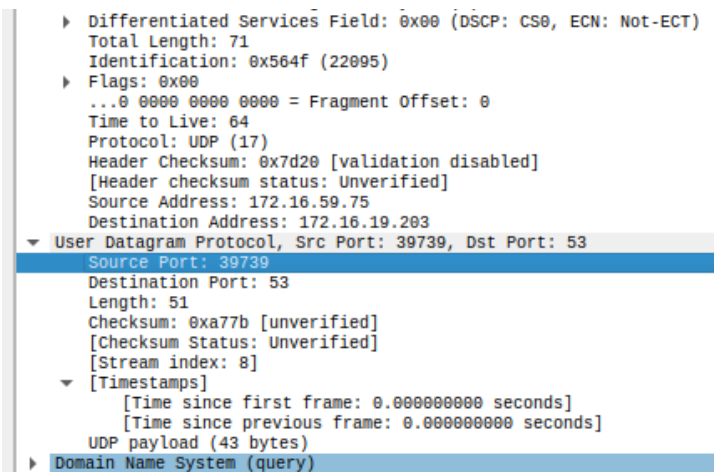


LAB3 > Analyzing UDP datagrams using wireshark

A. In the packet list pane, select the first DNS packet. In the packet detail pane, select the User Datagram Protocol. The UDP hexdump will be highlighted in the packet byte lane. Using the hexdump, Answer the following:

- the source port number.
39739
- the destination port number.
53
- the total length of the user datagram.
71
- the length of the data.
51
- whether the packet is directed from a client to a server or vice versa.
Client to server
- the application-layer protocol.
User datagram protocol
- whether a checksum is calculated for this packet or not.
Yes



```

  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 71
    Identification: 0x564f (22095)
  ▸ Flags: 0x00
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: UDP (17)
    Header Checksum: 0x7d20 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 172.16.59.75
    Destination Address: 172.16.19.203
  ▾ User Datagram Protocol, Src Port: 39739, Dst Port: 53
    Source Port: 39739
    Destination Port: 53
    Length: 51
    Checksum: 0xa77b [unverified]
    [Checksum Status: Unverified]
    [Stream index: 8]
  ▾ [Timestamps]
    [Time since first frame: 0.000000000 seconds]
    [Time since previous frame: 0.000000000 seconds]
    UDP payload (43 bytes)
  ▸ Domain Name System (query)
```

B. What are the source and destination IP addresses in the DNS query message? What are those addresses in the response message? What is the relationship between the two?

SRC: 172.16.59.75

DST: 172.16.19.203

In the response HTTP message, the address are the opposite i.e. source becomes the destination and the previous destination becomes the source.

C. What are the source and destination port numbers in the query message? What are those addresses in the response message? What is the relationship between the two? Which port number is a well-known port number?

QUERY: src: 39739 dst: 53
RESPONSE: src: 53 dst: 39739
53 is the standard port for DNS

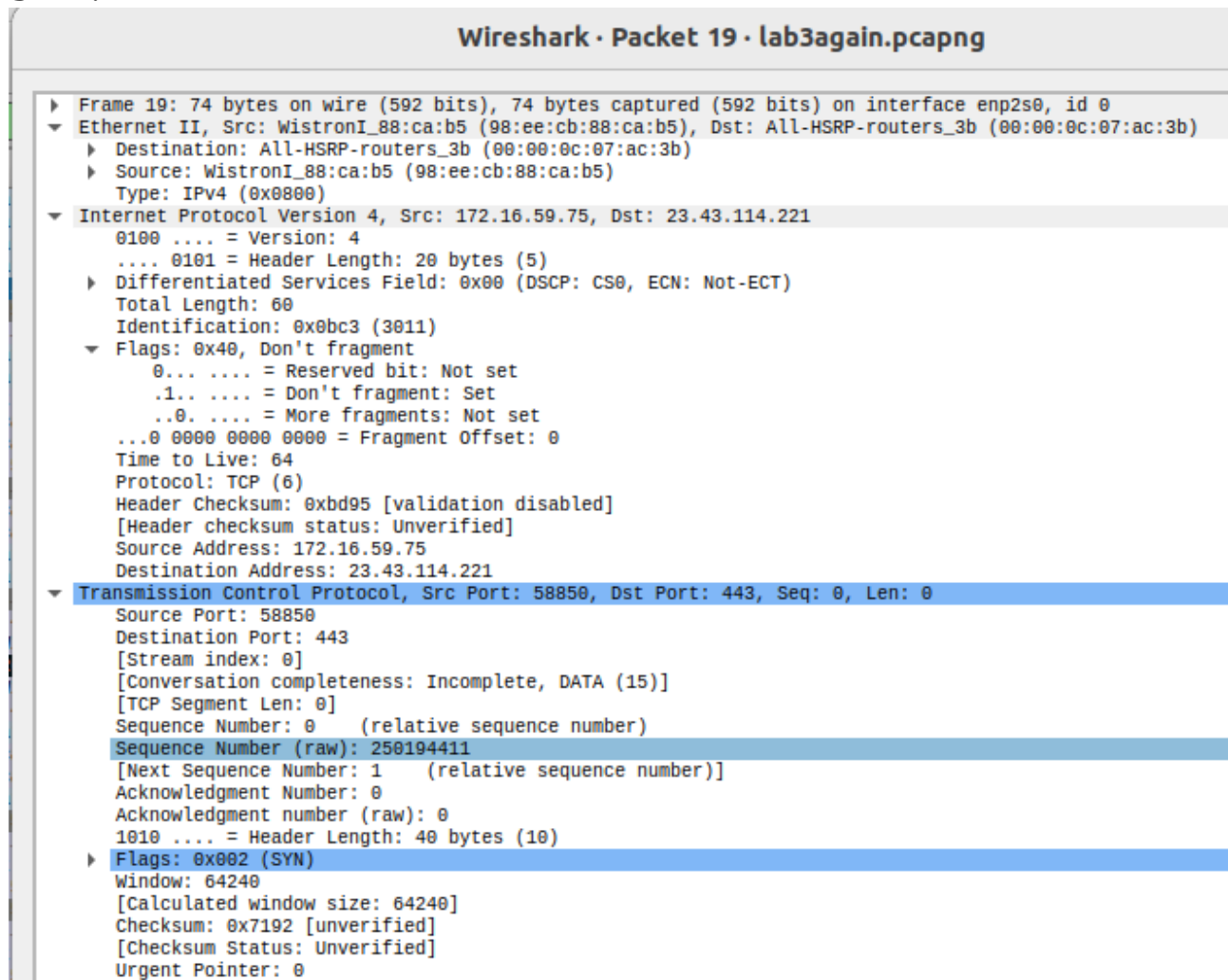
D. What is the length of the first packet? How many bytes of payload are carried by the first packet?

Length: 51

UDP Payload: 43 bytes

FLAGS:

SYN:

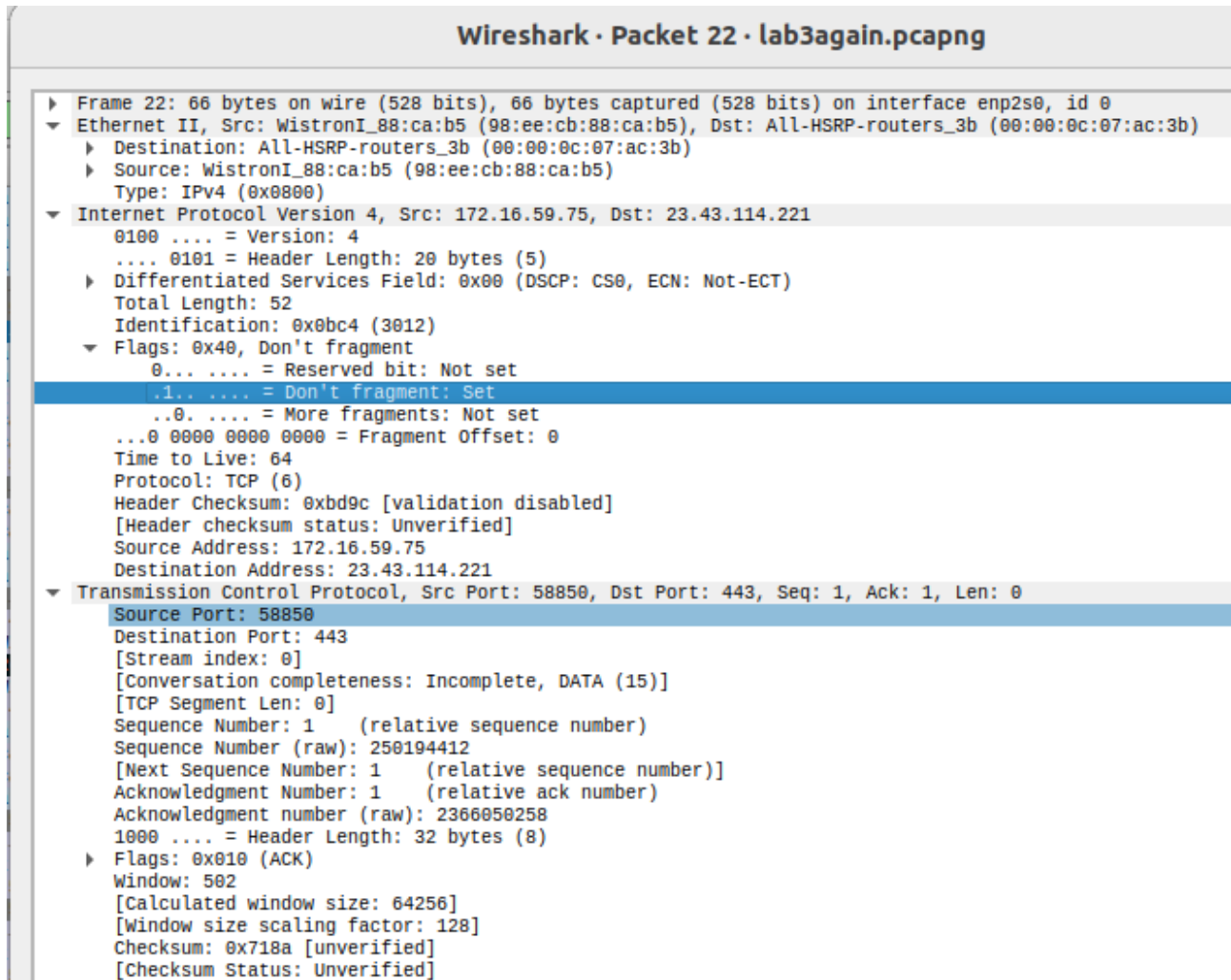


SYN ACK:

Wireshark · Packet 21 · lab3again.pcapng

- ▶ Frame 21: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface enp2s0, id 0
- ▼ Ethernet II, Src: Cisco_13:39:7f (cc:7f:76:13:39:7f), Dst: WistronI_88:ca:b5 (98:ee:cb:88:ca:b5)
 - ▶ Destination: WistronI_88:ca:b5 (98:ee:cb:88:ca:b5)
 - ▶ Source: Cisco_13:39:7f (cc:7f:76:13:39:7f)
 - Type: IPv4 (0x0800)
- ▼ Internet Protocol Version 4, Src: 23.43.114.221, Dst: 172.16.59.75
 - 0100 = Version: 4
 - 0101 = Header Length: 20 bytes (5)
 - ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
 - Total Length: 60
 - Identification: 0x0000 (0)
 - ▼ Flags: 0x40, Don't fragment
 - 0... = Reserved bit: Not set
 - .1.. = Don't fragment: Set
 - ..0. = More fragments: Not set
 - ...0 0000 0000 0000 = Fragment Offset: 0
 - Time to Live: 56
 - Protocol: TCP (6)
 - Header Checksum: 0xd158 [validation disabled]
 - [Header checksum status: Unverified]
 - Source Address: 23.43.114.221
 - Destination Address: 172.16.59.75
- ▼ Transmission Control Protocol, Src Port: 443, Dst Port: 58850, Seq: 0, Ack: 1, Len: 0
 - Source Port: 443
 - Destination Port: 58850
 - [Stream index: 0]
 - [Conversation completeness: Incomplete, DATA (15)]
 - [TCP Segment Len: 0]
 - Sequence Number: 0 (relative sequence number)
 - Sequence Number (raw): 2366050257
 - [Next Sequence Number: 1 (relative sequence number)]
 - Acknowledgment Number: 1 (relative ack number)
 - Acknowledgment number (raw): 250194412
 - 1010 = Header Length: 40 bytes (10)
 - ▶ Flags: 0x012 (SYN, ACK)
 - Window: 65160
 - [Calculated window size: 65160]
 - Checksum: 0xf3a7 [unverified]
 - [Checksum Status: Unverified]
 - Urgent Pointer: 0
 - ▶ Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale
 - ▶ [Timestamps]
 - ▶ [SEQ/ACK analysis]

ACK:



1. What are the socket addresses for each packet?

| | |
|------------------------------------|---------------------------------|
| SYN: Source: 172.16.59.75:58850 | Destination: 23.43.114.221:443 |
| SYN-ACK: Source: 23.43.114.221:443 | Destination: 172.16.59.75:58850 |
| ACK: Source: 172.16.59.75:58850 | Destination: 23.43.114.221:443 |

2. What flags are set in each packet?

0x002 : SYN flag is set
0x012: SYN and ACK flags are set
0x010: ACK flag is set

3. What are the sequence number and acknowledgment number of each packet?

SYN:
Sequence Number: 250194411(0) acknowledgment Number: 0(0)
SYN-ACK:
Sequence Number: 2366050257(0) acknowledgment Number: 250194412(1)
ACK:
Sequence Number: 250194412(1) acknowledgment Number: 2366050258(1)

4. What are the window size of each packet?

SYN: 64240

SYN-ACK: 65160

ACK: 64256

Part II: Data-Transfer Phase

The data-transfer phase starts with an HTTP GET request message and ends with an HTTP OK message.

Wireshark · Packet 7007 · lab3again.pc

- ▶ Frame 7007: 367 bytes on wire (2936 bits), 367 bytes captured (2936 bits) on interface enp2s0, id 0
- ▶ Ethernet II, Src: WistronI_88:ca:b5 (98:ee:cb:88:ca:b5), Dst: All-HSRP-routers_3b (00:00:0c:07:ac:3b)
- ▶ Internet Protocol Version 4, Src: 172.16.59.75, Dst: 34.107.221.82
- ▼ Transmission Control Protocol, Src Port: 56938, Dst Port: 80, Seq: 1, Ack: 1, Len: 301
 - Source Port: 56938
 - Destination Port: 80
 - [Stream index: 117]
 - [Conversation completeness: Incomplete, DATA (15)]
 - [TCP Segment Len: 301]
 - Sequence Number: 1 (relative sequence number)
 - Sequence Number (raw): 1635925582
 - [Next Sequence Number: 302 (relative sequence number)]
 - Acknowledgment Number: 1 (relative ack number)
 - Acknowledgment number (raw): 2713899354
 - 1000 = Header Length: 32 bytes (8)
 - ▶ Flags: 0x018 (PSH, ACK)
 - Window: 502
 - [Calculated window size: 64256]
 - [Window size scaling factor: 128]
 - Checksum: 0xe86c [unverified]
 - [Checksum Status: Unverified]
 - Urgent Pointer: 0
 - ▶ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
 - ▶ [Timestamps]
 - ▶ [SEQ/ACK analysis]
 - TCP payload (301 bytes)
- ▼ Hypertext Transfer Protocol
 - ▼ GET /canonical.html HTTP/1.1\r\n
 - ▶ [Expert Info (Chat/Sequence): GET /canonical.html HTTP/1.1\r\n]
 - Request Method: GET
 - Request URI: /canonical.html
 - Request Version: HTTP/1.1
 - Host: detectportal.firefox.com\r\n
 - User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/116.0\r\n
 - Accept: */*\r\n
 - Accept-Language: en-US,en;q=0.5\r\n
 - Accept-Encoding: gzip, deflate\r\n
 - Cache-Control: no-cache\r\n
 - Pragma: no-cache\r\n
 - Connection: keep-alive\r\n

```

▶ Frame 7016: 364 bytes on wire (2912 bits), 364 bytes captured (2912 bits) on interface enp2s0, id 0
▶ Ethernet II, Src: Cisco_13:39:7f (cc:7f:76:13:39:7f), Dst: WistronI_88:ca:b5 (98:ee:cb:88:ca:b5)
▶ Internet Protocol Version 4, Src: 34.107.221.82, Dst: 172.16.59.75
▼ Transmission Control Protocol, Src Port: 80, Dst Port: 56938, Seq: 1, Ack: 302, Len: 298

```

```

    Source Port: 80
    Destination Port: 56938
    [Stream index: 117]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 298]
    Sequence Number: 1 (relative sequence number)
    Sequence Number (raw): 2713899354
    [Next Sequence Number: 299 (relative sequence number)]
    Acknowledgment Number: 302 (relative ack number)
    Acknowledgment number (raw): 1635925883
    1000 .... = Header Length: 32 bytes (8)
    ▶ Flags: 0x018 (PSH, ACK)
    Window: 261
    [Calculated window size: 66816]
    [Window size scaling factor: 256]
    Checksum: 0x18e5 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
    ▶ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
    ▶ [Timestamps]
    ▶ [SEQ/ACK analysis]
    TCP payload (298 bytes)

```

```

▼ Hypertext Transfer Protocol
  ▼ HTTP/1.1 200 OK\r\n
    ▶ [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      Response Version: HTTP/1.1
      Status Code: 200
      [Status Code Description: OK]
      Response Phrase: OK
      Server: nginx\r\n
    ▶ Content-Length: 90\r\n
      Via: 1.1 google\r\n
      Date: Thu, 24 Aug 2023 13:33:35 GMT\r\n
      Age: 52108\r\n
      Content-Type: text/html\r\n
      Cache-Control: public, must-revalidate, max-age=0, s-maxage=3600\r\n
      \r\n
      [HTTP response 1/1]
      [Time since request: 0.032865203 seconds]
      [Request in frame: 7007]
      [Request URI: http://detectportal.firefox.com/canonical.html]
      File Data: 90 bytes
    ▶ Line-based text data: text/html (1 lines)

```

1. What TCP flags are set in the first data-transfer packet (HTTP GET message)?
ACK and PUSH flags
2. How many bytes are transmitted in this packet?
87 bytes
3. How often does the receiver generate an acknowledgment? To which acknowledgment rule (defined in Page 200 in the textbook) does your answer correspond to?
0
corresponds to NO CONNECTION ESTABLISHMENT(no delay)
4. How many bytes are transmitted in each packet? How are the sequence and acknowledgment numbers related to number of bytes transmitted?
87 bytes transmitted in GET request, 154 bytes received in response.
Sequence numbers increase as number of bytes transmitted increase.

5. What are the original window sizes that are set by the client and the server? Are these numbers expected? How do they change as more segments are received by the client?

```
Window: 502  
[Calculated window size: 64256]  
[Window size scaling factor: 128]
```

Window size will increase as long as hardware supports it to reduce congestion.
They increase.

6. Explain how the window size is used in flow control?

Sender can only send bytes that can fit in the receiver's buffer then must wait till the bytes have been acknowledged to prevent congestion in turn improving flow control.

7. What is the purpose of the HTTP OK message in the data transfer phase?

OK message means that the request succeeded

Part III: Connection Termination Phase

The data-transfer phase is followed by the connection termination phase. Note that some packets used in the connection-termination phase may have the source or sink protocol at the application layer.

1. How many TCP segments are exchanged for this phase?

4 segments are exchanged in the connection termination phase. (FIN, ACK, FIN, ACK)

2. Which end point started the connection termination phase?

Browser(sends first fin flag)

3. What flags are set in each of segments used for connection termination?

Segment 1 Client sends FIN flag to server to initiate termination indicating end of data sending process

Segment 2 : Server sends ACK flag to client to acknowledge FIN flag reception

Segment 3 (Server to Client):FIN flag

Segment 4 (Client to Server): ACK flag to indicate completion of termination process