CSC358 Wireshark Assignment 1 Solution

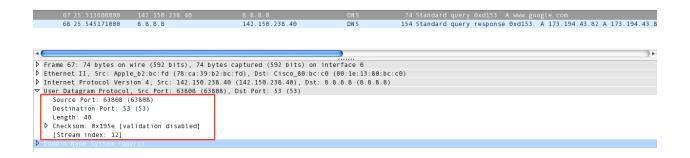
Part I. UDP

1. Select one UDP packet from your trace. From this packet, determine how many fields there are in the UDP header. (You shouldn't look in the textbook! Answer these questions directly from what you observe in the packet trace.) Name these fields.

Solution:

UDP header contains 4 fields:

1. source port; 2. destination port; 3. length; 4. checksum



2. By consulting the displayed information in Wireshark's packet content field for this packet, determine the length (in bytes) of each of the UDP header fields.

Solution:

The UDP header has a fixed length of 8 bytes. Each of these 4 header fields is 2 bytes long.

```
| Frame 67: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
| Ethernet II, Src: Apple_b2:bc:rd (78:ca:39:b2:bc:rd), Dst: Cisco_80:bc:c0 (00:le:13:80:bc:c0)
| Internet Protocol Version 4, Src: 142.150.238.40 (142.150.238.40), Dst: 8.8.8.8 (8.8.8.8)
| User Datagram Protocol, Src Port: 63808 (63808)
| Destination Port: 53 (53)
| Length: 40
| Destination Port: 53 (53)
| Length: 40
| Destination Name System (query)
| Domain Name System (query)
| Domain Name System (query)
| Domain Name System (query)
```

3. The value in the Length field is the length of what? (You can consult the text for this answer). What is the length of UDP payload for your selected packet.

Solution:

The length field specifies the number of bytes in the UDP segment (header plus data). An explicit length value is needed since the size of the data field may differ from one UDP segment to the next.

The length of UDP payload for selected packet is 32 bytes. 40 bytes - 8 bytes = 32 bytes.

```
▼ User Datagram Protocol, Src Port: 63808 (63808), Dst Port: 53 (53)

Source Port: 63808 (63808)

■ Destination Port: 53 (53)

Length: 40

□ Checksum. 0x195e [validation disabled]

[Stream index: 12]

▶ Domain Name System (query)
```

4. What is the maximum number of bytes that can be included in a UDP payload? (Hint: the answer to this question can be determined by your answer to 2. above)

Solution:

The maximum number of bytes that can be included in a UDP payload is $(2^16 - 1)$ bytes plus the header bytes. This gives 65535 bytes - 8 bytes = 65527 bytes.

5. What is the largest possible source port number?

Solution:

The largest possible source port number is $(2^{16} - 1) = 65535$.

6. What is the protocol number for UDP? Give your answer in both hexadecimal and decimal notation. (To answer this question, you'll need to look into the IP header.)

Solution:

The IP protocol number for UDP is 0x11 hex, which is 17 in decimal value.

```
▼ Internet Protocol Version 4, Src: 142.150.238.40 (142.150.238.40), Dst: 8.8.8.8 (8.8.8.8)
     Version: 4
     Header Length: 20 bytes
   Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
     Total Length: 60
     Identification: 0xa129 (41257)
   ▶ Flags: 0x00
     Fragment offset: 0
     Time to live: 64
   ▶ Header checksum: 0x4cb9 [validation disabled]
     Source: 142.150.238.40 (142.150.238.40)
     Destination: 8.8.8.8 (8.8.8.8)
     [Source GeoIP: Unknown]
0000 00 1e 13 80 bc c0 78 ca 39 b2 bc fd 08 00 45 00 0010 00 3c a1 29 00 00 40 11 4c b9 8e 96 ee 28 08 08 0020 08 08 f9 40 00 35 00 28 19 5e d1 53 01 00 00 01
                                                                    ....x. 9....E
0020 08 08 f9 40 00 35 00 28 19 5e d1 53 01 00 00 01 0030 00 00 00 00 00 00 03 77 77 77 06 67 6f 6f 67 6c
                                                                    ....w ww.googl
                                                                     e.com... ..
0040 65 03 63 6f 6d 00 00 01 00 01
```

7. Examine a pair of UDP packets in which the first packet is sent by your host and the second packet is a reply to the first packet. Describe the relationship between the port numbers in the two packets.

Solution:

The source port of the UDP packet sent by the host is the same as the destination port of the reply packet, and conversely the destination port of the UDP packet sent by the host is the same as the source port of the reply packet.

```
Frame 67: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
▶ Ethernet II, Src: Apple_b2:bc:fd (78:ca:39:b2:bc:fd), Dst: Cisco_80:bc:c0 (00:1e:13:80:bc:c0)
▶ Internet Protocol Version 4, Src: 142.150.238.40 (142.150.238.40), Dst: 8.8.8.8 (8.8.8.8)
▽ User Datagram Protocol, Src Port: 63808 (63808), Dst Port: 53 (53)
    Source Port: 63808 (63808)
    Destination Port: 53 (53)
    Length: 40

    ▷ Checksum: 0x195e [validation disabled]

    [Stream index: 12]
Domain Name System (query)
Frame 68: 154 bytes on wire (1232 bits), 154 bytes captured (1232 bits) on interface 0
▶ Ethernet II, Src: Cisco_80:bc:c0 (00:1e:13:80:bc:c0), Dst: Apple_b2:bc:fd (78:ca:39:b2:bc:fd)
▶ Internet Protocol Version 4, Src: 8.8.8.8 (8.8.8.8), Dst: 142.150.238.40 (142.150.238.40)
▽ User Datagram Protocol, Src Port: 53 (53), Dst Port: 63808 (63808)
    Source Port: 53 (53)
    Destination Port: 63808 (63808)
    Length: 120

    ▷ Checksum: 0x949b [validation disabled]

    [Stream index: 12]
Domain Name System (response)
```

Part II. TCP

1. What is the IP address of the client (the initiator of this TCP connection), and what is the server's IP address? From which port the client initiates the connection, and what is the port number used for this connection on the server side?

Solution:

Client:

IP address: 192.168.1.140

Port: 57678

```
▶ Ethernet II, Src: AsustekC_b3:01:84 (00:1d:60:b3:01:84), Dst: Actionte_2f:47:87 (00:26:62:2f:47:87)
▶ Internet Protocol Version 4, Src: 192.168.1.140 (192.168.1.140), Dst: 174.143.213.184 (174.143.213.184)
▽ Transmission Control Protocol, Src Port: 57678 (57678), Dst Port: 80 (80), Seq: 0, Len: 0
    Source Port: 57678 (57678)
Destination Port: 80 (80)
    [Stream index: 01
    [TCP Segment Len: 0]
    Sequence number: 0
                          (relative sequence number)
    Acknowledgment number: 0
    Header Length: 40 bytes
  Window size value: 5840
    [Calculated window size: 5840]
  ▶ Checksum: 0x8f47 [validation disabled]
    Urgent pointer: 0
  D Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale
```

Server:

IP address: 174.143.213.184

Port: 80

```
Frame 2: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
▶ Ethernet II, Src: Actionte_2f:47:87 (00:26:62:2f:47:87), Dst: AsustekC_b3:01:84 (00:1d:60:b3:01:84)
D Internet Protocol Version 4, Src: 174.143.213.184 (174.143.213.184) Dst: 192.168.1.140 (192.168.1.140)
  Transmission Control Protocol, Src Port: 80 (80), Dst Port: 57678 (57678), Seq: 0, Ack: 1, Len: 0
    Source Port: 80 (80)
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 0
                         (relative sequence number)
    Acknowledgment number: 1
                              (relative ack number)
    Header Length: 40 bytes
  Window size value: 5792
    [Calculated window size: 5792]
  D Checksum: 0x3e7c [validation disabled]
    Urgent pointer: 0
  DOptions: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale
  [SEQ/ACK analysis]
```

2. Roughly speaking, what is the RTT of this connection? (just need a rough number, answer within ± 10 ms will be fine). Roughly how many RTT does it take (in total) for this TCP connection to establish, tranfer data and close?

Solution:

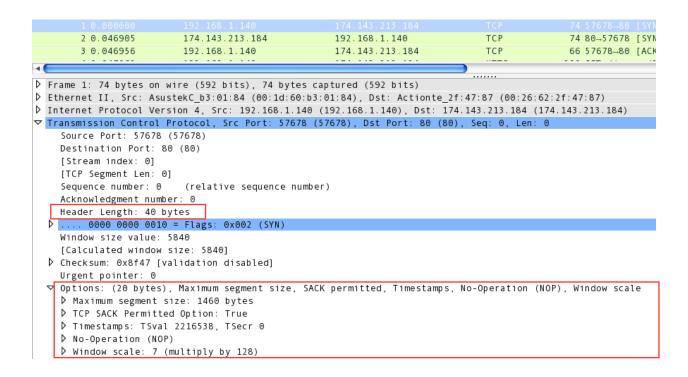
The RTT is approximately 47ms. It takes 5 RTT before the connect closes.

	1 0.000000				74 57678⊸80 [SYN] Seq=0 Win=5840 Len=0 M
	2 0.046905	174.143.213.184	192.168.1.140	TCP	74 80→57678 [SYN, ACK] Seq=0 Ack=1 Win=5
	3 0.046956	192.168.1.140	174.143.213.184	TCP	66 57678→80 [ACK] Seq=1 Ack=1 Win=5888 L
	4 0.047068	192.168.1.140	174.143.213.184	HTTP	200 GET /images/layout/logo.png HTTP/1.0

3. During the handshaking of this connection, what is the length of the TCP header? Is this TCP header the basic one? If not, what is the optional field(s) in the TCP header.

Solution:

The header length is 40 bytes in handshaking stage. The TCP header is not just the basic one, it has a few optional header fields: Maximum segment size, SACK permitted, Timestamps, NOP, Window scale.



4. Answer 3 after the handshaking stage. What is the length of maximum TCP payload of a packet after this tcp connection is established?

Solution:

The header length is 32 bytes in established stage. It has Timestamps and 2 NOP as optional headers.

The maximum length of payload of a packet in established mode is 1448 bytes (1500 - 20 IP header -32 TCP header).

```
TCP
        1 0.000000
                          192.168.1.140
                                                     174.143.213.184
                                                                                               74 57678→80
        2 0.046905
                          174.143.213.184
                                                     192.168.1.140
                                                                                TCP
                                                                                               74 80→57678
Frame 3: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
▶ Ethernet II, Src: AsustekC_b3:01:84 (00:1d:60:b3:01:84), Dst: Actionte_2f:47:87 (00:26:62:2f:47:87)
D Internet Protocol Version 4, Src: 192.168.1.140 (192.168.1.140), Dst: 174.143.213.184 (174.143.213.184)
▽ Transmission Control Protocol, Src Port: 57678 (57678), Dst Port: 80 (80), Seq: 1, Ack: 1, Len: 0
    Source Port: 57678 (57678)
    Destination Port: 80 (80)
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 1
                          (relative sequence number)
    Acknowledgment number: 1
                                (relative ack number)
    Header Length: 32 bytes
    .... 0000 0001 0000 = Flags: 0x010 (ACK)
    Window size value: 46
    [Calculated window size: 5888]
    [Window size scaling factor: 128]
  D Checksum: 0x83b4 [validation disabled]
    Urgent pointer: 0
  マ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
    No-Operation (NOP)
    No-Operation (NOP)
    D Timestamps: TSval 2216543, TSecr 835172936
  D [SEQ/ACK analysis]
```

5. How many HTTP data the server sends to the client during the 2nd RTT, 3rd RTT and 4th RTT (respectively)?

Solution:

2nd RTT: 3 packets * 1448 bytes HTTP data each packet = 4344 bytes

3rd RTT: 6 packets * 1448 bytes HTTP data each packet = 8688 bytes

4th RTT: 6 packets * 1448 bytes + 325 bytes = 9013 bytes

5 0.094268	174.143.213.184	192.168.1.140	TCP
6 0.096673	174.143.213.184	192.168.1.140	TCP
7 0.096702	192.168.1.140	174.143.213.184	TCP
8 0.096785	174.143.213.184	192.168.1.140	TCP
9 0.096789	192.168.1.140	174.143.213.184	TCP
10 0.100001	174.143.213.184	192.168.1.140	TCP
11 0.100023	192.168.1.140	174.143.213.184	TCP
L . /			
12/0.144237	174.143.213.184	192.168.1.140	TCP
13 0.144263	192.168.1.140	174.143.213.184	TCP
14 0.147063	174.143.213.184	192.168.1.140	TCP
15 0.147072	192.168.1.140	174.143.213.184	TCP
16.8.147184	174.143.213.184	192.168.1.140	TCP
17 0.147192	192.168.1.140	174.143.213.184	TCP
18 0.149480	174.143.213.184	192.168.1.140	TCP
19 0.149488	192.168.1.140	174.143.213.184	TCP
20 0.149603	174.143.213.184	192.168.1.140	TCP
21 0.149611	192.168.1.140	174.143.213.184	TCP
22 0.149726	174.143.213.184	192.168.1.140	TCP
23 0.149735	192.168.1.140	174.143.213.184	TCP
24 0.191646	174.143.213.184	192.168.1.140	TCP
25 0.191658	192.168.1.140	174.143.213.184	TCP
26.0.194472	174.143.213.184	192.168.1.140	TCP
27 0.194481	192.168.1.140	174.143.213.184	TCP
28 0. 196601	174.143.213.184	192.168.1.140	TCP
29 0.196609	192.168.1.140	174.143.213.184	TCP
30 0.196723	174.143.213.184	192.168.1.140	TCP
31 0.196731	192.168.1.140	174.143.213.184	TCP
(32 0.197342	174.143.213.184	192.168.1.140	TCP
33 0.197350	192.168.1.140	174.143.213.184	TCP
34 0.199928	174.143.213.184	192.168.1.140	TCP
35 0.199936	192.168.1.140	174.143.213.184	TCP
36 0 19 9950	174.143.213.184	192.168.1.140	HTTP
37 0.199955	192.168.1.140	174.143.213.184	TCP
38 0.200580	192.168.1.140	174.143.213.184	TCP

6. What is the initial buffer size (window size) advertised by the client? Has the buffer size on the client side ever become the bottleneck during the transferring of data? (Be careful when answer this question, show how you get your conclusion).

Solution:

The client's initial window size is 5840 bytes. The client's window size has never become the bottleneck. In the 2^{nd} RTT, the client's window size is 5840 bytes from the server's perspective, and the server only sends 3 * 1448 bytes = 4344 bytes < 5840 bytes. In the 3^{rd} RTT, the client's window size is 14592 bytes (the last advertised window size by client in the 2^{nd} RTT), and the server only sends 6 * 1448 bytes = 8688 bytes < 14592 bytes. In the 4^{th} RTT the client's window size is even larger.

