תרגיל 2

שאלות תיאורטיות

1. הריצו את התרגיל ובקשו בקשות GetClientToServerDelayEstimation רבות. האם ייתכן שהלקוח או השרת "ייתקעו"? אם לא, נמקו. אם כן, נמקו והציעו תיקון (ללא מימוש) בתוכנית שלכם בכדי שהלקוח והשרת לא "יתקעו".

<u>תשובה:</u>

אכן ייתכן כי השרת או הלקוח ייתקעו.

כמו שלמדנו, פרוטוקול UDP הנו פרוטוקול לא אמין ולא ניתן לדעת האם ההודעה הגיעה או לא או האם תקינה או לא.

בצד הלקוח ייתכן וכי ייתקע כי יחכה לתשובה מצד השרת שלא תגיע. (ייתכנו מספר סיבות שהודעה זו לא תגיע - למשל פאקטה שאבדה) ולכן ייתקע.

אחד הפתרונות הוא מעבר לפרוטוקול TCP שהרי פרוטוקול זה הוא פרוטוקול אמין הפותר סוגיות ובעיות אלו. בנוסף, TCP שהרי מאפשר שירותי non blocking. חשוב לציין כי אמנם מעבר לפרוטוקול TCP יפתור בעיה זו אך יגרע בדברים אחרים כמו שנלמד בשיעור כגון: מהירות וביצועים).

2 מבחינת ההשהיות השונות שאנו כבר מכירים (אותן 4 השהיות שנלמדו בכיתה), מה מבטא הגודל הממוצע אותו הלקוח מחשב על סמך תשובות השרת בבקשת GetClientToServerDelayEstimation? (המלצה: נתחו גודל זה מבחינה תאורטית ולאו דווקא על סמך התוצאות המתקבלות מההרצה בפועל שעלולות להטעות, בפרט כאשר מבוצעות על אותו המחשב).

<u>: תשובה</u>

הגודל הממוצע מבטא את סכום ה – Transmition/Process Delay.

האפליקציה מחשבת את הפרש הזמן בין הודעה לתשובה וממצעת את ההפרשים של כל הבקשות.

תיאור האפליקציה

<u>פרוטוקול</u>

סוג פרוטוקול התעבורה הוא UDP

:Client Side

בצד הלקוח אנו נציג תפריט עם 13 אפשרויות למשתמש (כולל יציאה).

כאשר האפליקציה תחכה לבחירתו של הלקוח ותפעל בהתאם לבחירת הלקוח – תשלח את הבקשה לשרת (מספר הבחירה של הלקוח).

במקרה בו הלקוח יבחר בתפריט את אפשרות 12 – ייפתח לו תפריט בחירה נוסף בו יצטרך לבחור עיר מסוימת. אפשרויות הבחירה של המשתמש יהיו בין המספרים 14-18 ואותה בחירה תשלח לשרת כך שלא יהיו כפילויות בין בקשות ולכל בקשה יש את המספר המזהה שלה. במידה והלקוח יזין מספר שאינו תואם לבחירות התפריט תוצג לו הודעת שגיאה ידידותית ויצטרך להזין מחדש את בחירתו

:התפריט ייראה בצורה הבאה

- . מπזיר את הזמן בתצורה של שנה, חודש, יום, שעה, דקה ושניה. Get Time .1
- . מחזיר את הזמן בתצורה של שעה, דקה ושניה ללא תאריך Get Time Without Date .2
 - .1.1.1970 מחזיר את הזמן בתצורה של שניות מאז Get Time Since Epoch .3
- -Get Client To Server Delay Estimation .4 החזר אומדן של ההשהיה בין הלקוח לשרת(ראו -Get Client To Server Delay הנחיות להלן)
 - (ראו הנחיות להלן) Round Trip Time מדידת Measure RTT.5
 - . מחזיר את הזמן בתצורה של שעה ודקה Get Time Without Date Or Seconds .6
 - .מחזיר את השנה בלבד Get Year .7
 - . מחזיר את היום והחודש Get Month And Day .8
- . מחזיר כמה שניות עברו מאז תחילת החודש הנוכחי Get Seconds Since Begining Of Month .9
- Get How Many Days Were Passed since Begining Of Current Year .10 מחזיר את מספר הימים שעברו מאז תחילת השנה הנוכחית.
 - אם שעון 0 אם שעון חורף 0 אם שעון חורף Get Day Light Savings .11
- Get Time Without Date In City .12 פותח תפריט חדש עם מספר ערים בעולם ויש לבחור אחת מהאפשרויות להצגת השעה הנוכחית באותה עיר. (ראו המשך הנחיות).
 - בירה ויציאה מהאפליקציה. Exit .13

באפשרות 4 בתפריט (Get Client To Server Delay Estimation) יישלחו לשרת 100 בקשות ולאחר מכן תקלוט מהשרת 100 תשובות ותחשב את הנדרש ותציג זאת כתשובה ללקוח.

באפשרות 5 בתפריט (Measure RTT) תשלח בקשה ונחכה לתשובה של השרת, כך 100 פעמים זוגות של שליחה ובקשה. האפליקציה תחשב את הנדרש ותציג את התשובה ללקוח.

באפשרות 12 ייפתח תפריט נוסף המציג למשתמש ערים מסוימות בעולם לבחירה ובחירה של שעון UTC עולמי באופן הבא:

- .מחזיר את השעה הנוכחית בבייג'ינג, סין China Baijing .14
- בסידני, אוסטרליה. Australia Sydney .15
- USA Washington .16 מחזיר את השעה הנוכחית בוושינגטון, ארצות הברית.
 - . מחזיר את השעה הנוכחית ברומא, איטליה Italy Rome .17
 - .UTC מחזיר את השעה עם הזמן האוניברסלי המתואם UTC .18

ומכאן, לאחר שיבחר עיר מסוימת (אחת מאפשרויות 14-17), או שעון עם זמן אוניברסלי (18) יקבל תשובה מהשרת התואמת לבחירתו ובכל בחירה אחרת תוצג לו הודעת שגיאה ידידותית כי המספרים שהכניס אינם תקינים ויחזור על תהליך הבחירה שוב.

:Server Side

בצד השרת יהיה בנוי switch-case אשר יחכה תמיד לבחירת הלקוח. לכל אפשרות כזו תהיה פונקציה אחידה שתטפל בצורה ייחודית לבקשת הלקוח.

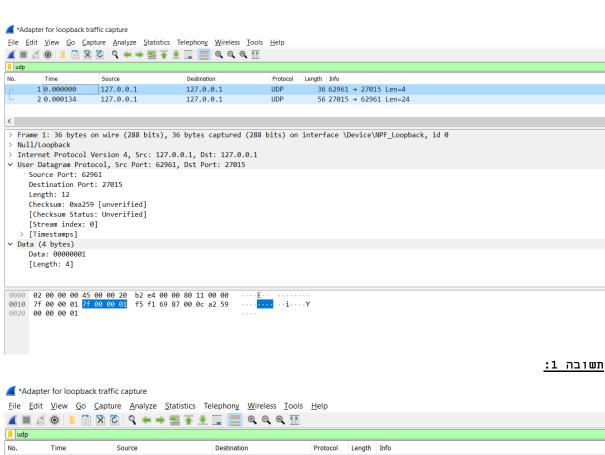
בבקשות 4/5 של הלקוח, השרת יחזיר 100 פעמים תשובות עבור הבקשות ללא ידיעה כי אמורות להגיע מהלקוח 100 בקשות רצופות.

השרת מטפל בבקשות עבור מספרים 1-11 ו- 14-18 (אופציה מספר 12 היא רק בצד הלקוח כדי לפתוח תפריט חדש עם אפשרויות נוספות 14-18 ואילו אופציה מספר 13 מטרתה סגירת האפליקציה).

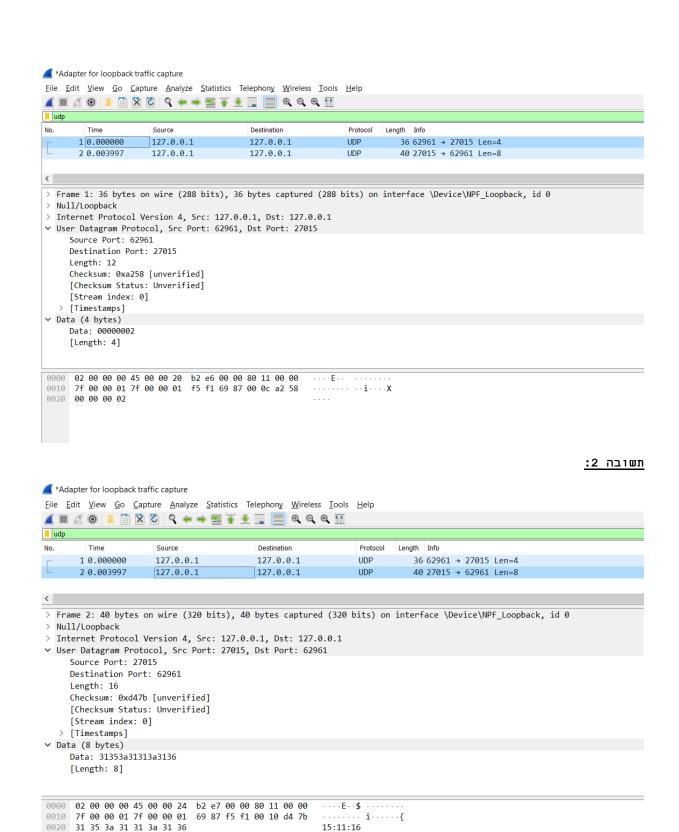
תשובת השרת לכל בקשה תהיה לכל היותר 255 בתים.

WIRESHARK תמונות מסך

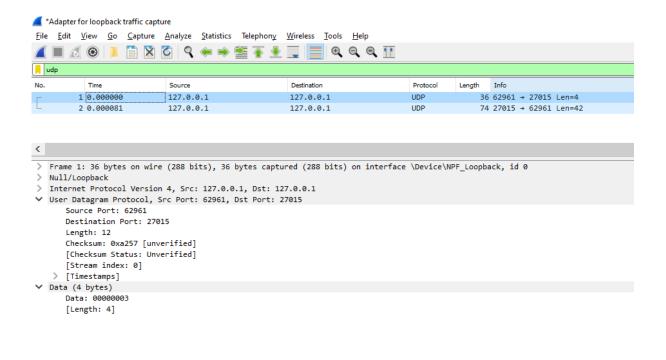
בקשה **1:**



```
1 0.000000
                    127.0.0.1
                                      127.0.0.1
                                                         UDP
                                                                  36 62961 → 27015 Len=4
      2 0.000134
                   127.0.0.1
                                      127.0.0.1
                                                         UDP
                                                                   56 27015 → 62961 Len=24
<
> Frame 2: 56 bytes on wire (448 bits), 56 bytes captured (448 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
∨ User Datagram Protocol, Src Port: 27015, Dst Port: 62961
    Source Port: 27015
    Destination Port: 62961
    Length: 32
    Checksum: 0xb583 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
  > [Timestamps]
V Data (24 bytes)
    Data: 4d6f6e204175672032342031353a30393a30322032303230
    [Length: 24]
     02 00 00 00 45 00 00 34 b2 e5 00 00 80 11 00 00
```

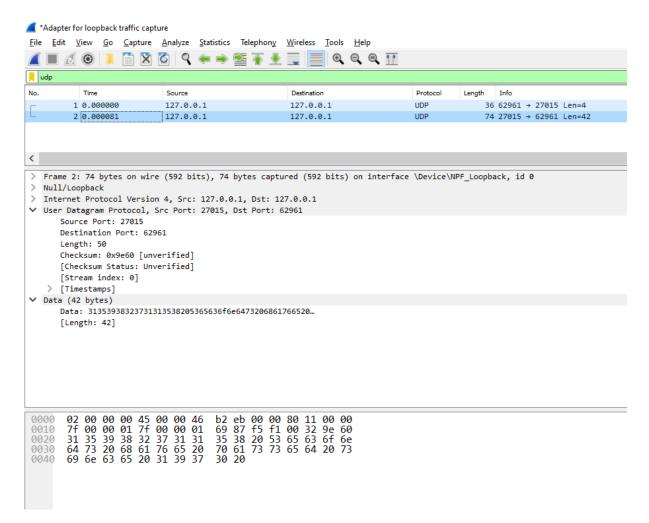


בקשה 3:

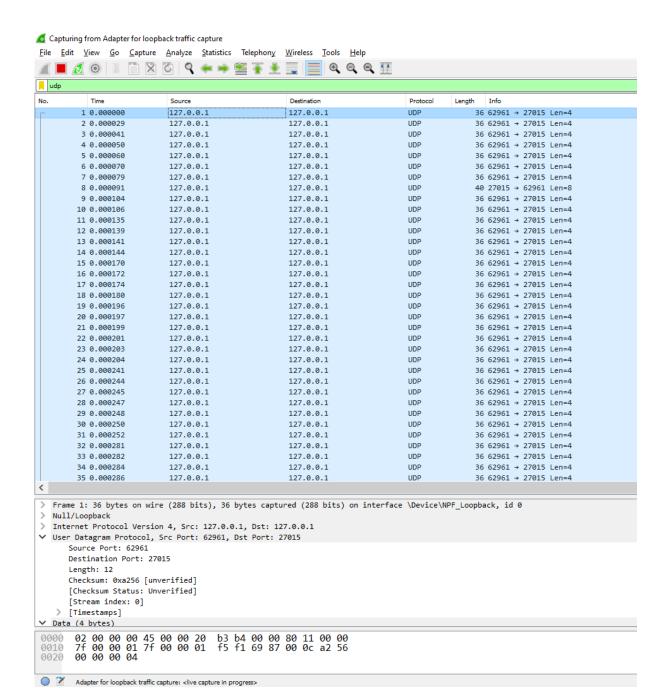


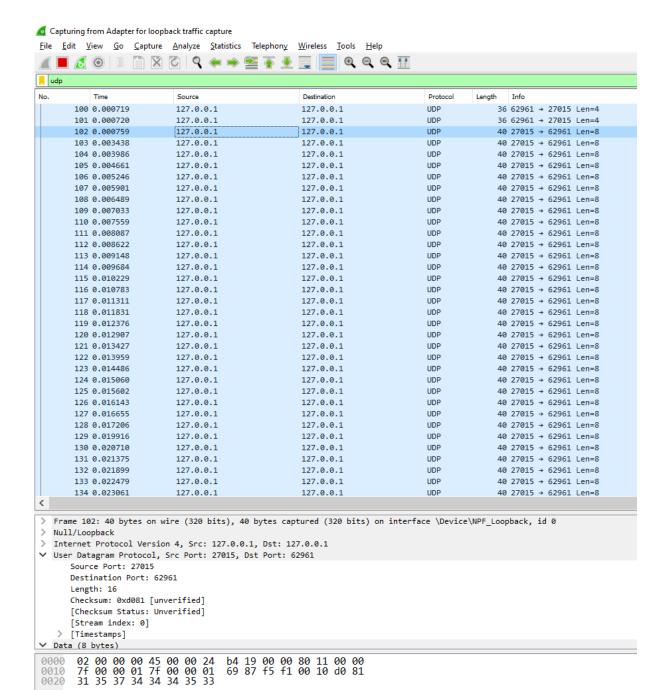
02 00 00 00 45 00 00 20 b2 ea 00 00 80 11 00 00 7f 00 00 01 7f 00 00 01 f5 f1 69 87 00 0c a2 57 00 00 00 03

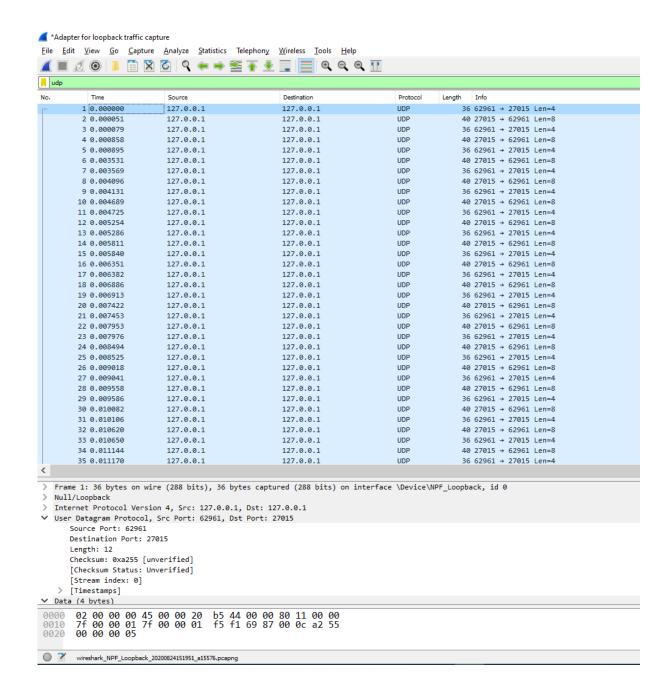
תשובה 3:



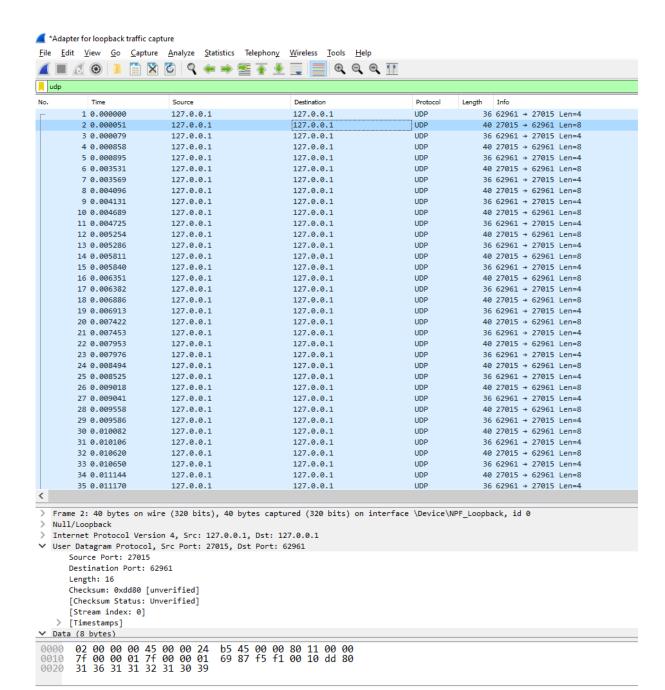
<u>:4</u> בקשה

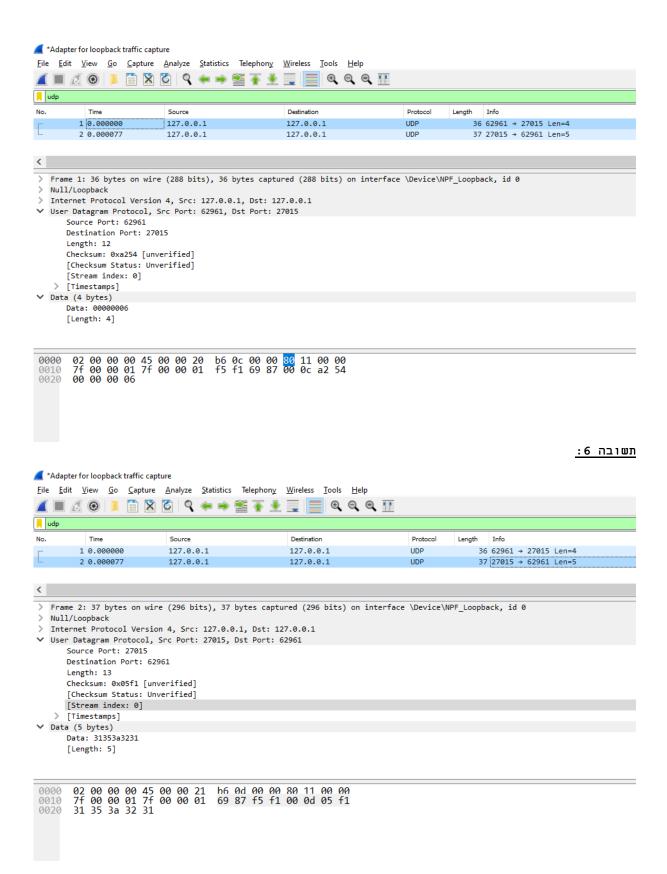


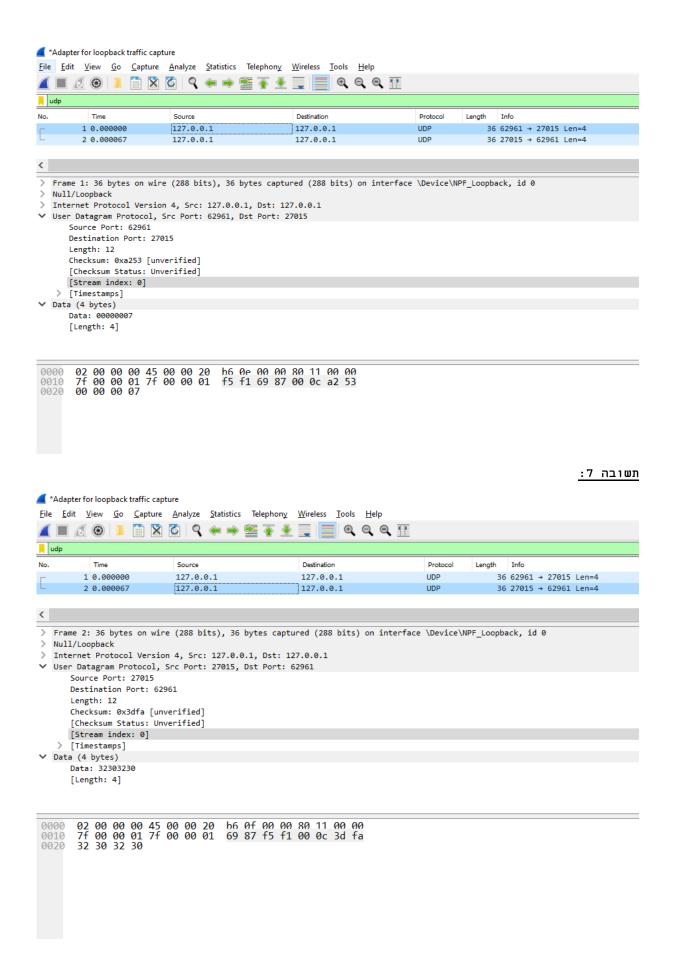


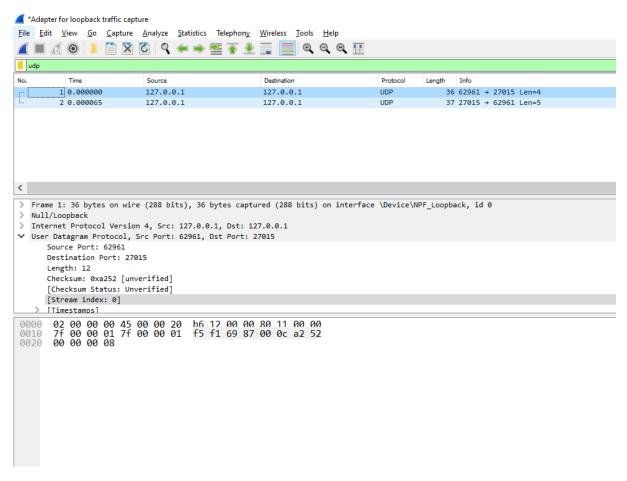


תשובה 5:

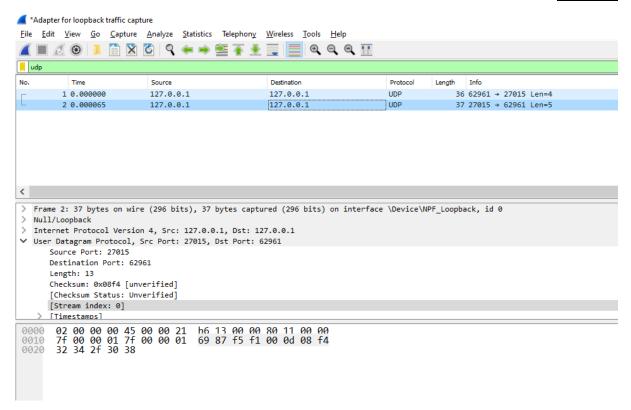


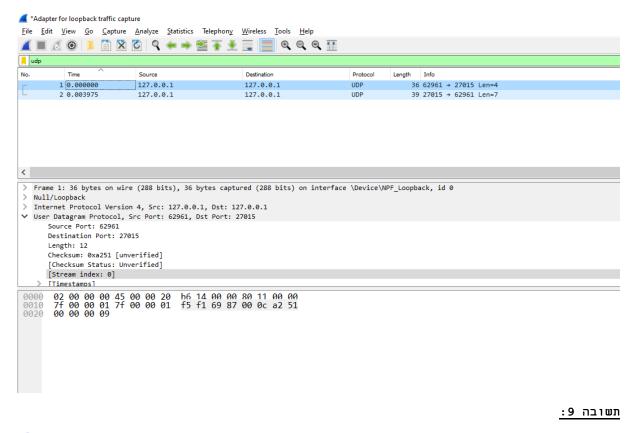


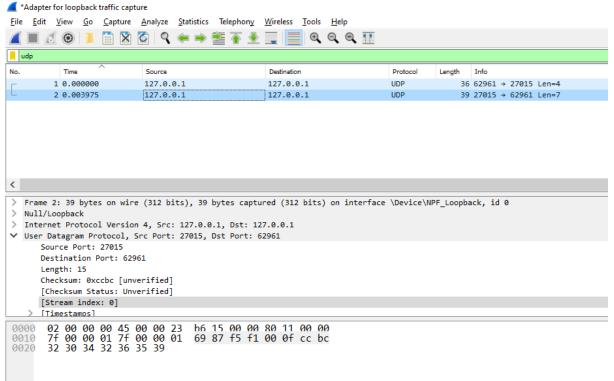


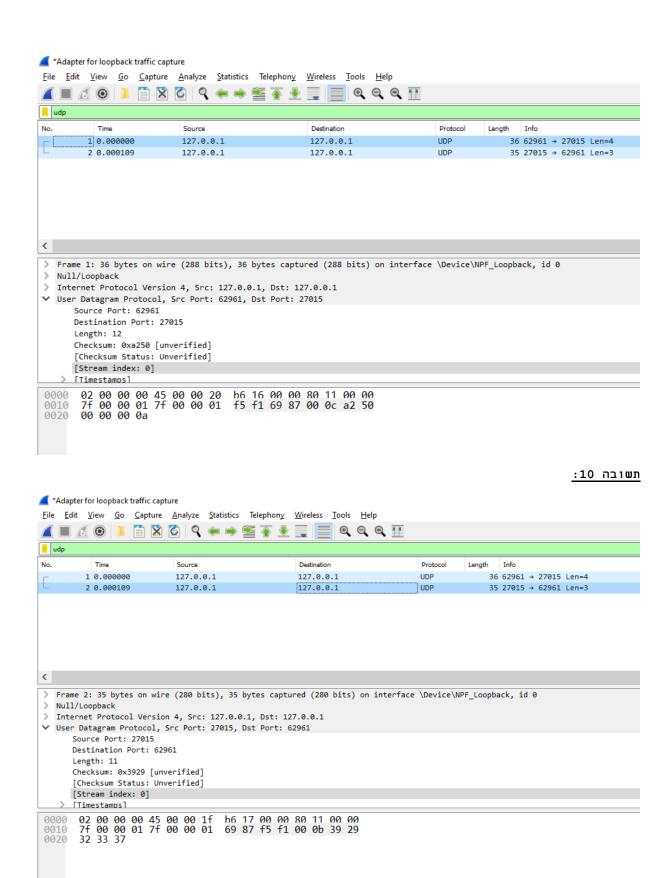


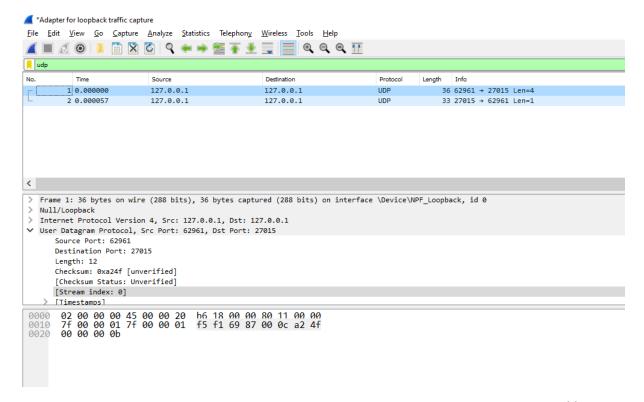
תשובה 8:



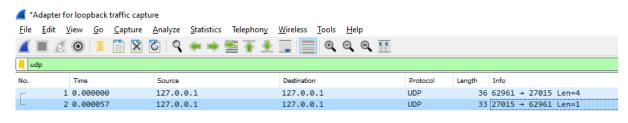


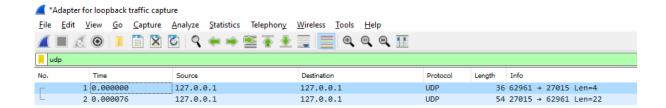


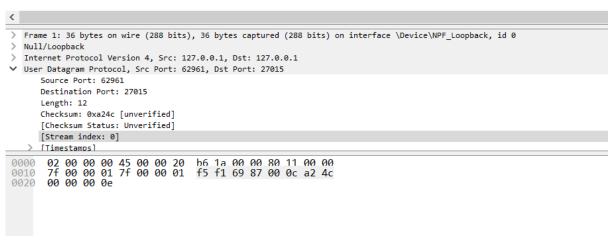




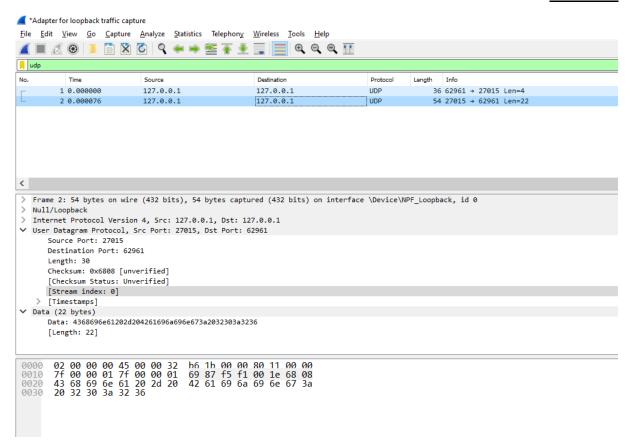
תשובה 11:

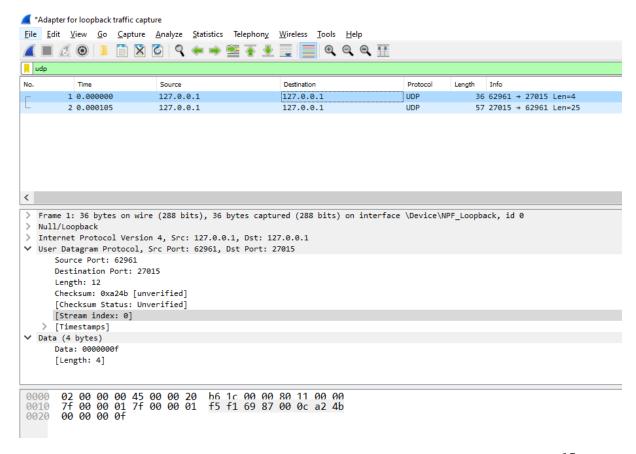






:14 תשובה

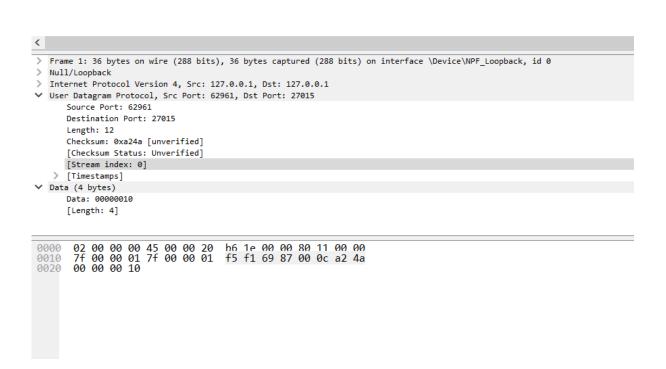




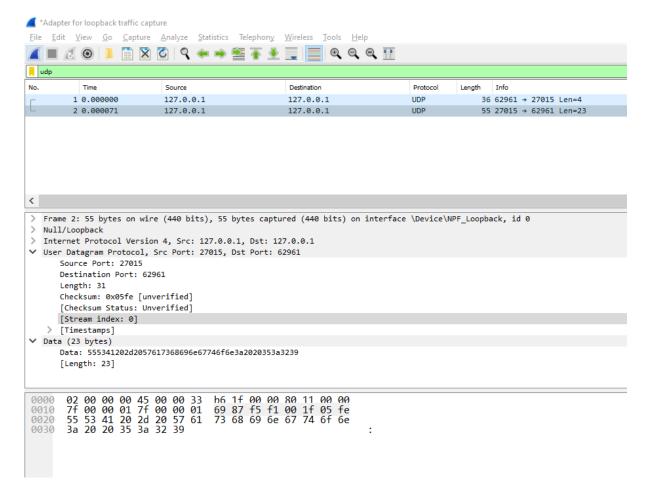
<u>תשובה 15:</u>



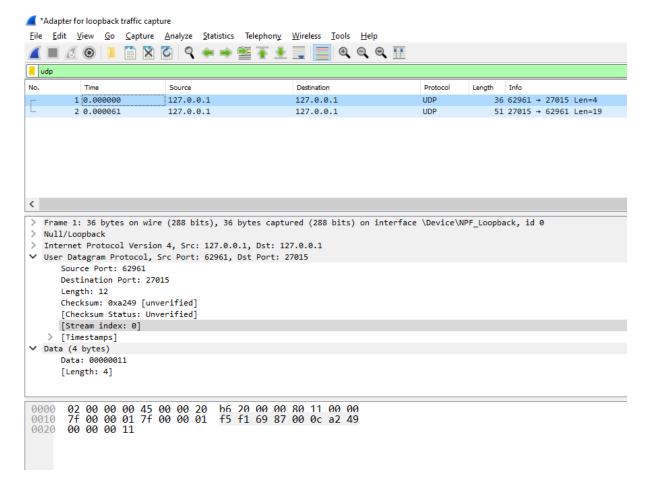




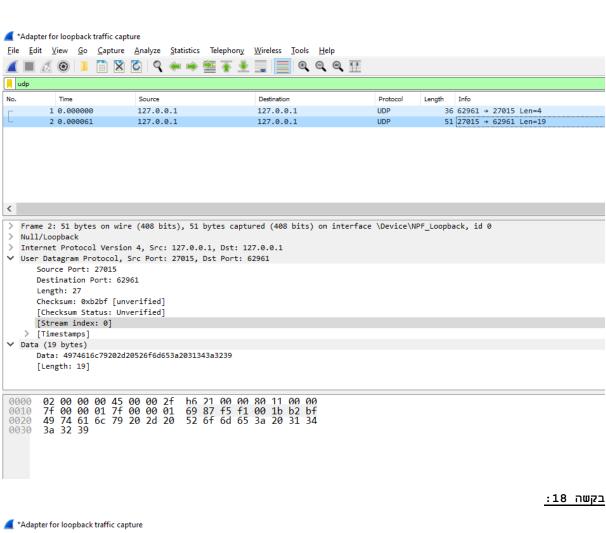
תשובה 16:

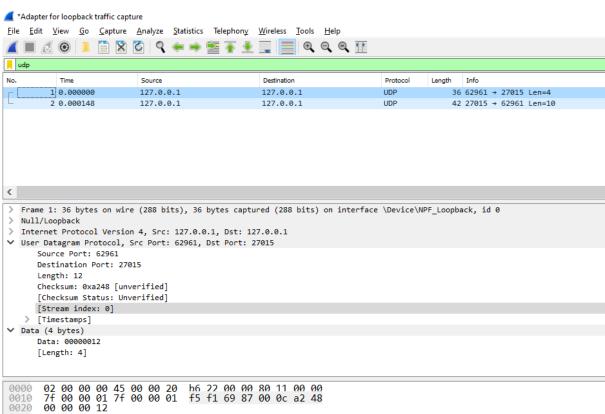


בקשה 17:



<u>תשובה 17:</u>





תשובה 18:

