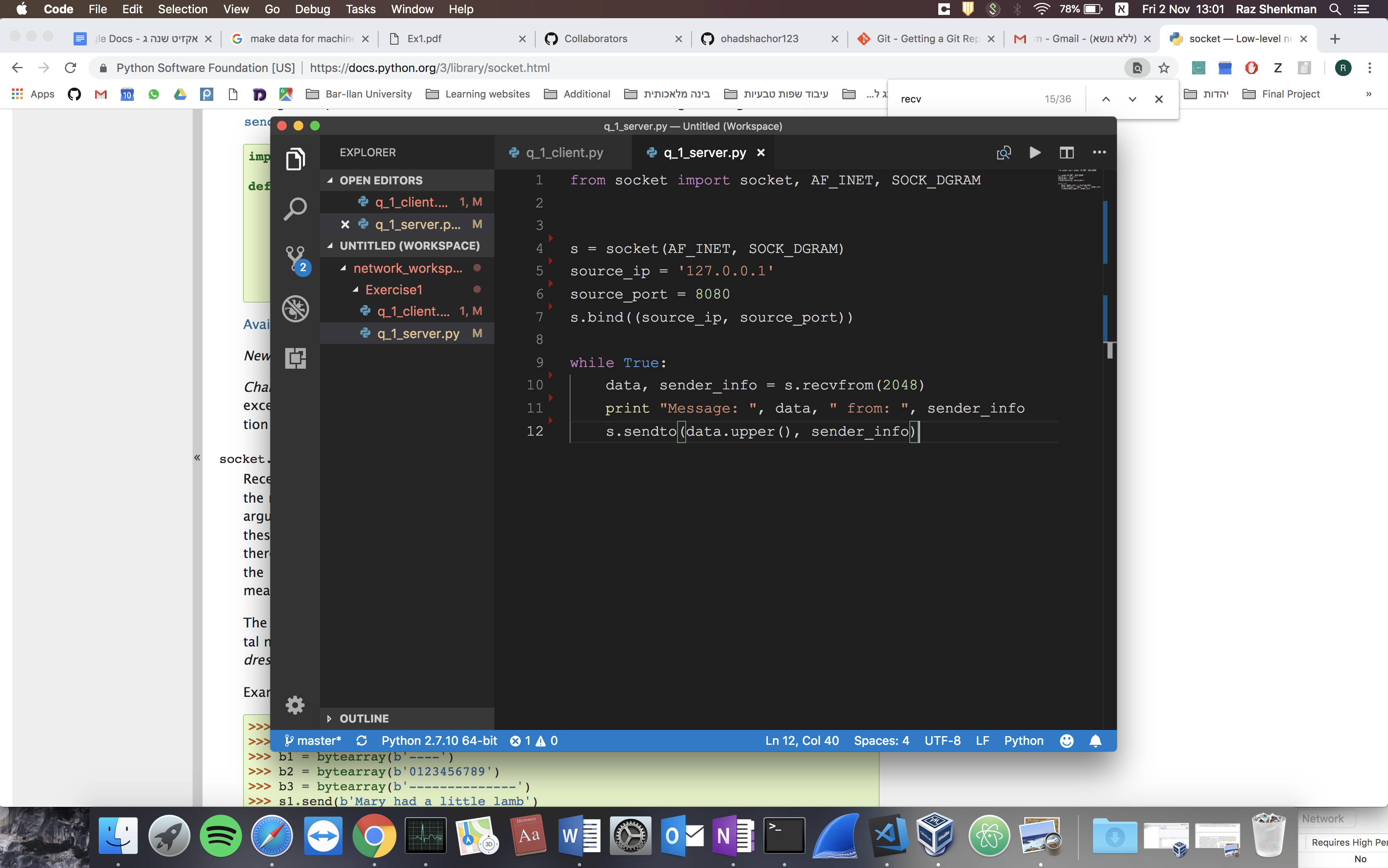
רשתות תקשורת תרגיל 1

אוהד שחור(322952433), רז שנקמן(311130777)

שאלה 1



קוד השרת:

**שורה 1:** מייבאים מספריית socket את המשאבים שנדרשים.

**שורה 4:** s הוא אובייקט מסוג socket כך ש:

Flag AF\_INET מסמן את הסוג protocol ip שהsocket יעבוד איתו, במקרה זה מדובר על ipV4. נשים לב שאם כתוב AF\_INET6 אז מתאימים את הsocket לעבוד עם ip protocol מסוג ipV6.

SOCK\_DGRAM- דגל שמסמן איזה פרוטוקול הsocket יעבוד איתו בשכבת התעבורה, כלומר tcp/udp/etc., כאן מדובר על datagram כלומר udp (עבור tcp נכתוב sock\_stream).

**שורה 5:** מגדירים את הip של השרת- כאן זהו שרת שעובד בlocalhost (ולכן הip הוא של localhost כלומר 127.0.0.1).

**שורה 6:** מגדירים את הport של השרת- כאן זה 8080, אפליקציה שתרצה לשלוח ספציפית לשרת זה תצטרך לשלוח לפורט 8080.

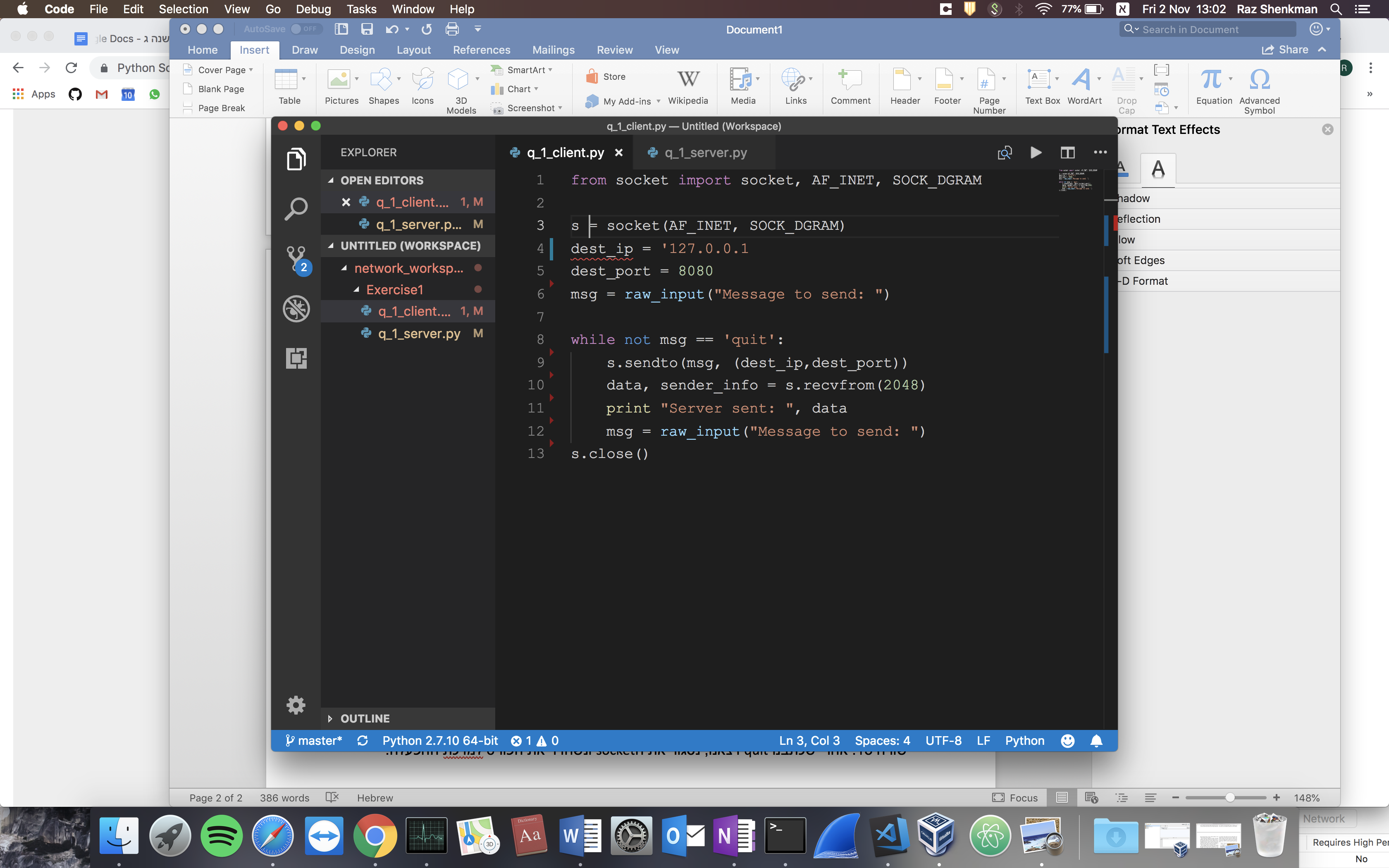
**שורה 7:** מקשרים את הsocket לip ולport, כלומר שומרים באובייקט הsocket שאלו יהיו נתוני ההתחברות אליו. ואם הport תפוס/ip שגוי- שורה זו תכשל. (כאן מערכת ההפעלה תנסה לתפוס את הפורט עבור הסוקט).

**שורה 10:** מקבלים מהסוקט (recvfrom- מחזירה בפורט (string, address) ככה שהstring הוא המידע שהתקבל, והaddress זה הכתובת של השולח). 2048 זוהי כמות הבייטים שהוא מסוגל לקבל. אם ישלחו יותר, השרת יקרוס.

**שורה 11:** מדפיסים את המידע שנשלח, וגם ממי (senderinfo כאן יהיה tuple של (ip,port)).

**שורה 12:** שולחים בחזרה דרך הsocket את המידע באותיות גדולות במתודה sendto (מקבלת את המידע ואת המידע למי לשלוח).

קוד הלקוח:



**שורה 1:** מייבאים מספריית socket את המשאבים שנדרשים.

**שורה 3:** s הוא אובייקט מסוג socket כך ש:

Flag AF\_INET מסמן את הסוג protocol ip שהsocket יעבוד איתו, במקרה זה מדובר על ipV4. נשים לב שאם כתוב AF\_INET6 אז מתאימים את הsocket לעבוד עם ip protocol מסוג ipV6.

SOCK\_DGRAM- דגל שמסמן איזה פרוטוקול הsocket יעבוד איתו בשכבת התעבורה, כלומר tcp/udp/etc., כאן מדובר על datagram כלומר udp (עבור tcp נכתוב sock\_stream).

**שורה 4:** מגדירים את הip של השרת- כאן זהו שרת שעובד בlocalhost (ולכן הip הוא של localhost כלומר 127.0.0.1). לשרת זה נרצה לשלוח.

**שורה 5:** מגדירים את הport של השרת- כאן זה 8080. לשרת זה נרצה לשלוח.

**שורה 8:** נשלח/נקבל הודעות חדשות עד שנשלח quit (ואז הלקוח יפסיק לעבוד והסוקט יסגר)

**שורה 9:** שולחים לשרת באמצעות sendto.

**שורה 10:** מקבלים מהשרת מידע (המידע כאן מפורט יותר בקוד של השרת, זה חוזר על אותו דבר).

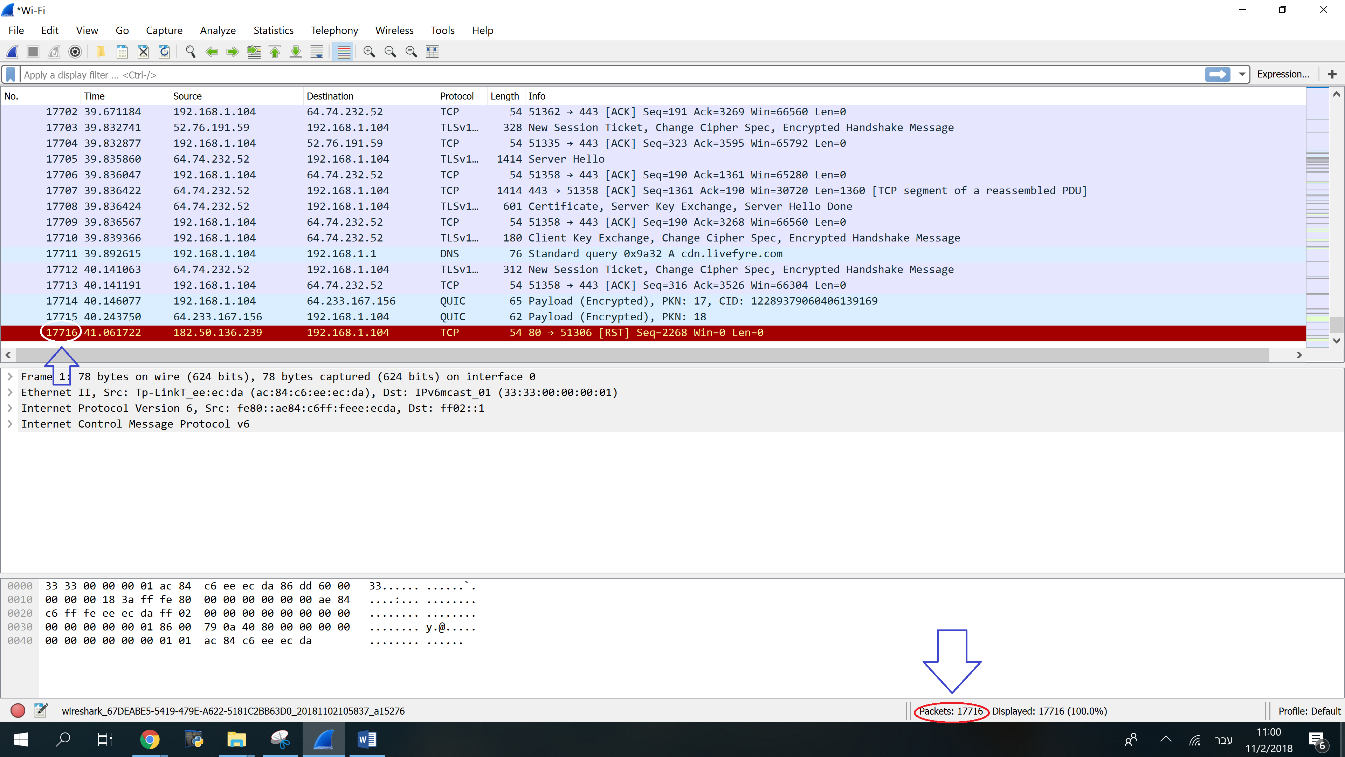
**שורה 11:** מדפיסים רק את המידע שהשרת שלח בחזרה (בלי הפרטים).

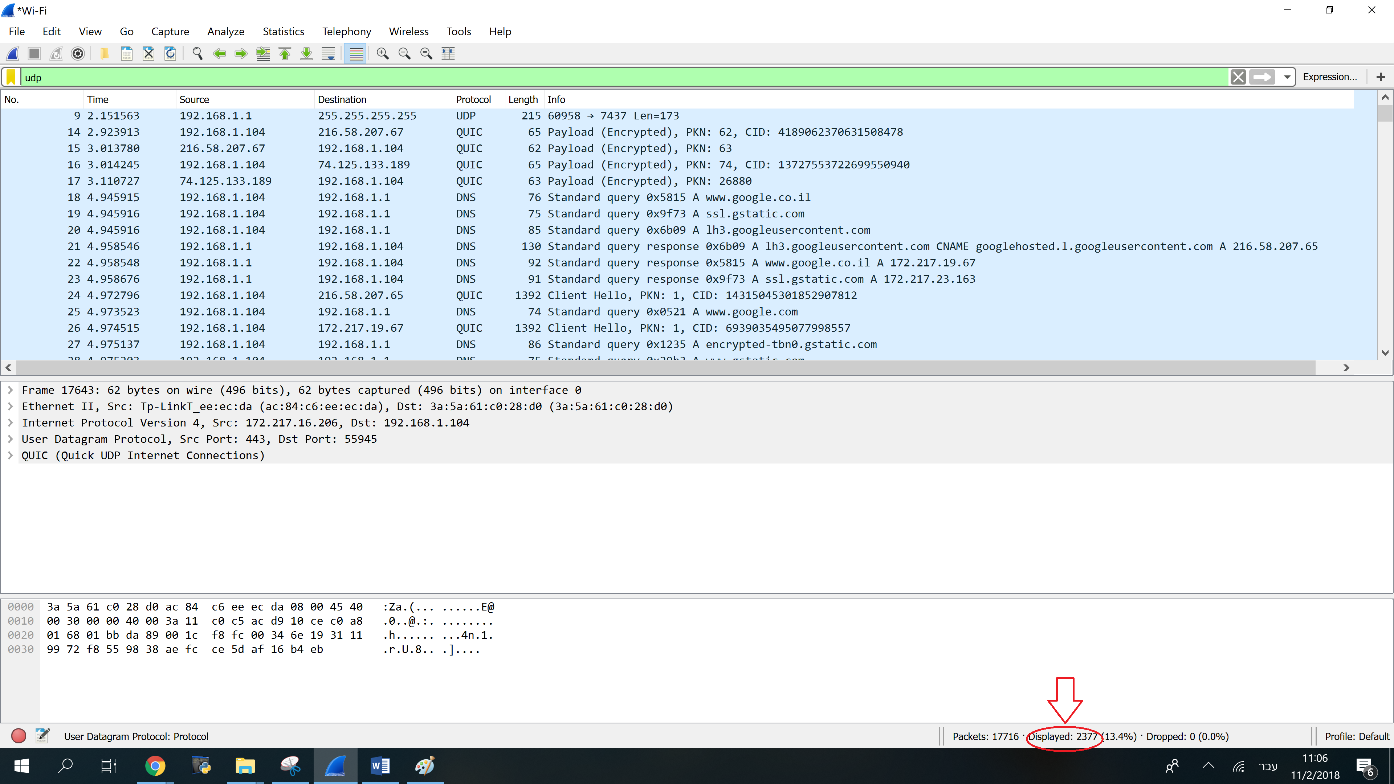
**שורה 12:** מקבלים קלט מהמקלדת (raw\_input).

**שורה** **13:** אחרי שכתבנו quit ויצאנו, נסגור את הsocket ונשחרר את הפורט למרכת ההפעלה.

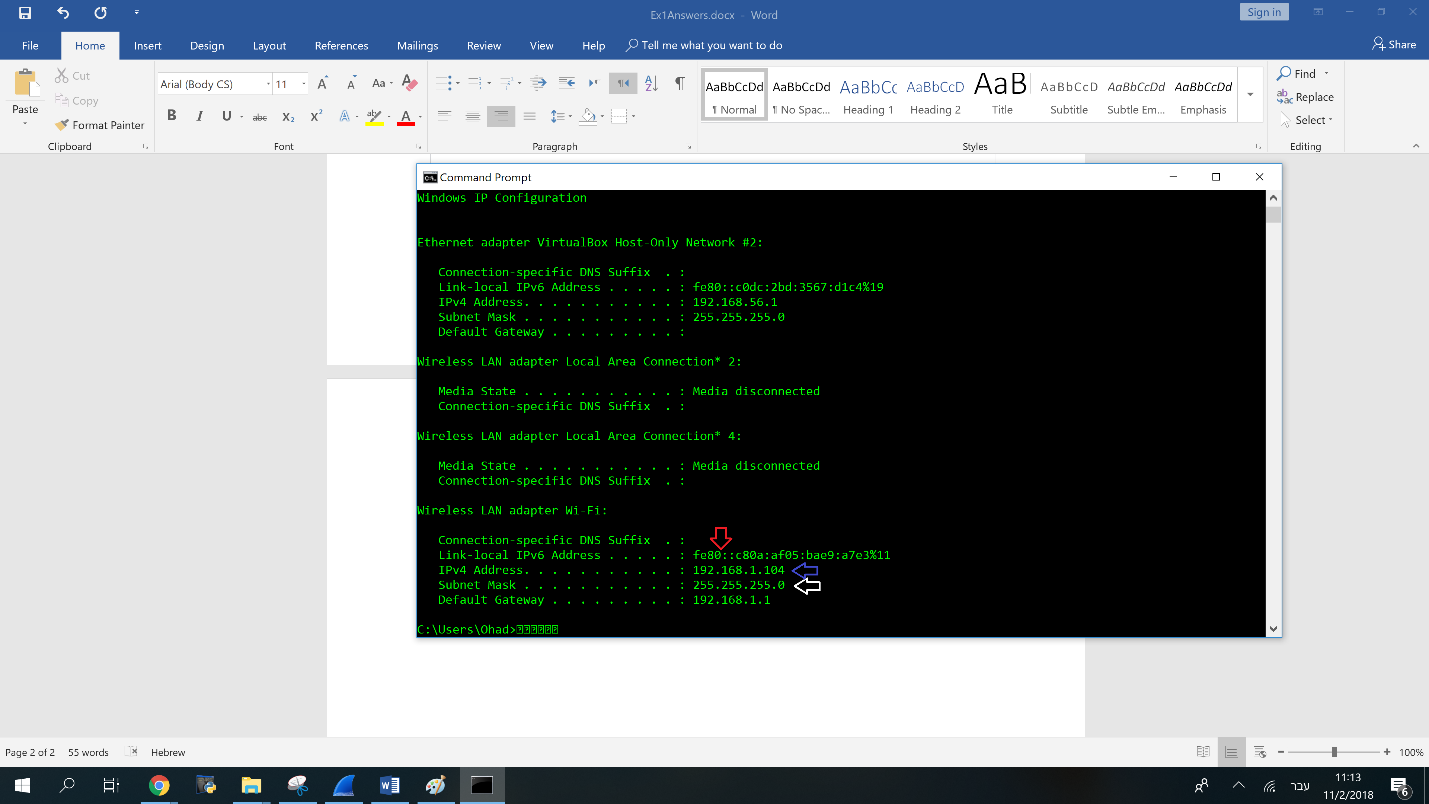
שאלה 2

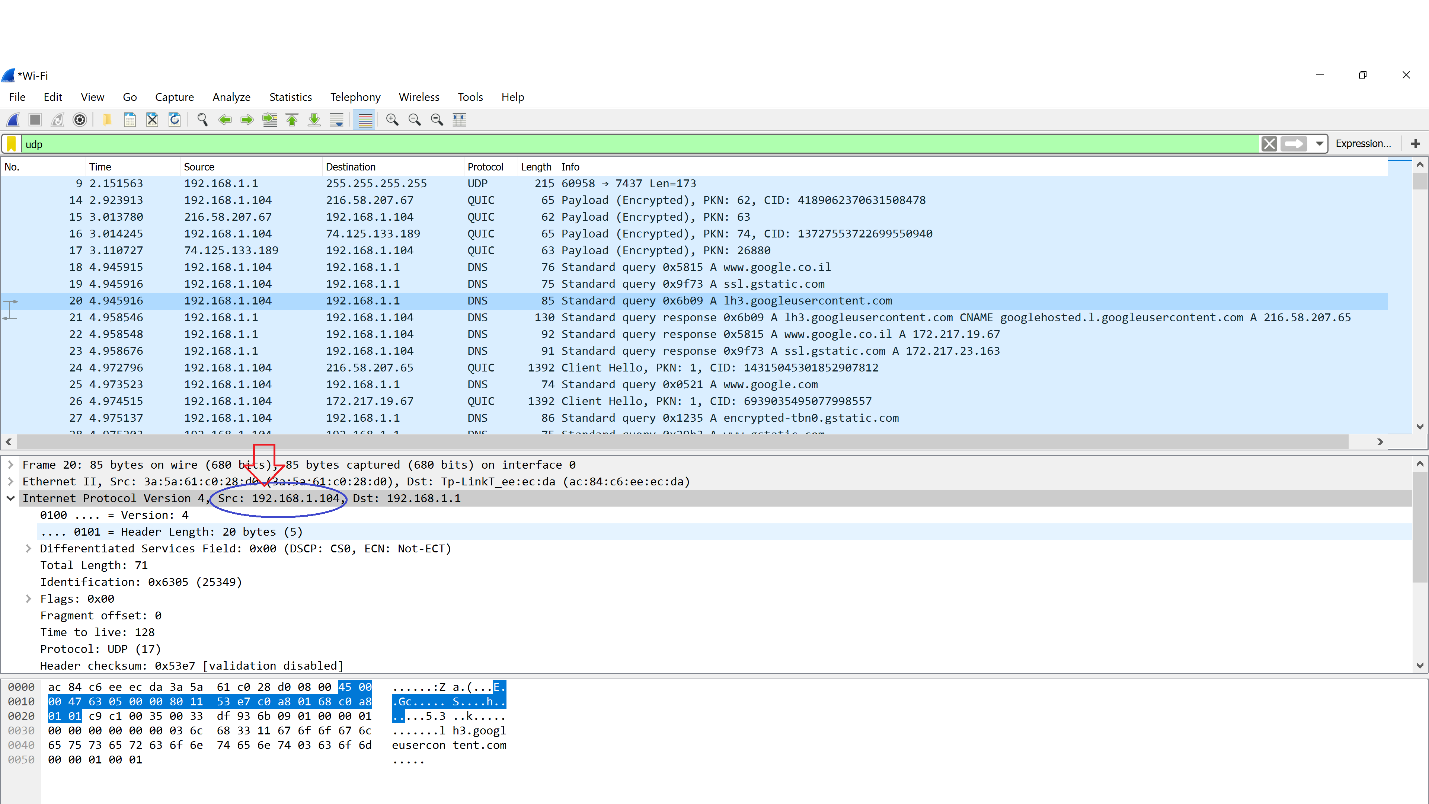
א. גלשנו למספר אתרים והוספנו 17716 חבילות. צילום מסך:



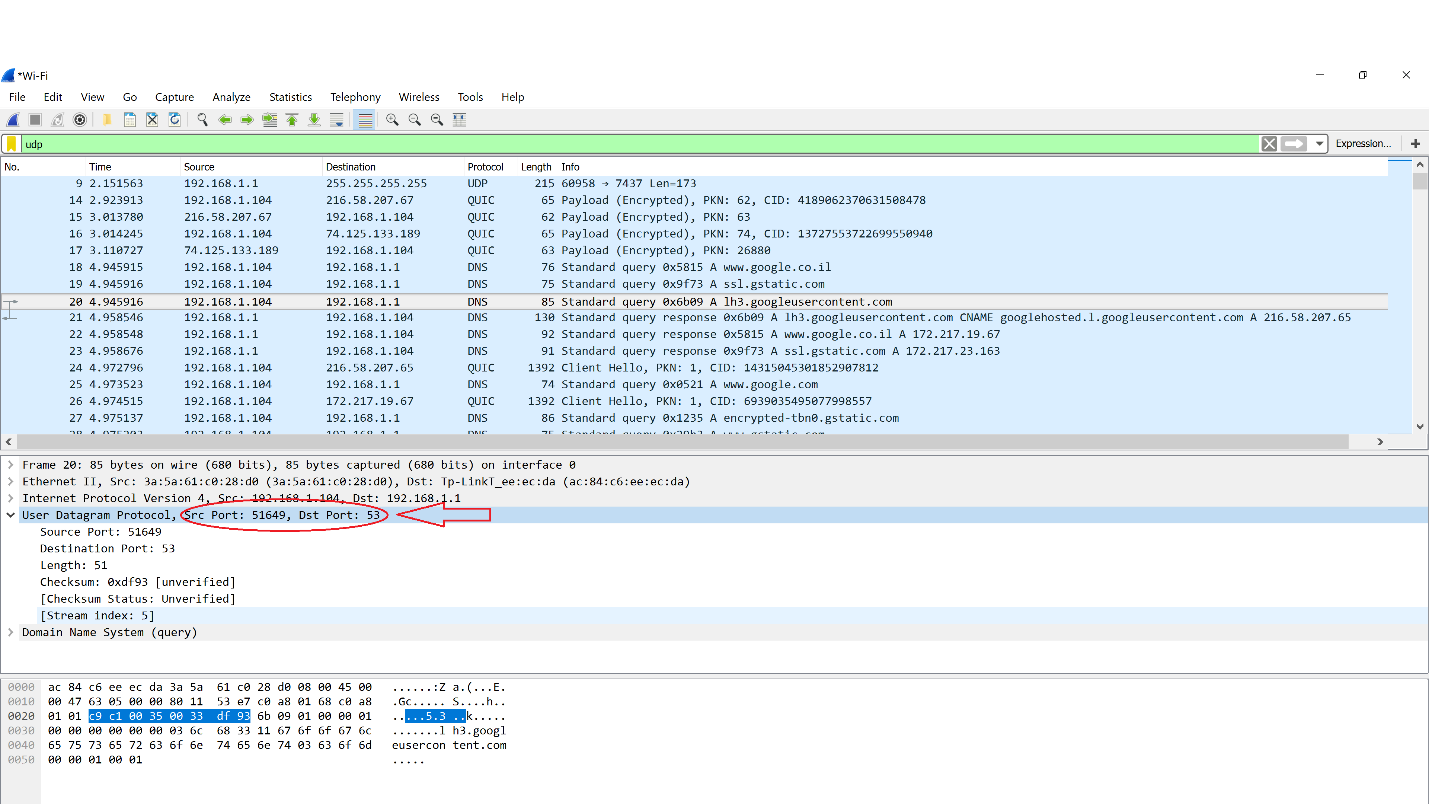
ב. כתבנו בשורת הסינון udp ולמטה(באדום) רואים שיש 2377 כאלו.

ג. כתובת ה ipv4 של המחשב בתוך ה LAN היא 192.168.1.104 (בכחול) עם subnet mask של 24 ביטים(, בלבן 255.255.255.0). כתובת ה ipv6 היא fe80::c80a:af05:bae9:a7e3%11 (באדום)

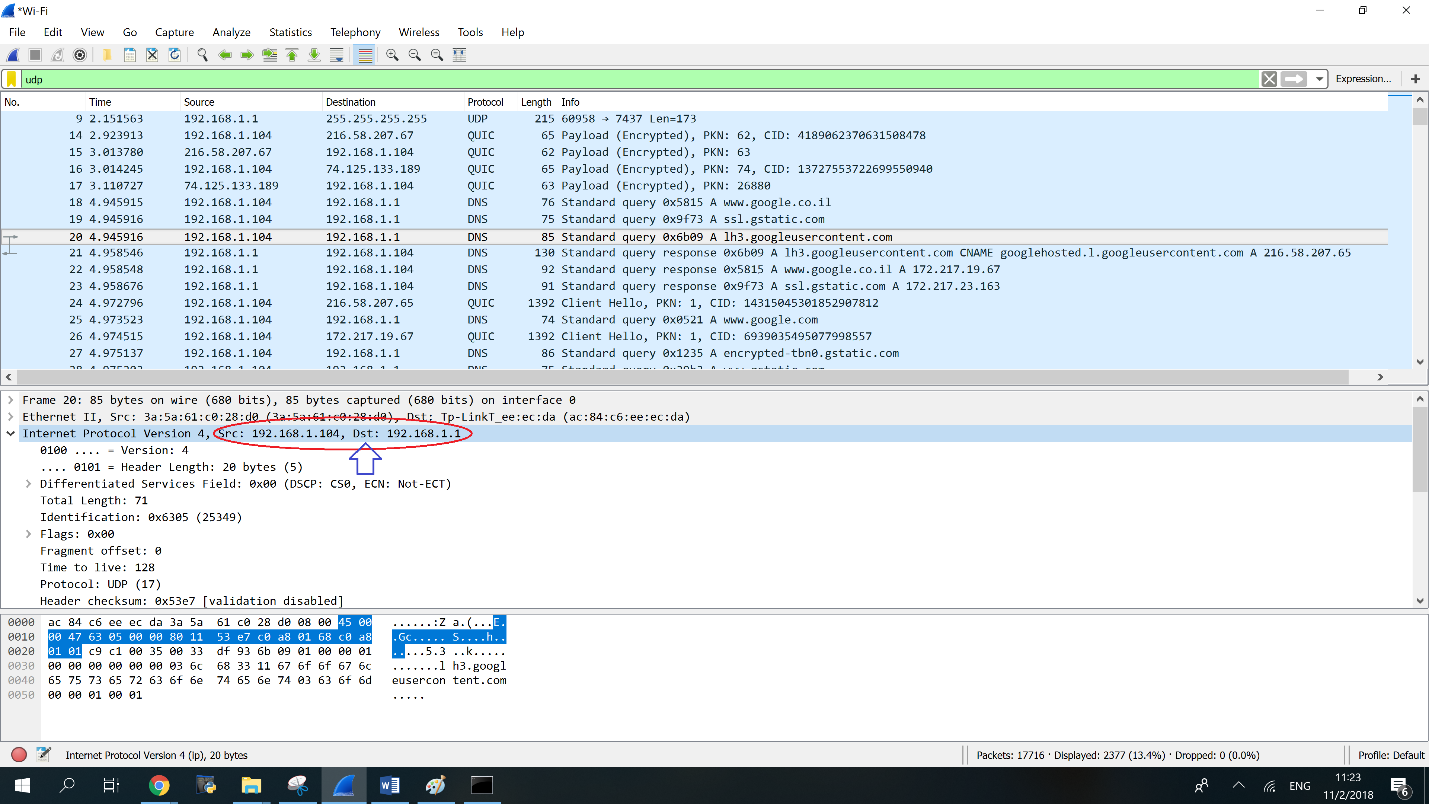


ד. בחרנו בחבילה 20. לאחר התבוננות, ראיונ שהחבילה נשלחה מאיתנו מכיוון ש ה source ip בחבילה זהה לכתובת של המחשב שלנו שראינו בסעיף הקודם.

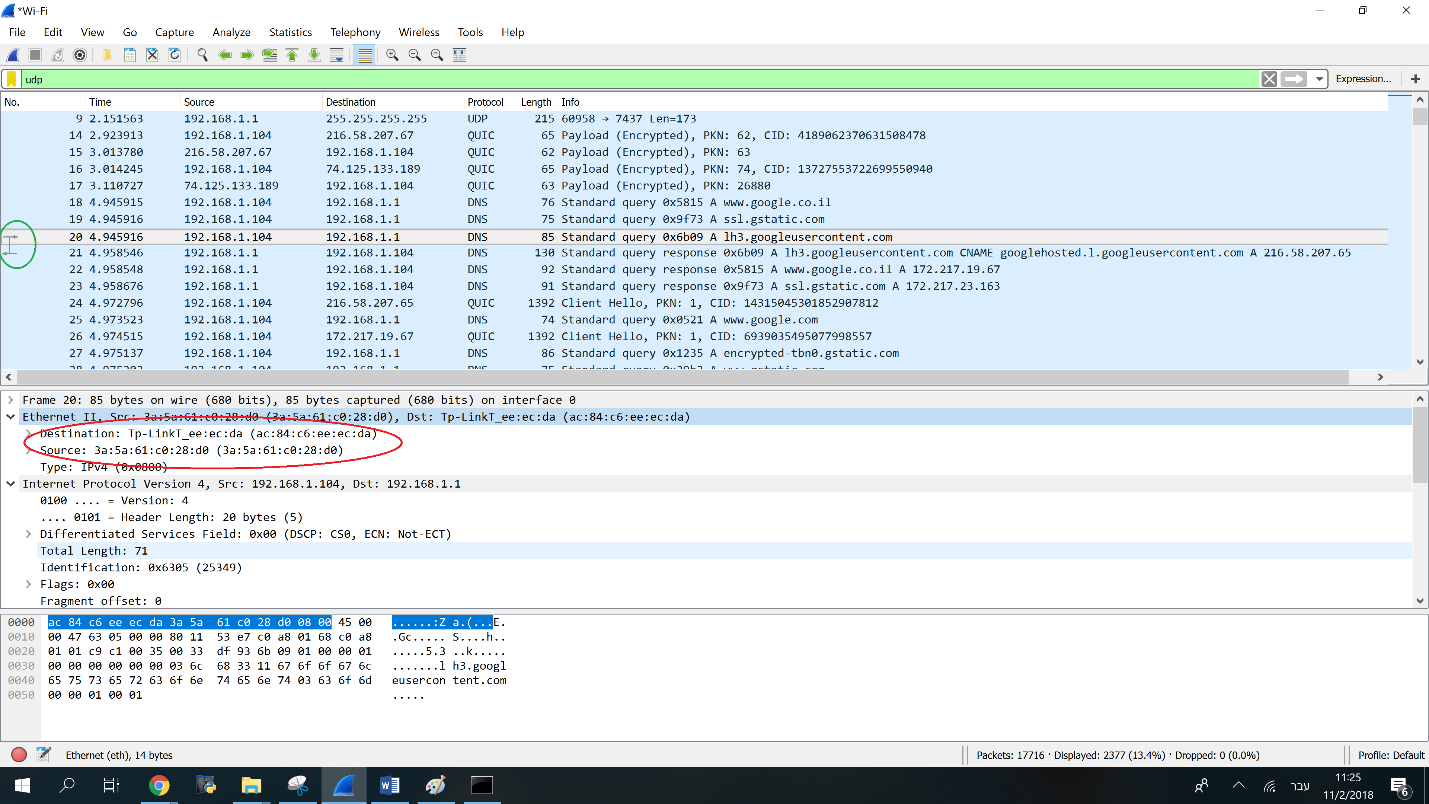
נסתכל בheader של UDP, החבילה נשלחה מפורט 51649(מהלקוח) אל פורט 53(פורט ידוע ל DNS) בשרת.



החבילה נשלחה מ source ip 192.168.1.104 אל destination ip 192.168.1.1 כמו שניתן לראות ב header של שכבת הרשת.

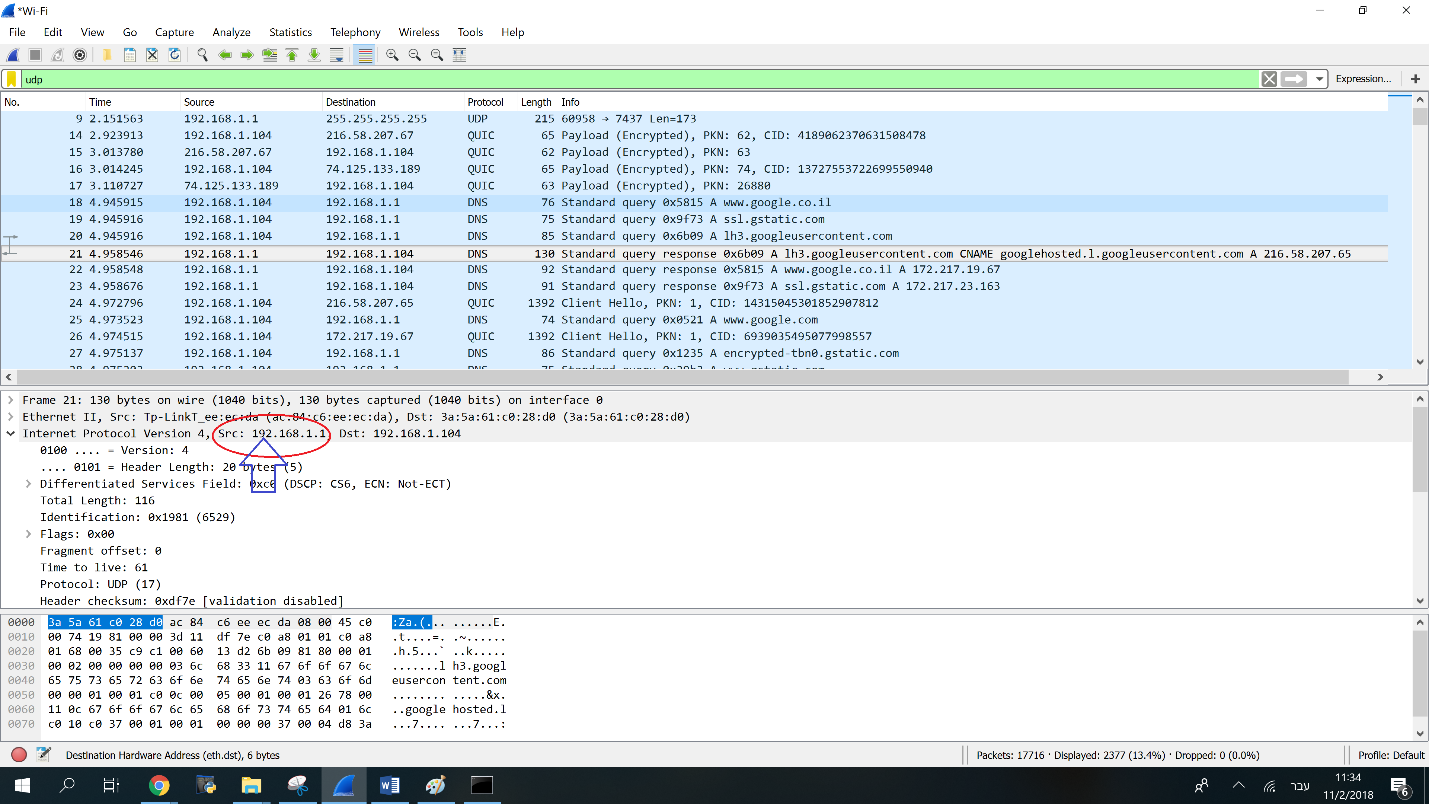


כמו שניתן לראות ב header של שכבת הקו כתובת ה MAC של השולח היא 3a:5a:61:c0:28:d0 וכתובת ה MAC של היעד היא ac:84:c6:ee:ec:da

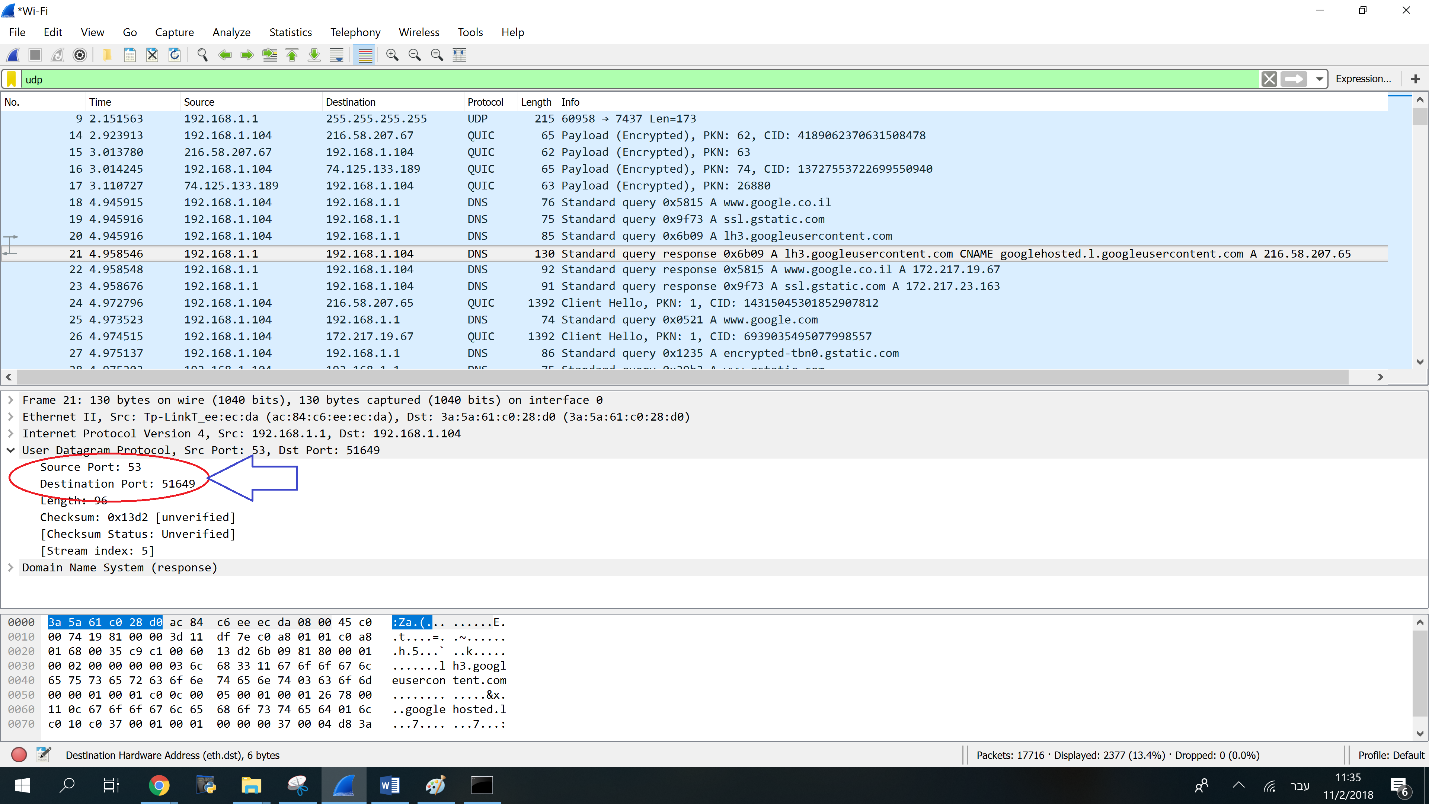


ה. בעיגול הירוק בתמונה למעלה קל לראות את החץ ש wireshark יוצר, כלומר הוא אומר לנו שפקטה 21 היא התשובה לפקטה 20 שניתחנו. ננתח באותו תהליך את פקטה 21.

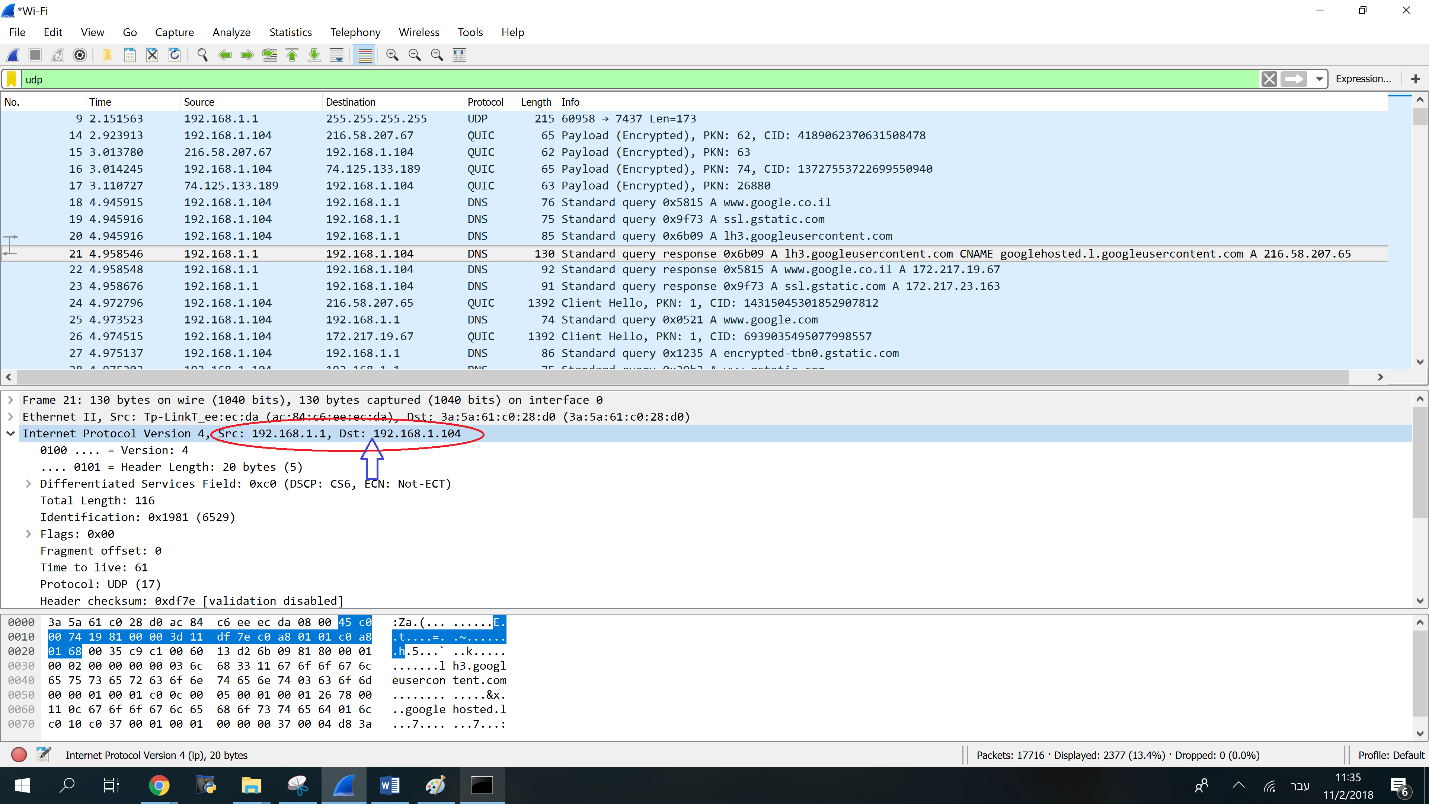
החבילה לא נשלחה מהמחשב שלנו אלא ממכשיר אחר, מכיוון ש ה source ip שונה מהכתובת Ip של המחשב שלנו.



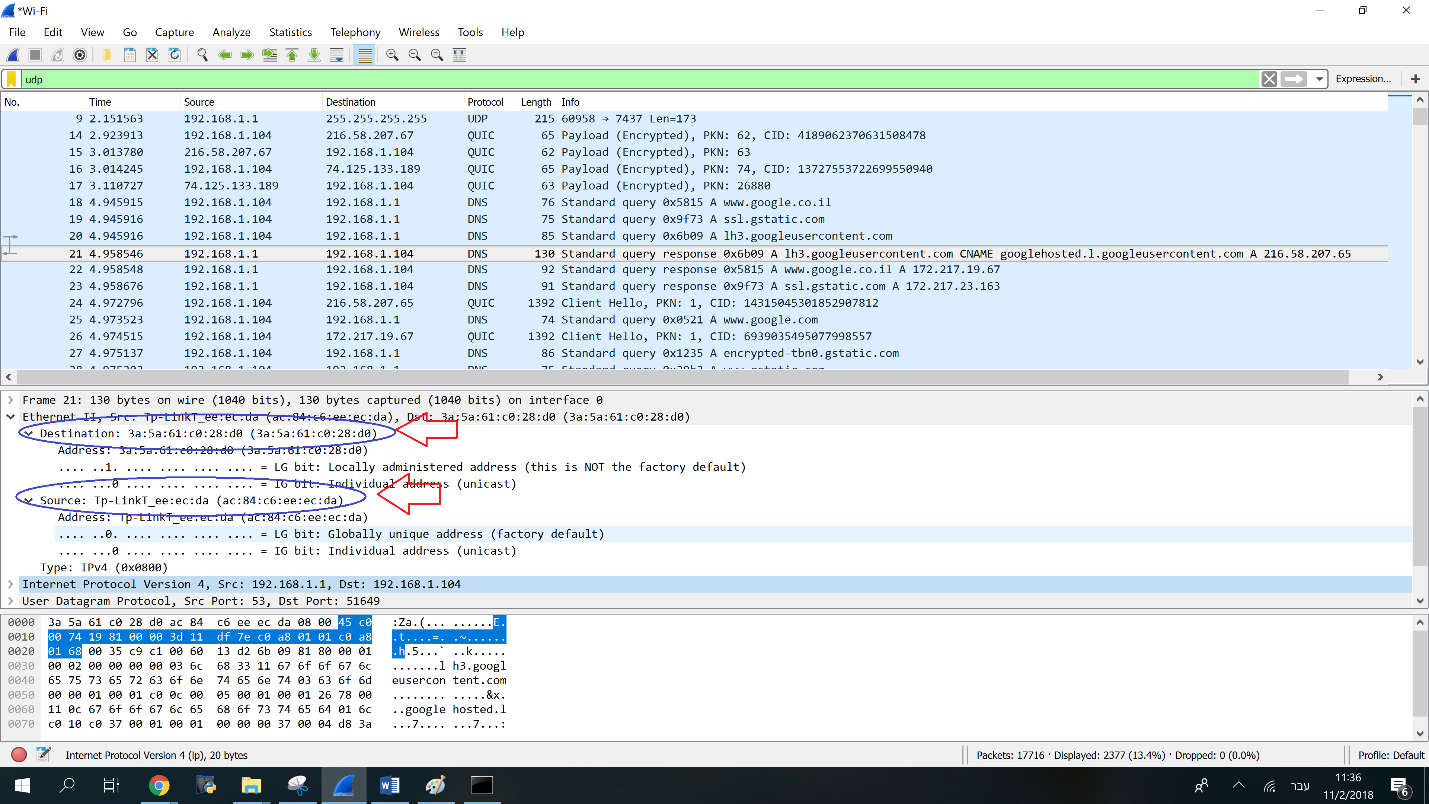
נסתכל בheader של UDP, החבילה נשלחה מפורט 53(מהשרת DNS) אל פורט 51649(אל הלקוח).



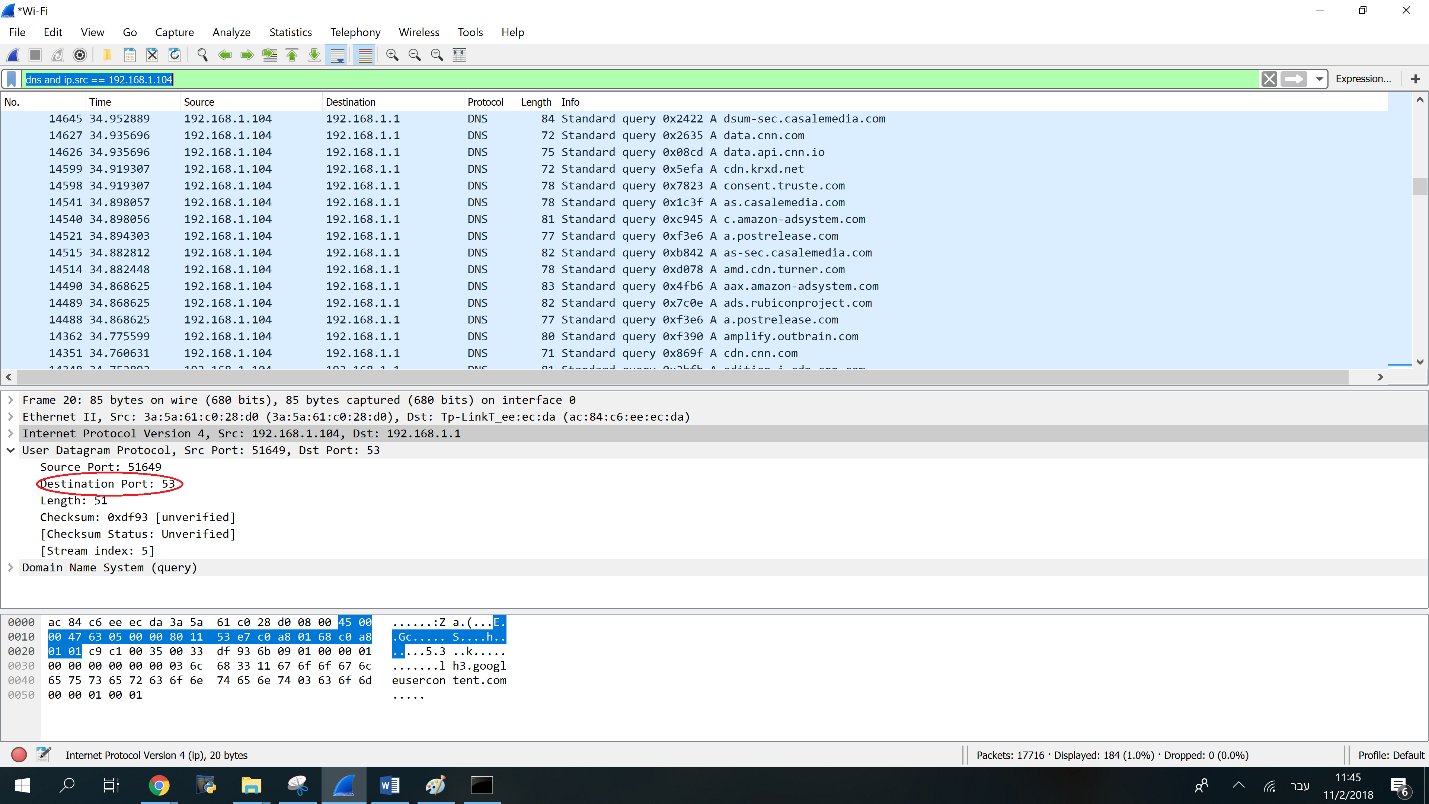
החבילה נשלחה מ source ip 192.168.1.1 אל destination ip 192.168.1.104 כמו שניתן לראות ב header של שכבת הרשת.



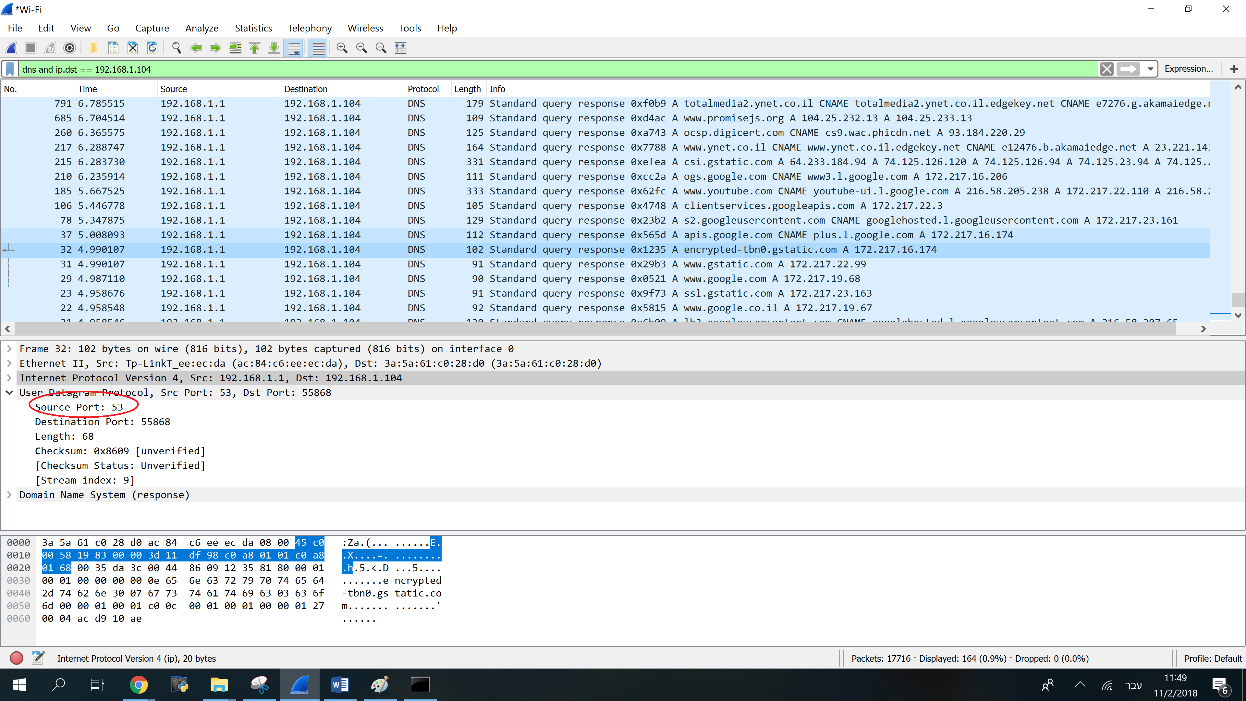
כמו שניתן לראות ב header של שכבת הקו כתובת ה MAC של השולח היא ac:84:c6:ee:ec:da וכתובת ה MAC של היעד היא 3a:5a:61:c0:28:d0



ו. נפלטר את החבילות באמצעות כתיבת "dns and ip.src == 192.168.1.104" בשורת הפילטר. שמנו לב שהפורט 53 חוזר תמיד כפורט יעד. זה כמובן בגלל ששרתי DNS מאזינים לפי קונבנציה לפורט 53. לכן כל פעם שנשלח חבילה מהמחשב שלנו לשרת DNS זה יהיה **כנראה** לפורט יעד 53.



ו2. נפלטר את החבילת באמצעות כתיבת " dns and ip.dst == 192.168.1.104" בשורת הפילטר. שמים לב שהפורט 53 חוזר תמיד כפורט מקור. כלומר בכל חבילות ה DNS שנשלחות אלינו, השרת DNS שולח אותן מפורט 53, כנראה כי זו הקונבציה וזה מחזק את התשובה הקודמת.



.

ז. יודעים ש DNS עובד בשיטת UDP ולכן אפשר פשוט לסנן לפי פורט 53, כלומר לכתוב " udp.port == 53" ונקבל את כל החבילות שאחד מהפורטים(יעד או מקור) הוא 53 ומניחים לפי הקונבנציה שכל אלו יהיו חבילות DNS.

