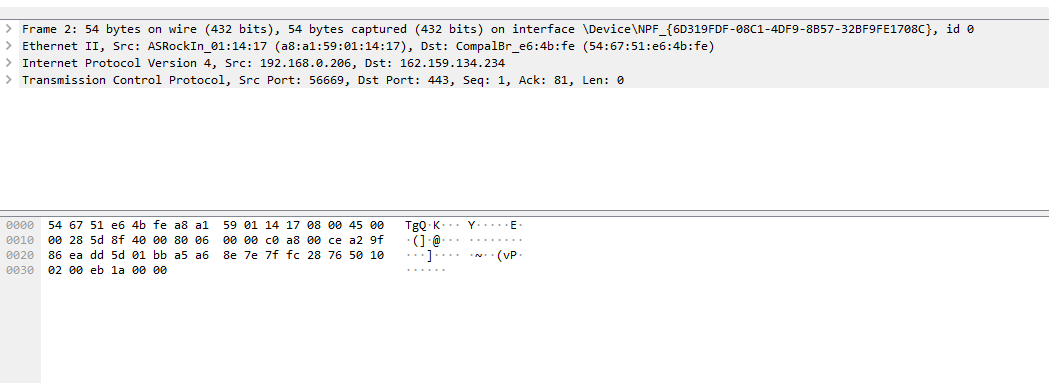
# Question 1 – TCP Header

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.

|  |  |  |  |
| --- | --- | --- | --- |
| Source Port | | Destination Port | |
| Sequence Number | | | |
| Ach. Number | | | |
| Data Offset | Reserved | Flags | Window |
| Check Sum | | Urgent Pointer | |
| Options | | Padding | |
| Data | | | |



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



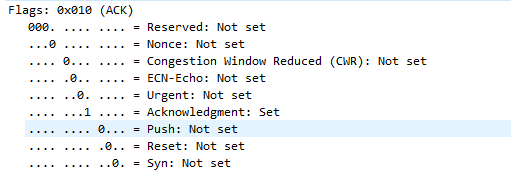
**Sequence Number:** 32 bits total, shows the first number assigned to the first byte of data in current message.



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

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

* URG
* ACK
* PSH
* RST
* SYN
* FIN



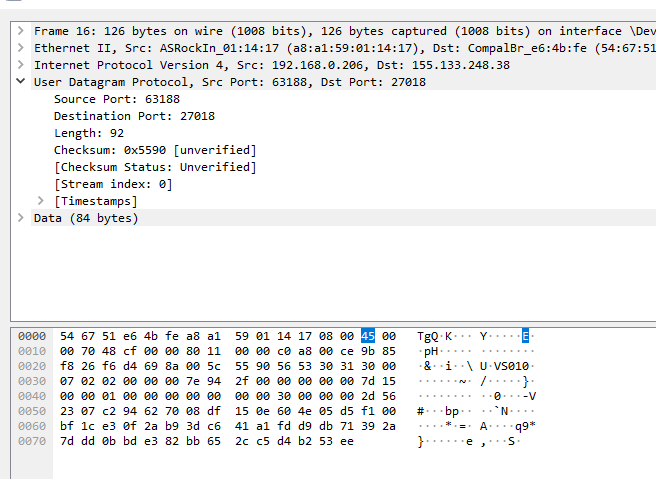
**Window size:** Buffer space for incoming data on users side.

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

**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.



|  |  |
| --- | --- |
| Source Port | Destination Port |
| Length | Checksum |

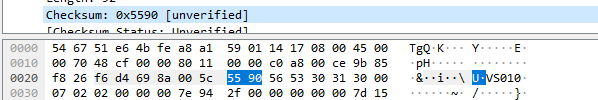
**Source Port & Destination Port :** Address if receiving and sending ports, 16 bits each, 32 altogether.

**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.



**55 90 = 101 0101 1001 0000**

**Pseudo header**

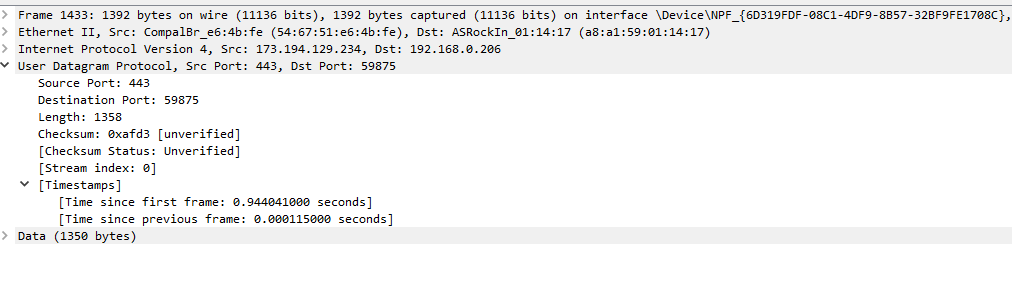
|  |  |  |  |
| --- | --- | --- | --- |
| **Pseudo header** | **Decimal** | **Binary** | **Hex** |
| **SOURCE IP** | 192.168.0.206 | 1100 0000 1010 1000  0000 0000 1100 1110 |  |
| **DEST IP** | 155.133.248.38 | 1001 1011 1000 0101  1111 1000 0010 0110 | 9B . 85  F8. 26 |
| **UDP PROT** | 0/17 | 0000 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**](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?



“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?

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