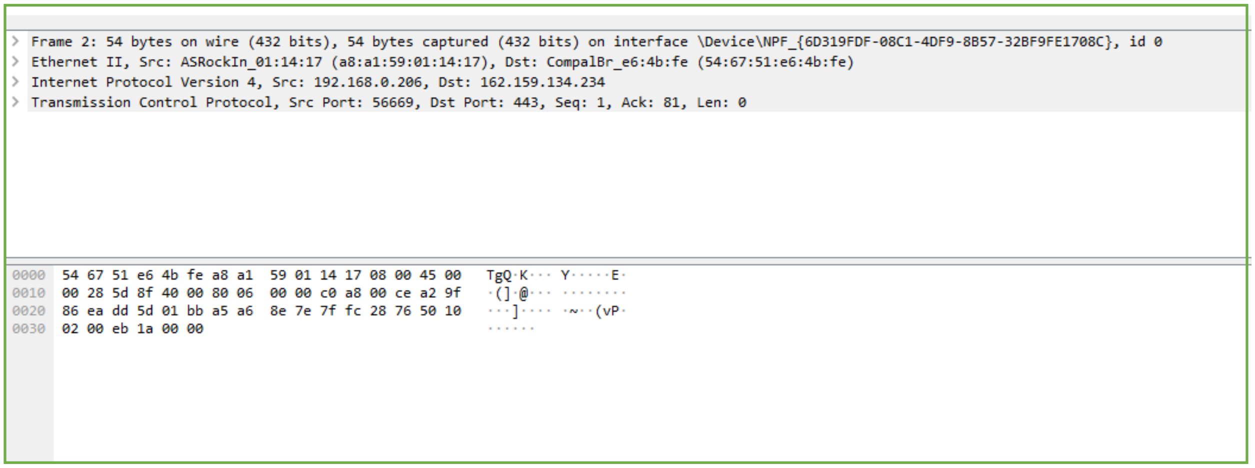
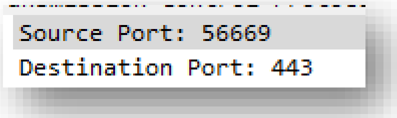
Question 1 – TCP Header

1. Draw an TCP header. Capture packets using wireshark and explain the fields for a particular TCP packet captured. Try to explain the purpose of each field.



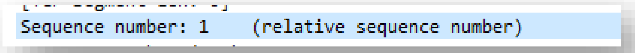
```
> Frame 2: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF_{6D319FDF-08C1-4DF9-8B57-32BF9FE1708C}, id 0
> Ethernet II, Src: ASRockIn_01:14:17 (a8:a1:59:01:14:17), Dst: CompalBr_e6:4b:fe (54:67:51:e6:4b:fe)
> Internet Protocol Version 4, Src: 192.168.0.206, Dst: 162.159.134.234
> Transmission Control Protocol, Src Port: 56669, Dst Port: 443, Seq: 1, Ack: 81, Len: 0

0000  54 67 51 e6 4b fe a8 a1  59 01 14 17 08 00 45 00   TgQ·K··· Y·····E·
0010  00 28 5d 8f 40 00 80 06  00 00 c0 a8 00 ce a2 9f   ·(]·@··· ········
0020  86 ea dd 5d 01 bb a5 a6  8e 7e 7f fc 28 76 50 10   ···]···· ·~··(vP·
0030  02 00 eb 1a 00 00                                  ······
```

<table>
<tr><td colspan="2">Source Port</td><td colspan="2">Destination Port</td></tr>
<tr><td colspan="4">Sequence Number</td></tr>
<tr><td colspan="4">Ach. Number</td></tr>
<tr><td>Data Offset</td><td>Reserved</td><td>Flags</td><td>Window</td></tr>
<tr><td colspan="2">Check Sum</td><td colspan="2">Urgent Pointer</td></tr>
<tr><td colspan="2">Options</td><td colspan="2">Padding</td></tr>
<tr><td colspan="4">Data</td></tr>
</table>



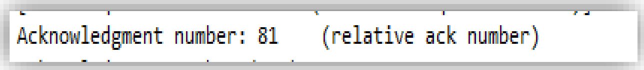
```
Source Port: 56669
Destination Port: 443
```

Source & Destination Ports: 16 bits each, 32 bits total, show the end point of a connection



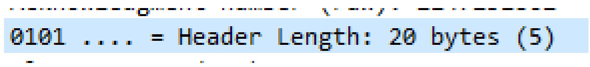
```
Sequence number: 1    (relative sequence number)
```

Sequence Number: 32 bits total, shows the first number assigned to the first byte of data in current message.



```
Acknowledgment number: 81    (relative ack number)
```

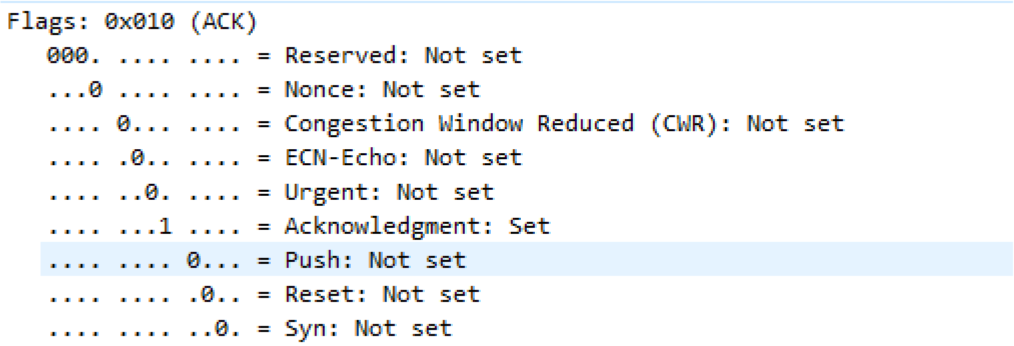
Acknowledgement Number: value of the next sequence number that the sender is expecting to receive



```
0101 .... = Header Length: 20 bytes (5)
```

Header Length / Data Offset: Length can vary with options as can header, tells how many 32 bit words are contained in the TCP

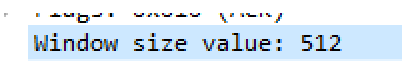
Reserved Field: Must be zero for future use, not sure what.



```
Flags: 0x010 (ACK)
   000. .... .... = Reserved: Not set
   ...0 .... .... = Nonce: Not set
   .... 0... .... = Congestion Window Reduced (CWR): Not set
   .... .0.. .... = ECN-Echo: Not set
   .... ..0. .... = Urgent: Not set
   .... ...1 .... = Acknowledgment: Set
   .... .... 0... = Push: Not set
   .... .... .0.. = Reset: Not set
   .... .... ..0. = Syn: Not set
```

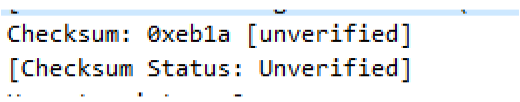
Flags: 6 bits, each represents a Boolean value for various flags.

- URG
- ACK
- PSH
- RST
- SYN
- FIN



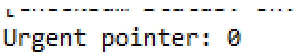
```
Window size value: 512
```

Window size: Buffer space for incoming data on users side.



```
Checksum: 0xeb1a [unverified]
[Checksum Status: Unverified]
```

Checksum: Indicates if package was damaged. My example hasn't verified

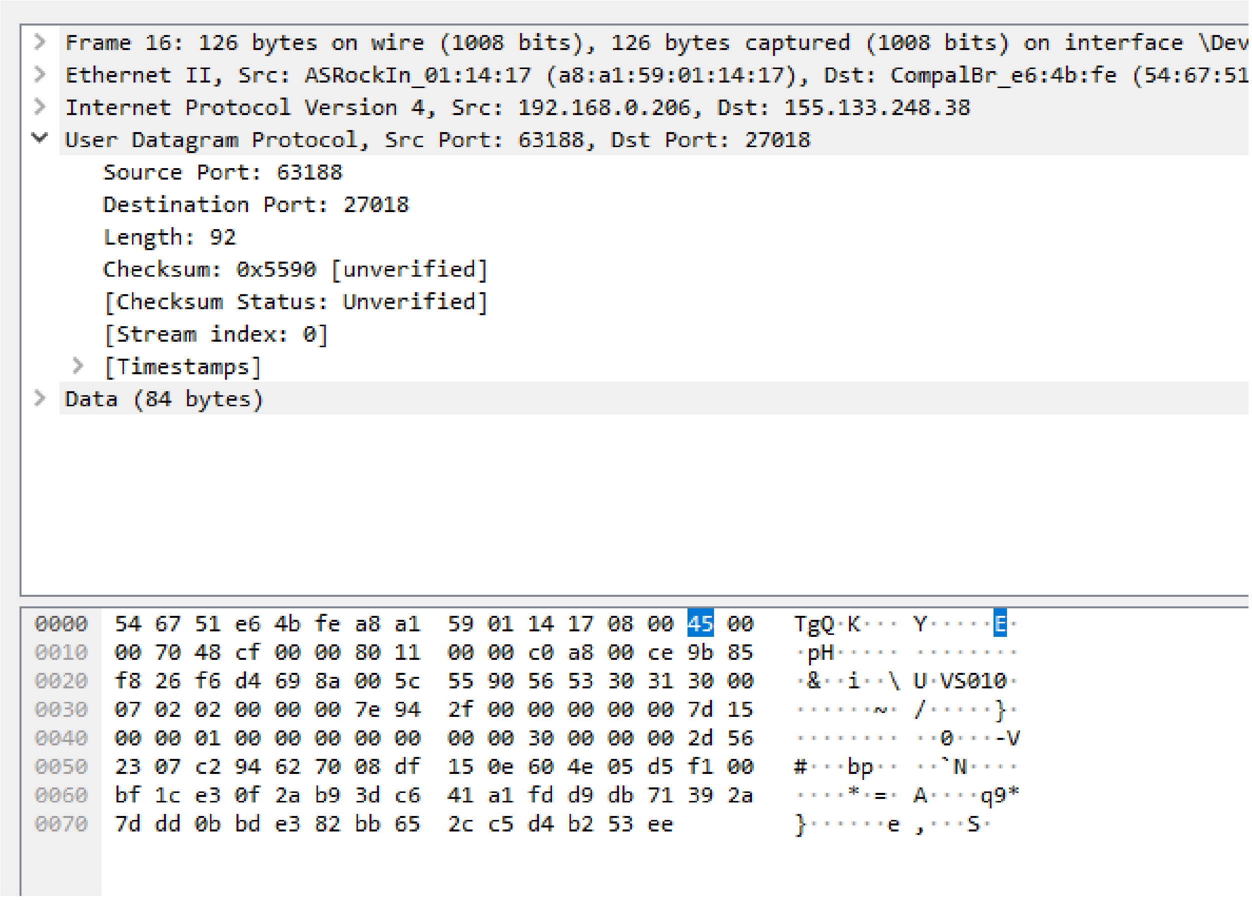


```
Urgent pointer: 0
```

Pointer: Points to first critical data point.

Question 2 – UDP Header

Draw an UDP header. Capture packets using wireshark and explain the fields for a particular UDP packet captured. Try to explain the purpose of each field.



```
> Frame 16: 126 bytes on wire (1008 bits), 126 bytes captured (1008 bits) on interface \Dev
> Ethernet II, Src: ASRockIn_01:14:17 (a8:a1:59:01:14:17), Dst: CompalBr_e6:4b:fe (54:67:51
> Internet Protocol Version 4, Src: 192.168.0.206, Dst: 155.133.248.38
v User Datagram Protocol, Src Port: 63188, Dst Port: 27018
     Source Port: 63188
     Destination Port: 27018
     Length: 92
     Checksum: 0x5590 [unverified]
     [Checksum Status: Unverified]
     [Stream index: 0]
   > [Timestamps]
> Data (84 bytes)
```

```
0000  54 67 51 e6 4b fe a8 a1  59 01 14 17 08 00 45 00   TgQ·K··· Y·····E·
0010  00 70 48 cf 00 00 80 11  00 00 c0 a8 00 ce 9b 85   ·pH····· ········
0020  f8 26 f6 d4 69 8a 00 5c  55 90 56 53 30 31 30 00   ·&··i··\ U·VS010·
0030  07 02 02 00 00 00 7e 94  2f 00 00 00 00 00 7d 15   ······~· /·····}·
0040  00 00 01 00 00 00 00 00  00 00 30 00 00 00 2d 56   ········ ··0···-V
0050  23 07 c2 94 62 70 08 df  15 0e 60 4e 05 d5 f1 00   #···bp·· ··`N····
0060  bf 1c e3 0f 2a b9 3d c6  41 a1 fd d9 db 71 39 2a   ····*·=· A····q9*
0070  7d dd 0b bd e3 82 bb 65  2c c5 d4 b2 53 ee         }······e ,···S·
```

Source Port	Destination Port
Length	Checksum

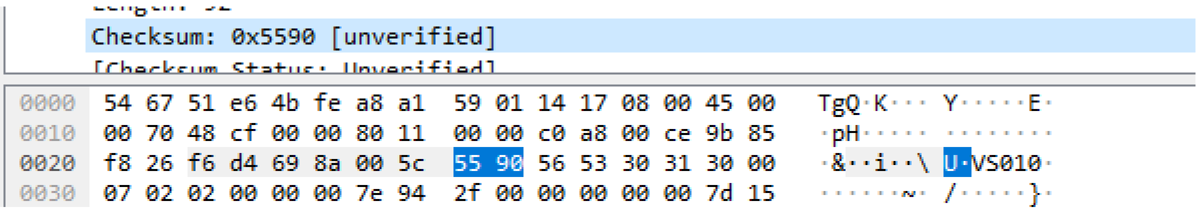
Source Port & Destination Port : Address if receiving and sending ports, 16 bits each, 32 altogether.

Length: tells the size of the header + data.

Checksum: Used to verify if packet has been damaged.

Question 3 – Packet Verification

Capture a UDP packet, verify the checksum using 16-bit One's Complement Sum algorithm.



```
        Length: 92
        Checksum: 0x5590 [unverified]
        [Checksum Status: Unverified]

0000  54 67 51 e6 4b fe a8 a1  59 01 14 17 08 00 45 00   TgQ·K···  Y·····E·
0010  00 70 48 cf 00 00 80 11  00 00 c0 a8 00 ce 9b 85   ·pH·····  ········
0020  f8 26 f6 d4 69 8a 00 5c  55 90 56 53 30 31 30 00   ·&··i··\  U·VS010·
0030  07 02 02 00 00 00 7e 94  2f 00 00 00 00 00 7d 15   ······~·  /·····}·
```

55 90 = 101 0101 1001 0000

Pseudo header

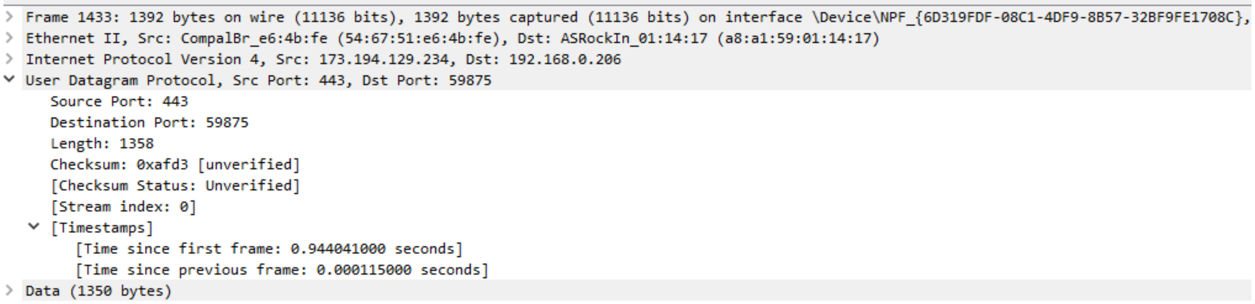
Pseudo header	**Decimal**	**Binary**	**Hex**
SOURCE IP	192.168.0.206	1100 0000 1010 1000 0000 0000 1100 1110	
DEST IP	155.133.248.38	1001 1011 1000 0101 1111 1000 0010 0110	9B . 85 F8. 26
UDP PROT	0/17	0000 0000 0001 0001	
LENGTH	**92**	**0000 0000 0101 1100**	
END OF PSEUDO	**5841484129**	**0 1010 1110 0001 01101 1111 1001 0110 0001**	
UDP source	63188	1111 0110 1101 0100	F6D4
UDP dest	27018	110 1001 1000 1010	698A
UDP Length	92	101 1100	5C
UDP Data	132	1000 0100	84
Add them all up			
Total		0101011100001011110101101010011111	1 5C2F 5A9F
Add 1	1		
		1 0101 1100 0010 1111 0101 1010 1010 0000	
Interchange 0'1 and 1s		0 1010 0011 1101 0000 1010 0101 0101 1111	A3D0 A55F

I think I did something wrong regarding exceeding 16bit limit and calculation

Followed this https://www.securitynik.com/2015/08/calculating-udp-checksum-with-taste-of.html

Question 4

Capture packets from a streaming application. Does this application use UDP or TCP? If both, what UDP packets are mainly used for?



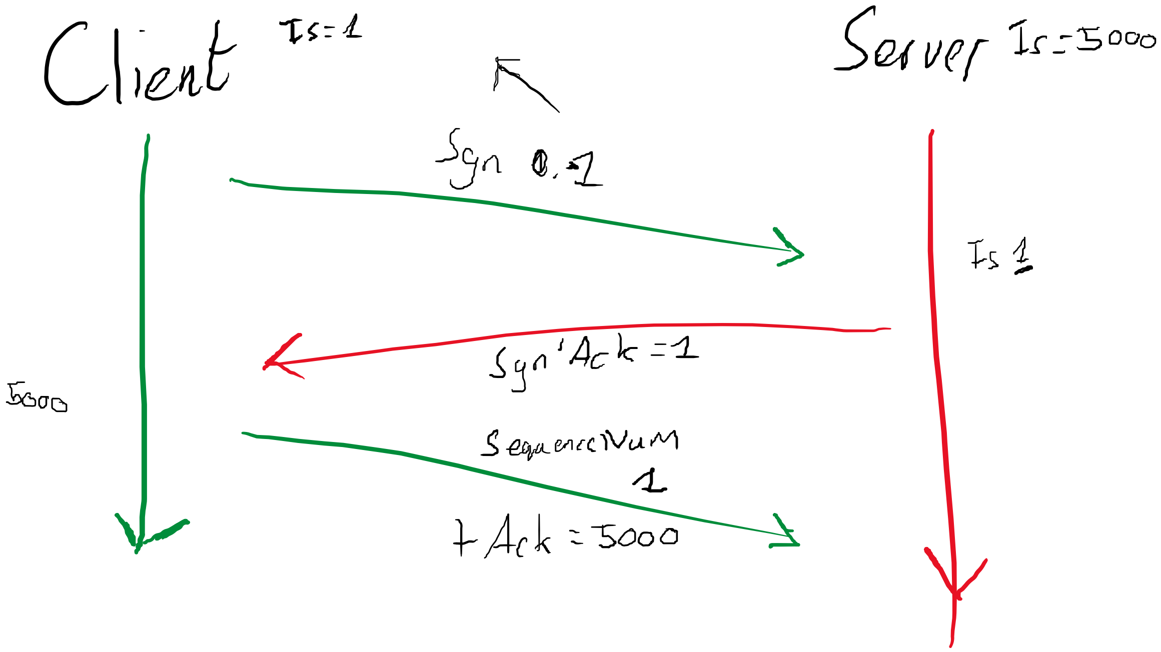
```
> Frame 1433: 1392 bytes on wire (11136 bits), 1392 bytes captured (11136 bits) on interface \Device\NPF_{6D319FDF-08C1-4DF9-8B57-32BF9FE1708C},
> Ethernet II, Src: CompalBr_e6:4b:fe (54:67:51:e6:4b:fe), Dst: ASRockIn_01:14:17 (a8:a1:59:01:14:17)
> Internet Protocol Version 4, Src: 173.194.129.234, Dst: 192.168.0.206
v User Datagram Protocol, Src Port: 443, Dst Port: 59875
    Source Port: 443
    Destination Port: 59875
    Length: 1358
    Checksum: 0xafd3 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
  v [Timestamps]
      [Time since first frame: 0.944041000 seconds]
      [Time since previous frame: 0.000115000 seconds]
> Data (1350 bytes)
```

"For streaming media to flash player Real Time Streaming Protocol(RTSP) is used. The play button on flash player acts as RTSP invoker for media being called and media is streamed via UDP packets"

A UDP packet is being used to transfer media.

Question 5

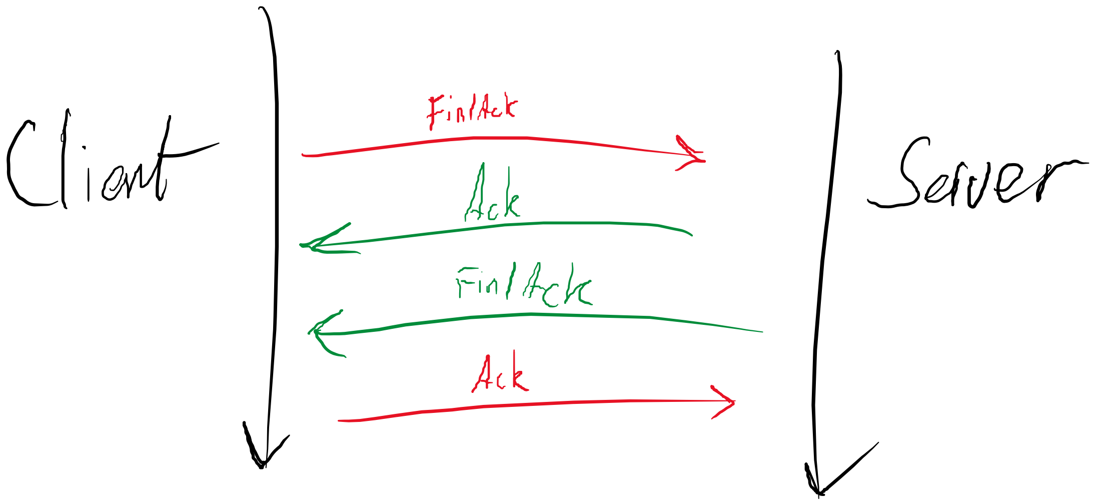
What's TCP 3-Way Handshake? Draw a diagram to illustrate the process using real packets captured in a TCP session. Fill in the values of some key fields of the packets.



1. Client sends start of sequence (is =1 in fig) it subtracts one and makes synchronise flag to 1. The server expectes the syc to be 1.
2. The server notes the sync number and notes it is less than the sequence number of the client and sends back 1, the clients sequence number, sets both syn and ack to 1. The server has picked initial sequence number of 5000, reduces it by one and sends it back to the client
3. Now both sides know what the initial sequence numbers will be for a TCP exchange.

Question 6

What's TCP 4-Way teardown? Draw a diagram to illustrate the process using real packets captured in a TCP session. Fill in the values of some key fields of the packets.



Clients wants to close connection so it sends a fin set to 1 to the server, the client now waits for a response from the server.

Server sends ack (acknowledgment) back to the client.

The client enters wait_2 state and waits for a new response from the server with the fin bit set to 1

When the client receives this it sends it back to the server as acknowledgment and the connection closes.

Bonus

Find two interview questions about TCP, and provide the answer. please provide the reference.

What are TCP and UDP?

Lab 5 – C00225954 Ashleigh Henry

Transmission control protocol, and User Datagram Protocol. TCP is a connection oriented protocol and is bi directional and a bit slower than UDP as UDP is faster and used for smallers amounts of data sent as packets. https://www.educba.com/computer-network-interview-questions/

Explain the range of TCP/ip Classes

CLASS A = 1 to 126

CLASS B = 128 to 191

CLASS C = 192 to 223

http://intquestionsandans.blogspot.com/p/tcpip.html