

1. Select one UDP packet from your trace. From this packet, determine how many fields there are in the UDP header. (You shouldn't look in the textbook! Answer these questions directly from what you observe in the packet trace.) Name these fields.

No.	Time	Source	Destination	Protocol	Length	Info
3	0.051550	2800:3f0:4001:827::...	2001:12f0:618:16b:1...	UDP	87	443 → 49568 Len=25
13	1.702610	2800:3f0:4001:827::...	2001:12f0:618:16b:1...	UDP	87	443 → 49568 Len=25
41	4.947014	2800:3f0:4001:827::...	2001:12f0:618:16b:1...	UDP	87	443 → 49568 Len=25
67	8.894934	20.190.173.144	10.14.69.26	TCP	66	443 → 50562 [ACK] Seq=1 Ack=2 Win=2052 Len=0 SLE=1 SRE=2
38	4.473825	2620:1ec:46::33	2001:12f0:618:16b:1...	TCP	74	443 → 50574 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
65	8.568368	2620:1ec:21::16	2001:12f0:618:16b:1...	TCP	74	443 → 50590 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
63	7.370282	204.79.197.219	10.14.69.26	TCP	66	443 → 50594 [ACK] Seq=1 Ack=2 Win=16384 Len=0 SLE=1 SRE=2
7	1.282227	2603:105c:1400:1...	2001:12f0:618:16b:1...	TCP	86	443 → 50670 [ACK] Seq=1 Ack=2 Win=2045 Len=0 SLE=1 SRE=2

>	Frame 3: 87 bytes on wire (696 bits), 87 bytes captured (696 bits) on interface \Device\NPF_{974E18D6-0849-479E-B9AA-4DC9F63482A4}, id 0
>	Ethernet II, Src: Alcatel_94:c3:cd (94:24:e1:94:c3:cd), Dst: Inventus_5f:e7:34 (a4:63:a1:5f:e7:34)
>	Internet Protocol Version 6, Src: 2800:3f0:4001:827::200a, Dst: 2001:12f0:618:16b:194:3775:37d3:85fb
>	User Datagram Protocol, Src Port: 443, Dst Port: 49568
>	Source Port: 443
>	Destination Port: 49568
>	Length: 33
>	Checksum: 0x01c0 [unverified]
>	[Checksum Status: Unverified]
>	[Stream index: 0]
>	[Timestamps]
>	UDP payload (25 bytes)
>	Data (25 bytes)

0000	a4 63 a1 5f e7 34 94 24	e1 94 c3 cd 86 dd 60 00	..4\$.....
0010	00 00 00 21 11 38 28 00	03 f0 40 01 08 27 00 00	...8(....@..
0020	00 00 00 00 20 0a 20 01	12 f0 06 18 01 6b 01 94k..
0030	37 75 37 d3 85 fb 01 bb	c1 a0 00 21 01 c0 57 16	7u7.....!..W
0040	6b 09 30 f5 c2 1b c7 f5	95 80 9c a9 55 81 11 6c	k:0.....U..1
0050	f7 8a 07 99 95 bf ca	

O header do UDP tem quatro campos: Source Port, Destination Port, Length e 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.

Cada um dos UDP campos header tem comprimento de 2 bytes.

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.

O valor do comprimento do campo é a soma dos 8 bytes do header, mais os 42 data bytes encapsulados.

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)

O número máximo de bytes que podem ser incluídos num UDP payload é $2^{16} - 1$ menos os bytes do header. Isso dá $65535 - 8 = 65527$ bytes.

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

O maior source port possível é $2^{16} - 1 = 65535$.

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 (see Figure 4.13 in the text, and the discussion of IP header fields).

O número de protocolo IP para o UDP é 0x11hex, que é 17 em decimal.

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.

A soma de verificação UDP é calculada como o complemento de um de 16 bits da soma do complemento de um de um pseudocabeçalho de informações do cabeçalho IP, do cabeçalho UDP e dos dados. Isto é preenchido conforme necessário com zero bytes no final para fazer um múltiplo de dois bytes. Se a soma de verificação for calculada como 0, deve ser definido como 0xFFFF.