# Wireshark Lab: UDP and TCP

In this lab, you will explore the UDP and TCP transport protocols using packet traces provided in a zip file. Use Wireshark to examine the traces and answer the questions below. Whenever possible, provide annotated printouts of relevant packets.

## UDP Analysis

Q1: Select one UDP packet from `http-ethereal-trace-5` (which contains some UDP packets carrying SNMP messages). Determine the length (in bytes) of each field in the UDP header by examining the details in Wireshark. List each field and its length.

| | |
|---|---|
| Source Port | 2 bytes |
| Destination Port | 2 bytes |
| Length | 2 bytes |
| Checksum | 2 bytes |

Q2: Examine the value in the "Length" field of the UDP header. What does this length represent? Verify your answer by comparing it to the combined lengths of the header and payload.

The length field in the UDP header represents the total length of the UDP datagram, including the header and payload. This is confirmed by adding the lengths of the UDP header fields to the length of the payload to get the total length of the UDP datagram.

Q3: What is the maximum number of bytes that can be included in a UDP payload? *(Hint: Use the field lengths identified in Question Q2 to help determine this.)*

Since the length field in the UDP header is 2 bytes long, the maximum value that can be represented is $2^{16} - 1 = 65535$ bytes. Subtracting the 8-byte header length gives a maximum payload size of $65535 - 8 = 65527$ bytes.

Q4: What is the highest possible source port number for UDP? *(Hint: Refer to your answer from Q2 for the relevant field length.)*

The source port field in the UDP header is 2 bytes long, so the highest possible source port number is $2^{16} - 1 = 65535$.

Q5: Find the protocol number for UDP in the IP header containing the UDP segment. Provide the protocol number in both hexadecimal and decimal formats.

$$\text{Protocol Number } = 11_{16} = 17_{10}$$

Q6: Find a pair of UDP packets where one is a reply to the other. Describe the relationship between the source and destination port numbers in these packets. The source and destination ports swap between the two packets.

# TCP Analysis

In this section, you will analyze a TCP transfer recorded in `tcp-ethereal-trace-1`. This trace captures the upload of a 150KB file from a client to a server, demonstrating TCP sequence numbers, acknowledgments, congestion control, and flow control.

Q7: What is the IP address and TCP port number used by the client to transfer the file to the server?

The client's IP address is `192.168.1.102`, and uses port `1161` for the transfer.

Q8: Identify the IP address of the server and the port number it uses for sending and receiving TCP segments.

The server's IP address is `128.119.245.12`, and it uses port `80` for both sending and receiving segments.

Q9: Locate the SYN segment initiating the connection to the server. What is the sequence number of this segment? What field indicates this segment as a SYN?

The sequence number of the SYN segment is `232129012`. The SYN flag in the TCP header indicates that this segment is a SYN (second to last bit in the flags field).

Q10: Find the SYNACK segment sent by the server. Record its sequence number, the Acknowledgement field value, and explain how the server calculated this acknowledgment.

The sequence number of the SYNACK segment is `883061785`. The Acknowledgement field value is `232129013`. The server calculated this acknowledgment by adding 1 to the sequence number of the SYN segment received from the client.

Q11: What is the sequence number of the TCP segment containing the HTTP POST command? *(Hint: Look in the DATA field for "POST".)*

`232293053`

Q12: Starting with the HTTP POST, record the sequence numbers of the first six TCP segments, the time each segment was sent, and the time each ACK was received. For each segment, calculate the Round-Trip Time (RTT), which is the time difference between when a segment is sent and when its ACK is received. What is the average RTT for the first 3 segments sent by the client?

| Segment | Sequence Number | Sent Time | RTT | Length |
|---|---|---|---|---|
| 1 | 232293053 | 1093095865 | 0.158489000 | 20 + 50 |
| 2 | 883061786 | 1093095865 | N/A | 20 |
| 3 | 883061786 | 1093095866 | N/A | 20 |
| 4 | 883061786 | 1093095866 | N/A | 20 |
| 5 | 883061786 | 1093095866 | N/A | 20 + 730 |
| 6 | 164091 | 1093095866 | N/A | 20 |

Q13: What is the length (in bytes) of each of the first six TCP segments? Refer above.

Q14: Examine the advertised buffer space at the receiver by inspecting the "Window Size" field in the TCP header of each ACK packet sent by the receiver. Record the smallest value found in the "Window Size" field across all packets. This minimum value represents the least amount of buffer space available at any point during the connection. Based on your

observations, explain whether there were times when a lack of buffer space restricted the sender's ability to transmit data. *(Hint: Sort the packets by the source IP address.)*

The smallest window size observed was 5840 bytes, and it increased throughout the connection. The sender's ability to transmit data was not restricted by the receiver's buffer space, as the window size was always sufficient to accommodate the data being sent.

Q15: Are any TCP segments retransmitted in this trace? If so, what are the packet numbers? If not, how did you determine that there were no retransmissions?

There are no retransmissions in this trace. This was determined by examining the sequence numbers of the transmitted segments and comparing them to the acknowledgment numbers received from the receiver. All segments were acknowledged without any retransmissions.

Q16: How much data does the receiver typically acknowledge in each ACK? Identify cases where the receiver acknowledges every other segment.

The receiver seems to be acknowledging about 3000 bytes of data in each ACK. The receiver almost always acknowledges every other segment.