Network Programming Lab 01

Due date: Friday, September 8 at 11:59 PM on Submitty

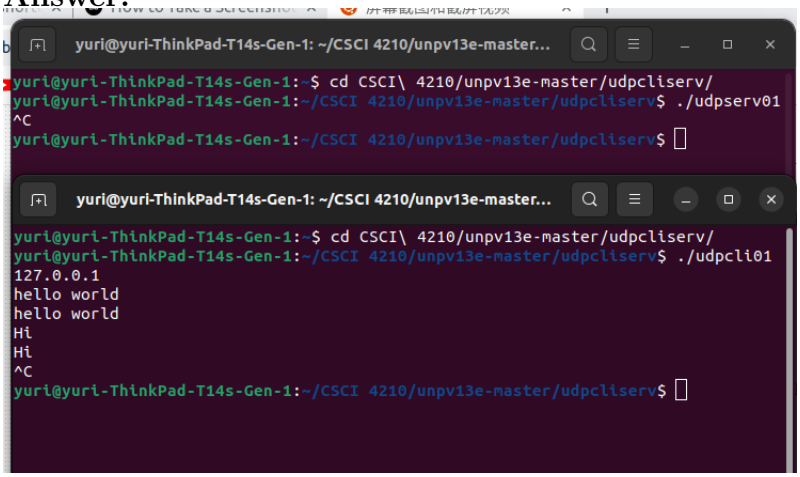
Initial Capture:

1. Write down how many different protocols are visible with the filter active.
   Answer:
   There is 1 protocol visible with the filter active, which is UDP.

2. Write down how many UDP datagrams should your program have sent?
   Answer:
   There are 4 datagrams the program sent

3. How many should it have received?
   Answer:
   It should receive 2 datagrams.

4. Make sure to record the input and output from your terminal.
   Answer:



5. How many datagrams in Wireshark appear to be from either your udpserv01 or udpcli01 programs? (Write down this number)
   Answer:

   (a) From udpserv01: 2 datagrams

   (b) From udpcli01: 2 datagrams

SWitching to Loopback:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.1	UDP	54	36031 → 9877 Len=12
2	0.000095194	127.0.0.1	127.0.0.1	UDP	54	9877 → 36031 Len=12
3	2.961803126	127.0.0.1	127.0.0.1	UDP	45	36031 → 9877 Len=3
4	2.961918225	127.0.0.1	127.0.0.1	UDP	45	9877 → 36031 Len=3

1. Do the numbers match up?
   Answer:
   Yes, the numbers match up.
   There are 4 UDP datagrams in total, 2 of which are from udpserv01, and 2 of which are from udpcli01.

Examining Packet Content:

1. What was the port number on the client side?
   Answer:
   The client Port number is 36031.

2. What was the port number on the server side?
   Answer:
   The server port number is 8977.

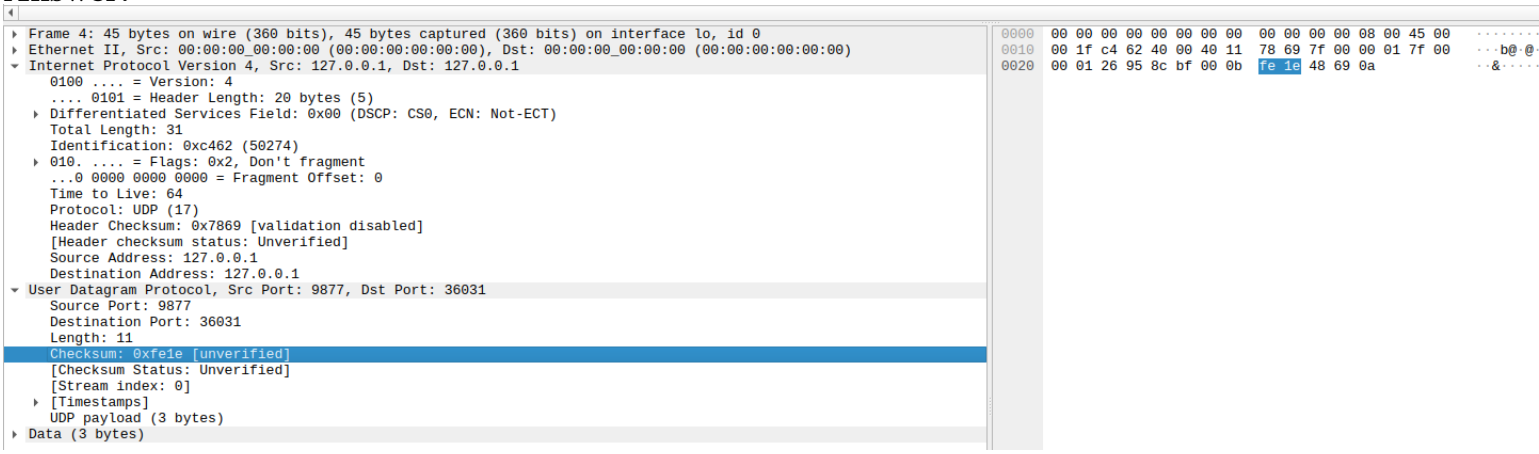
3. How large is the UDP header?
   Answer:
   The UDP header has a fixed length of 8 bytes.

4. How large is the application data? (Answer this for just one of your packets)
   Answer:
   The size of application data is 3 bytes (for NO. 4).

5. How large are all the headers in one packet? Give just a single total number. Answer this for any one of your packets)
   Answer:
   The size of all headers in one packet is $45 - 3 = 42$ bytes (for NO. 4).

Internet Checksums

Answer:



*: Using the datagram that sends out the string "Hi" to the client (NO 12).

Pseudo Header:
IP Source: 7f 00 00 01
IP Dest: 7f 00 00 01
Protocol: 00 11
UDP Length: 00 0b

Header:
UDP header: 26 95 8c bf 00 0b 00 00
Data: 48 69 0a (00)
Thus, we have:

	Parts	Hex	Binary
	Header		
	UDP Source	26 95	0010 0110 1001 0101
	UDP Dest	8c bf	1000 1100 1011 1111
	UDP Len	00 0b	0000 0000 0000 1011
	UDP checkSum	00 00	0000 0000 0000 0000
	Data	(00) 48	0000 0000 0100 1000
'+		69 0a	0110 1001 0000 1010
	Psedu Header		
	Source IP	7f 00	0111 1111 0000 0000
		00 01	0000 0000 0000 0001
	Dest IP	7f 00	0111 1111 0000 0000
		00 01	0000 0000 0000 0001
	Protocol	00 11	0000 0000 0001 0001
	Length	00 0b	0000 0000 0000 1011

Where we have the sum of all these binaries is $0001\ 1010\ 1101\ 0001_2$. After adding the prechecksum ($1110\ 0101\ 0010\ 1110_2 = E52E_{16}$) Sadly, it does not match up with the UDP checksum ($FE1E_{16}$).

* Note: The temporary calculation is in the column "result" below the snapshot. It computes the sum of the binary in the current line and the result in the previous line (curry modification included).

	Parts	Hex	Binary	Result
	Header			
	UDP Source	26 95	0010 0110 1001 0101	
	UDP Dest	8c bf	1000 1100 1011 1111	1011 0011 0101 0100
	UDP Len	00 0b	0000 0000 0000 1011	1011 0011 0101 1111
	UDP checkSum	00 00	0000 0000 0000 0000	1011 0011 0101 1111
	Data	(00) 48	0000 0000 0100 1000	1011 0011 1010 0111
'+		69 0a	0110 1001 0000 1010	0001 1100 1011 0010
				0001 1100 1011 0010
	Psedu Header			
	Source IP	7f 00	0111 1111 0000 0000	1001 1011 1011 0010
		00 01	0000 0000 0000 0001	1001 1011 1011 0011
	Dest IP	7f 00	0111 1111 0000 0000	0001 1010 1011 0100
		00 01	0000 0000 0000 0001	0001 1010 1011 0101
	Protocol	00 11	0000 0000 0001 0001	0001 1010 1100 0110
	Length	00 0b	0000 0000 0000 1011	0001 1010 1101 0001