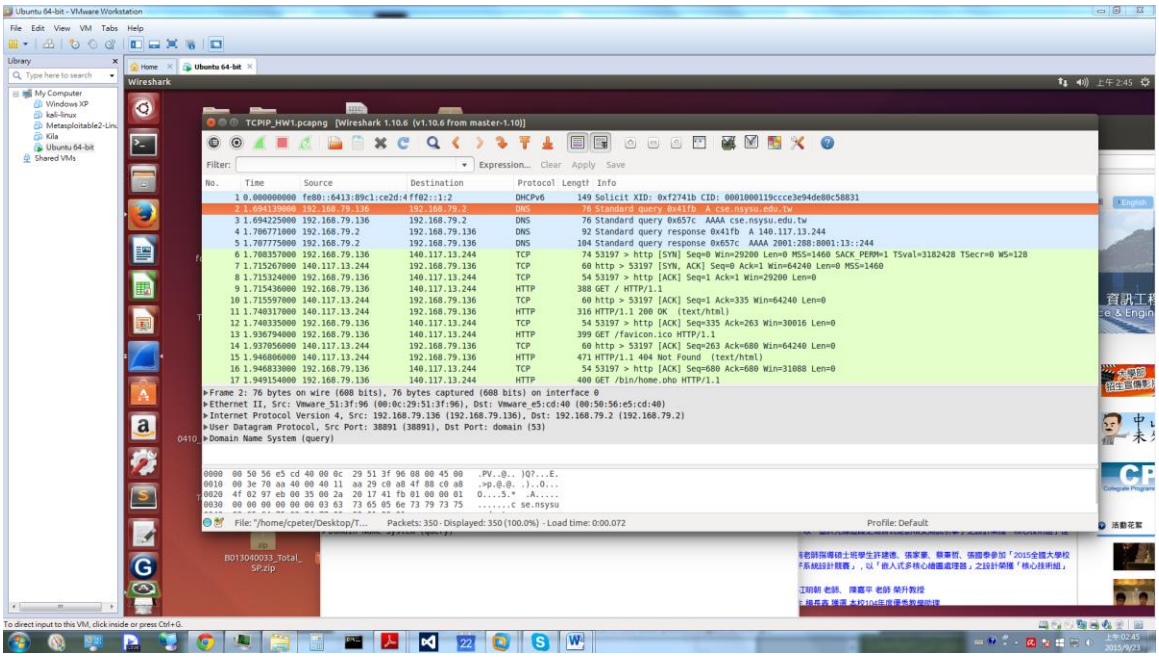
高等電腦網路
HW1

B013040033
雷皓博

環境 VMware中的Ubuntu 14.04



1. Find the first DNS request packet sent by the client.(Request for **cse.nsysu.edu.tw**)

You can find a record like below on Wireshark. And you can answer the question.

(1) Examine the Ethernet

a. What is the Ethernet address of the source and destination?

```
▶Frame 2: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface 0
▼Ethernet II, Src: Vmware_51:3f:96 (00:0c:29:51:3f:96), Dst: Vmware_e5:cd:40 (00:50:56:e5:cd:40)
  ▶Destination: Vmware_e5:cd:40 (00:50:56:e5:cd:40)
  ▶Source: Vmware_51:3f:96 (00:0c:29:51:3f:96)
```

b. What is the content of the type field in the Ethernet frame?

```
Type: IP (0x0800)
```

(2) Examine the Internet Protocol

a. What is the IP address of the source and destination?

```
Internet Protocol Version 4, Src: 192.168.79.136 (192.168.79.136), Dst: 192.168.79.2 (192.168.79.2)
```

b. What is the header length? What is the total packet length?

```
Header length: 20 bytes
Total Length: 62
```

c. Identify the protocol type field. What is the number and type of the protocol in the payload?

```
▼Flags: 0x02 (Don't Fragment)
    0... .... = Reserved bit: Not set
    .1.. .... = Don't fragment: Set
    ..0. .... = More fragments: Not set
  Fragment offset: 0
  Time to live: 64
  Protocol: UDP (17)
```

(3) Examine the User Datagram Protocol
a. Identify the client ephemeral port number and the server well-known port number .

```
User Datagram Protocol, Src Port: 38891 (38891), Dst Port: domain (53)
```

b. What type of application layer protocol is in the payload?
DNS

(4) Examine the Domain Name System (query)
a. What field indicates whether the message is a query or a response?

```
Flags: 0x0100 Standard query
```

```
0000  00 50 56 e5 cd 40 00 0c  29 51 3f 96 08 00 45 00   .PV..@.. )Q?...E.
0010  00 3e 70 aa 40 00 40 11  aa 29 c0 a8 4f 88 c0 a8   .>p.@.@. .)..O...
0020  4f 02 97 eb 00 35 00 2a  20 17 41 fb 01 00 00 01   O....5.*  .A.....
0030  00 00 00 00 00 00 03 63  73 65 05 6e 73 79 73 75   .......c se.nsysu
0040  03 65 64 75 02 74 77 00  00 01 00 01              .edu.tw. ....
```

b. What is the query transaction ID?

```
Transaction ID: 0x41fb
```

c. Identify the fields that carry the type and class of the query.

DNS class. 'IN' refers to 'Internet'
Type:A IPv4 address record, Returns a 32-bit IP address , which typically maps a domain's hostname to an IP address

```
▶cse.nsysu.edu.tw: type A, class IN
```

```
0000  00 50 56 e5 cd 40 00 0c  29 51 3f 96 08 00 45 00   .PV..@.. )Q?...E.
0010  00 3e 70 aa 40 00 40 11  aa 29 c0 a8 4f 88 c0 a8   .>p.@.@. .)..O...
0020  4f 02 97 eb 00 35 00 2a  20 17 41 fb 01 00 00 01   O....5.*  .A.....
0030  00 00 00 00 00 00 03 63  73 65 05 6e 73 79 73 75   ......c se.nsysu
0040  03 65 64 75 02 74 77 00  00 01 00 01              .edu.tw. ....
```

2. Find the DNS response packet which is response to the DNS request packet from the above question. You can find a record like below on Wireshark. And you can answer the question.
(1) Examine the Ethernet.
a. What is the Ethernet address of the source and destination?

```
Ethernet II, Src: Vmware_e5:cd:40 (00:50:56:e5:cd:40), Dst: Vmware_51:3f:96 (00:0c:29:51:3f:96)
```

b. What is the content of the type field in the Ethernet frame?

```
Type: IP (0x0800)
```

(2) Examine the Internet Protocol & Domain Name System (response)
a. What is the IP address of the source and destination?

```
Internet Protocol Version 4, Src: 192.168.79.2 (192.168.79.2), Dst: 192.168.79.136 (192.168.79.136)
```

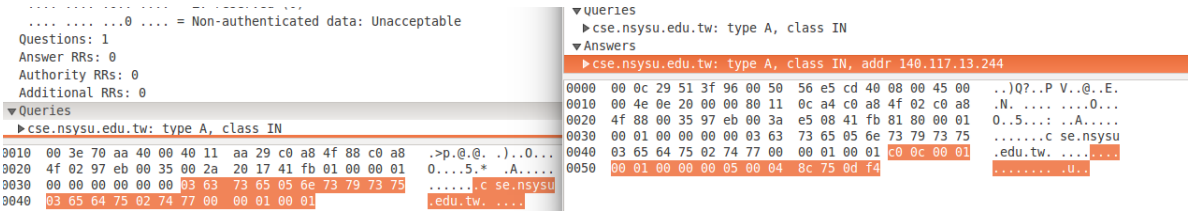
b. What is the header length? What is the total packet length? Is it longer than the query?

```
Header length: 20 bytes
Total Length: 78
```

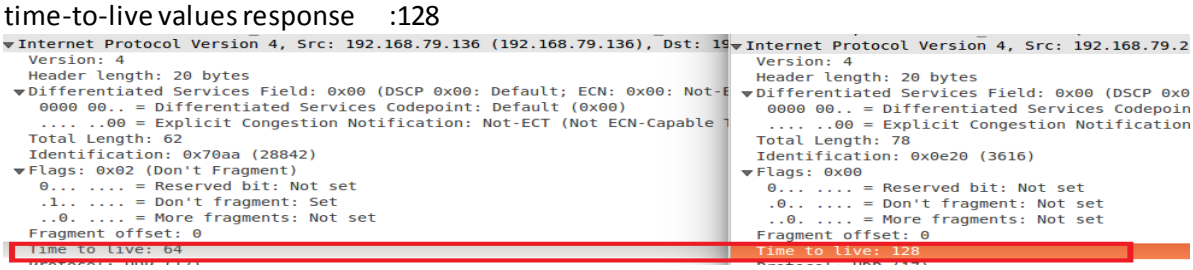
Yes,78>62.

c. How many answers are provided in the response message? Compare the answers and their time-to-live values.

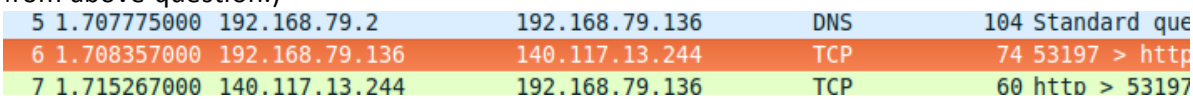
1 answers



time-to-live values query :64
time-to-live values response :128



3. Find the first TCP packet sent by client. (The destination IP address is response from above question.)



You can find three record like below on Wireshark. It's TCP three-way handshake.

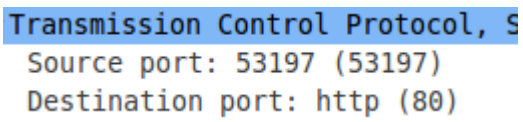
Figure: TCP three-way handshake

(1) Examine the Transmission Control Protocol.

a. What are the ephemeral port number used by the client and the well-known port number used by the server?

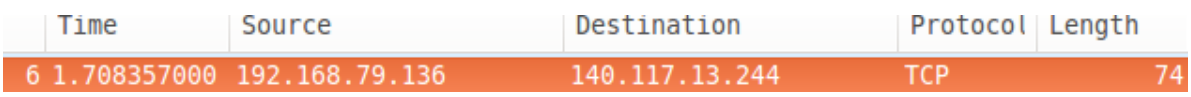
ephemeral port number used by the client :53197

well-known port number used by the server :80

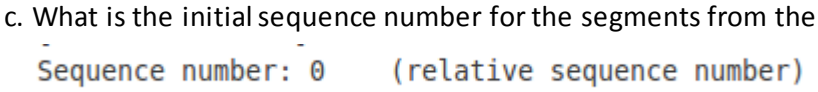


b. What is the length of the TCP segment?

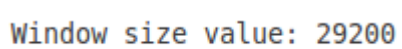
74



c. What is the initial sequence number for the segments from the client to the server?



d. What is the initial window size?



e. What is the maximum segment size?

```
Options: (20 bytes), Maximum segment size,
▶Maximum segment size: 1460 bytes
```

f. Find the hex character that contains the SYN flag bit

```
▼Flags: 0x002 (SYN)
    000. .... .... = Reserved: Not set
    ...0 .... .... = Nonce: Not set
    .... 0... .... = Congestion Window Reduced (CWR): Not set
    .... .0.. .... = ECN-Echo: Not set
    .... ..0. .... = Urgent: Not set
    .... ...0 .... = Acknowledgment: Not set
    .... .... 0... = Push: Not set
    .... .... .0.. = Reset: Not set
  ▶ .... .... ..1. = Syn: Set
    .... .... ...0 = Fin: Not set
0000  00 50 56 e5 cd 40 00 0c  29 51 3f 96 08 00 45 00   .PV..@.. )Q?...E.
0010  00 3c ae 14 40 00 40 06  e2 0d c0 a8 4f 88 8c 75   .<..@.@. ....O..u
0020  0d f4 cf cd 00 50 ab 4f  21 48 00 00 00 00 a0 02   .....P.O !H.....
```

Part 2 Probing the Internet (ICMP, PING, Traceroute)

Protocol Analysis Questions

To answer the following questions, start Wireshark and open the packet capture file created above.

1. Ping Captured.

(1) Find the first ICMP Echo Request packet.

a. First, examine the Internet Protocol. What is the Time-to-Live?

```
Time to live: 64
Protocol: ICMP (1)
```

Time To Live 指一個封包在經過一個網路時，可傳遞的最長距離（躍點數）。

b. Next examine the Internet Control Message Protocol. What is the ICMP message type?

```
Internet Control Message Protocol
Type: 8 (Echo (ping) request)
```

c. What is the message identifier and sequence number?

Identifier

```
0f 67
```

sequence numbe

```
00 01
```

BE(big endian),LE(little endian)

```
Identifier (BE): 3943 (0x0f67)
Identifier (LE): 26383 (0x670f)
Sequence number (BE): 1 (0x0001)
Sequence number (LE): 256 (0x0100)
```

(2) Find the first ICMP Echo Reply packet.

a. Now examine the Internet Control Message Protocol. What is the ICMP message type?

```
Internet Control Message Protocol
Type: 0 (Echo (ping) reply)
```

2. Traceroute Captured.

```
cpeter@ubuntu: ~
peter@ubuntu:~$ sudo traceroute -q 1 -I 8.8.8.8
raceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
1  192.168.79.2 (192.168.79.2)  0.257 ms
2  192.168.1.1 (192.168.1.1)  1.262 ms
3  h254.s98.ts.hinet.net (168.95.98.254)  5.948 ms
4  nkn1-3311.hinet.net (168.95.220.214)  5.901 ms
5  SKC1-3011.hinet.net (220.128.24.142)  8.694 ms
6  TNE1-3011.hinet.net (220.128.24.29)  7.587 ms
7  TNE1-3301.hinet.net (220.128.26.193)  6.561 ms
8  74.125.49.50 (74.125.49.50)  13.119 ms
9  209.85.243.30 (209.85.243.30)  16.352 ms
0  209.85.240.153 (209.85.240.153)  16.287 ms
1  209.85.247.57 (209.85.247.57)  15.475 ms
2  *
```

(1) Find the first ICMP Echo Request packet.

a. Examine the Internet Protocol. What are the source and destination addresses?

```
Internet Protocol Version 4, Src: 192.168.79.136 (192.168.79.136), Dst: 8.8.8.8 (8.8.8.8)
```

b. What are the protocol type and the Time-to-Live in the IP packet?

```
▼Time to live: 1
  ▶[Expert Info (Note/Sequence): "Time To Live" only 1]
  Protocol: ICMP (1)
```

c. Next, examine the Internet Control Message Protocol. What is the ICMP message type?

```
Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
```

What are the message identifier and sequence number?

BE(big endian),LE(little endian)

```
  Identifier (BE): 3216 (0x0c90)
  Identifier (LE): 36876 (0x900c)
  Sequence number (BE): 1 (0x0001)
  Sequence number (LE): 256 (0x0100)
```

(2) Find an ICMP Time-to-live exceeded packet.

a. Examine the Internet Protocol. What are the source and destination addresses?

```
3 0.000243000 192.168.79.2 192.168.79.136 ICMP 102 Time-to-live exceeded (Time to live exceeded in transit)
▶Frame 3: 102 bytes on wire (816 bits), 102 bytes captured (816 bits) on interface 0
▶Ethernet II, Src: Vmware_e5:cd:40 (00:50:56:e5:cd:40), Dst: Vmware_51:3f:96 (00:0c:29:51:3f:96)
▼Internet Protocol Version 4, Src: 192.168.79.2 (192.168.79.2), Dst: 192.168.79.136 (192.168.79.136)
```

b. Next, examine the Internet Control Message Protocol. What is the ICMP message type?

```
Internet Control Message Protocol
  Type: 11 (Time-to-live exceeded)
```

Part 3 Measuring Network Bandwidth

Server IP : 140.117.171.226

1. Measure the bandwidth for Transmission Control Protocol

Type “iperf -c 140.117.171.226 -t 10 -i 2”

```
cpeter@ubuntu:~$ iperf -c 140.117.171.226 -t 10 -i 2
------------------------------------------------------------
Client connecting to 140.117.171.226, TCP port 5001
TCP window size: 85.0 KByte (default)
------------------------------------------------------------
[  3] local 192.168.79.136 port 35047 connected with 140.117.171.226 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 2.0 sec  1.00 MBytes  4.19 Mbits/sec
[  3]  2.0- 4.0 sec  0.00 Bytes  0.00 bits/sec
[  3]  4.0- 6.0 sec  0.00 Bytes  0.00 bits/sec
[  3]  6.0- 8.0 sec  0.00 Bytes  0.00 bits/sec
[  3]  8.0-10.0 sec  0.00 Bytes  0.00 bits/sec
[  3]  0.0-10.2 sec  1.12 MBytes   926 Kbits/sec
cpeter@ubuntu:~$
```

2. Adjust the window size for Transmission Control Protocol. See what’s different.

Type “iperf -c 140.117.171.226 -w 2000 -t 10 -i 2”

```
cpeter@ubuntu:~$ iperf -c 140.117.171.226 -w 2000 -t 10 -i 2
WARNING: TCP window size set to 2000 bytes. A small window size
will give poor performance. See the Iperf documentation.
------------------------------------------------------------
Client connecting to 140.117.171.226, TCP port 5001
TCP window size: 4.50 KByte (WARNING: requested 1.95 KByte)
------------------------------------------------------------
[  3] local 192.168.79.136 port 35048 connected with 140.117.171.226 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 2.0 sec   512 KBytes  2.10 Mbits/sec
[  3]  2.0- 4.0 sec   128 KBytes   524 Kbits/sec
[  3]  4.0- 6.0 sec  0.00 Bytes  0.00 bits/sec
[  3]  6.0- 8.0 sec  0.00 Bytes  0.00 bits/sec
[  3]  8.0-10.0 sec  0.00 Bytes  0.00 bits/sec
[  3]  0.0-17.0 sec   768 KBytes   370 Kbits/sec
cpeter@ubuntu:~$
```

3. Measure the bandwidth for User Datagram Protocol

Type “iperf -c 140.117.171.226 -u -t 10 -i 2”

```
cpeter@ubuntu:~$ iperf -c 140.117.171.226 -u -t 10 -i 2
------------------------------------------------------------
Client connecting to 140.117.171.226, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 192.168.79.136 port 54849 connected with 140.117.171.226 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 2.0 sec   257 KBytes  1.05 Mbits/sec
[  3]  2.0- 4.0 sec   256 KBytes  1.05 Mbits/sec
[  3]  4.0- 6.0 sec   256 KBytes  1.05 Mbits/sec
[  3]  6.0- 8.0 sec   257 KBytes  1.05 Mbits/sec
[  3]  8.0-10.0 sec   256 KBytes  1.05 Mbits/sec
[  3]  0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec
[  3] Sent 893 datagrams
[  3] WARNING: did not receive ack of last datagram after 10 tries.
cpeter@ubuntu:~$
```

4. Adjust the bandwidth for User Datagram Protocol.Measure the package lost rate or any else happened.

Type “iperf -c 140.117.171.226 -u -t 10 -i 2 -b 512G”

```
cpeter@ubuntu:~$ iperf -c 140.117.171.226 -u -t 10 -i 2 -b 512G
------------------------------------------------------------
Client connecting to 140.117.171.226, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 192.168.79.136 port 48995 connected with 140.117.171.226 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 2.0 sec   181 MBytes   761 Mbits/sec
[  3]  2.0- 4.0 sec   182 MBytes   765 Mbits/sec
[  3]  4.0- 6.0 sec   180 MBytes   755 Mbits/sec
[  3]  6.0- 8.0 sec   179 MBytes   751 Mbits/sec
[  3]  8.0-10.0 sec   179 MBytes   753 Mbits/sec
[  3]  0.0-10.0 sec   903 MBytes   757 Mbits/sec
[  3] Sent 643885 datagrams
[  3] WARNING: did not receive ack of last datagram after 10 tries.
cpeter@ubuntu:~$
```