**מטלה 6**

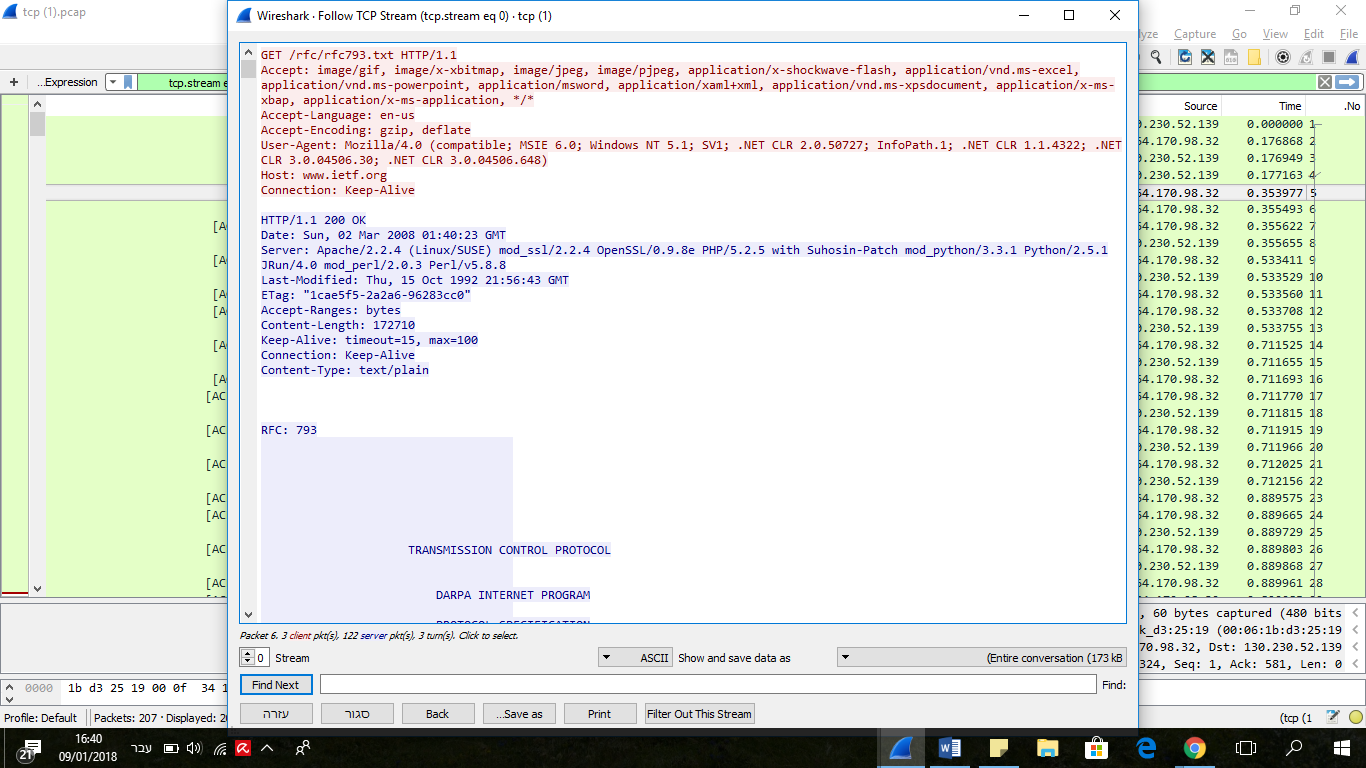
**מגישות:**

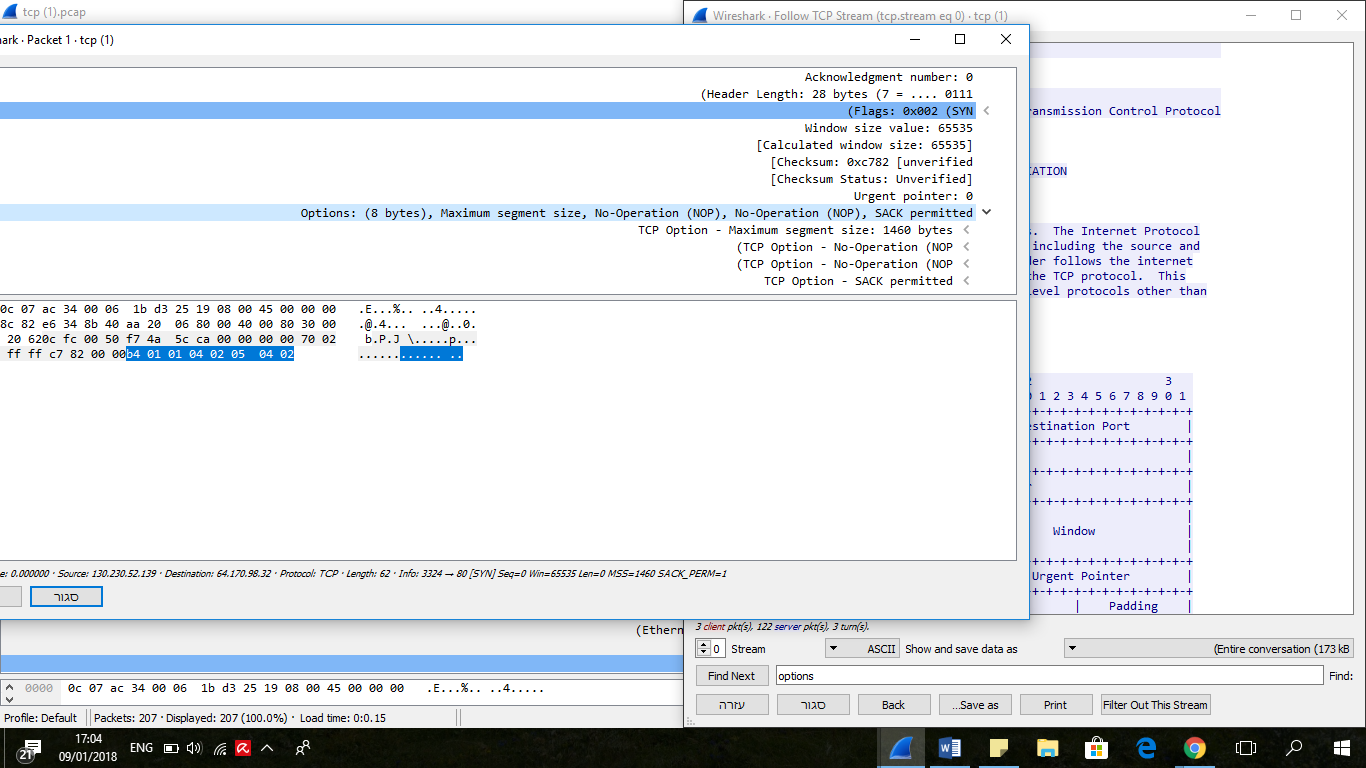
**מירב בוים 206489155**

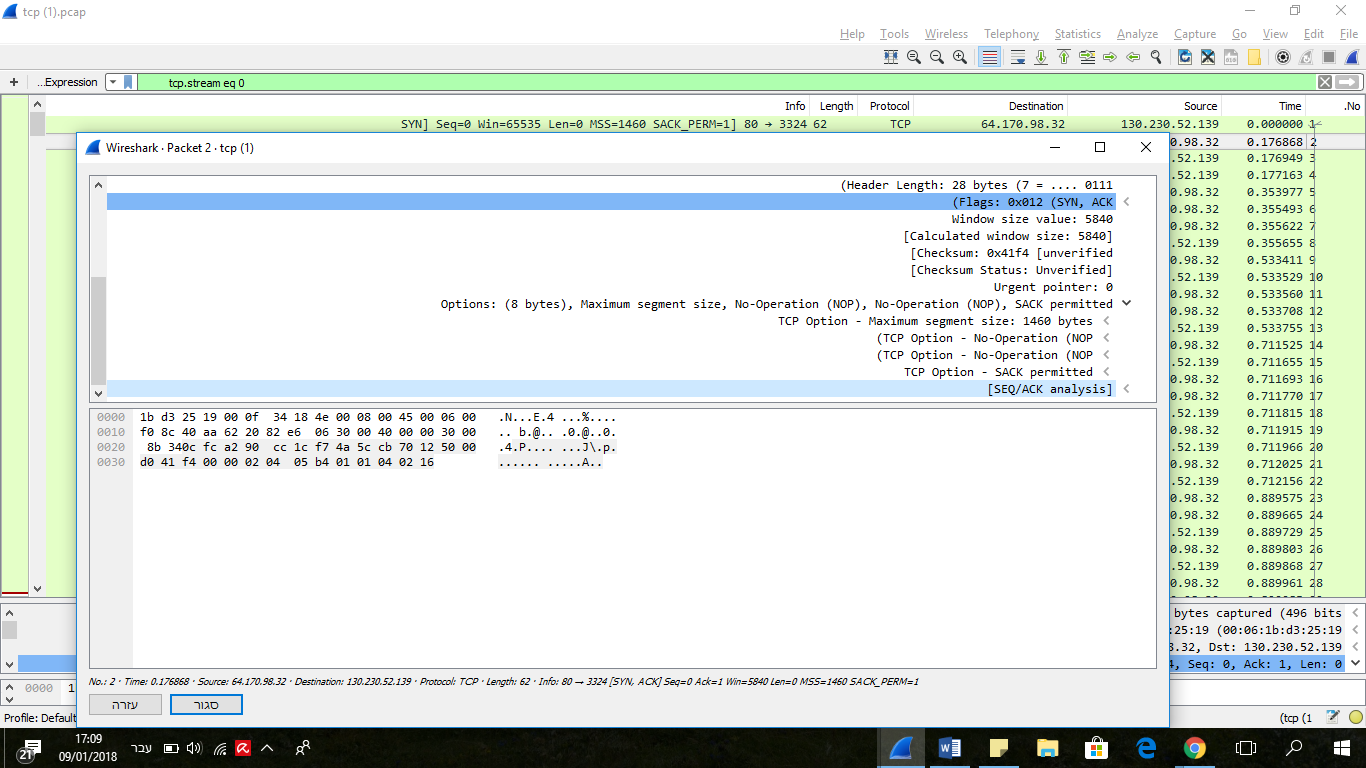
**נעמה שטאובר 316162114**

**רעות דביר 204121446**

**חלק ראשון:**

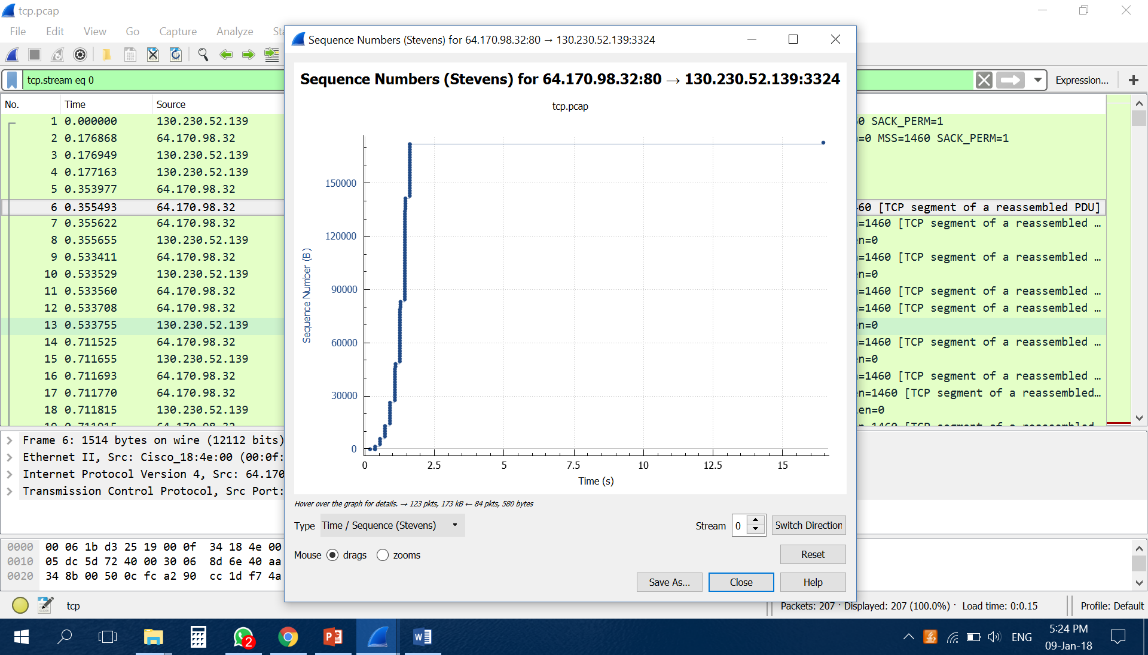
1.

2. בשימוש ע"י המחשב: Nop ,Maximum segment size (MSS), SACK permitted

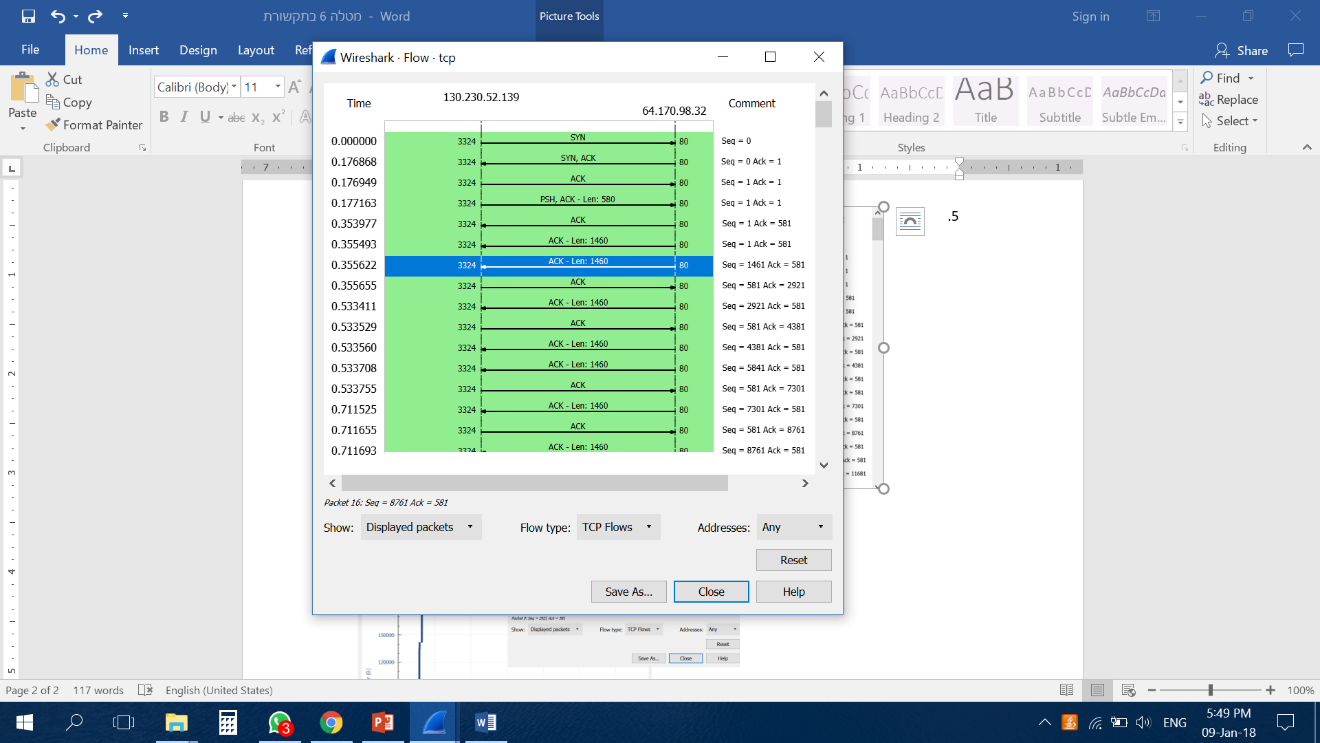
בשימוש ע"י השרת: Nop ,Maximum segment size (MSS), SACK permitted

3. MSS של השרת : 1460 MSS של המחשב: 1460

4. גודל החלון ההתחלתי של השרת הוא: 6960



5.



בפקטה הראשונה כל צד מתחיל עם seq אפס וגם ה- ack הוא אפס, כי עדיין אין צד משלים לשיחה.

בפקטה השנייה השרת מגיב ללקוח עם seq אפס, כי מדובר בחבילה הראשונה ב- session זה, וב - ack של 1. ack מוגדר 1 כדי לציין את קבלת הדגל SYN של הלקוח בפקטה הראשונה.

בפקטה השלישית הלקוח מגיב ל- seq של השרת עם אפס ועם מספר Ack של 1. הלקוח כולל את ה- seq שלו 1.

הפקטה הרביעית היא הפקטה הראשונה שבה יש מטען בפועל. ה- seq נשאר ב 1, בגלל ששום נתונים לא הועברו מאז המנה האחרונה. מספר ההודאה נשאר גם הוא 1, כי לא התקבלו נתונים מהשרת.

הפקטה החמישית נשלחת על ידי השרת כדי לאשר את הנתונים שנשלחו על ידי הלקוח בפקטה 4 בעוד השכבות העליונות מעבדות את בקשת HTTP.

מהפקטה שישית והלאה הלקוח של seq יישאר יציב ב- 581, כי אין לו נתונים להעביר מעבר לבקשה הראשונית 580 bytes. ה- seq של השרת, לעומת זאת, ממשיך לגדול כאשר הוא שולח יותר סיגמנטים של תגובת HTTP.

6. א. Ack

ב. השדות שלא משתנים:

**Urgent data pointer**: אם דגל URG נושא ערך "1", הערך בשדה זה מציין את ההיסט למיקום מידע דחוף.

**Reserved**: שדה השמור לשימוש עתידי (אמור להכיל אפסים).

הם לא משתנים כי הם לא בשימוש.

ג. השדות שמשתנים:

**Port source**: שדה המכיל את מספר הפורט במחשב המקור.

**Port destination**: שדה המכיל את מספר הפורט במחשב היעד.

**Sequence number**: הלקוח הוא משתמש ב- seq כדי לעקוב אחר כמה נתונים שה- TCP session שלח. מספר זה כלול בכל פקטה המועברת, ומוכר על ידי המארח כמספר אישור כדי ליידע את המארח המשגר כי הנתונים המועברים התקבלו בהצלחה.

**Acknowledgement number** : שדה מכיל את המספר הסידורי של ההודעה הבאה לה מצפה התחנה ומהווה אישור קבלה על כל ההודעות בעלות מספר סידורי הקטן ממנו.

**Header length** : שדה זה מכיל את גודל ה- header של החבילה הנוכחית שהוא גם ההיסט מתחילת החבילה עד לתחילת המידע.

**Flags** : סיביות בקרה המציינות מצבי חיבור שונים או מידע על אופן הטיפול בחבילה.

**Window size**: מספר הבתים אותם יכול המחשב לקלוט החל ממספר ההודעה שצוין בשדה ACK.

**Checksum** : מספר האימות של הפתיח וחלק מהפתיח של שכבת ה IP.

**Options** : שדה אופציונלי שיכול להכיל בין היתר את גודל הסגמנט המרבי (Maximum Segment Size, MSS) לשימוש בפרוטוקול.

**Data** : שדה שתפקידו למלא את הפתיח ב-"0" על מנת להגיע לכפולה שלמה של 32 סיביות.

7. בפקטות 1עד 3 החיבור של ה- TCP שגוי. ה- ACK לא נקלט משום שאחרי בקשת ה- SYN השרת לא הגיב וביקש חיבור SYN-ACK ולכן, ה- ACK לא נלכד. כדי שהחיבור יעבוד הלקוח צריך לשלוח בקשת חיבור SYN והשרת יגיב SYN-ACK ורק אז ה- ACK של הלקוח ילכד והחיבור יעבוד.

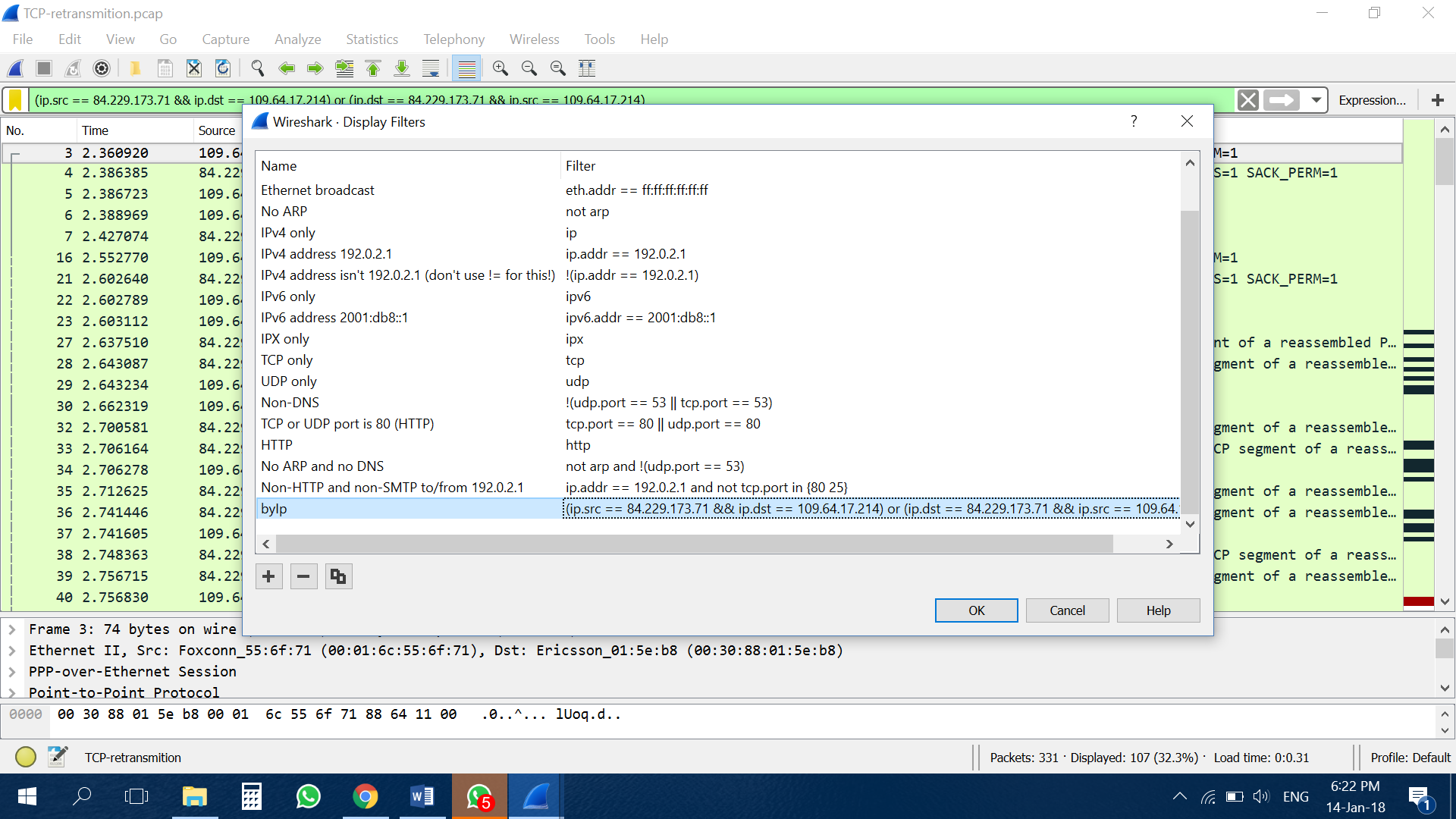
בפקטות 15 עד 17 ה- ACK בשליחת SYN-ACK הוא 0 . כדי שהחיבור יעבוד כמו שצריך הערך שלו צריך להיות 1.

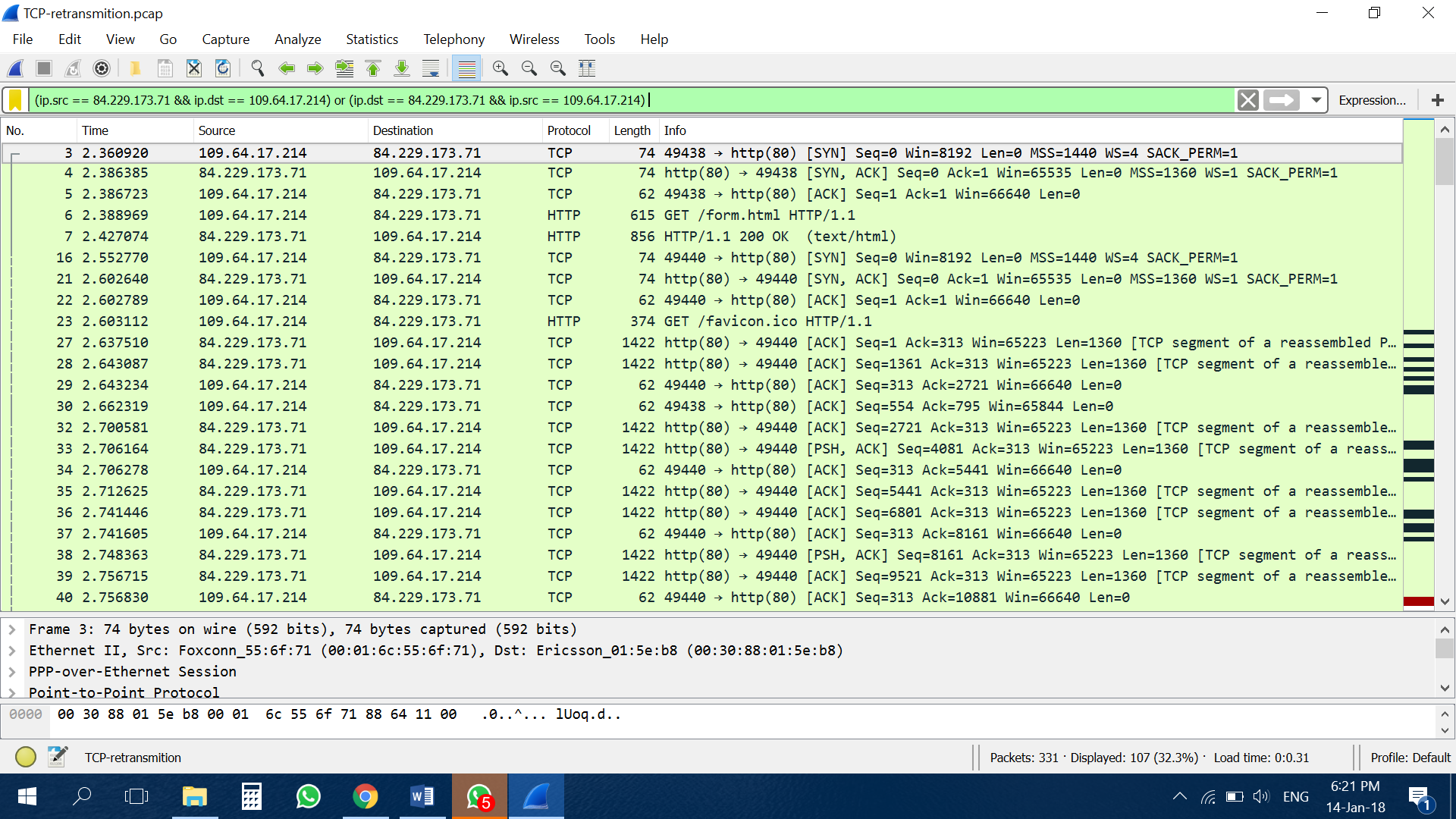
8. בגלל שבפועל הפקטות הגיעו אבל לא בסדר שהם נשלחו אז זה מראה כאילו לא התקבל תשובה מהשרת למרות שכל המידע הגיע מהשרת.

9.העברנו מHex לAscii וזה לא עובד ולא נותן תווים הגיוניים.

10. אם TCP מאבד פקטות החיבור מתעכב, משום שהפורמט שלו מאפשר העברת נתונים מלאה.

ל- UDP לא אכפת מאבדות בשכבת התחבורה ברשת ולכן הוא ימשיך לשלוח נתונים כרגיל.

11.



12.

13. בפקטות מספר 174 209 221 היה חיבור מחדש בגלל שנאבד הנתונים בדרך

**חלק שלישי:**

1. rar/wrar371.exe

2. גודל החלון המקסימלי של הלקוח: 65535.

גודל החלון של השרת בתחילת השיחה: 5840.

3. בין 1.2 שניות לבין 3.9 שניות חלון ה- TCP שנקוב ע"י המקבל מלא ולכן ערך ה- Seq נשאר אותו דבר.

4. הסגמנט של keep-alive מכיל bytes0.

וזה מתאים למפרט של : RFC 1122

"מימוש יכול לשלוח סגמנט keep-alive ללא נתונים (נקרא גם alive -window probe) .

עם זאת ,הוא יכול גם להיות מוגדר כך שישלח בית אחד של זבל".

5.הערך התחלתי של ה-RTO הוא 0.267903 לפי החישוב והנתונים במטלה:

RTO=(1.483208-1.21527)-RTT=(1.483208-1.21527)-0.0003500

6. השרת לא מגדיל בצורה מעריכת את המרווחים בין חבילות ה- keep-alive, אלא ה RTO משתנה

מזמן 1.21527 עד 3.895820.