**Lab 2 UDP**

**Objectives**

In this lab, we’ll take a look at the UDP transport protocol. As we saw in Chapter 3 of the textbook, UDP is a streamlined, no-frills protocol. Please first review section 3.3 in the text before doing this lab. Because UDP is simple and sweet, we’ll be able to cover it pretty quickly in this lab.

At this stage, you should be a Wireshark expert. Thus, we are not going to spell out the steps as explicitly as in earlier labs. In particular, we are not going to provide example screenshots for all the steps.

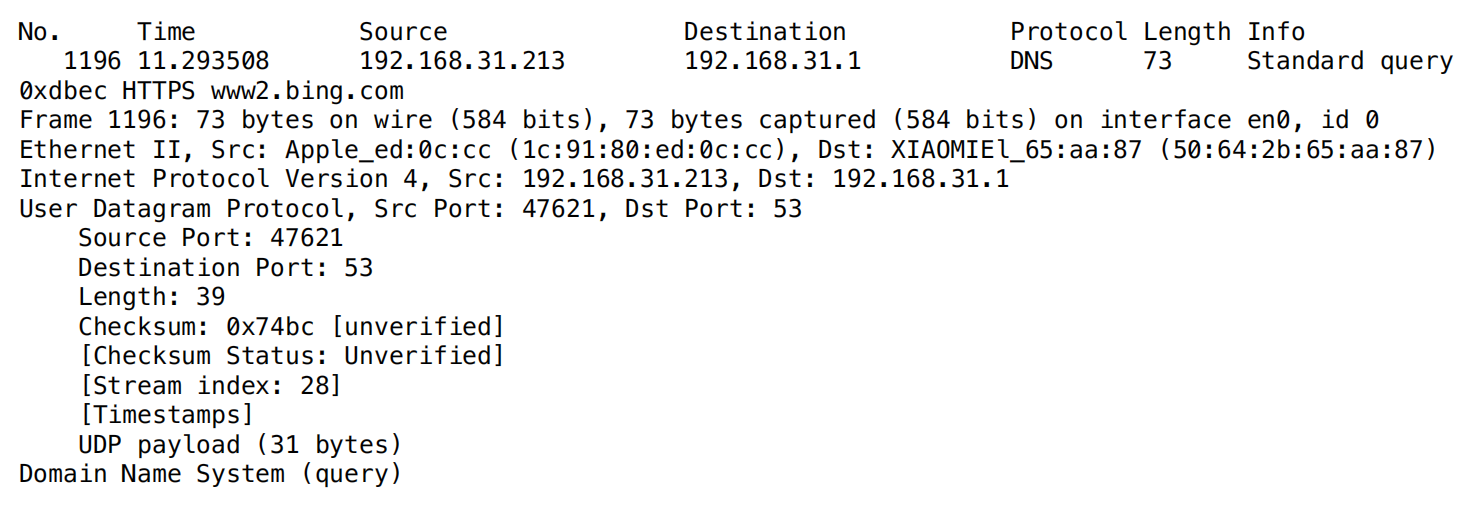
**Procedures**

Start capturing packets in Wireshark and then do something that will cause your host to send and receive several UDP packets. It’s also likely that just by doing nothing (except capturing packets via Wireshark) that some UDP packets sent by others will appear in your trace. In particular, the Simple Network Management Protocol (SNMP – see section 5.7 in the text) sends SNMP messages inside of UDP, so it’s likely that you’ll find some SNMP messages (and therefore UDP packets) in your trace.

After stopping packet capture, set your packet filter so that Wireshark only displays the UDP packets sent and received at your host. Pick one of these UDP packets and expand the UDP fields in the details window.

When answering a question below, please include an electronic printout of the packet(s) within the trace that you used to answer the question below, with annotation to explain your answer. To print a packet, use *File->Print to pdf file*, choose *Selected packet only*, choose *Packet summary line,* and select the minimum amount of packet detail that you need to answer the question.

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



**There are four fields: Source Port, Destination Port, Length, and CheckSum**

1. 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.

**The UDP header has a fixed length of 8 bytes. Each of these 4 header fields is 2 bytes long.**

1. 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 length field specifies the number of bytes in the UDP segment (header plus data). An explicit length value is needed since the size of the data field may differ from one UDP segment to the next. The length of UDP payload for selected packet is 31 bytes. 39 bytes - 8 bytes = 31 bytes.**

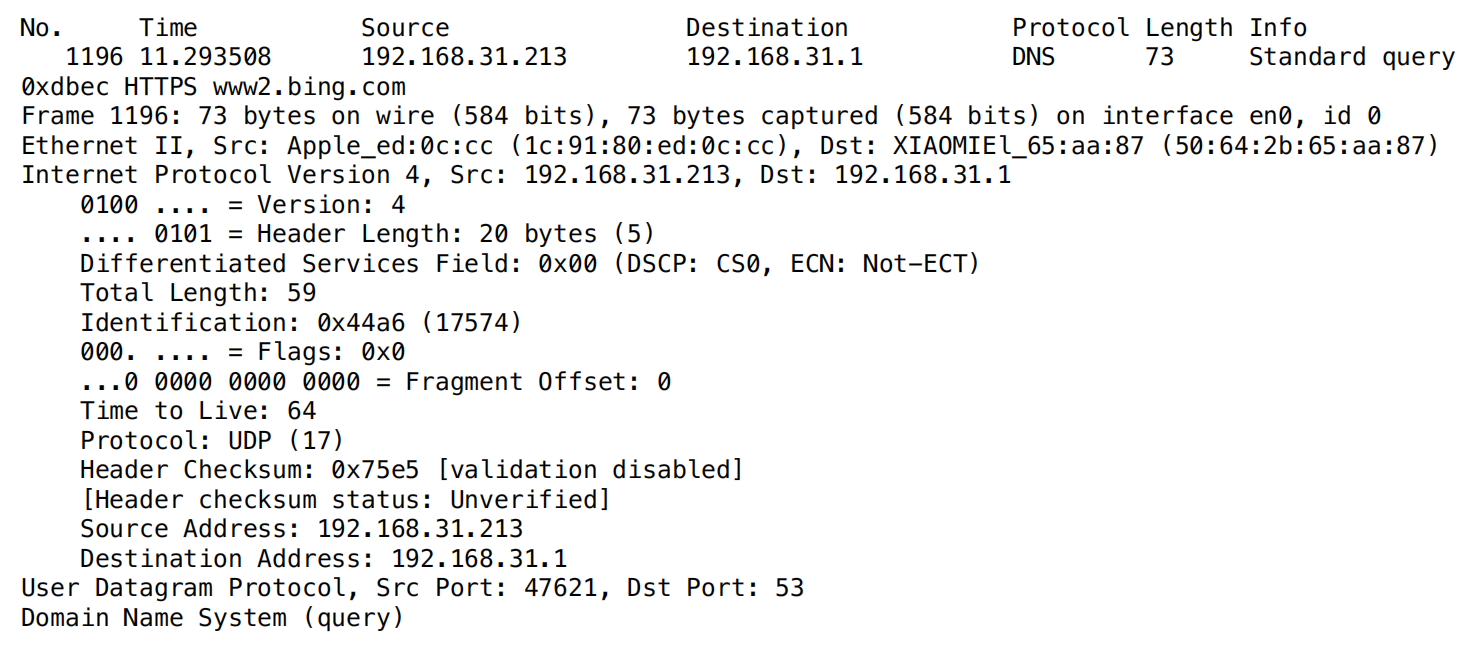
1. What is the maximum number of bytes that can be included in a UDP payload? (Hint: the answer to this question can be determined by your answer to 2. above)

**The maximum number of bytes that can be included in a UDP payload is (2^16 – 1) bytes plus the header bytes. This gives 65535 bytes – 8 bytes = 65527 bytes.**

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

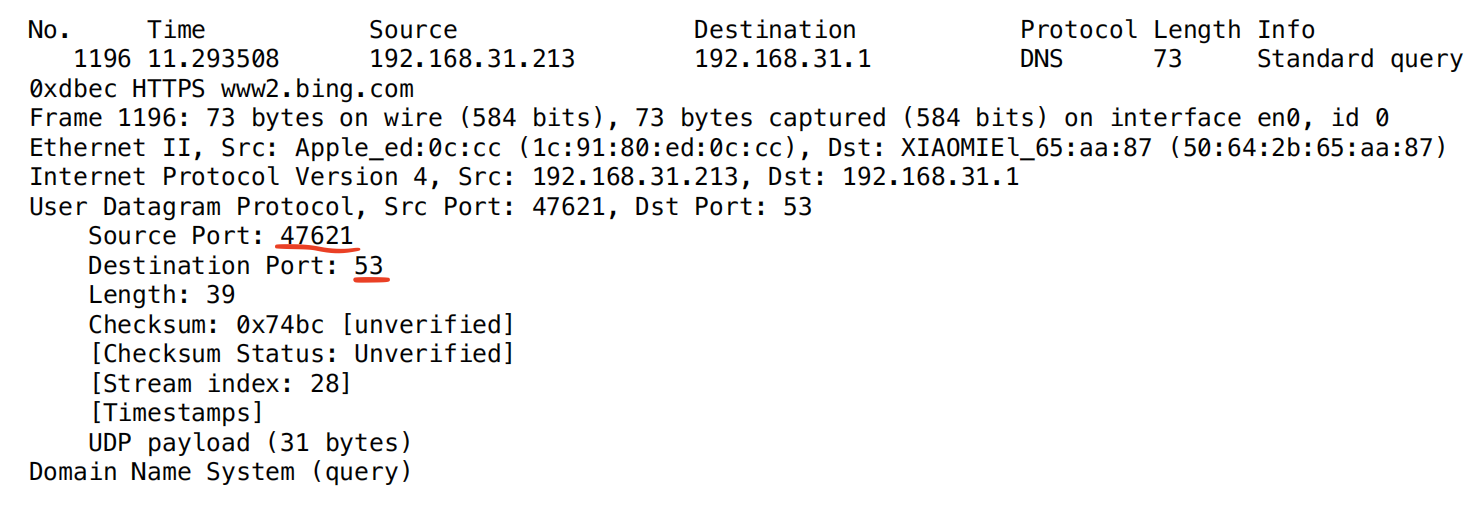
**The largest possible source port number is (2^16 – 1) = 65535.**

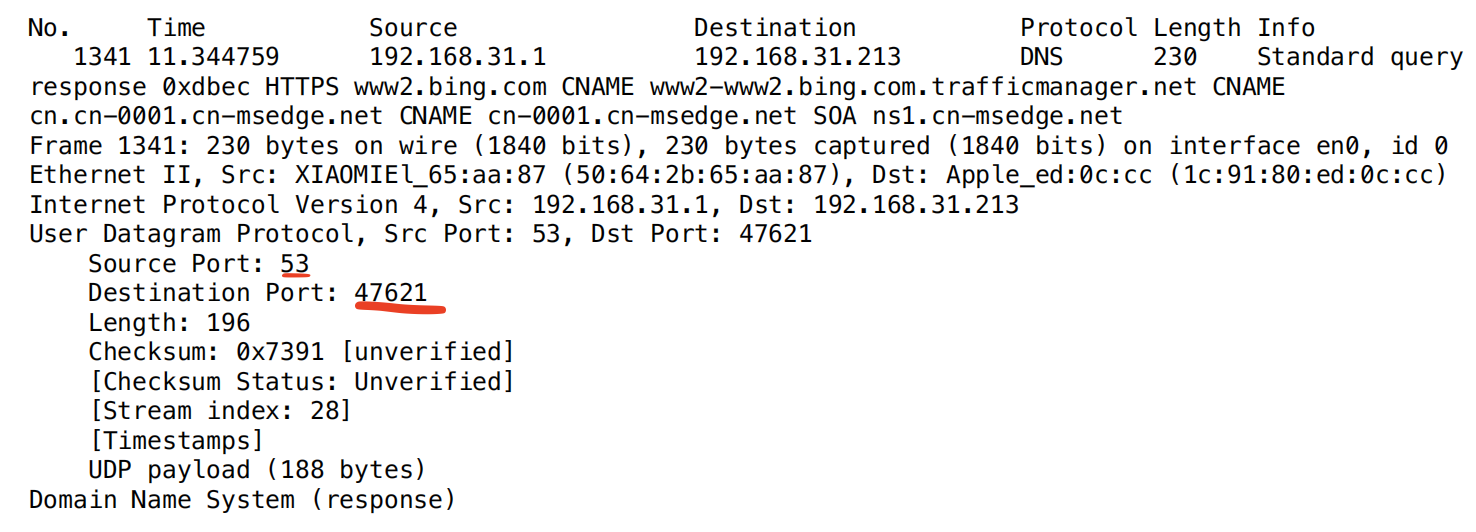
1. What is the protocol number for UDP? Give your answer in both hexadecimal and decimal notation. To answer this question, you’ll need to look into the Protocol field of the IP datagram containing this UDP segment (see Figure 4.13 in the text, and the discussion of IP header fields).



**The IP protocol number for UDP is 0x11 hex, and that translates to 17 in decimal.**

1. 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 source port of the UDP packet sent by the host is the same as the destination port of the reply packet. Likewise, the destination port of the UDP packet sent by the host is the same as the source port of the response packet.**