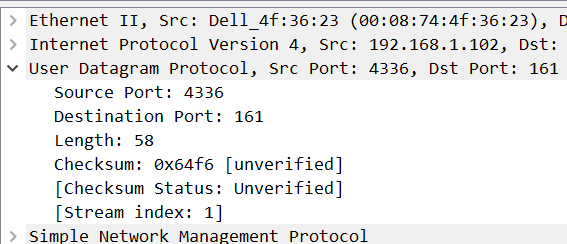
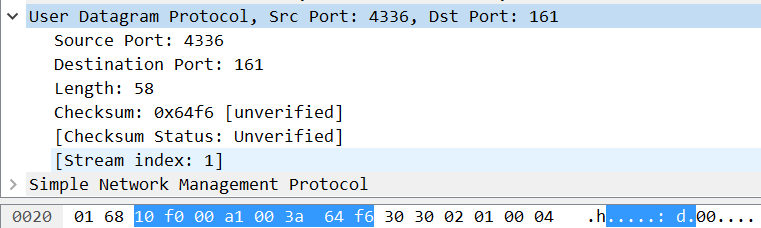
Lab09-Wireshark UDP Lab

***1. Select one UDP packet from your trace. From this packet, determine how many fields there are in the UDP header. Name these fields.***



The UDP header contains 4 fields: source port, destination port, length, and 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.***



Each of the UDP header fields is 2 bytes long.

***3. The value in the Length field is the length of what? (You can consult the text for this answer). Verify your claim with your captured UDP packet.***

The value in the length field is the sum of the 8 header bytes, plus the 42 encapsulated data bytes.

***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)***

The maximum number of bytes that can be included in a UDP payload is 216 – 1 less the header bytes. This gives 65535 – 8 = 65527 bytes.

***5. What is the largest possible source port number? (Hint: see the hint in 4.)***

The largest possible source port number is 216 – 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 Protocol field of the IP datagram containing this UDP segment.***

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

***7. Examine a pair of UDP packets in which your host sends the first UDP packet and the second UDP packet is a reply to this first UDP packet. (Hint: for a second***

***packet to be sent in response to a first packet, the sender of the first packet should be the destination of the second packet). Describe the relationship between the port numbers in the two packets.***

The UDP checksum is calculated as the 16-bit one’s complement of the one’s complement sum of a pseudo header of information from the IP header, the UDP header, and the data. This is padded as needed with zero bytes at the end to make a multiple of two bytes. If the checksum is computed to be 0, it must be set to 0xFFFF.