Predicting the contents of an IP header

This is an example of how to format your results for Lab 2. Here, we predict and observe the IP header of a TCP/IP packet. In the lab, you will predict the UDP header and payload of a UDP packet (ignoring the IP header).

Just to be clear, this is TCP, but you should do UDP.

Predicted Contents

Field name and contents in decimal	*Field contents in hexadecimal*
Version: 4 bits (0.5 byte): 4	0x4 (0b0100)
Header length: 4 bits (0.5 byte): 20	0x14? 0x4? 0x0? (How to fit in just 0.5 bytes?)
Type of service: 8 bits (1 byte): unknown	0x--
Datagram Length: 16 bits (2 bytes): 20 bytes (IP header) + 20 bytes (TCP header) + 0 bytes (TCP data) = 40. (The first TCP message sent contains no data because it is part of a 3-way handshake to set up the connection.)	0x00 28
16-bit identifier, flags, and 13-bit fragmentation offset: 32 bits (8 bytes): unknown	0x-- -- -- -- -- -- -- --
Time-to-live: 8 bits (1 byte): 255?	0xff?
Protocol: 8 bits (1 byte): TCP (6)	0x06
Header checksum: 16 bits (2 bytes): unknown	0x--
Source IP address: 32 bits (4 bytes): 192.168.1.15	0xc0 a8 01 0f
Destination IP address: 32 bits (4 bytes): 98.142.108.75	0x62 8e ac 2b
Options: 0 bits: none	
Data: 160 bits (20 bytes): TCP header	(not included in this example – do include the payload in your lab)

Complete Predicted Packet

Prediction: `40 -- 00 28 -- -- -- -- -- -- -- -- ff 06 -- c0 a8 01 0f 62 8e ac 2b`

Complete Actual Packet

Actual: `45 00 00 34 47 ca 40 00 80 06 22 69 c0 a8 01 0f 62 8e 6c 4b`

Actual Contents

Field name and contents in decimal	*Field contents in hexadecimal*
Version: 4 bits (0.5 byte): 4 (same as predicted)	0x4
Header length: 4 bits (0.5 byte): 5 * (32-bit words) = 5*4 bytes = 20 bytes	0x5
Type of service: 8 bits (1 byte): 0	0x00
Datagram Length: 16 bits (2 bytes): 52 = 20 bytes (IP header) + 32 bytes (TCP header) + 0 bytes data. The TCP header was longer than expected because it contained 12 bytes in the "options" field, to set the Maximum Segment Size (MSS), Window Scale, and SACK Permitted.	0x00 34
16-bit identifier, flags, and 13-bit fragmentation offset: 32 bits (**4** bytes): I computed the byte size wrong. 32/8 = 4, not 8!	0x47 ca 40 00
Time-to-live: 8 bits (1 byte): 128 (about half what I predicted)	0x80
Protocol: 8 bits (1 byte): TCP (6) (same as predicted)	0x06
Header checksum: 16 bits (2 bytes): 0x2269 (I only included one blank byte -- instead of two -- --)	0x22 69
Source IP: 32 bits (4 bytes): 192.168.1.15 (same as predicted)	0xc0 a8 01 0f
Destination IP: 32 bits (4 bytes): 98.142.108.75 (error converting to hex)	0x62 8e 6c 4b
Options: 0 bits: None	
Data: 256 bits (32 bytes): TCP header (see above)	(not included in this example)