

שאלה 2:

סעיף א TCP

tcp.stream eq 0

No.	Time	Source	Destination	Protocol	Length	Source Port	Destination Port	Info
3	1.210...	10.0.0.10	10.0.0.7	TCP	74	49818	12345	49818 → 12345 [SYN, ECN, CWR] Seq=2022050827 Win=29200 Len=0 MSS=1460 SACK_PER...
4	1.210...	10.0.0.7	10.0.0.10	TCP	66	12345	49818	49818 → 12345 [SYN, ACK] Seq=2490432537 Ack=2022050828 Win=65535 Len=0 MSS=146...
5	1.210...	10.0.0.10	10.0.0.7	TCP	54	49818	12345	49818 → 12345 [ACK] Seq=2022050828 Ack=2490432538 Win=29312 Len=0
6	1.210...	10.0.0.10	10.0.0.7	TCP	7354	49818	12345	49818 → 12345 [ACK] Seq=2022050828 Ack=2490432538 Win=29312 Len=7300
7	1.210...	10.0.0.10	10.0.0.7	TCP	7354	49818	12345	49818 → 12345 [ACK] Seq=2022058128 Ack=2490432538 Win=29312 Len=7300
8	1.210...	10.0.0.7	10.0.0.10	TCP	60	12345	49818	12345 → 49818 [ACK] Seq=2490432538 Ack=2022052288 Win=525568 Len=0
9	1.210...	10.0.0.10	10.0.0.7	TCP	454	49818	12345	49818 → 12345 [PSH, ACK] Seq=2022065428 Ack=2490432538 Win=29312 Len=400
10	1.210...	10.0.0.7	10.0.0.10	TCP	60	12345	49818	12345 → 49818 [ACK] Seq=2490432538 Ack=2022053748 Win=525568 Len=0
11	1.210...	10.0.0.7	10.0.0.10	TCP	60	12345	49818	12345 → 49818 [ACK] Seq=2490432538 Ack=2022055208 Win=525568 Len=0
12	1.210...	10.0.0.7	10.0.0.10	TCP	60	12345	49818	12345 → 49818 [ACK] Seq=2490432538 Ack=2022066668 Win=525568 Len=0
13	1.210...	10.0.0.7	10.0.0.10	TCP	60	12345	49818	12345 → 49818 [ACK] Seq=2490432538 Ack=2022058128 Win=525568 Len=0
14	1.210...	10.0.0.7	10.0.0.10	TCP	60	12345	49818	12345 → 49818 [ACK] Seq=2490432538 Ack=2022059588 Win=525568 Len=0
15	1.210...	10.0.0.7	10.0.0.10	TCP	60	12345	49818	12345 → 49818 [ACK] Seq=2490432538 Ack=2022061048 Win=525568 Len=0
16	1.210...	10.0.0.7	10.0.0.10	TCP	60	12345	49818	12345 → 49818 [ACK] Seq=2490432538 Ack=2022062508 Win=525568 Len=0
17	1.210...	10.0.0.7	10.0.0.10	TCP	60	12345	49818	12345 → 49818 [ACK] Seq=2490432538 Ack=2022063968 Win=525568 Len=0
18	1.210...	10.0.0.7	10.0.0.10	TCP	60	12345	49818	12345 → 49818 [ACK] Seq=2490432538 Ack=2022065428 Win=525568 Len=0
19	1.210...	10.0.0.7	10.0.0.10	TCP	60	12345	49818	12345 → 49818 [ACK] Seq=2490432538 Ack=2022065828 Win=525056 Len=0
20	1.217...	10.0.0.7	10.0.0.10	TCP	60	12345	49818	12345 → 49818 [PSH, ACK] Seq=2490432538 Ack=2022065828 Win=525056 Len=1
21	1.217...	10.0.0.10	10.0.0.7	TCP	54	49818	12345	49818 → 12345 [ACK] Seq=2022065828 Ack=2490432539 Win=29312 Len=0
22	1.217...	10.0.0.10	10.0.0.7	TCP	54	49818	12345	49818 → 12345 [FIN, ACK] Seq=2022065828 Ack=2490432539 Win=29312 Len=0
23	1.217...	10.0.0.7	10.0.0.10	TCP	60	12345	49818	12345 → 49818 [ACK] Seq=2490432539 Ack=2022065829 Win=525056 Len=0
24	1.217...	10.0.0.7	10.0.0.10	TCP	60	12345	49818	12345 → 49818 [FIN, ACK] Seq=2490432539 Ack=2022065829 Win=525056 Len=0
25	1.217...	10.0.0.10	10.0.0.7	TCP	54	49818	12345	49818 → 12345 [ACK] Seq=2022065829 Ack=2490432540 Win=29312 Len=0

Frame 9: 454 bytes on wire (3632 bits), 454 bytes captured (3632 bits) on interface 0
 Ethernet II, Src: PcsCompu_7d:1b:f7 (08:00:27:7d:1b:f7), Dst: Giga-Byt_54:ec:28 (40:8d:5c:54:ec:28)
 Internet Protocol Version 4, Src: 10.0.0.10, Dst: 10.0.0.7
 Transmission Control Protocol, Src Port: 49818, Dst Port: 12345, Seq: 2022065428, Ack: 2490432538, Len: 400
 Source Port: 49818
 Destination Port: 12345
 [Stream index: 0]
 [TCP Segment Len: 400]
 Sequence number: 2022065428
 [Next sequence number: 2022065828]
 Acknowledgment number: 2490432538
 0101 = Header Length: 20 bytes (5)
 Flags: 0x018 (PSH, ACK)
 Window size value: 229
 [Calculated window size: 29312]
 [Window size scaling factor: 128]
 Checksum: 0x15bb [unverified]
 [Checksum Status: Unverified]
 Urgent pointer: 0
 [SEQ/ACK analysis]
 [Timestamps]
 TCP payload (400 bytes)
 Data (400 bytes)

0020 00 07 c2 9a 30 39 78 86 45 14 94 70 fc 1a 50 18 ...09x. E..p..P
 0030 00 e5 15 bb 00 00 41 41 41 41 41 41 41 41 41 41AA AAAAAAAAAA
 0040 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAA AAAAAAAAAA
 0050 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAA AAAAAAAAAA
 0060 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAA AAAAAAAAAA
 0070 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAA AAAAAAAAAA
 0080 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAA AAAAAAAAAA
 0090 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAA AAAAAAAAAA
 00a0 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAA AAAAAAAAAA
 00b0 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAA AAAAAAAAAA
 00c0 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAA AAAAAAAAAA
 00d0 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAA AAAAAAAAAA

Flags (12 bits) (tcp.flags), 2 bytes Packets: 27 · Displayed: 23 (85.2%) · Dropped: 0 (0.0%) Profile: Default

ניתן לראות את תהליך ה-Syn-Ack בהתחלה, ולאחר מכן תהליך שליחת החבילות. אפשר לראות כאן בעצם דברים סותרים בצורה מסימת. מצד אחד בעת ה-Syn-Ack הלקוח מקבל מידע שה-MSS של השרת הוא 1460, ואילו הלקוח יוזם ושולח חבילות בגודל 7300. ההסבר לתופעה היא שבמקום שמערכת ההפעלה תפרק את החבילה לפי ה-MSS ותעביר את החבילות המפורקות לכרטיס הרשת לשליחה, הפירוק מתבצע בכרטיס הרשת עצמו, כך שה WIRESHARK אינו מספיק "לתפוס" את החבילות המפורקות לפי שהן כבר יוצאות. עם זאת, ניתן עדיין לראות את השינוי בSeq Number בין שתי החבילות של ה-7300, שההפרש ביניהן הוא גודל החבילה. השרת מחזיר Ack על גודל החבילות שהתקבל, ואז הקליינט שולח את החבילה האחרונה בגודל 400, ו מקבל גם לעיה ACK.

בסופו של דבר, לאחר אישור קבלת כל החבילות (לפי מספר ה-Ack) ניתן לראות שה-Ack לחבילה האחרונה, מספרו 2022065828, כמספר ה-Seq של החבילה האחרונה שנשלחה מהקליינט (בגודל 400, שה-Seq שלה הוא 2022065428) + גודלה, 400 בתים.

Wireshark · Flow · enp0s3

Time	10.0.0.10	10.0.0.7	Comment
1.210370631	49818	12345	Seq = 2022050827 SYN, ECN, CWR
1.210615703	49818	12345	Seq = 2490432537 Ack = 2022050828 SYN, ACK
1.210630199	49818	12345	Seq = 2022050828 Ack = 2490432538 ACK
1.210694889	49818	12345	Seq = 2022050828 Ack = 2490432538 ACK - Len: 7300
1.210704382	49818	12345	Seq = 2022058128 Ack = 2490432538 ACK - Len: 7300
1.210757170	49818	12345	Seq = 2490432538 Ack = 2022052288 ACK
1.210762910	49818	12345	Seq = 2022065428 Ack = 2490432538 PSH, ACK - Len: 400
1.210772731	49818	12345	Seq = 2490432538 Ack = 2022053748 ACK
1.210774571	49818	12345	Seq = 2490432538 Ack = 2022055208 ACK
1.210775785	49818	12345	Seq = 2490432538 Ack = 2022056668 ACK
1.210876079	49818	12345	Seq = 2490432538 Ack = 2022058128 ACK
1.210879651	49818	12345	Seq = 2490432538 Ack = 2022059588 ACK
1.210881012	49818	12345	Seq = 2490432538 Ack = 2022061048 ACK
1.210882251	49818	12345	Seq = 2490432538 Ack = 2022062508 ACK
1.210883494	49818	12345	Seq = 2490432538 Ack = 2022063968 ACK
1.210885010	49818	12345	Seq = 2490432538 Ack = 2022065428 ACK
1.210886888	49818	12345	Seq = 2490432538 Ack = 2022065828 ACK
1.217401126	49818	12345	Seq = 2490432538 Ack = 2022065828 PSH, ACK - Len: 1
1.217416530	49818	12345	Seq = 2022065828 Ack = 2490432539 ACK
1.217530580	49818	12345	Seq = 2022065828 Ack = 2490432539 FIN, ACK
1.217572915	49818	12345	Seq = 2490432539 Ack = 2022065829 ACK
1.217844936	49818	12345	Seq = 2490432539 Ack = 2022065829 FIN, ACK
1.217854175	49818	12345	Seq = 2022065829 Ack = 2490432540 ACK

Packet 3: Seq = 2022050827

☒ Limit to display filter Flow type: TCP Flows Addresses: Any

זהו ה-Flow Graph של התעבורה, כפי שנאמר הפירוק לסמגנטים קטנים יותר אינו נראה כאן כיוון שהפירוק התבצע בכרטיס הרשת (אולם עדיין בוצע פירוק מסוים לחבילות של 7300, 7300, 400)

סעיף ב TCP

The image shows a Wireshark capture of a TCP stream. The top pane displays a list of packets with columns for No., Time, Source, Destination, Protocol, Length, Source Port, and Destination Port. The middle pane shows the details of the selected packet (No. 325), including Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol. The bottom pane shows the raw packet data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Source Port	Dest. Port	Info
19	2.157026545	10.0.0.7	10.0.0.10	TCP	66	12345	51324	[SYN, ACK] Seq=3245010832 Ack=1857157348 Win=65535
20	2.157042725	10.0.0.10	10.0.0.7	TCP	54	51324	12345	[ACK] Seq=1857157348 Ack=3245010833 Win=29312 Len=0
21	2.157181169	10.0.0.10	10.0.0.7	TCP	55	51324	12345	[PSH, ACK] Seq=1857157348 Ack=3245010833 Win=29312 Len=0
22	2.157232360	10.0.0.7	10.0.0.10	TCP	60	12345	51324	[ACK] Seq=3245010833 Ack=1857157349 Win=525568 Len=0
23	2.157238187	10.0.0.10	10.0.0.7	TCP	55	51324	12345	[PSH, ACK] Seq=1857157349 Ack=3245010833 Win=29312 Len=0
24	2.157307343	10.0.0.7	10.0.0.10	TCP	60	12345	51324	[ACK] Seq=3245010833 Ack=1857157350 Win=525568 Len=0
33	4.159922924	10.0.0.10	10.0.0.7	TCP	55	51324	12345	[PSH, ACK] Seq=1857157350 Ack=3245010833 Win=29312 Len=0
34	4.160078790	10.0.0.7	10.0.0.10	TCP	60	12345	51324	[ACK] Seq=3245010833 Ack=1857157351 Win=525568 Len=0
35	4.160089963	10.0.0.10	10.0.0.7	TCP	55	51324	12345	[PSH, ACK] Seq=1857157351 Ack=3245010833 Win=29312 Len=0
36	4.160177389	10.0.0.7	10.0.0.10	TCP	60	12345	51324	[ACK] Seq=3245010833 Ack=1857157352 Win=525568 Len=0
37	4.160688781	10.0.0.7	10.0.0.10	TCP	60	12345	51324	[PSH, ACK] Seq=3245010833 Ack=1857157352 Win=525568 Len=0
38	4.160694138	10.0.0.10	10.0.0.7	TCP	54	51324	12345	[ACK] Seq=1857157352 Ack=3245010834 Win=29312 Len=0
54	6.162936886	10.0.0.10	10.0.0.7	TCP	55	51324	12345	[PSH, ACK] Seq=1857157352 Ack=3245010834 Win=29312 Len=0
55	6.163766784	10.0.0.7	10.0.0.10	TCP	60	12345	51324	[ACK] Seq=3245010834 Ack=1857157353 Win=525568 Len=0
56	6.163820496	10.0.0.10	10.0.0.7	TCP	55	51324	12345	[PSH, ACK] Seq=1857157353 Ack=3245010834 Win=29312 Len=0
57	6.163858643	10.0.0.7	10.0.0.10	TCP	60	12345	51324	[PSH, ACK] Seq=3245010834 Ack=1857157353 Win=525568 Len=0
58	6.163864666	10.0.0.10	10.0.0.7	TCP	54	51324	12345	[ACK] Seq=1857157354 Ack=3245010835 Win=29312 Len=0
59	6.164018408	10.0.0.7	10.0.0.10	TCP	60	12345	51324	[ACK] Seq=3245010835 Ack=1857157354 Win=525568 Len=0
60	6.165079179	10.0.0.7	10.0.0.10	TCP	60	12345	51324	[PSH, ACK] Seq=3245010835 Ack=1857157354 Win=525568 Len=0
61	6.165100189	10.0.0.10	10.0.0.7	TCP	54	51324	12345	[ACK] Seq=1857157354 Ack=3245010836 Win=29312 Len=0
89	8.166281506	10.0.0.10	10.0.0.7	TCP	55	51324	12345	[PSH, ACK] Seq=1857157354 Ack=3245010836 Win=29312 Len=0
90	8.167042711	10.0.0.7	10.0.0.10	TCP	60	12345	51324	[ACK] Seq=3245010836 Ack=1857157355 Win=525568 Len=0
91	8.167072962	10.0.0.10	10.0.0.7	TCP	55	51324	12345	[PSH, ACK] Seq=1857157355 Ack=3245010836 Win=29312 Len=0
92	8.167131857	10.0.0.7	10.0.0.10	TCP	60	12345	51324	[PSH, ACK] Seq=3245010836 Ack=1857157355 Win=525568 Len=0
93	8.167141246	10.0.0.10	10.0.0.7	TCP	54	51324	12345	[ACK] Seq=1857157356 Ack=3245010837 Win=29312 Len=0
94	8.167711838	10.0.0.7	10.0.0.10	TCP	60	12345	51324	[ACK] Seq=3245010837 Ack=1857157356 Win=525568 Len=0
95	8.177779874	10.0.0.10	10.0.0.7	TCP	60	12345	51324	[PSH, ACK] Seq=3245010837 Ack=1857157356 Win=525568 Len=0
96	8.177819860	10.0.0.10	10.0.0.7	TCP	54	51324	12345	[ACK] Seq=1857157356 Ack=3245010838 Win=29312 Len=0
116	10.171400368	10.0.0.10	10.0.0.7	TCP	55	51324	12345	[PSH, ACK] Seq=1857157356 Ack=3245010838 Win=29312 Len=0
117	10.171932177	10.0.0.7	10.0.0.10	TCP	60	12345	51324	[ACK] Seq=3245010838 Ack=1857157357 Win=525568 Len=0
118	10.171964427	10.0.0.10	10.0.0.7	TCP	55	51324	12345	[PSH, ACK] Seq=1857157357 Ack=3245010838 Win=29312 Len=0
119	10.172022794	10.0.0.7	10.0.0.10	TCP	60	12345	51324	[PSH, ACK] Seq=3245010838 Ack=1857157357 Win=525568 Len=0

Frame 325: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
 Ethernet II, Src: PcsCompu_7d:1b:f7 (08:00:27:7d:1b:f7), Dst: Giga-Byt_54:ec:28 (40:8d:5c:54:ec:28)
 Internet Protocol Version 4, Src: 10.0.0.10, Dst: 10.0.0.7
 Transmission Control Protocol, Src Port: 51324, Dst Port: 12345, Seq: 1857157371, Len: 0

0000 40 8d 5c 54 ec 28 08 00 27 7d 1b f7 08 00 45 00 @ \T (. . .) E .
 0010 00 28 00 00 40 00 40 06 26 c0 0a 00 00 0a 0a 00 (. &
 0020 00 07 c8 7c 30 39 6e b1 f8 fb 00 00 00 00 50 04 . . . | 09n - p .
 0030 00 00 3b 6d 00 00 . . . m . . .

- לקראת הסוף הייתה קפיצה של חבילה עם פלאג RST, כלומר שהחיבור התנתק באופן לא צפוי בזמן שלקליינט עדיין יש מה לשלוח (הוא זה שיזם את בקשת הRST) אבל מפאת חוסר זמן לא הספקתי לתקן את הבאג, הבעיה היא בספירת מספרי הA שהתקבלו לצורך התנתקות חלקה יותר מהשרת.

בכל מקרה, ניתן לראות שבמקרה הזה אין סגמנטציה של ההודעה ויש תקשורת דו כיוונית בין השרת ללקוח. על 2 הA הראשונים השרת רק מחזיר ACK ולא יותר (כפי שנדרש) לאחר מכן אחרי קבלת שני A השרת מחזיר B (כמובן שלכל A הוא מחזיר ACK) כאשר הקליינט שולח A הוא מעלה כל פעם את הSEQ שלו ב1, והשרת מחזיר ACK NUMBER שהוא הSEQ שנשלח אליו + גודל הPAYLOAD שהוא 1.

סעיף א UDP

The screenshot displays the Wireshark network protocol analyzer interface. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The top toolbar contains various icons for file operations, capture control, and analysis. The main display area is divided into three panes:

- Packet List Pane (Top):** Shows a list of captured packets. The selected packet is #48, which is a UDP packet from 10.0.0.10 to 10.0.0.10. The packet is fragmented and has a length of 1514 bytes. The details pane shows it is a UDP packet with a source port of 52486 and a destination port of 12345.
- Packet Details Pane (Middle):** Shows the hierarchical structure of the selected packet. It includes Ethernet II (Source: PcsCompu7d:1b:f7, Destination: Giga-Byt_54:ec:28), Internet Protocol Version 4 (Source: 10.0.0.10, Destination: 10.0.0.7), and Data (1480 bytes). The data field is expanded, showing a hexadecimal dump and its corresponding ASCII representation.
- Packet Bytes Pane (Bottom):** Shows the raw packet data in hexadecimal and ASCII. The data is displayed as a series of hexadecimal values and their corresponding ASCII characters.

The status bar at the bottom indicates that 69 packets are displayed, with 12 (17.4%) shown and 0 (0.0%) dropped. The profile is set to Default.

ניתן לראות כאן את הפירוק לפרגמנטים של החבילה ששלחנו. ניתן לראות כי הoffset הוא בקפצות של 1480, אם נחלק אותו ב8 נקבל את האופסט האמיתי. כמו כן אפשר לראות את הID של החבילה (שבכל פרגמנט זהה, הרי הם שייכים כולם לאותה חבילה) אם ניקח לדוגמה את הפרגמנט הראשון

```

0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 1500
Identification: 0x6447 (25671)
▶ Flags: 0x2000, More fragments
Time to live: 64
Protocol: UDP (17)
Header checksum: 0xdcb9 [validation disabled]
[Header checksum status: Unverified]
Source: 10.0.0.10
Destination: 10.0.0.7
Reassembled IPv4 in frame: 48
Data (1480 bytes)

```

ניתן לראות שדלוק הדלק של More Fragments, כלומר יש עוד פרגמנטים לשלוח אחריו.

*קובץ pcap המצורף לחלק הזה שונה מהתצלום כיוון ששכחתי לשמור את pcap שצילמתי כאן, אבל ההרצה של שניהם היא לפי אותם קבצי שרת וקליינט.

סעיף ב UDP

The image shows a Wireshark packet capture window titled "enp0s3". The packet list pane displays a series of 20 UDP packets. The selected packet (No. 20) is expanded in the packet details pane, showing the following structure:

- Frame 20: 43 bytes on wire (344 bits), 43 bytes captured (344 bits) on interface 0
- Ethernet II, Src: PcsCompu_7d:1b:f7 (08:00:27:7d:1b:f7), Dst: Giga-Byt_54:ec:28 (40:8d:5c:54:ec:28)
- Internet Protocol Version 4, Src: 10.0.0.10, Dst: 10.0.0.7
- User Datagram Protocol, Src Port: 37118, Dst Port: 12345
 - Source Port: 37118
 - Destination Port: 12345
 - Length: 9
 - Checksum: 0x142b [unverified]
 - [Checksum Status: Unverified]
 - [Stream index: 3]
- Data (1 byte)
 - Data: 41
 - [Length: 1]

The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII. The ASCII column displays the characters "@.T.(. ')....E:" for the first few bytes.

כעת אנחנו שולחים את ה A באינטרוולים דרך UDP. במצב זה לא מתבצעת פרגמנטציה כיוון שהחבילה מספיק קטנה כדי לעבור באופן ישיר.

ניתן לראות בוויירשארק שנשלח B מהשרת אחרי כל הודעת A ואז A בהתאמה, לא כולל הפעם הראשונה ששלחו AA.

לעומת TCP, כאן כל הודעה מועבדת בפני עצמה ואין מצב שבו נקבל למשל AA או BB כמו ב TCP.