מטלה 3 תקשורת

מגישות:

נעמה שטאובר 316162114

מירב בוים 206489155

רעות דביר 204121446

1. פרוטוקול HTTP נמצא בשכבת APPLICATION.
2. כאשר המרנו את החבילה מ HEX ל ASCII:

ÿÔ)ÔEA@1¼À¨eÀ¨gÜþPg¾t$u&öP@)½#GET

/form.html HTTP/1.1

Accept: \*/\*

Accept-Language: en-us

User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; InfoPath.3; .NET4.0C; .NET4.0E)

Accept-Encoding: gzip, deflate

Host: 192.168.2.103

Connection: Keep-Alive

**GET-** שיטת הבקשה, מילה המתארת מה מבקש הלקוח מהשרת לעשות.

**/form.html-** כתובת המשאב המבוקש, מציינת לשרת לאיזה אובייקט נרצה לגשת.

**HTTP/1.1-** גרסת הפרוטוקול לפיה נבנתה הבקשה.

**Accept-Language: en-us-** מציינת לשרת שהתשובה לבקשה יכולה להכיל מידע באנגלית.

**User-Agent-** מכילה מחרוזת אופיינית המאפשרת לפרוטוקולי הרשת לזהות את סוג היישום, מערכת ההפעלה, ספק התוכנה או גרסת התוכנה של סוכן המשתמש המבקש תוכנה.

**Accept-Encoding-** סוג המידע שאנו תומכים בקבלה שלו (תומך ב gzip, deflate).

**Host-** כתובת האתר אליו מפנים את הבקשה.

**Connection-** קובע אם החיבור לרשת נשאר פתוח לאחר השלמת העסקה הנוכחית.

GET

GET

OK

GET

OK

GET

OK

OK

POST

Server 2

212.179.74.179

Server 1

84.229.173.71

Client

109.64.17.214

GET

OK

TCP

Spurious Retransmission

CONTINUE

OK

GET

OK

GET

GET

OK

GET

OK

GET

OK

NOT FOUND

GET

GET

OK

1. סוגי הבקשות שזיהינו בtrace:

GET בפקטות: 6,15,23,72,140,143,144,148,237,241,283,291

POST בפקטה: 66.

סוגי הבקשות הנוספים המוגדרים בפרוטוקול HTTP:

HEAD, PUT, DELETE, OPTIONS, TRACE, CONNECT.

1. לכל בקשה יש ייחודיות בבקשה:

**GET-** מיועדת לקבלת אובייקט שנמצא על השרת.

**HEAD-** מבקשת מהשרת לשלוח את כל שדות הכותרת שהיו נשלחים לבקשת GET, אך בלי האובייקט עצמו.

**POST-** בקשות המכילות גוף הודעה.

**PUT-** מבקשת מהשרת לשמור את גוף ההודעה המצורף לבקשה בתור אובייקט, שכתובתו היא הכתובת שניתנה בתחילת הבקשה.

**DELETE-** מבקשת מהשרת למחוק את האובייקט שכתובתו מצוינת בתחילת הבקשה.

**OPTIONS-** מבקשת מהשרת מידע על דרכי התקשורת האפשריות ביחס לאובייקט מסוים או ביחס לשרת עצמו.

**-TRACE** מבקש מהשרת לשלוח את הבקשה בדיוק כפי שקיבל אותה.

**-CONNECT**הפרוטוקול אינו מגדיר את השימוש ב-CONNECT, אך שומר את השיטה הזו לשימוש עבור שרתי פרוקסי שמסוגלים להתנהג כמו תעלות SSL.

1. מתי כדאי להשתמש בכל סוג בקשה:

**GET**- כאשר רוצים לקבל אובייקט שנמצא על השרת, בקשות GET הן הנפוצות ביותר.

**HEAD-** כאשר רוצים לבדוק קישורים שבורים או זמני שינויים של אובייקטים מבלי לבקש את כל האובייקט.

**POST-** כאשר רוצים לשלוח מחרוזות לשרת לשם עיבוד.

**PUT-** כאשר רוצים שהשרת ישמור את גוף ההודעה בתור אובייקט.

**DELETE-** כאשר רוצים שהשרת ימחק אובייקט מסוים.

**OPTIONS-** כאשר רוצים לקבל שיטת בקשה לגבי אובייקט מסוים.

**-TRACE** כאשר רוצים לוודא שההודעה הועברה בתחנות הביניים כמו שצריך.

**CONNECT-** כאשר משתמשים בשרתי פרוקסי.

1. סוגי התשובות שזיהנו:

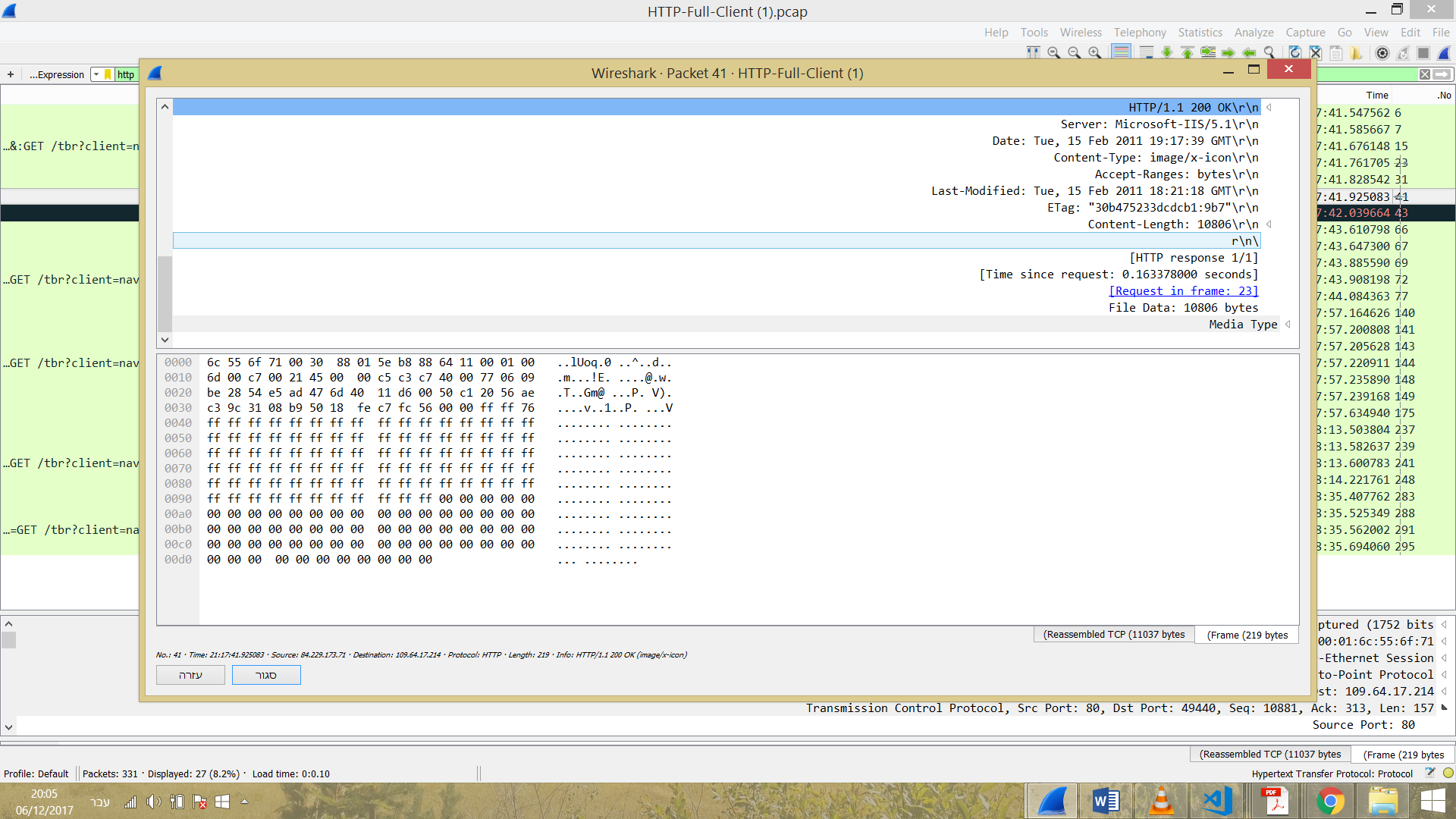
**100 CONTINUE:** בפקטה 67 עבור 66.

**404 NOT FOUND:** בפקטה 288 עבור 283.

**200 OK:** בפקטה 7 עבור 6, בפקטה 31 עבור 15, בפקטה41 עבור 23, בפקטה 69 עבור 66, בפקטה 77 עבור 72, בפקטה 141 עבור 140, בפקטה 149 עבור 143, בפקטה 175 עבור 144, בפקטה 239 עבור 237, בפקטה 248 עבור 241, בפקטה 295 עבור 291.

זיהינו את הבקשה לתשובה בעזרת המידע בשכבת המידע על הhttp.

מספר סוגי התשובות המוגדרות בפרוטוקול HTTP: 200, 204, 206, 301, 302, 303, 304, 307, 400 401, 403, 404, 405, 500, 503, **סה"כ** 15 סוגי תשובות.



1. פירוט סוגי התשובות:

**200 –** “הכל בסדר”: התוכן שנתבקש אמור להגיע ב-body

**204 – No Content:** כאשר אין ערך החזרה. הדפדפן לא ינווט לכתובת החדשה אלא יישאר בכתובת האחרונה שהיה.

**206 – Partial Content:** כמו 200, רק משמש עבור בקשות של range בתוך המסמך, המבוצעות על בסיס ה header שנקרא Content-Range.

**301, 302 או 303 –** הדפדפן מציין שה URL למשאב זה הוחלף, וה URL החדש מצויים ב header בשם Location. דפדפנים יפנו אוטומטית לכתובת החדשה, מבלי שהמשתמש מודע לכך, מה שנקרא “Client-Side ReDirect”. אם הפעולה הייתה פעולת POST, הדפדפן ייגש ל URL החדש בפעולת GET (כדי למנוע שליחה כפולה של הנתונים). דפים שמציגים למשתמש הודעה “…You are being redirected” מחזירים בעצם קוד 200, ופעולת ה Redirect היא בעצם אפליקטיבית.

**304 – not modified:** מציין שגרסת העמוד הנמצא אצל הלקוח היא הגרסה המעודכנת ביותר שלו, ולכן התשובה אינה כוללת את האובייקט. כך מתאפשר שימוש במנגנוני המטמון של HTTP.

**307 – Temporary Redirect**: כמו 301-303 רק ללא ביטול פעולת ה POST.

**400 – bad request:** השרת מציין שמשהו לא בסדר בבקשה שנשלחה, אולי הפורמט ואולי תוכן הבקשה. הפירוט אמור להגיע בהודעת השגיאה.

**401 – Unauthorized:** השרת לא מזהה את המשתמש ולכן אינו יכול לאפשר לו לגשת למשאב שהגישה אליו מוגבלת. קוד זה יזניק חלון “שם/סיסמא” או יהיה נקודת הפתיחה לסט קריאות מורכב כחלק מהפעלת פרוטוקולים לאימות המשתמש כגון Kerberos או SAML.

**403 – Forbidden:** השרת מזהה את המשתמש, אך למשתמש אין הרשאות לגשת למשאב.

**404 –** לא קיים משאב בשם שתואר ב URL.

**405 –** **Method not allowed:** פעולת ה HTTP שביקשנו (למשל Delete) לא זמינה למשאב הספציפי. ייתכן והיא תהיה זמינה עבור משאב (URL) אחר.

**500 – Internal Error:** תקלה פנימית בשרת, שיכולה להיות בעצם כל דבר. התקלה לרוב תכתוב לקובצי הלוג של השרת ולא תשלח (מטעמי בטיחות) למשתמש.

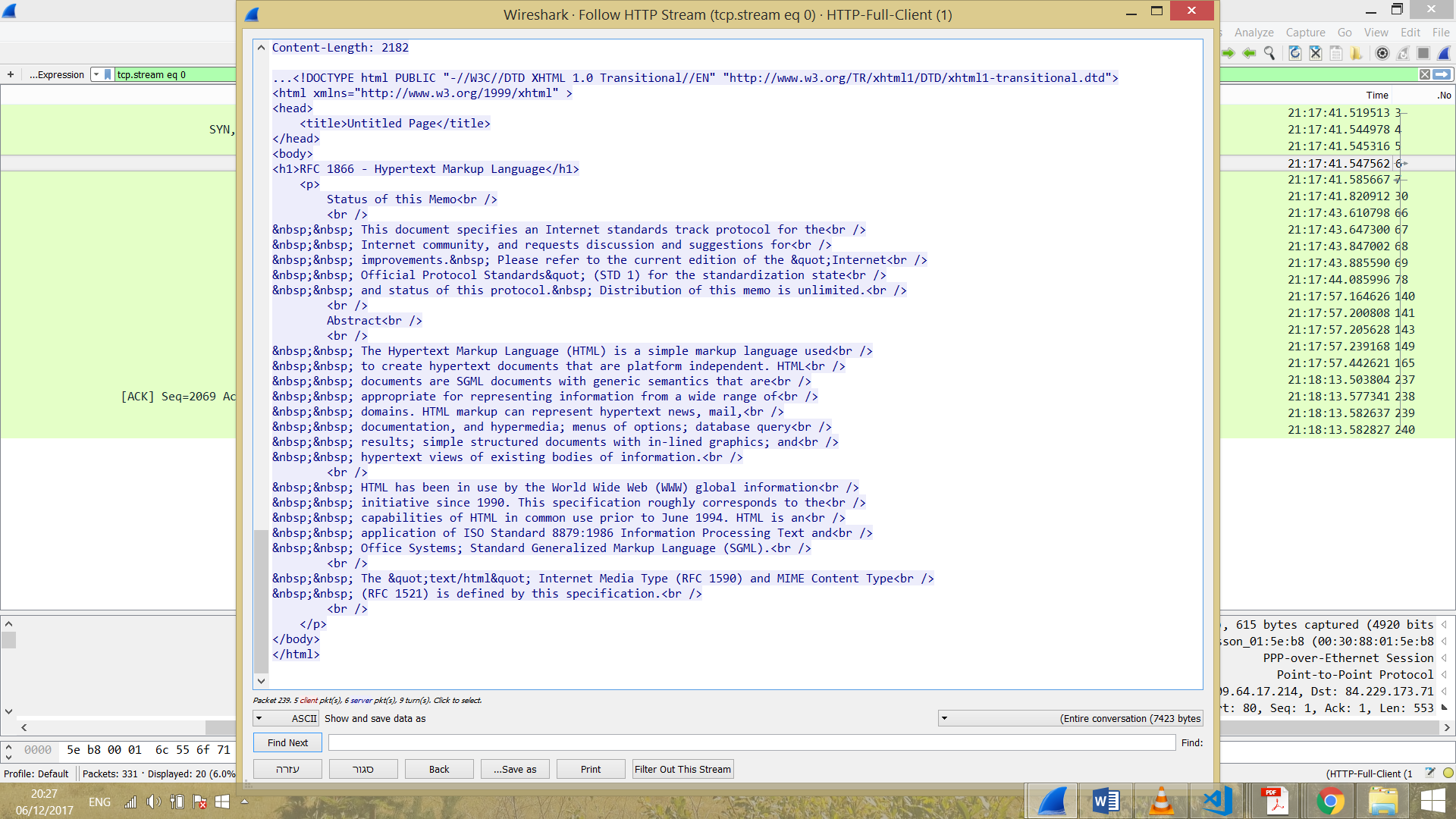
**503 –** **Service Unavailable:** כמו 500, אך בא להדגיש שמדובר בתקלה זמנית בגלל שאיזה service בדיוק נמצא ב maintainable או restart, למשל.

1. פרוטוקול HTTP נמצא בתוך [**TCP/IP protocols**](https://www.ibm.com/support/knowledgecenter/SSGMCP_5.3.0/com.ibm.cics.ts.internet.doc/topics/dfhtl23.html?view=kc), הוא נמצא בתוכם מכיוון שTCP / IP היא משפחה של פרוטוקולי תקשורת המשמשים לחיבור מערכות מחשב ברשת.

TCP / IP מוסיף לhttp את האמינות והרציפות בשליחת המידע.

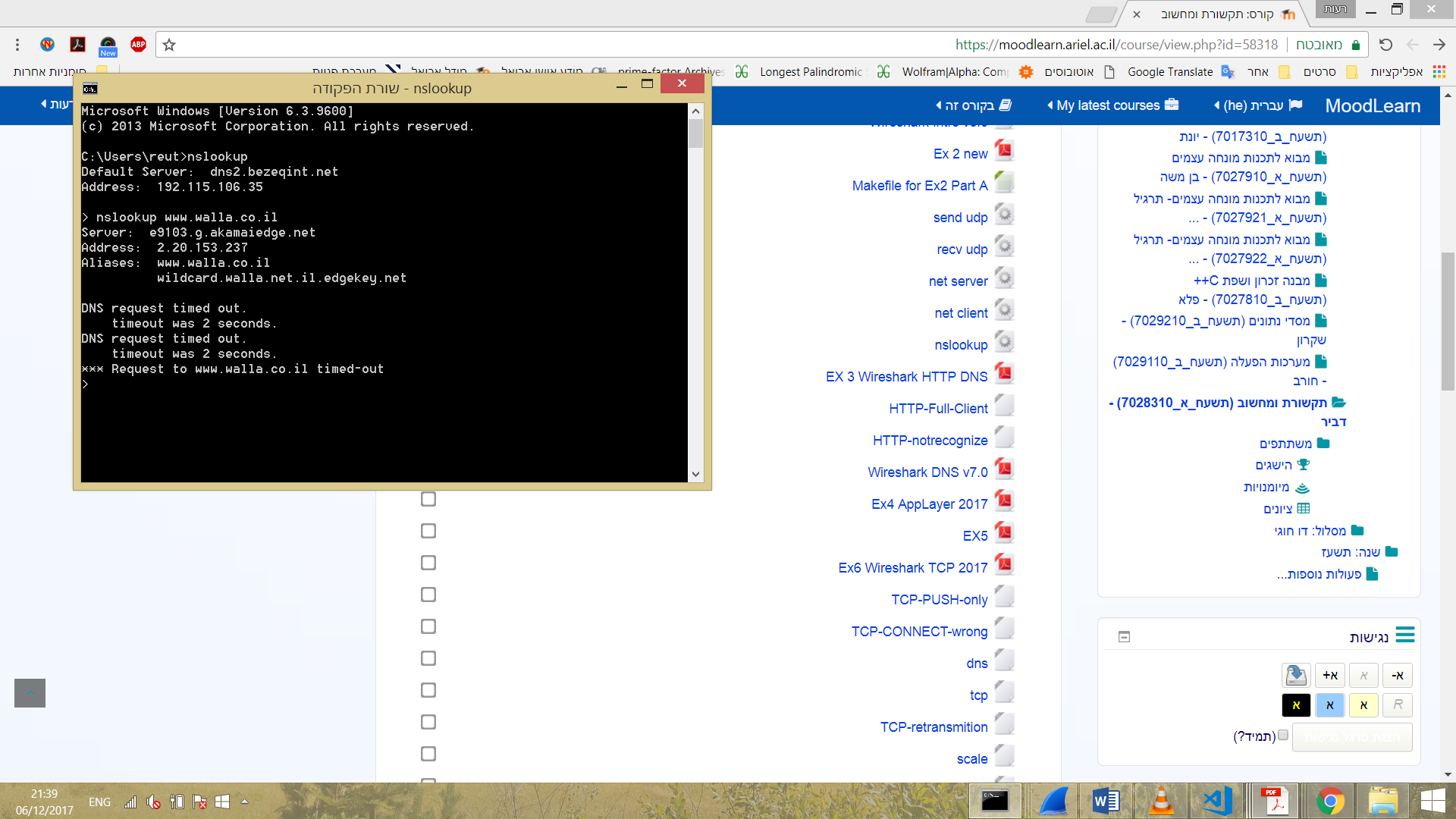
1. תוכן השיחה:

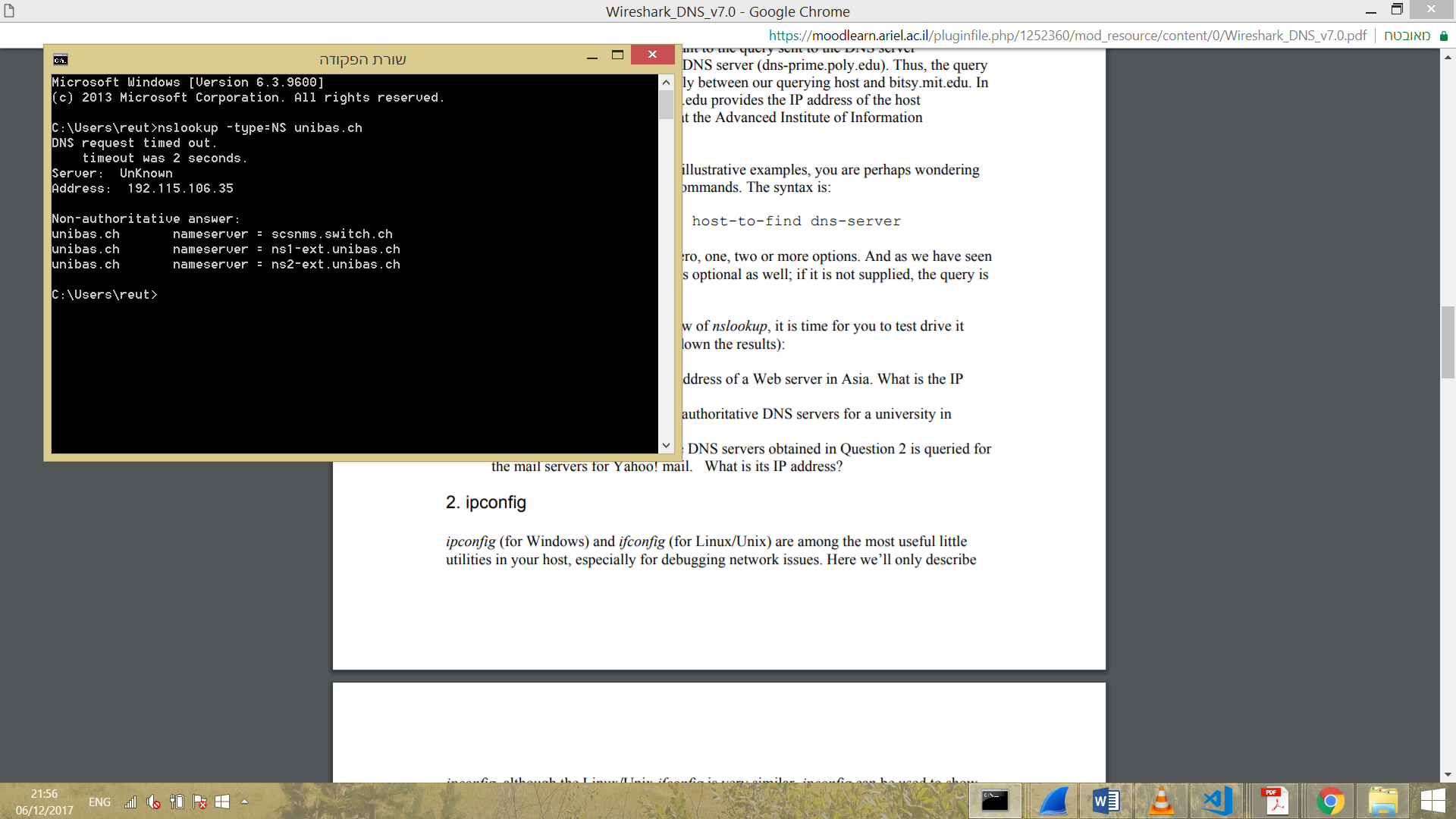
* הלקוח שולח בקשה לשרת
* השרת מחזיר לו דף למילוי פרטים
* הלקוח ממלא את הפרטים שלו (כמו שם: מיקי מאוס) ושולח לשרת
* השרת מאשר את הפרטים שקיבל
* בצד הלקוח- הפרטים נשמרים בcookie, והלקוח מבקש קובץ HTML
* השרת שולח ללקוח תמונה
* הלקוח שומר את התמונה בcookie, ומבקש פרטים עיצוביים לגבי התמונה
* השרת שולח מידע עיצובי לגבי התמונה
* הלקוח שומר את המידע העיצובי בcookie ומבקש דף html
* השרת שולח דף html



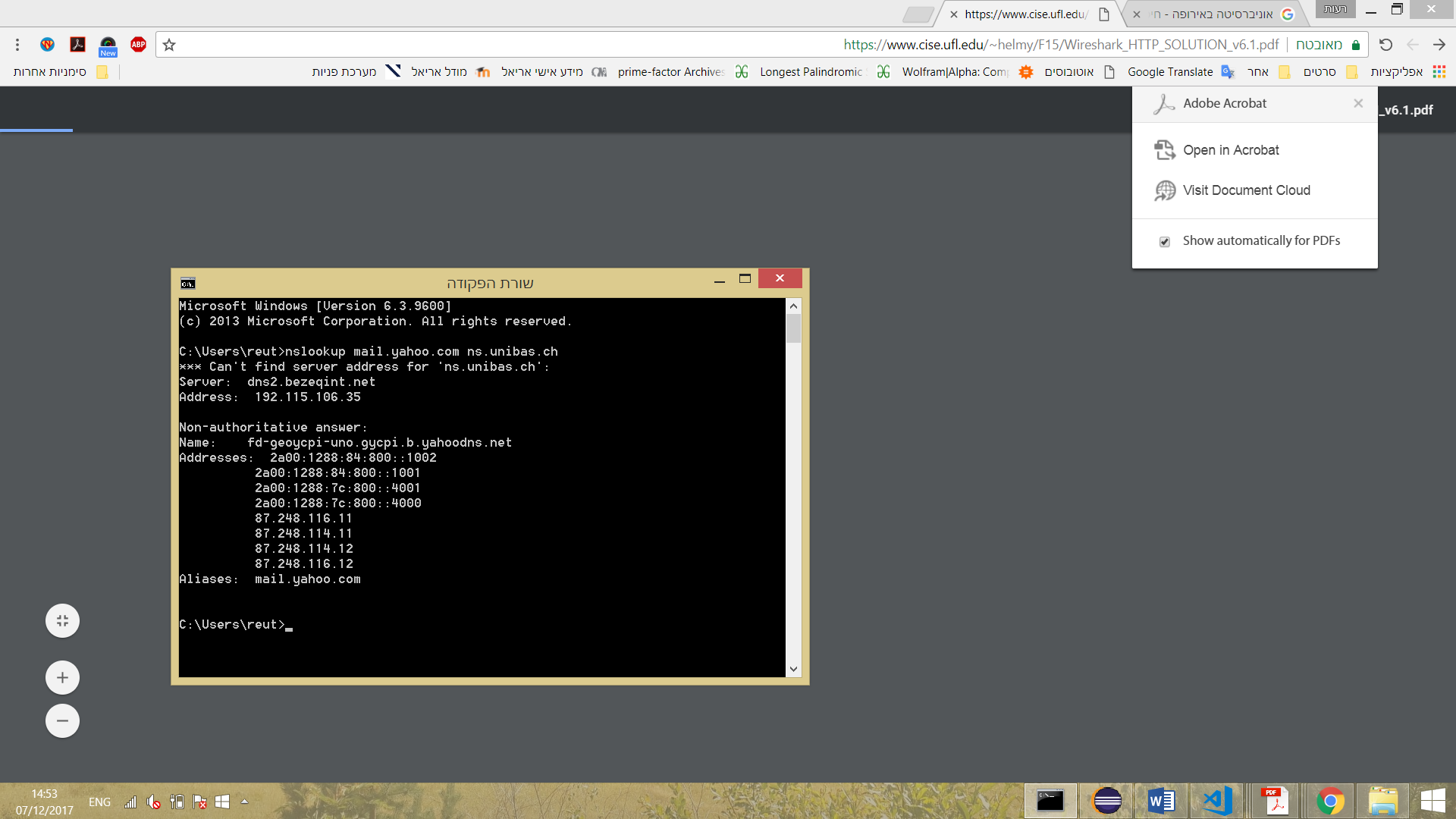
1. יש שתי בקשות מכיוון שכאשר רוצים לקבל את התמונה המבוקשת- מקבלים מידע עליה משני דפים: html ו css.
2. ללקוח יש את הגרסה העדכנית של דף הhtml שהוא מבקש ולכן השרת לא שולח לו כלום, ולא נענה לבקשתו, לכן הwireshark לא מתייחס לזה בתור שיחה.
3. Dns 15 שאלות, בדפים הבאים:
4. הכנסנו לnslookup את האתר [www.walla.co.il](http://www.walla.co.il),

קיבלנו את ה**ip** 2.20.153.238

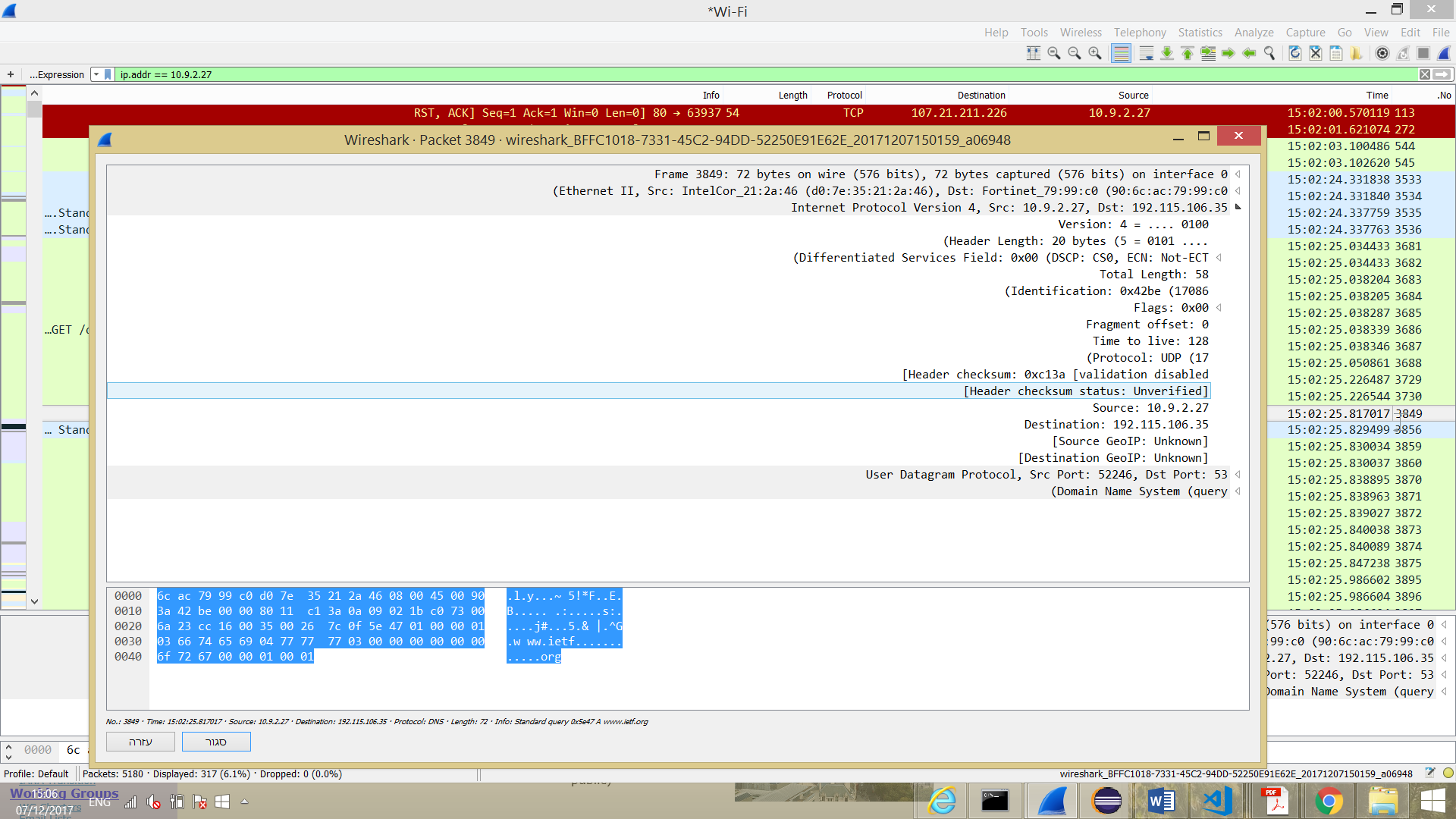




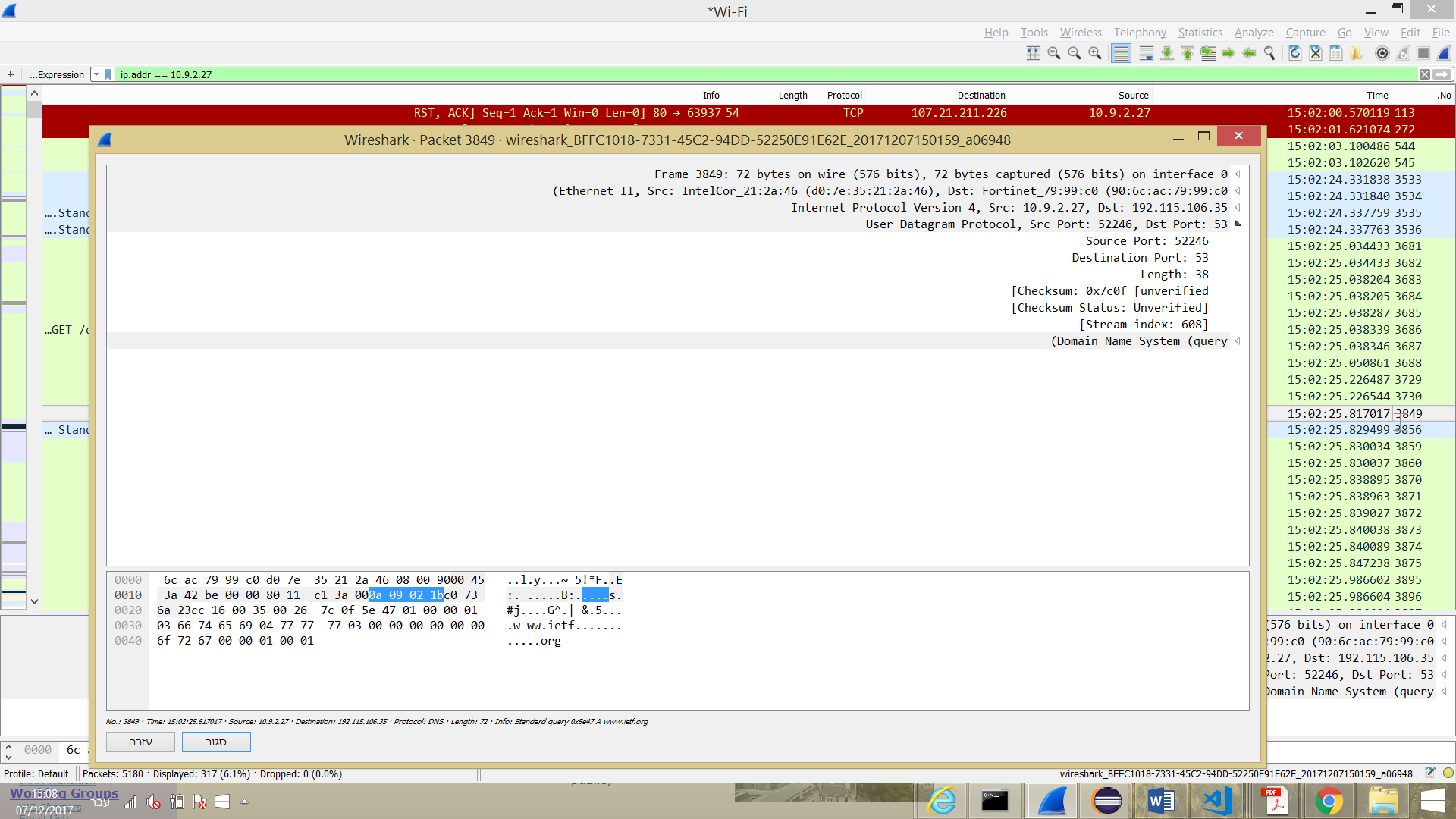
1. הip של ה DNS SERVER הוא 192.115.106.35



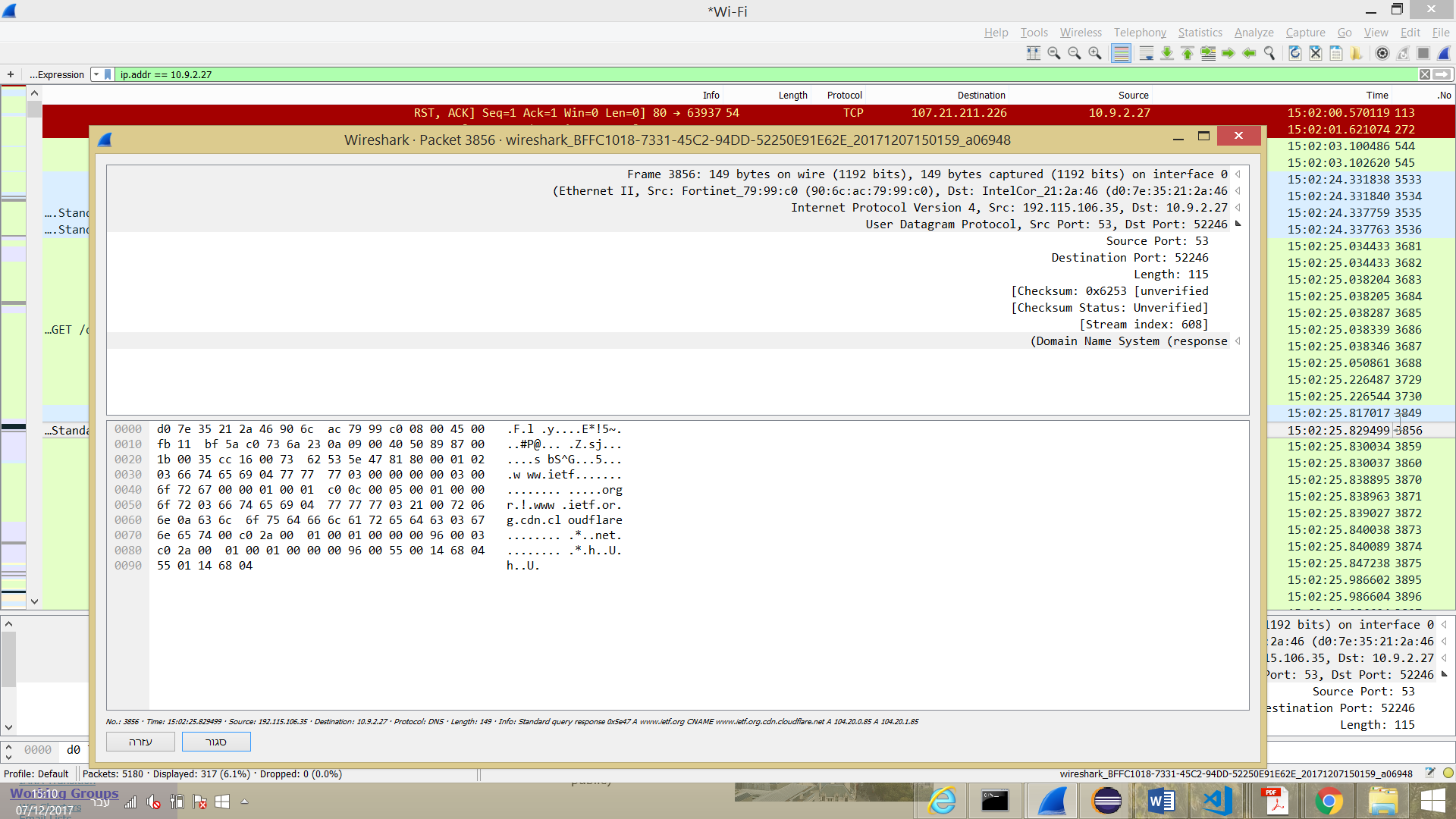
1. ההודעות נשלחו ב UDP



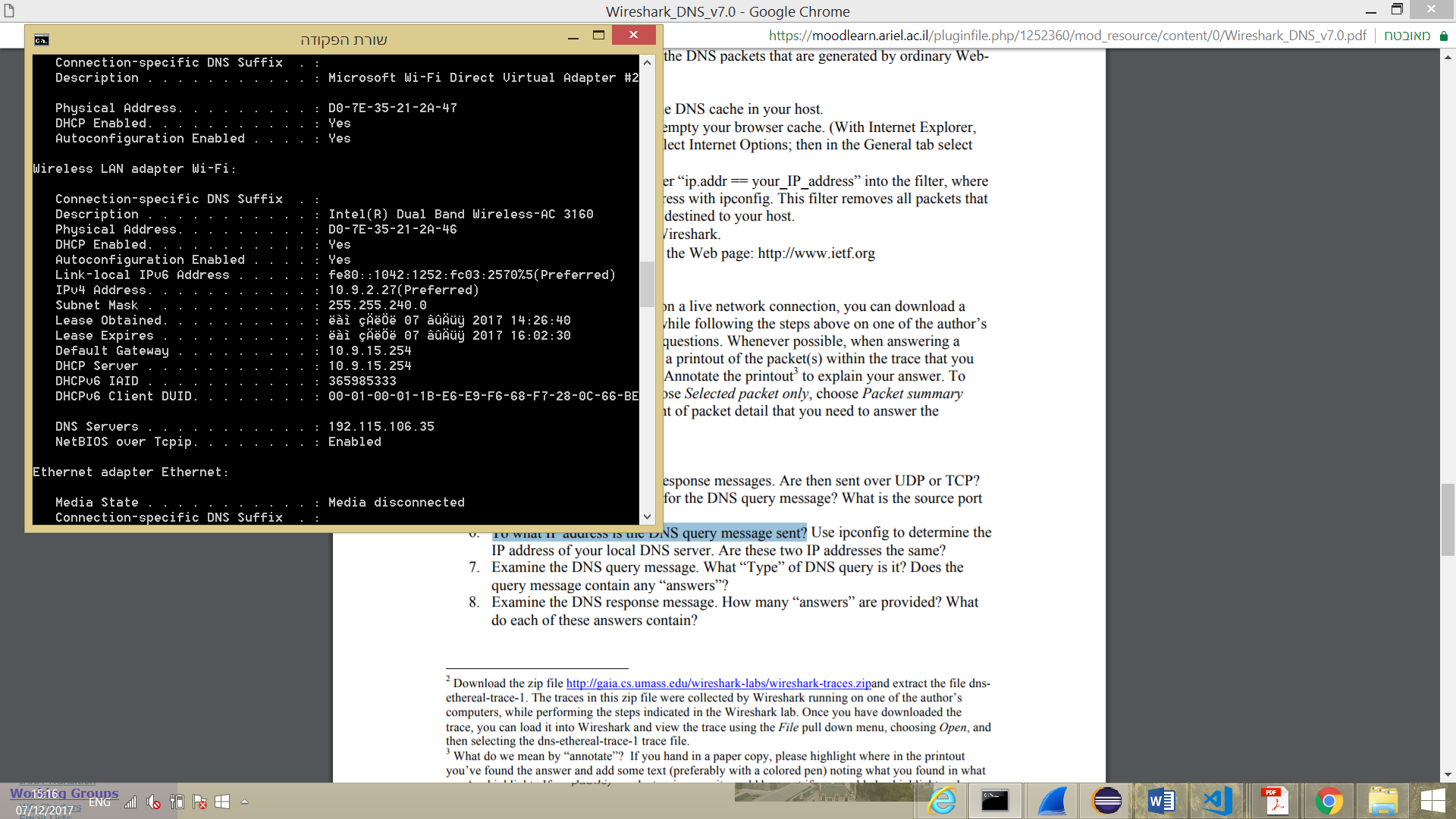
1. Port היעד של הquery הוא 53

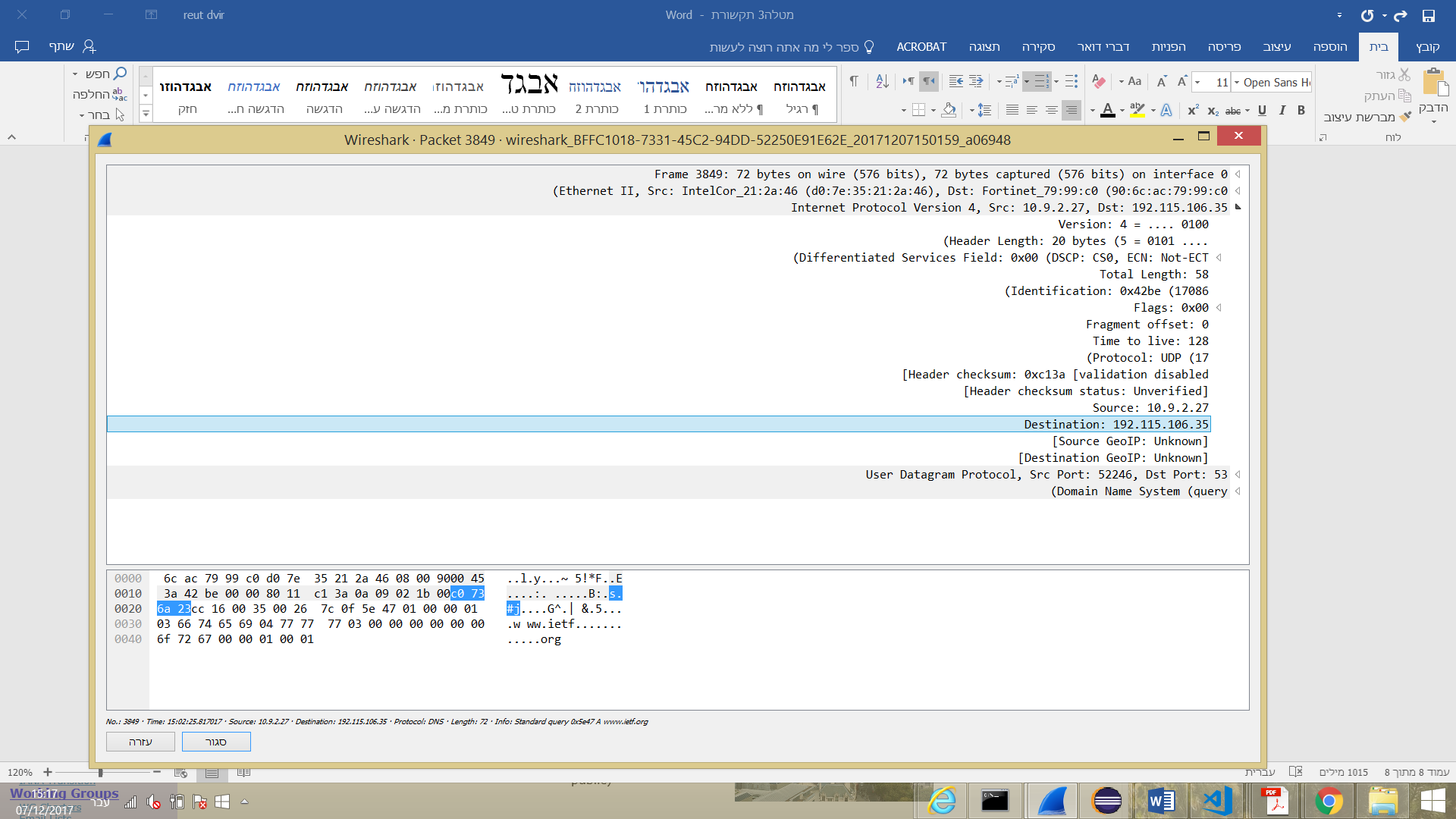


Port המקור של ה response הוא 53

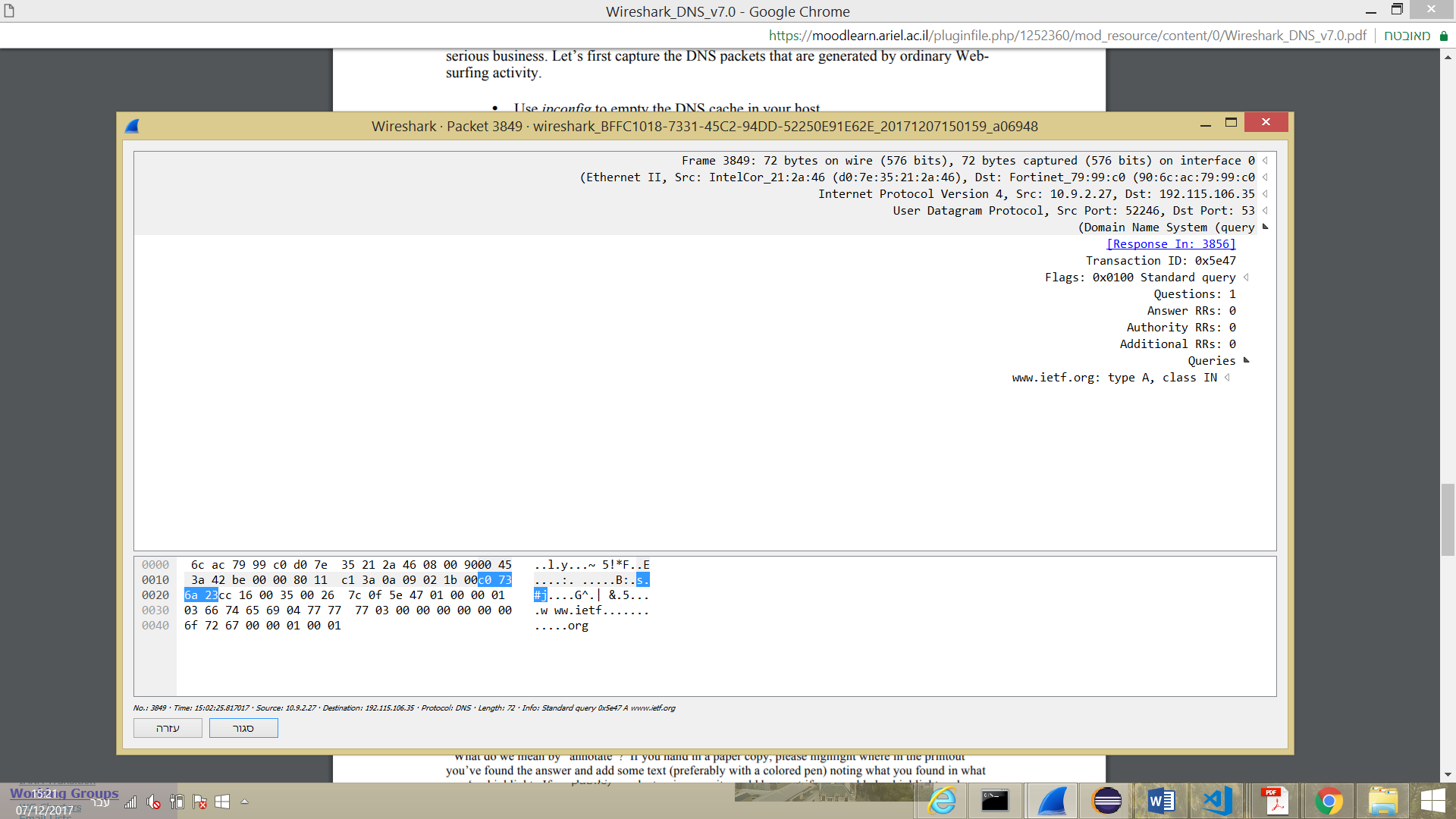


1. זה נשלח לip 192.115.106.35 לפי ה ip config זהו השרת DNS



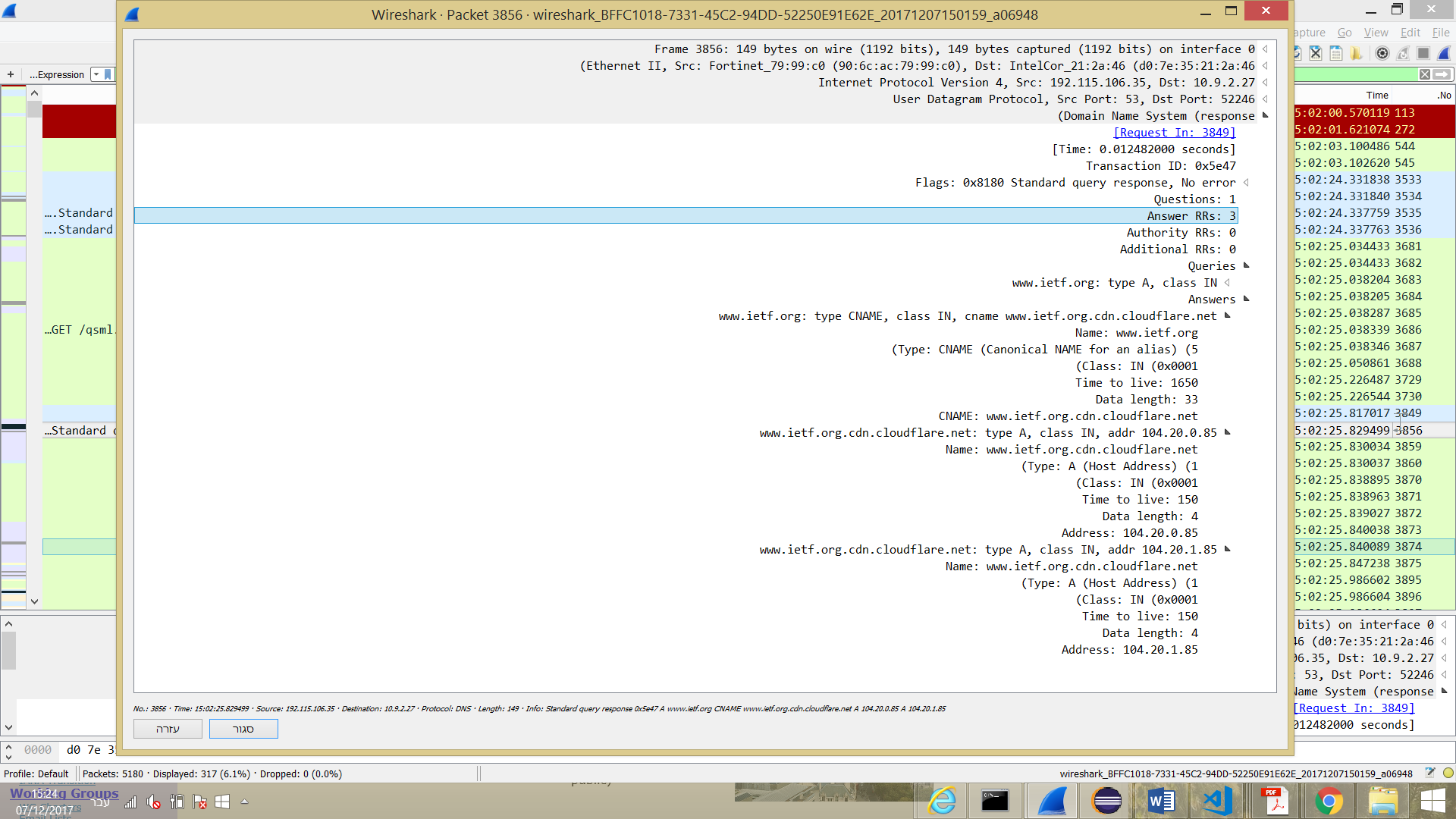


1. ה DNS query הוא מסוג A, והוא לא מכיל תשובות



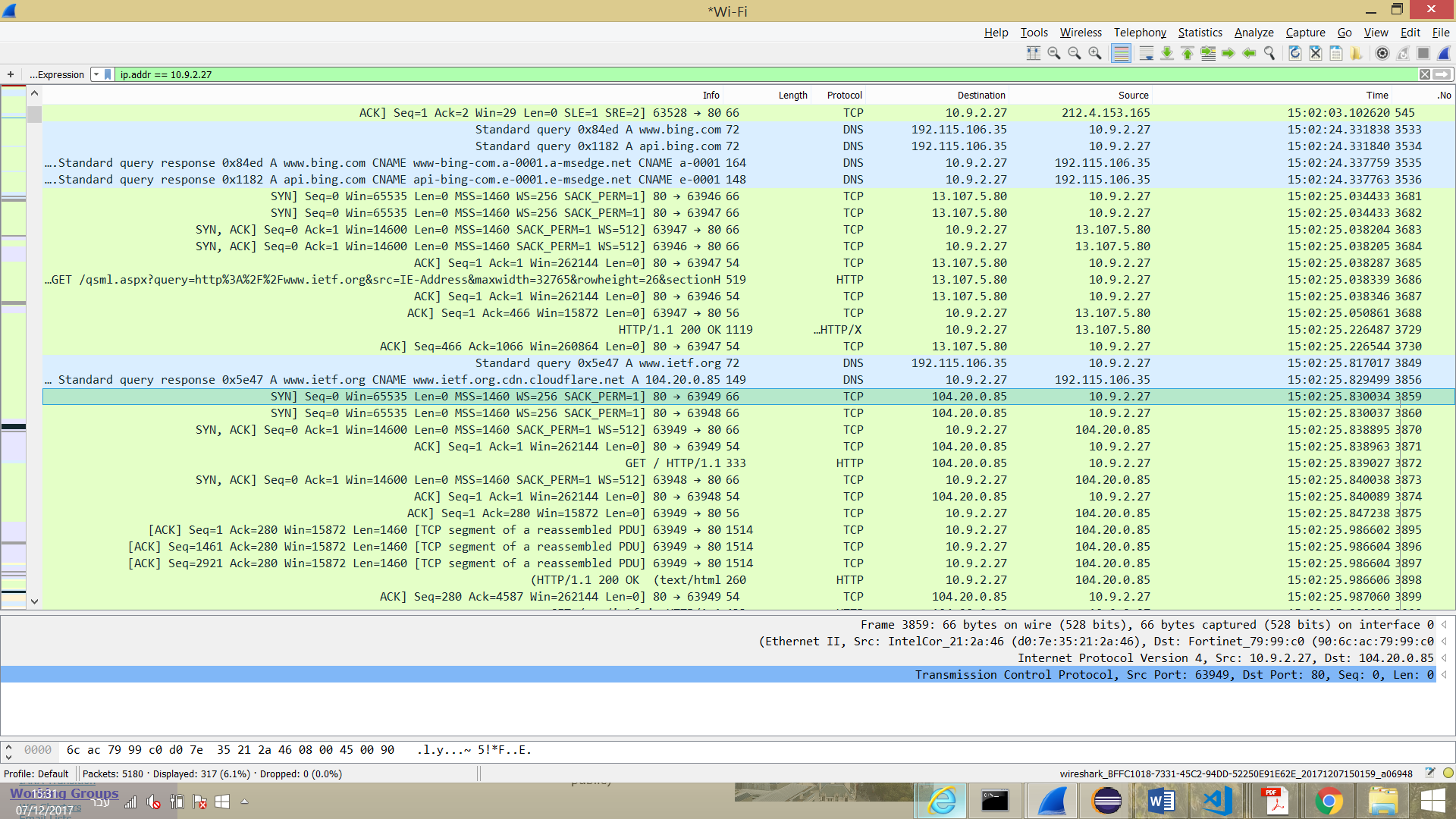
1. התקבלו 3 תשובות, והן מכילות מידע על השם של הhost, סוג הכתובת, class,

Time to live, data length, ip address.

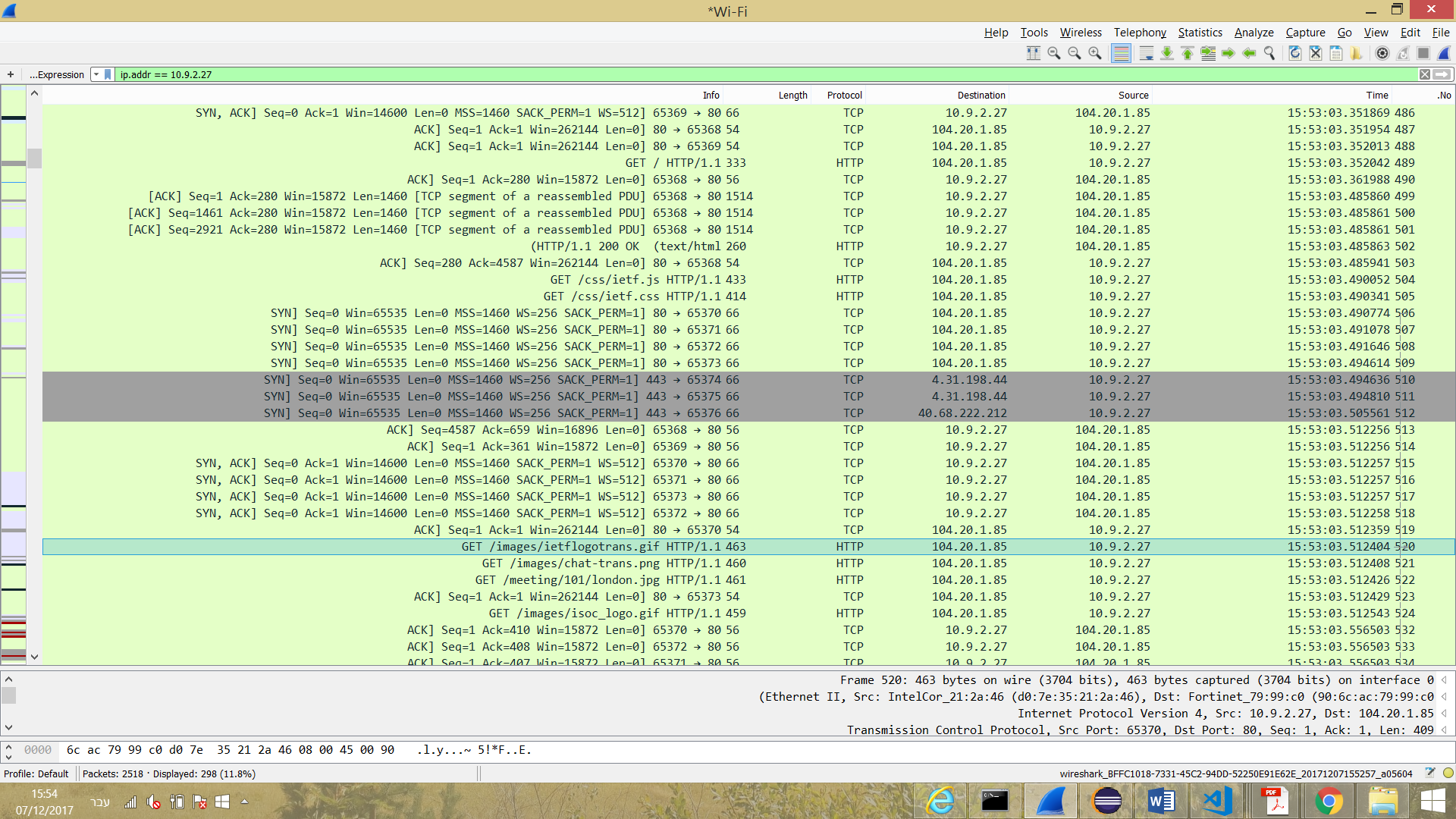


1. הודעות syn הראשונה התקבלה בפקטה 3859, הip של היעד הוא 104.20.0.85

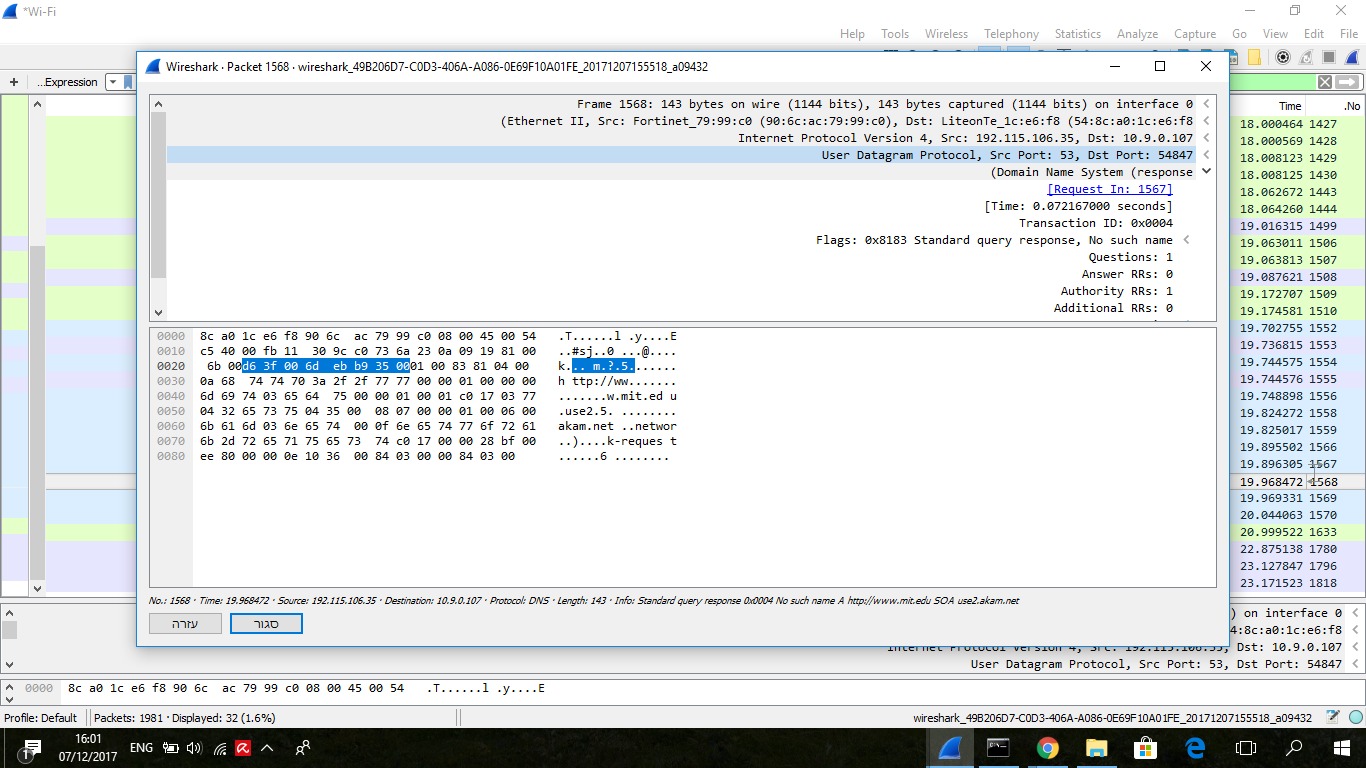
את הip הזה קיבלנו מהודעת הresponse.



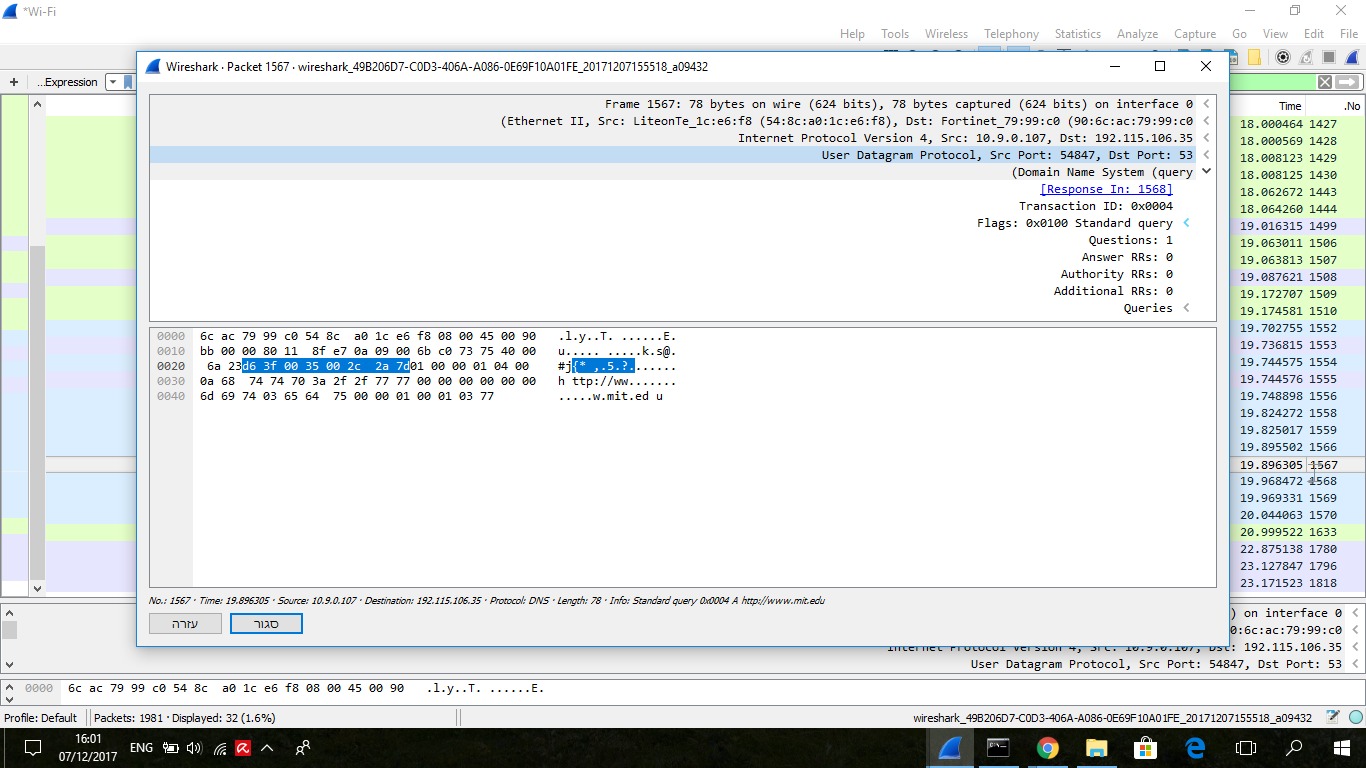
1. לא הייתה עוד בקשת DNS query לפני בקשת הget של התמונה.



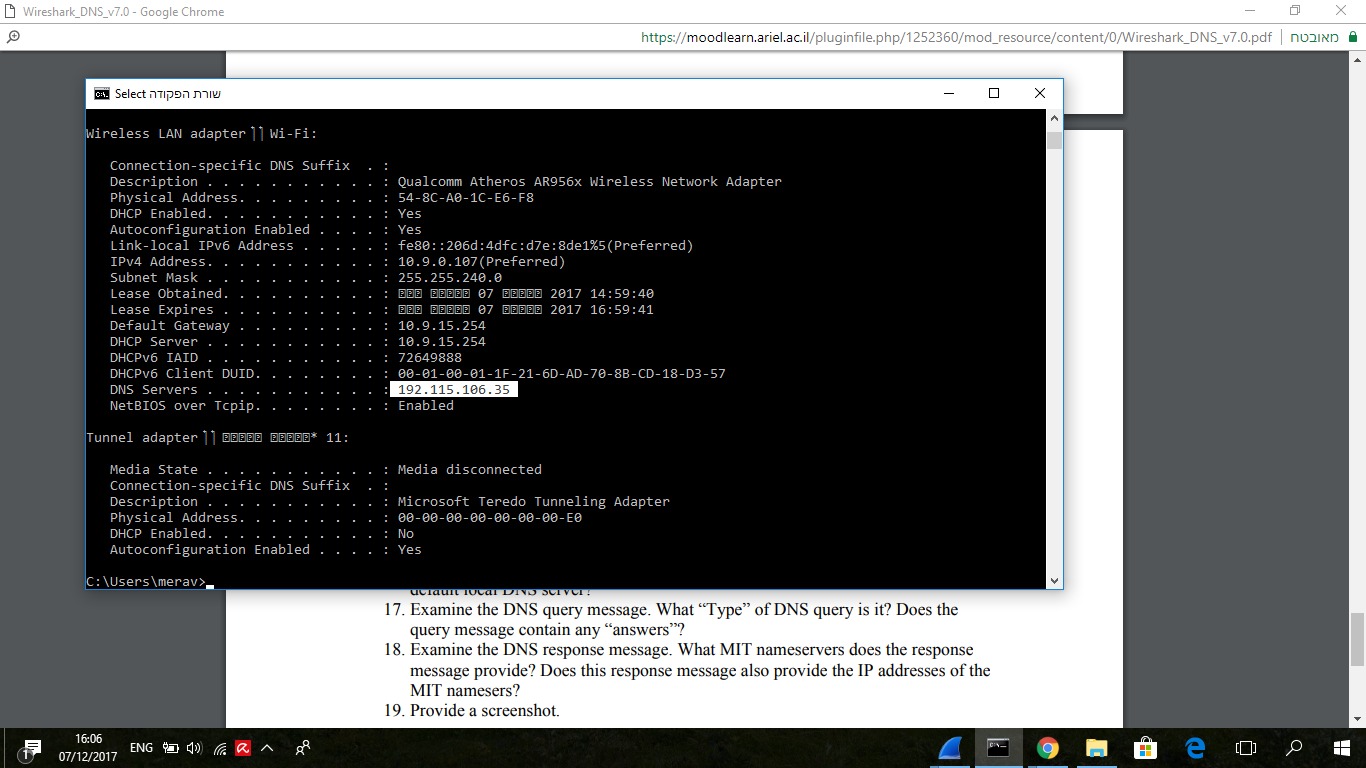
1. Port המקור של ה response הוא 53

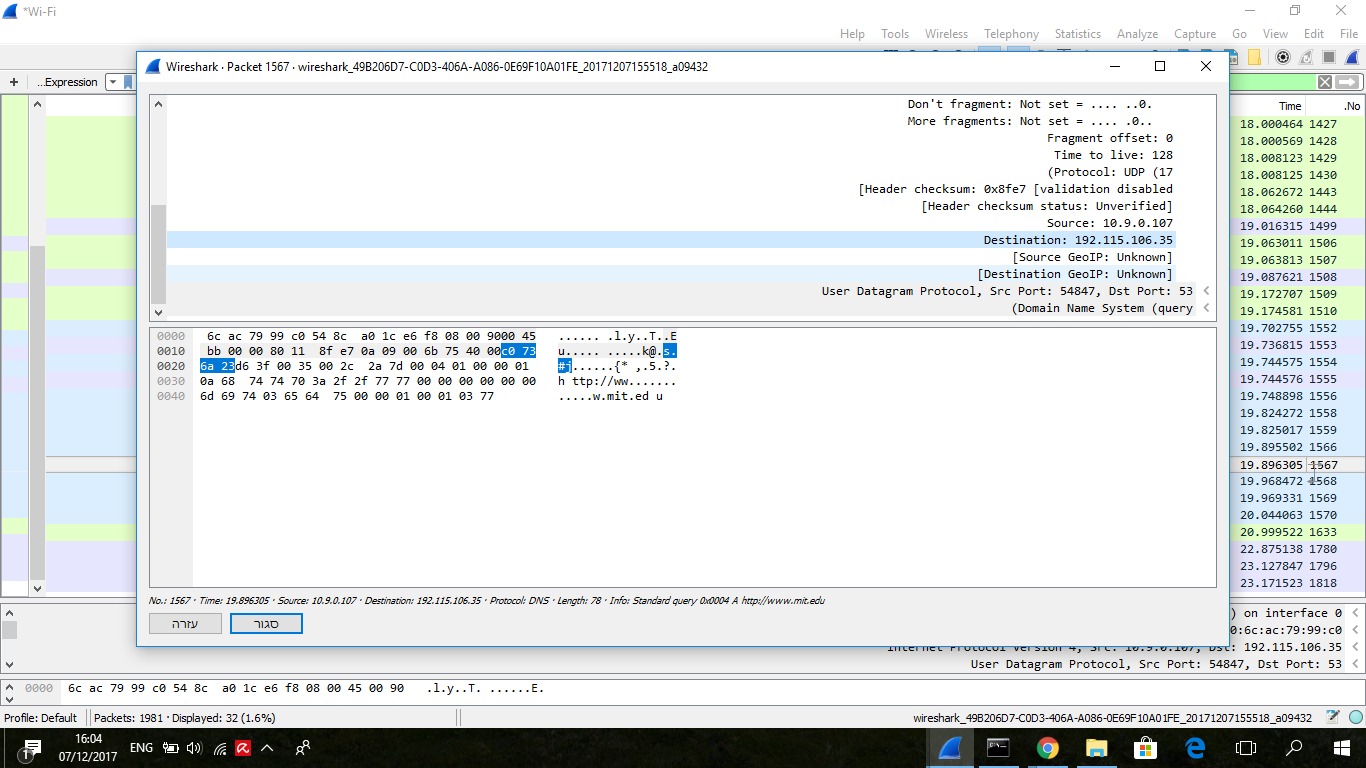


Port היעד של הquery הוא 53

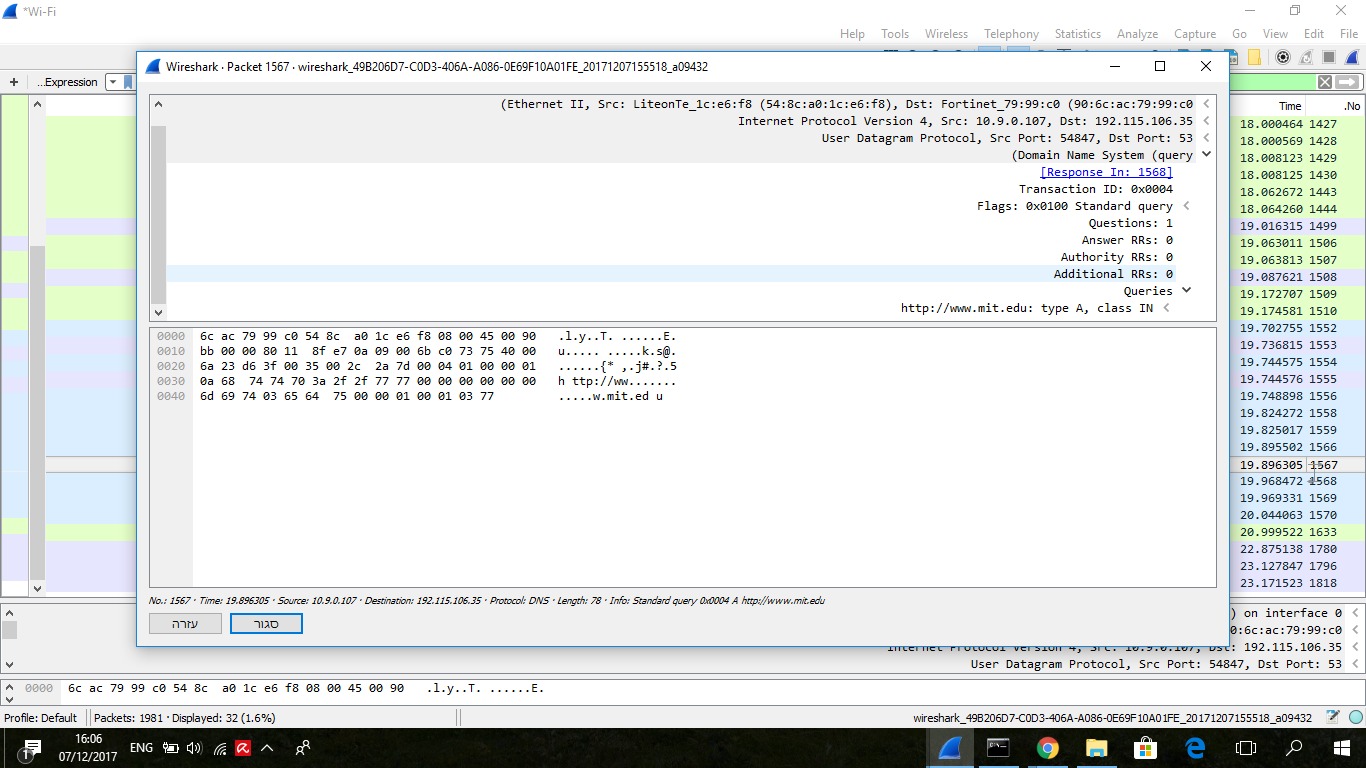


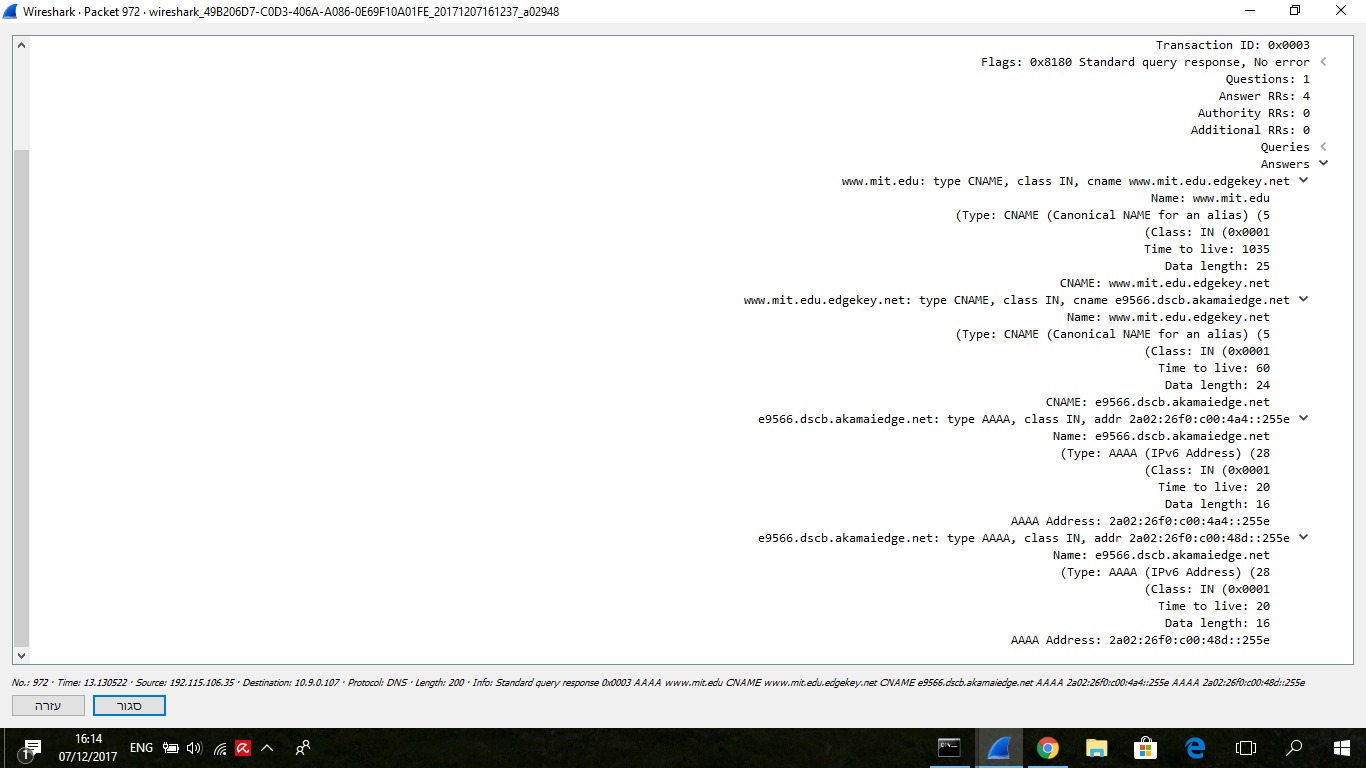
1. זה נשלח לip 192.115.106.35 לפי ה ip config זהו השרת DNS





1. ה DNS query הוא מסוג A, והוא לא מכיל תשובות



1.  התקבלו 4 תשובות, והן מכילות מידע על השם של הhost, סוג הכתובת, class,

Time to live, data length, ip address, cname.