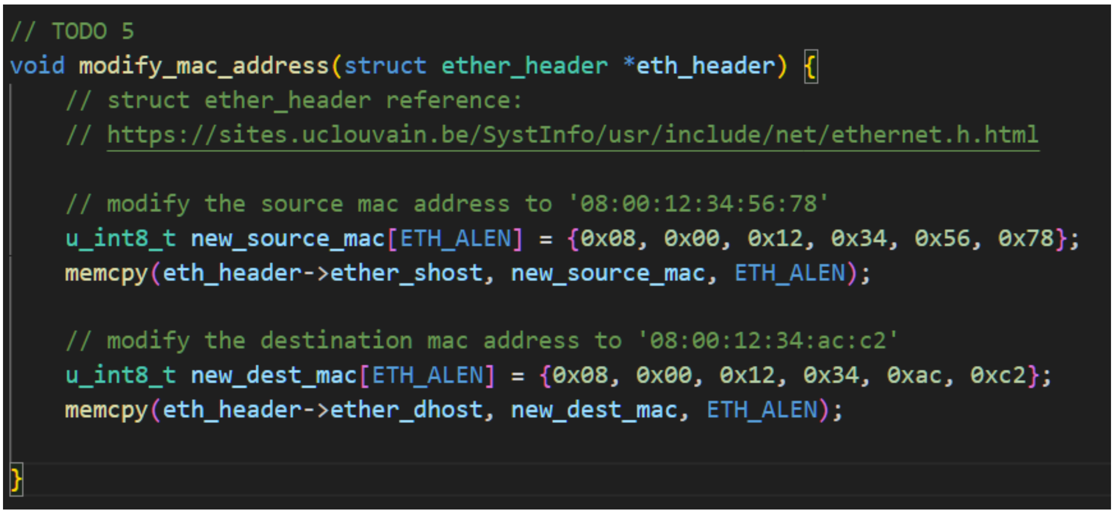
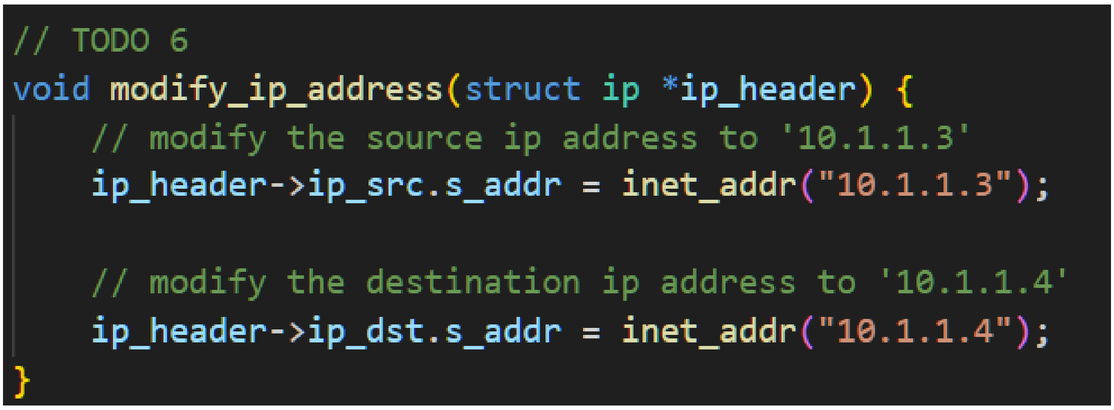
Function Explanations



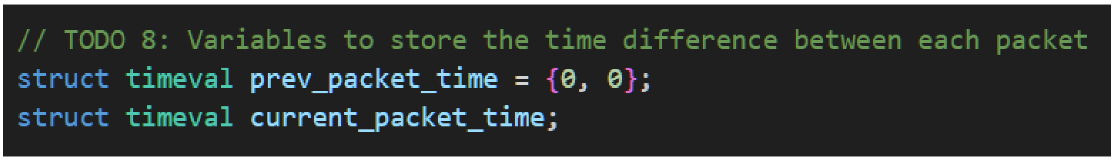
```c
// TODO 5
void modify_mac_address(struct ether_header *eth_header) {
    // struct ether_header reference:
    // https://sites.uclouvain.be/SystInfo/usr/include/net/ethernet.h.html

    // modify the source mac address to '08:00:12:34:56:78'
    u_int8_t new_source_mac[ETH_ALEN] = {0x08, 0x00, 0x12, 0x34, 0x56, 0x78};
    memcpy(eth_header->ether_shost, new_source_mac, ETH_ALEN);

    // modify the destination mac address to '08:00:12:34:ac:c2'
    u_int8_t new_dest_mac[ETH_ALEN] = {0x08, 0x00, 0x12, 0x34, 0xac, 0xc2};
    memcpy(eth_header->ether_dhost, new_dest_mac, ETH_ALEN);

}
```

為了修改 source mac address，我先將 **'08:00:12:34:56:78'** 存入 new_source_mac，再用 ‘memcpy’ 將 new_source_mac 中的資料複製到 eth_header->ether_shost (shost 代表 source eth addr)。修改 destination mac address 也用相似方式，先將 **'08:00:12:34:ac:c2'** 存入 new_dest_mac，再用 ‘memcpy’ 將 new_dest_mac 中的資料複製到 eth_header->ether_dhost (dhost 代表 destination eth addr)。



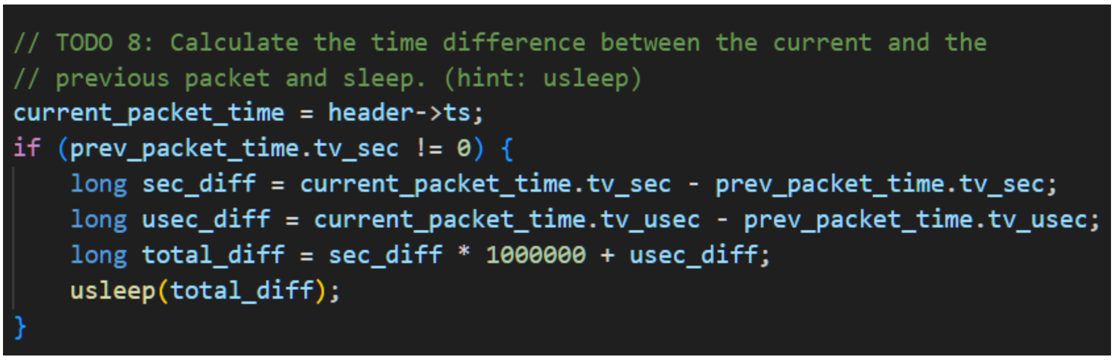
```c
// TODO 6
void modify_ip_address(struct ip *ip_header) {
    // modify the source ip address to '10.1.1.3'
    ip_header->ip_src.s_addr = inet_addr("10.1.1.3");

    // modify the destination ip address to '10.1.1.4'
    ip_header->ip_dst.s_addr = inet_addr("10.1.1.4");
}
```

要更動 ip address，我們需要用 ‘inet_addr’ 將人類可讀的 IP 位址轉為機器可以處理的格式，並分別存入 ip_header 中的 ip_src.s_addr 和 ip_dst.s_addr。



```c
// TODO 8: Variables to store the time difference between each packet
struct timeval prev_packet_time = {0, 0};
struct timeval current_packet_time;
```

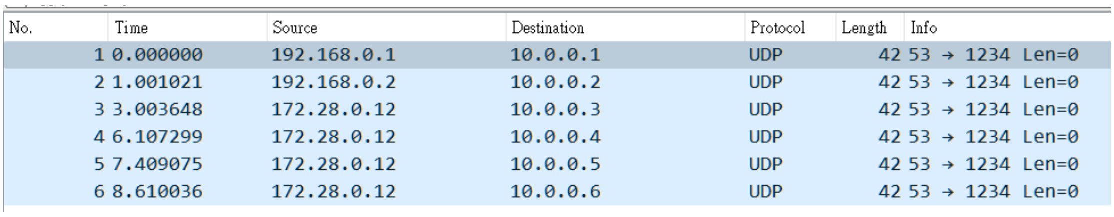
一開始把 prev_packet_time 初始化為 {0, 0}。



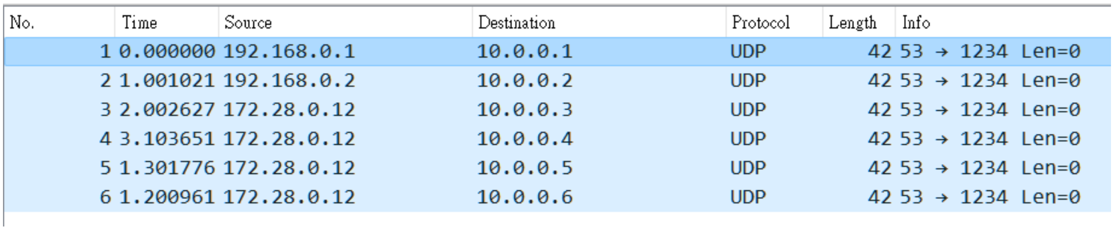
```
// TODO 8: Calculate the time difference between the current and the
// previous packet and sleep. (hint: usleep)
current_packet_time = header->ts;
if (prev_packet_time.tv_sec != 0) {
    long sec_diff = current_packet_time.tv_sec - prev_packet_time.tv_sec;
    long usec_diff = current_packet_time.tv_usec - prev_packet_time.tv_usec;
    long total_diff = sec_diff * 1000000 + usec_diff;
    usleep(total_diff);
}
```

每一輪把一個 packet 中 timestamp 存入 current_packet_time，並檢查 prev_packet_time 是否為零(是零的話則為第一個封包)，若 prev_packet_time 不為零(非第一個封包)，則 ”此封包與前一個封包的時間差” 需與 pcap file 中相同。為了計算時間差，我將 current_packet_time 減去 prev_packet_time (struct timeval 內有兩種時間單位不同的紀錄、須分別計算)， 並把單位都換為 microsecond。最後用 ‘usleep’ 這個 function 延遲送出修改後的封包。

Outcome Screenshot with test.pcap

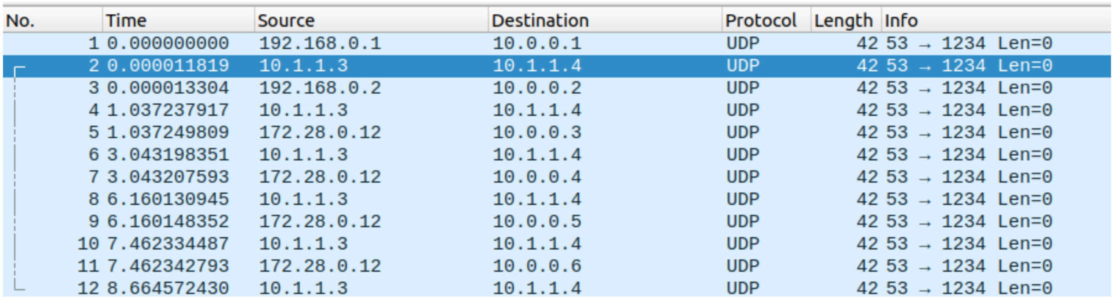


No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.1	10.0.0.1	UDP	42	53 → 1234 Len=0
2	1.001021	192.168.0.2	10.0.0.2	UDP	42	53 → 1234 Len=0
3	3.003648	172.28.0.12	10.0.0.3	UDP	42	53 → 1234 Len=0
4	6.107299	172.28.0.12	10.0.0.4	UDP	42	53 → 1234 Len=0
5	7.409075	172.28.0.12	10.0.0.5	UDP	42	53 → 1234 Len=0
6	8.610036	172.28.0.12	10.0.0.6	UDP	42	53 → 1234 Len=0



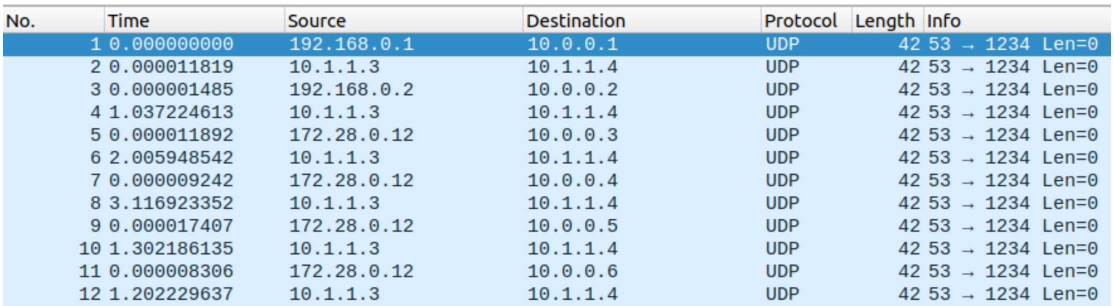
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.1	10.0.0.1	UDP	42	53 → 1234 Len=0
2	1.001021	192.168.0.2	10.0.0.2	UDP	42	53 → 1234 Len=0
3	2.002627	172.28.0.12	10.0.0.3	UDP	42	53 → 1234 Len=0
4	3.103651	172.28.0.12	10.0.0.4	UDP	42	53 → 1234 Len=0
5	1.301776	172.28.0.12	10.0.0.5	UDP	42	53 → 1234 Len=0
6	1.200961	172.28.0.12	10.0.0.6	UDP	42	53 → 1234 Len=0

Original packets in test.pcap (timestamp=[0, 1, 3, 6, 7, 8], time_diff=[1.001, 2.002, 3.103, 1.301, 1.200])



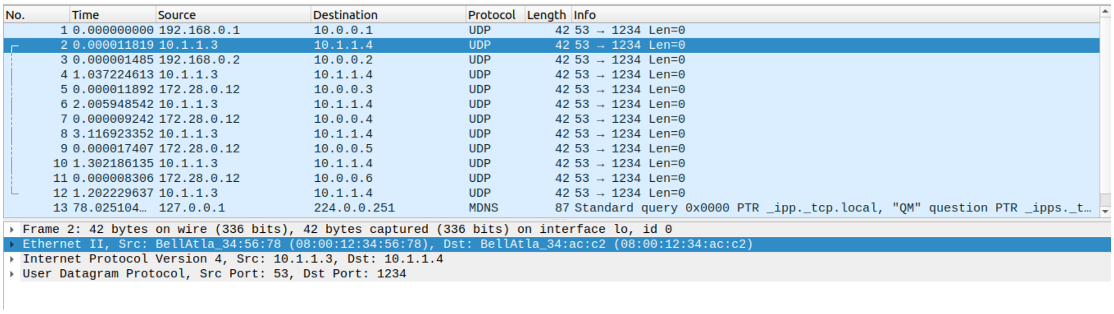
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.0.1	10.0.0.1	UDP	42	53 → 1234 Len=0
2	0.000011819	10.1.1.3	10.1.1.4	UDP	42	53 → 1234 Len=0
3	0.000013304	192.168.0.2	10.0.0.2	UDP	42	53 → 1234 Len=0
4	1.037237917	10.1.1.3	10.1.1.4	UDP	42	53 → 1234 Len=0
5	1.037249809	172.28.0.12	10.0.0.3	UDP	42	53 → 1234 Len=0
6	3.043198351	10.1.1.3	10.1.1.4	UDP	42	53 → 1234 Len=0
7	3.043207593	172.28.0.12	10.0.0.4	UDP	42	53 → 1234 Len=0
8	6.160130945	10.1.1.3	10.1.1.4	UDP	42	53 → 1234 Len=0
9	6.160148352	172.28.0.12	10.0.0.5	UDP	42	53 → 1234 Len=0
10	7.462334487	10.1.1.3	10.1.1.4	UDP	42	53 → 1234 Len=0
11	7.462342793	172.28.0.12	10.0.0.6	UDP	42	53 → 1234 Len=0
12	8.664572430	10.1.1.3	10.1.1.4	UDP	42	53 → 1234 Len=0

Wireshark (View/Time Display Format = Second Since Beginning of Capture) 顯示的第 2、4、6、8、10、12 個紀錄為修改後的封包，那些封包的 timestamp 秒數與 test.pcap 的 timestamp 相符。



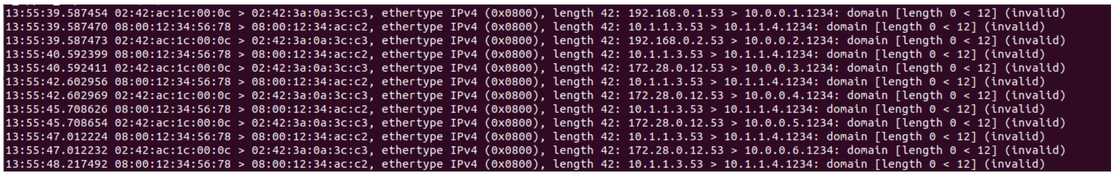
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.0.1	10.0.0.1	UDP	42	53 → 1234 Len=0
2	0.000011819	10.1.1.3	10.1.1.4	UDP	42	53 → 1234 Len=0
3	0.000001485	192.168.0.2	10.0.0.2	UDP	42	53 → 1234 Len=0
4	1.037224613	10.1.1.3	10.1.1.4	UDP	42	53 → 1234 Len=0
5	0.000011892	172.28.0.12	10.0.0.3	UDP	42	53 → 1234 Len=0
6	2.005948542	10.1.1.3	10.1.1.4	UDP	42	53 → 1234 Len=0
7	0.000009242	172.28.0.12	10.0.0.4	UDP	42	53 → 1234 Len=0
8	3.116923352	10.1.1.3	10.1.1.4	UDP	42	53 → 1234 Len=0
9	0.000017407	172.28.0.12	10.0.0.5	UDP	42	53 → 1234 Len=0
10	1.302186135	10.1.1.3	10.1.1.4	UDP	42	53 → 1234 Len=0
11	0.000008306	172.28.0.12	10.0.0.6	UDP	42	53 → 1234 Len=0
12	1.202229637	10.1.1.3	10.1.1.4	UDP	42	53 → 1234 Len=0

Wireshark (View/Time Display Format = Second Since Previous Displayed Packet) 顯示的時間差與 test.pcap 誤差都在 0.1s 以內。|1.037-1.001|=0.036 < 0.1、|2.005-2.002|=0.003 < 0.1、|3.116-3.103|=0.013 < 0.1、|1.302-1.301|=0.001 < 0.1、|1.202-1.200|=0.002 < 0.1 五次誤差皆小於 0.1。

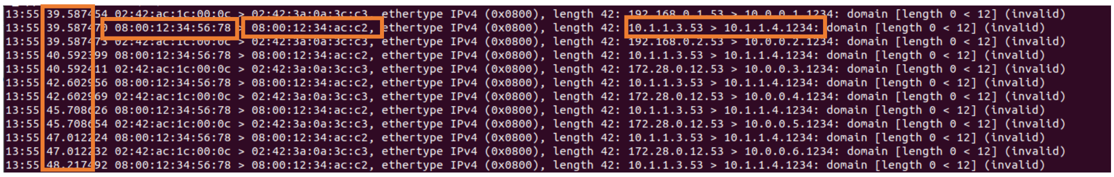


```
No.   Time        Source        Destination   Protocol Length Info
    1 0.000000000 192.168.0.1   10.0.0.1      UDP        42 53 → 1234 Len=0
    2 0.000011819 10.1.1.3      10.1.1.4      UDP        42 53 → 1234 Len=0
    3 0.000001485 192.168.0.2   10.0.0.2      UDP        42 53 → 1234 Len=0
    4 1.037224613 10.1.1.3      10.1.1.4      UDP        42 53 → 1234 Len=0
    5 0.000011892 172.28.0.12   10.0.0.3      UDP        42 53 → 1234 Len=0
    6 2.005948542 10.1.1.3      10.1.1.4      UDP        42 53 → 1234 Len=0
    7 0.000009242 172.28.0.12   10.0.0.4      UDP        42 53 → 1234 Len=0
    8 3.116923352 10.1.1.3      10.1.1.4      UDP        42 53 → 1234 Len=0
    9 0.000017407 172.28.0.12   10.0.0.5      UDP        42 53 → 1234 Len=0
   10 1.302186135 10.1.1.3      10.1.1.4      UDP        42 53 → 1234 Len=0
   11 0.000008306 172.28.0.12   10.0.0.6      UDP        42 53 → 1234 Len=0
   12 1.202229637 10.1.1.3      10.1.1.4      UDP        42 53 → 1234 Len=0
   13 78.025104…  127.0.0.1     224.0.0.251   MDNS       87 Standard query 0x0000 PTR _ipp._tcp.local, "QM" question PTR _ipps._t…

▸ Frame 2: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface lo, id 0
▸ Ethernet II, Src: BellAtla_34:56:78 (08:00:12:34:56:78), Dst: BellAtla_34:ac:c2 (08:00:12:34:ac:c2)
▸ Internet Protocol Version 4, Src: 10.1.1.3, Dst: 10.1.1.4
▸ User Datagram Protocol, Src Port: 53, Dst Port: 1234
```

第二個封包的 mac address 也有依作業要求將 source mac address 修改為 08:00:12:34:56:78 和將 destination mac address 修改為 08:00:12:34:ac:c2。



```
13:55:39.587454 02:42:ac:1c:00:0c > 02:42:3a:0a:3c:c3, ethertype IPv4 (0x0800), length 42: 192.168.0.1.53 > 10.0.0.1.1234: domain [length 0 < 12] (invalid)
13:55:39.587470 08:00:12:34:56:78 > 08:00:12:34:ac:c2, ethertype IPv4 (0x0800), length 42: 10.1.1.3.53 > 10.1.1.4.1234: domain [length 0 < 12] (invalid)
13:55:39.587473 02:42:ac:1c:00:0c > 02:42:3a:0a:3c:c3, ethertype IPv4 (0x0800), length 42: 192.168.0.2.53 > 10.0.0.2.1234: domain [length 0 < 12] (invalid)
13:55:40.592399 08:00:12:34:56:78 > 08:00:12:34:ac:c2, ethertype IPv4 (0x0800), length 42: 10.1.1.3.53 > 10.1.1.4.1234: domain [length 0 < 12] (invalid)
13:55:40.592411 02:42:ac:1c:00:0c > 02:42:3a:0a:3c:c3, ethertype IPv4 (0x0800), length 42: 172.28.0.12.53 > 10.0.0.3.1234: domain [length 0 < 12] (invalid)
13:55:42.602956 08:00:12:34:56:78 > 08:00:12:34:ac:c2, ethertype IPv4 (0x0800), length 42: 10.1.1.3.53 > 10.1.1.4.1234: domain [length 0 < 12] (invalid)
13:55:42.602969 02:42:ac:1c:00:0c > 02:42:3a:0a:3c:c3, ethertype IPv4 (0x0800), length 42: 172.28.0.12.53 > 10.0.0.4.1234: domain [length 0 < 12] (invalid)
13:55:45.708626 08:00:12:34:56:78 > 08:00:12:34:ac:c2, ethertype IPv4 (0x0800), length 42: 10.1.1.3.53 > 10.1.1.4.1234: domain [length 0 < 12] (invalid)
13:55:45.708654 02:42:ac:1c:00:0c > 02:42:3a:0a:3c:c3, ethertype IPv4 (0x0800), length 42: 172.28.0.12.53 > 10.0.0.5.1234: domain [length 0 < 12] (invalid)
13:55:47.012224 08:00:12:34:56:78 > 08:00:12:34:ac:c2, ethertype IPv4 (0x0800), length 42: 10.1.1.3.53 > 10.1.1.4.1234: domain [length 0 < 12] (invalid)
13:55:47.012232 02:42:ac:1c:00:0c > 02:42:3a:0a:3c:c3, ethertype IPv4 (0x0800), length 42: 172.28.0.12.53 > 10.0.0.6.1234: domain [length 0 < 12] (invalid)
13:55:48.217492 08:00:12:34:56:78 > 08:00:12:34:ac:c2, ethertype IPv4 (0x0800), length 42: 10.1.1.3.53 > 10.1.1.4.1234: domain [length 0 < 12] (invalid)
```



```
13:55:39.587454 02:42:ac:1c:00:0c > 02:42:3a:0a:3c:c3, ethertype IPv4 (0x0800), length 42: 192.168.0.1.53 > 10.0.0.1.1234: domain [length 0 < 12] (invalid)
13:55:39.587470 08:00:12:34:56:78 > 08:00:12:34:ac:c2, ethertype IPv4 (0x0800), length 42: 10.1.1.3.53 > 10.1.1.4.1234: domain [length 0 < 12] (invalid)
13:55:39.587473 02:42:ac:1c:00:0c > 02:42:3a:0a:3c:c3, ethertype IPv4 (0x0800), length 42: 192.168.0.2.53 > 10.0.0.2.1234: domain [length 0 < 12] (invalid)
13:55:40.592399 08:00:12:34:56:78 > 08:00:12:34:ac:c2, ethertype IPv4 (0x0800), length 42: 10.1.1.3.53 > 10.1.1.4.1234: domain [length 0 < 12] (invalid)
13:55:40.592411 02:42:ac:1c:00:0c > 02:42:3a:0a:3c:c3, ethertype IPv4 (0x0800), length 42: 172.28.0.12.53 > 10.0.0.3.1234: domain [length 0 < 12] (invalid)
13:55:42.602956 08:00:12:34:56:78 > 08:00:12:34:ac:c2, ethertype IPv4 (0x0800), length 42: 10.1.1.3.53 > 10.1.1.4.1234: domain [length 0 < 12] (invalid)
13:55:42.602969 02:42:ac:1c:00:0c > 02:42:3a:0a:3c:c3, ethertype IPv4 (0x0800), length 42: 172.28.0.12.53 > 10.0.0.4.1234: domain [length 0 < 12] (invalid)
13:55:45.708626 08:00:12:34:56:78 > 08:00:12:34:ac:c2, ethertype IPv4 (0x0800), length 42: 10.1.1.3.53 > 10.1.1.4.1234: domain [length 0 < 12] (invalid)
13:55:45.708654 02:42:ac:1c:00:0c > 02:42:3a:0a:3c:c3, ethertype IPv4 (0x0800), length 42: 172.28.0.12.53 > 10.0.0.5.1234: domain [length 0 < 12] (invalid)
13:55:47.012224 08:00:12:34:56:78 > 08:00:12:34:ac:c2, ethertype IPv4 (0x0800), length 42: 10.1.1.3.53 > 10.1.1.4.1234: domain [length 0 < 12] (invalid)
13:55:47.012232 02:42:ac:1c:00:0c > 02:42:3a:0a:3c:c3, ethertype IPv4 (0x0800), length 42: 172.28.0.12.53 > 10.0.0.6.1234: domain [length 0 < 12] (invalid)
13:55:48.217492 08:00:12:34:56:78 > 08:00:12:34:ac:c2, ethertype IPv4 (0x0800), length 42: 10.1.1.3.53 > 10.1.1.4.1234: domain [length 0 < 12] (invalid)
```

Tcpdump (timestamp=[39.587, 39.587, 39.587, 40.592, 40.592, 42.602, 42.602, 45.708, 45.708, 47.012, 47.012, 48.217], time_diff=[1.005, 2.010, 3.106, 1.304, 1.205]) 與原始 時間差[1.001, 2.002, 3.103, 1.301, 1.200] 誤差都在 0.1s 以內。