$.....」，與轉換成字元後的結果相符：



以下我們查看 ping reply 封包中的 Data，我們可以發現 reply 的內容與 request 完全一樣，但是紅框處的 source 和 destination 與 request 是正好相反的：



補充：如果使用 Windows 中的 Ping 的話，Data 內容會是小寫英文字母順序，與 Linux 中的 ping 不一樣，如下圖顯示：



(B)

```
192.168.0.174          192.168.0.1            DNS
192.168.0.1            192.168.0.174          DNS
```

這邊我們抓到一組 DNS query 和 response 封包，source 和 destination 分別是我的電腦 IP 及 router（DNS server）的 IP。以下我們觀察封包詳細內容：

```
▾ Queries
  ▾ mqtt5.earthquake.tw: type A, class IN
      Name: mqtt5.earthquake.tw
```

由此請求內容可猜測，這應該是電腦中「地牛 Wake up」應用程式連線時，電腦向 DNS server 請求轉換 IP。

網路資料表示，DNS query 使用 port 53，視資料大小使用 UDP 或 TCP 協定（參考資料），這裡資料大小尚未超過 UDP 大小限制，所以採用 UDP 協定傳送：

```
    Protocol: UDP (17)
    Header checksum: 0x6032 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.0.174
    Destination: 192.168.0.1
▾ User Datagram Protocol, Src Port: 39195, Dst Port: 53
    Source Port: 39195
    Destination Port: 53
```

以下我們查看 response 的內容：

```
    Protocol: UDP (17)
    Header checksum: 0xb89c [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.0.1
    Destination: 192.168.0.174
▾ User Datagram Protocol, Src Port: 53, Dst Port: 39195
    Source Port: 53
    Destination Port: 39195
```

```
▾ Answers
  ▾ mqtt5.earthquake.tw: type A, class IN, addr 139.162.122.233
      Name: mqtt5.earthquake.tw
      Type: A (Host Address) (1)
      Class: IN (0x0001)
      Time to live: 30 (30 seconds)
      Data length: 4
      Address: 139.162.122.233
0000  00 0e c6 49 30 56 10 be  f5 d8 f5 98 08 00 45 00   ···I0V·· ·····E·
0010  00 51 00 00 40 00 40 11  b8 9c c0 a8 00 01 c0 a8   ·Q··@·@· ········
0020  00 ae 00 35 99 1b 00 3d  87 dc a5 4c 80 80 00 01   ···5···= ···L····
0030  00 01 00 00 00 00 05 6d  71 74 74 35 0a 65 61 72   ·······m qtt5·ear
0040  74 68 71 75 61 6b 65 02  74 77 00 00 01 00 01 c0   thquake· tw·····
0050  0c 00 01 00 01 00 00 00  1e 00 04 8b a2 7a e9      ········ ···z·
```

DNS server 使用的 port 依舊為 53，協定依舊是 UDP（未超過大小限制）。回應中多了 Answers 區段紀錄該網址相對應的 IP。

（C）

一、ICMP

```
Source           Destination     Prot  Length Info
49.216.89.103    192.168.0.174   ICMP      70 Destination unreachable (Port unreachable)
220.133.16.107   192.168.0.174   ICMP      70 Destination unreachable (Port unreachable)
49.217.174.77    192.168.0.174   ICMP      70 Destination unreachable (Port unreachable)
114.36.33.218    192.168.0.174   ICMP      70 Destination unreachable (Port unreachable)
220.133.16.107   192.168.0.174   ICMP      70 Destination unreachable (Port unreachable)
49.217.174.77    192.168.0.174   ICMP      70 Destination unreachable (Port unreachable)
114.36.33.218    192.168.0.174   ICMP      70 Destination unreachable (Port unreachable)
220.133.16.107   192.168.0.174   ICMP      70 Destination unreachable (Port unreachable)
114.41.35.151    192.168.0.174   ICMP      70 Destination unreachable (Port unreachable)
49.217.174.77    192.168.0.174   ICMP      70 Destination unreachable (Port unreachable)
114.36.33.218    192.168.0.174   ICMP      70 Destination unreachable (Port unreachable)
220.133.16.107   192.168.0.174   ICMP      70 Destination unreachable (Port unreachable)
114.41.35.151    192.168.0.174   ICMP      70 Destination unreachable (Port unreachable)
49.217.174.77    192.168.0.174   ICMP      70 Destination unreachable (Port unreachable)
114.36.33.218    192.168.0.174   ICMP      70 Destination unreachable (Port unreachable)
220.133.16.107   192.168.0.174   ICMP      70 Destination unreachable (Port unreachable)
114.41.35.151    192.168.0.174   ICMP      70 Destination unreachable (Port unreachable)
114.36.33.218    192.168.0.174   ICMP      70 Destination unreachable (Port unreachable)
220.133.16.107   192.168.0.174   ICMP      70 Destination unreachable (Port unreachable)
114.41.35.151    192.168.0.174   ICMP      70 Destination unreachable (Port unreachable)
114.36.33.218    192.168.0.174   ICMP      70 Destination unreachable (Port unreachable)
220.133.16.107   192.168.0.174   ICMP      70 Destination unreachable (Port unreachable)
114.41.35.151    192.168.0.174   ICMP      70 Destination unreachable (Port unreachable)
114.36.33.218    192.168.0.174   ICMP      70 Destination unreachable (Port unreachable)
220.133.16.107   192.168.0.174   ICMP      70 Destination unreachable (Port unreachable)
114.41.35.151    192.168.0.174   ICMP      70 Destination unreachable (Port unreachable)
114.36.33.218    192.168.0.174   ICMP      70 Destination unreachable (Port unreachable)
```

ICMP 為「網際控制訊息協定」，可以用來偵測網路狀況及回報錯誤，當 Gateway 或是 router 發現某個封包轉送次數過多（TTL 值已為 0），無法成功抵達目的，則會回傳一個 ICMP 封包給來源裝置，回報錯誤訊息。在這邊我們可以看到許多 ICMP 封包，皆為其他裝置向我的電腦回報錯誤（destination 為 192.168.0.174）。我們挑選幾個來檢視內容：

```
▾ Internet Control Message Protocol
    Type: 3 (Destination unreachable)
    Code: 3 (Port unreachable)
    Checksum: 0xa058 [correct]
    [Checksum Status: Good]
    Unused: 00000000
  ▸ Internet Protocol Version 4, Src: 192.168.0.174, Dst: 220.133.16.107
  ▾ User Datagram Protocol, Src Port: 47393, Dst Port: 60756
      Source Port: 47393
      Destination Port: 60756
```

```
▾ Internet Control Message Protocol
    Type: 3 (Destination unreachable)
    Code: 3 (Port unreachable)
    Checksum: 0xd44f [correct]
    [Checksum Status: Good]
    Unused: 00000000
  ▸ Internet Protocol Version 4, Src: 192.168.0.174, Dst: 49.217.174.77
  ▾ User Datagram Protocol, Src Port: 47393, Dst Port: 59162
      Source Port: 47393
      Destination Port: 59162
```

```
▾ Internet Control Message Protocol
    Type: 3 (Destination unreachable)
    Code: 3 (Port unreachable)
    Checksum: 0x2527 [correct]
    [Checksum Status: Good]
    Unused: 00000000
  ▸ Internet Protocol Version 4, Src: 192.168.0.174, Dst: 114.41.35.151
  ▾ User Datagram Protocol, Src Port: 47393, Dst Port: 51803
      Source Port: 47393
      Destination Port: 51803
```

封包內 ICMP 區段紀錄的錯誤資訊表示，這些都是從本機 port 47393 對於不同目標請求失敗的回報訊息，於是引起我們好奇心，想要了解為什麼這些封包都是從 port 47393 發送。

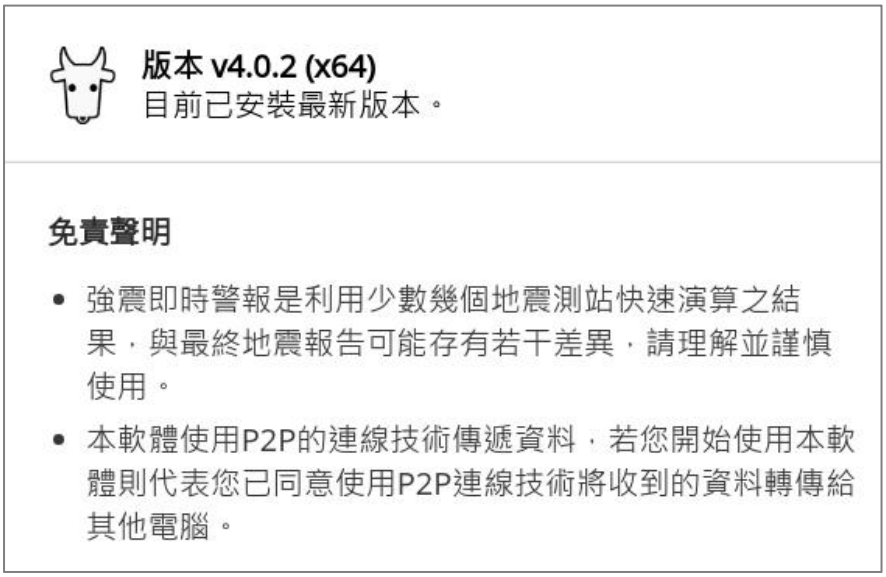
以下我們可以看到許多從 port 47393 使用 UDP 協定發送封包的紀錄：

192.168.0.174	192.168.225.116	UDP	114 47393 → 61262 Len=72
192.168.0.174	49.217.174.77	UDP	114 47393 → 59162 Len=72
192.168.0.174	192.168.43.227	UDP	114 47393 → 64784 Len=72
192.168.0.174	49.216.89.103	UDP	114 47393 → 54134 Len=72
192.168.0.174	192.168.1.182	UDP	114 47393 → 60107 Len=72
192.168.0.174	111.242.253.45	UDP	114 47393 → 60107 Len=72
192.168.0.174	192.168.1.163	UDP	114 47393 → 65468 Len=72
192.168.0.174	122.116.67.29	UDP	114 47393 → 65468 Len=72
192.168.0.174	192.168.100.3	UDP	114 47393 → 63811 Len=72
192.168.0.174	123.194.11.98	UDP	114 47393 → 11311 Len=72
192.168.0.174	172.105.211.118	UDP	106 47393 → 41417 Len=64
192.168.0.174	172.105.211.118	UDP	106 47393 → 41417 Len=64
192.168.0.174	192.168.1.109	UDP	114 47393 → 49891 Len=72
192.168.0.174	111.252.62.78	UDP	114 47393 → 49891 Len=72
192.168.0.174	192.168.1.109	UDP	114 47393 → 60384 Len=72
192.168.0.174	114.39.195.161	UDP	114 47393 → 60384 Len=72
192.168.0.174	192.168.5.105	UDP	114 47393 → 56193 Len=72
192.168.0.174	114.47.44.83	UDP	114 47393 → 56193 Len=72

我們在 Linux 中用 netstat 指令搭配 grep 來篩選檢視 port 使用狀況：

```
wilson@wilson:~$ netstat -a -p | grep 47393
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
udp        0      0 0.0.0.0:47393           0.0.0.0:*                           2248/oxwu --hidden
```

這裡可以看到 port 47393 被行程 2248 佔用（名為 oxwu 的應用）。由此可知這些封包也來自「地牛 Wake Up」這個軟體：



官方文件表示軟體使用 P2P 交換地震資料，我們猜測這些 UDP 封包為使用 P2P 技術連線時，向其他使用者交換資料的結果。而封包轉送失敗時就會收到 ICMP 封包回報錯誤訊息。

二、ARP

解釋：每台設備中都存在一個 ARP Table，用來查詢每個 IP 相對應的 MAC 位址，如果可以在 ARP table(cache)中查詢到，就會直接將 Mac 位址填入封包中並進行轉送；但如果目前沒有資料，設備就會向其他裝置廣播 ARP 封包詢問某 IP 位址對應之 MAC 位址。以下為實驗截圖：

AsixElec_49:30:…	Broadcast	ARP	42 Who has 192.168.0.150? Tell 192.168.0.174
AsixElec_49:30:…	Broadcast	ARP	42 Who has 192.168.0.103? Tell 192.168.0.174
AsixElec_49:30:…	Broadcast	ARP	42 Who has 192.168.0.150? Tell 192.168.0.174
AsixElec_49:30:…	Broadcast	ARP	42 Who has 192.168.0.103? Tell 192.168.0.174
AsixElec_49:30:…	Broadcast	ARP	42 Who has 192.168.0.103? Tell 192.168.0.174
D-LinkIn_d8:f5:…	AsixElec_49:30:…	ARP	60 Who has 192.168.0.174? Tell 192.168.0.1
AsixElec_49:30:…	D-LinkIn_d8:f5:…	ARP	42 192.168.0.174 is at 00:0e:c6:49:30:56
AsixElec_49:30:…	Broadcast	ARP	42 Who has 192.168.0.12? Tell 192.168.0.174
AsixElec_49:30:…	Broadcast	ARP	42 Who has 192.168.0.12? Tell 192.168.0.174
AsixElec_49:30:…	Broadcast	ARP	42 Who has 192.168.0.12? Tell 192.168.0.174
AsixElec_49:30:…	Broadcast	ARP	42 Who has 192.168.0.17? Tell 192.168.0.174
AsixElec_49:30:…	Broadcast	ARP	42 Who has 192.168.0.17? Tell 192.168.0.174

從此截圖中可以看到電腦向其他裝置廣播 ARP 封包請求「who has...Tell 192.168.0.174」。也可以看見接收到 router 發出的 ARP 封包、還有本機的回應（紅框處）。

以下我們做了兩個實驗，第一個是有找到該主機位址(未知主機 1)，第二則是找不到的(未知主機 2)。在一開始我們 ping 了未知主機的 IP 位址，然後使用 wireahark 擷取 ARP 的封包：

```
C:\Users\user>ping 192.168.100.30

Ping 192.168.100.30 (使用 32 位元組的資料):
要求等候逾時。
要求等候逾時。
要求等候逾時。
要求等候逾時。

192.168.100.30 的 Ping 統計資料:
    封包: 已傳送 = 4，已收到 = 0, 已遺失 = 4 (100% 遺失)，
```

以下兩個封包，第一個是從我主機(Mac address: 3C-7C-3F-59-11-0E)廣播出去找尋 IP 位址為 192.168.100.30 的未知主機 1。而第二則則是找到後，未知主機 2 回覆我他的 Mac address 為 40:ee:15:a9:73:31。

ASUSTekC_59:11:0e	Broadcast	ARP	42 Who has 192.168.100.30? Tell 192.168.100.21
ZioncomE_a9:73:31	ASUSTekC_59:11:0e	ARP	60 192.168.100.30 is at 40:ee:15:a9:73:31

接著我們 ping 了另一個未知主機的 IP，然後也用 wireshark 擷取 ARP 封包：

```
C:\Users\user>ping 192.168.100.150

Ping 192.168.100.150 (使用 32 位元組的資料):
回覆自 192.168.100.21: 目的地主機無法連線。
回覆自 192.168.100.21: 目的地主機無法連線。
回覆自 192.168.100.21: 目的地主機無法連線。
回覆自 192.168.100.21: 目的地主機無法連線。

192.168.100.150 的 Ping 統計資料:
    封包: 已傳送 = 4，已收到 = 4, 已遺失 = 0 (0% 遺失)，
```

下圖可以看到，本機為了尋找 IP 192.168.100.150 相對應的 MAC 位址，所以一直向外廣播，但無法得到該主機的回覆：

```
Source              Destination         Protocol  Length  Info
ASUSTekC_59:11:0e   Broadcast           ARP          42 Who has 192.168.100.150? Tell 192.168.100.21
ASUSTekC_59:11:0e   Broadcast           ARP          42 Who has 192.168.100.150? Tell 192.168.100.21
ASUSTekC_59:11:0e   Broadcast           ARP          42 Who has 192.168.100.150? Tell 192.168.100.21
ASUSTekC_59:11:0e   Broadcast           ARP          42 Who has 192.168.100.150? Tell 192.168.100.21
ASUSTekC_59:11:0e   Broadcast           ARP          42 Who has 192.168.100.150? Tell 192.168.100.21
ASUSTekC_59:11:0e   Broadcast           ARP          42 Who has 192.168.100.150? Tell 192.168.100.21
ASUSTekC_59:11:0e   Broadcast           ARP          42 Who has 192.168.100.150? Tell 192.168.100.21
ASUSTekC_59:11:0e   Broadcast           ARP          42 Who has 192.168.100.150? Tell 192.168.100.21
ASUSTekC_59:11:0e   Broadcast           ARP          42 Who has 192.168.100.150? Tell 192.168.100.21
ASUSTekC_59:11:0e   Broadcast           ARP          42 Who has 192.168.100.150? Tell 192.168.100.21
```

下圖為該封包的內容，可以看到目的為廣播時，內容會填入「ff:ff:ff:ff:ff:ff」，而來源則是本機之 MAC 位址：

```
> Frame 3: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_{9052AB87-C6DD-4110-82B0-9658D3E9F641}, id 0
v Ethernet II, Src: ASUSTekC_59:11:0e (3c:7c:3f:59:11:0e), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  > Source: ASUSTekC_59:11:0e (3c:7c:3f:59:11:0e)
    Type: ARP (0x0806)
> Address Resolution Protocol (request)

0000  ff ff ff ff ff ff 3c 7c  3f 59 11 0e 08 06 00 01   ······<| ?Y······
0010  08 00 06 04 00 01 3c 7c  3f 59 11 0e c0 a8 64 15   ······<| ?Y····d·
0020  00 00 00 00 00 00 c0 a8  64 96                     ········ d·
```