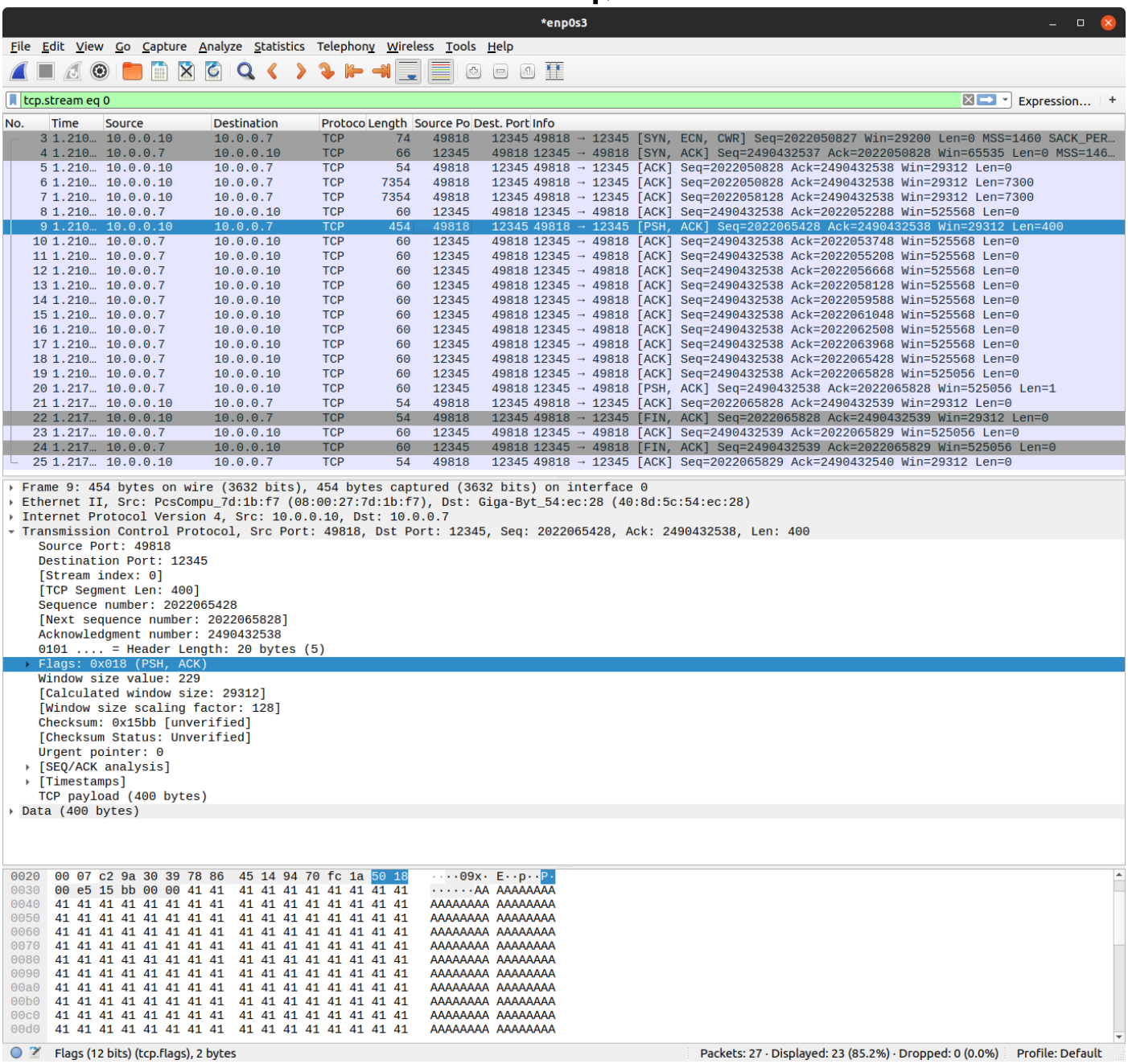
מבוא לרשתות תקשורת – תרגיל 2

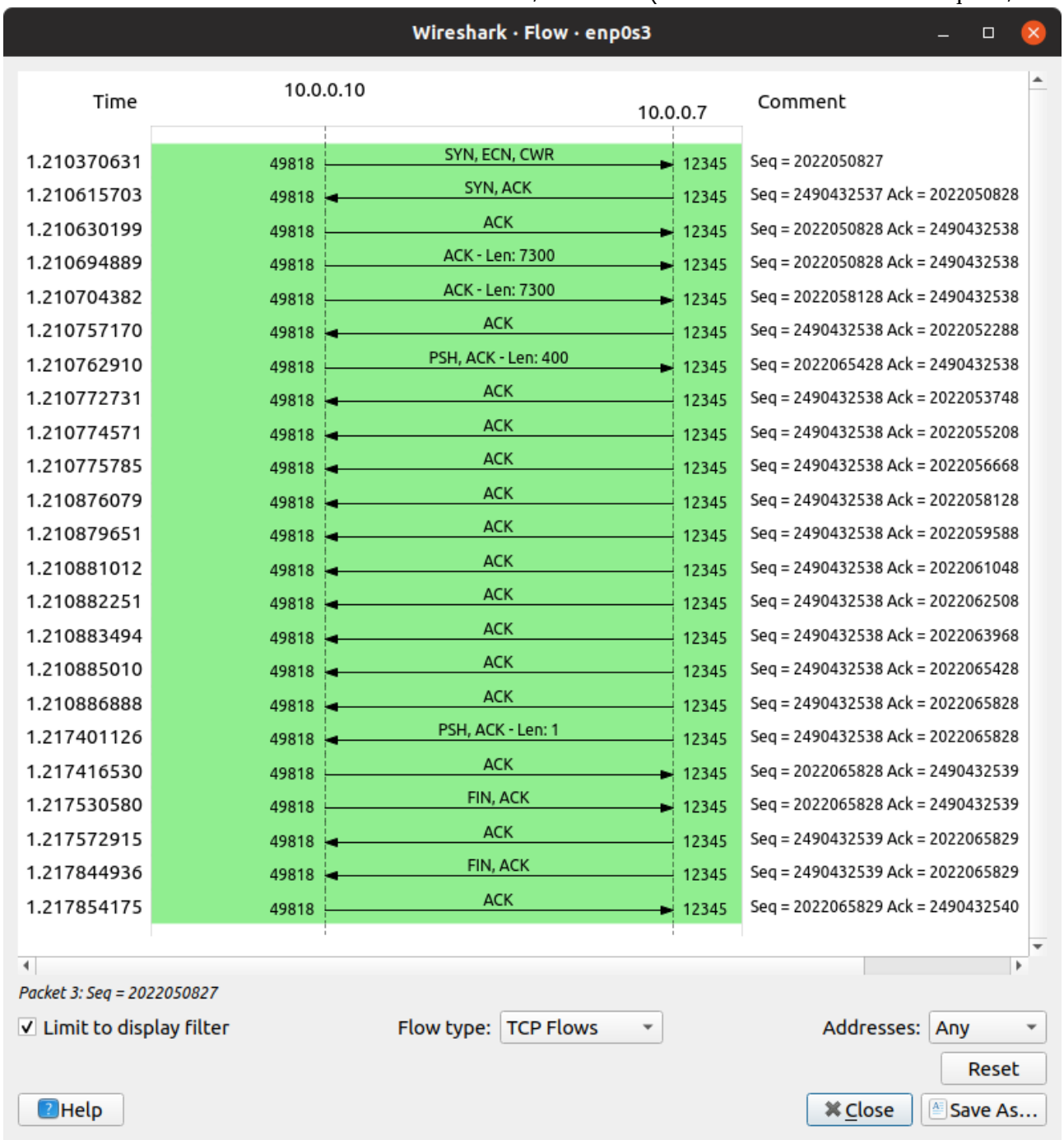
שאלה 2:

סעיף א TCP



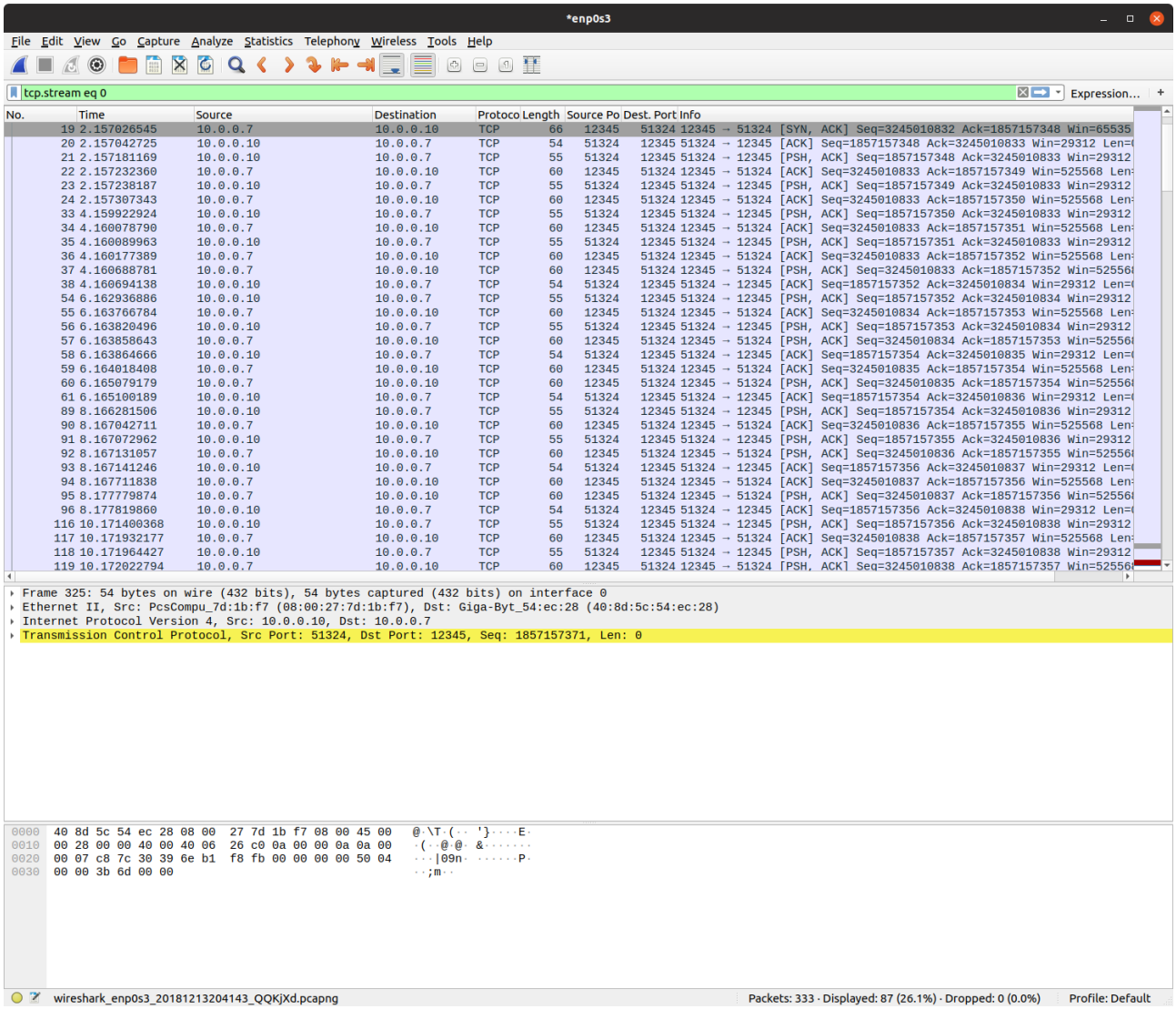
ניתן לראות את תהליך הSyn-Ack בהתחלה, ולאחר מכן תהליך שליחת החבילות.
אפשר לראות כאן בעצם דברים סותרים בצורה מסוימת. מצד אחד בעת הSyn-Ack הלקוח מקבל מידע שהMSS של השרת הוא 1460, ואילו הלקוח יוזם ושולח חבילות בגודל 7300.
ההסבר לתופעה היא שבמקום שמערכת ההפעלה תפרק את החבילה לפי הMSS ותעביר את החבילות המפורקות 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 ה7300, 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 הAck) ניתן לראות שהAck לחבילה האחרונה, מספרו 2022065828, כמספר הSeq של החבילה האחרונה שנשלחה מהקליינט (בגודל 400, שהSeq שלה הוא 2022065428) + גודלה, 400 בתים.



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

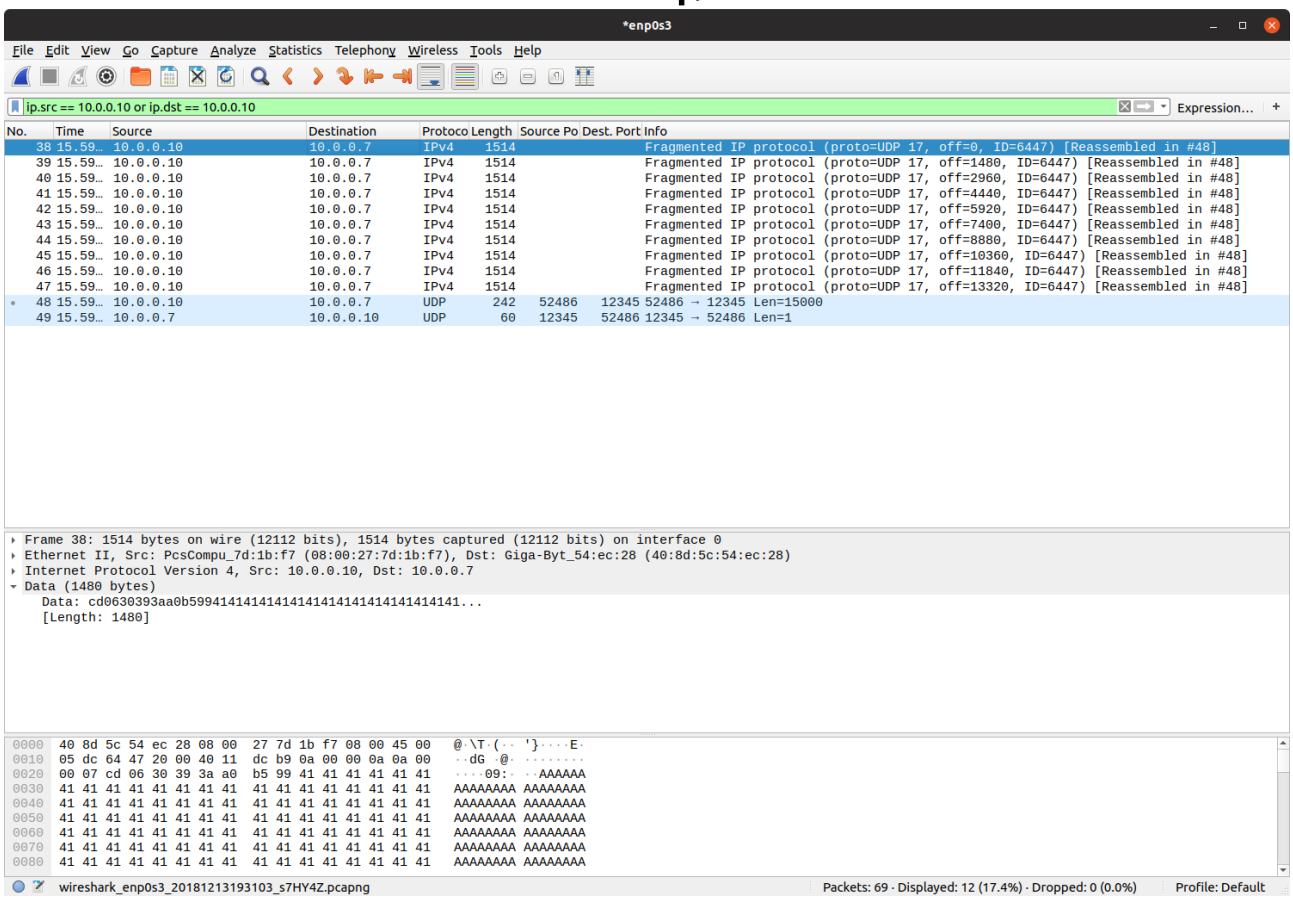
סעיף ב TCP



– 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



ניתן לראות כאן את הפירוק לפרגמנטים של החבילה ששלחנו. ניתן לראות כי הoffset הוא בקפיצות 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)

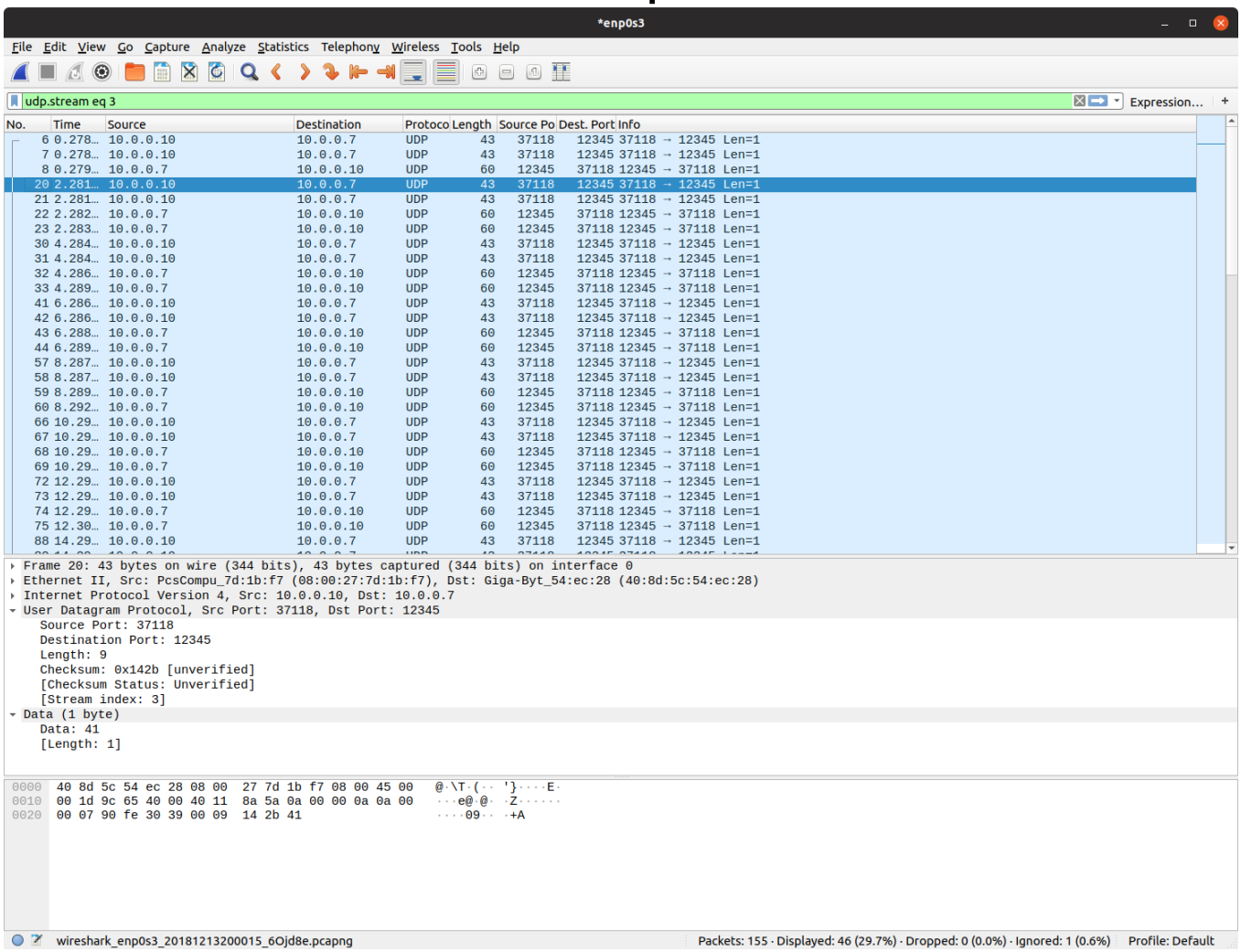
אם ניקח לדוגמה את הפרגמנט הראשון

```
    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
```

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

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

סעיף ב UDP



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

ניתן לראות בווירשארק 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 בTCP.