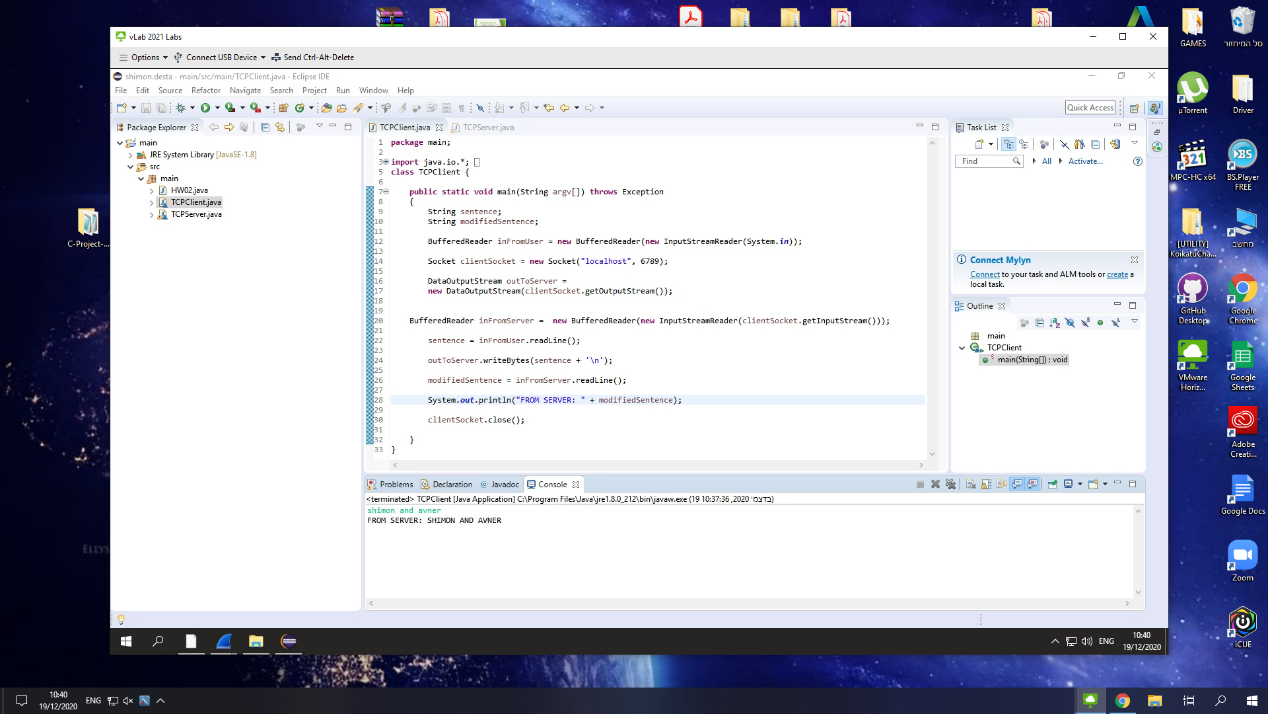
מגישים:   
אבנר לוי 308063395  
שמעון דסטה 203670286

**Lab 1 – Sockets**

1. המסך של הקליינט:  
  
  
  
  
2. כשאפליקציה פותחת סוקט כלשהו, הסוקט מכיל 5 שדות והם: באיזה פרוטוקול לעבוד, ה-ip של המחשב שיוצר את הסוקט, הפורט של המחשב שיוצר את הסוקט, ה-ip של המחשב שאיתו מתקשרים, והפורט של המחשב שאיתו מתקשרים.

כתובת IP – כתובת באורך קבוע שמשמשת כנקודת זיהוי של תחנת קצה (מחשב לדוג').  
Logical Port – תהליך ספציפי שדרכו תוכנות יכולות להעביר נתונים, המספר הם בין 0 ל-65536.  
סוג חיבור – באיזה פרוטוקול בשכבת התעבורה המחשבים יתקשרו.

3. על שדה ה-ip מערכת ההפעלה של ה-client אחראית וגם על הפורט.

על שדה ה-ip של השרת מערכת ההפעלה של השרת אחראית וגם על הפורט שלו.

(ה-client משיג את כתובת ה-ip של השרת באמצעות DNS)

על שדה הפרוטוקול אחראית האפליקציה.

4. Source port – 2 bytes.  
Destination port – 2 bytes.  
Source IP – 4 bytes.  
Destination IP – 4 bytes.

5. נשתמש בפרוטוקול ה-UDP כיוון שהוא יותר מהיר מפרוטוקול ה-TCP, מאחר והוא לא מחכה לתשובה מהיעד שהמידע הגיע.

6.א. המשמעות של המספר 6789 זה הפורט של האפליקציה שהקליינט רוצה לתקשר איתו.

ב. שייך לשרת

ג. לא, ניתן לתת כל פורט שהוא. (לא פורט סטנדרטי)

7 א. שרת שעובד ב-udp שולח הודעה ולא מחכה לתשובה, לכן אין צורך בעוד סוקטים (זאת אומרת נפתח רק סוקט אחד). לעומת זאת, ב-TCP נפתח סוקט גם לשליחה וגם לקבלת התשובה מהיעד

ב. השרת יצטרך n סוקטים.

8. א. משום שיש צורך שהשרת יפתח קודם כל welcome סוקט

ב. משום שבפרוטוקול ה-udp אין יצירת קשר בפתיחת התוכנית, אלא רק בעת שליחת המידע.

9. לא תיווצר תקשורת ביניהם

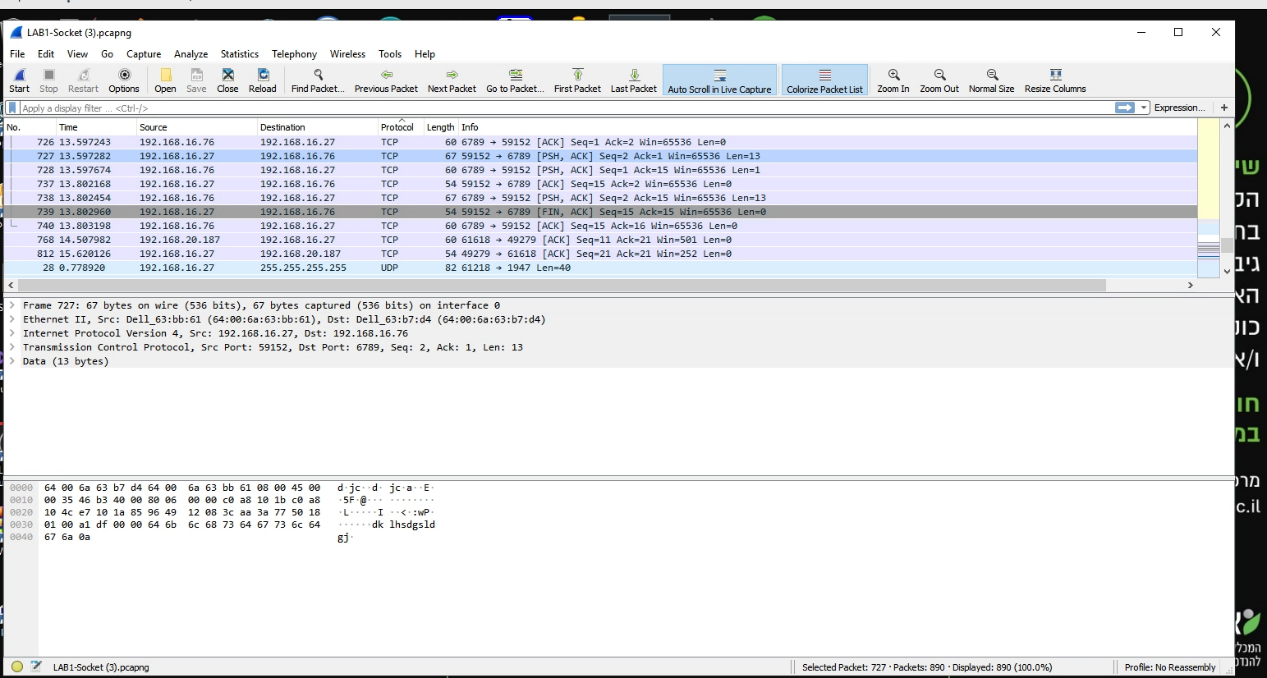
10. Source port number: y.  
Destination port number: x.

11. שני הסגמנטים יכוונו לאותו סוקט, מכיוון שעובדים בפרוטוקול ה-UDP, ניתן לשלוח את כל הסגמנטים לאותו הסוקט, המארח ידע להבדיל בין שניהם ע"י מספר הפורט של השולחים.

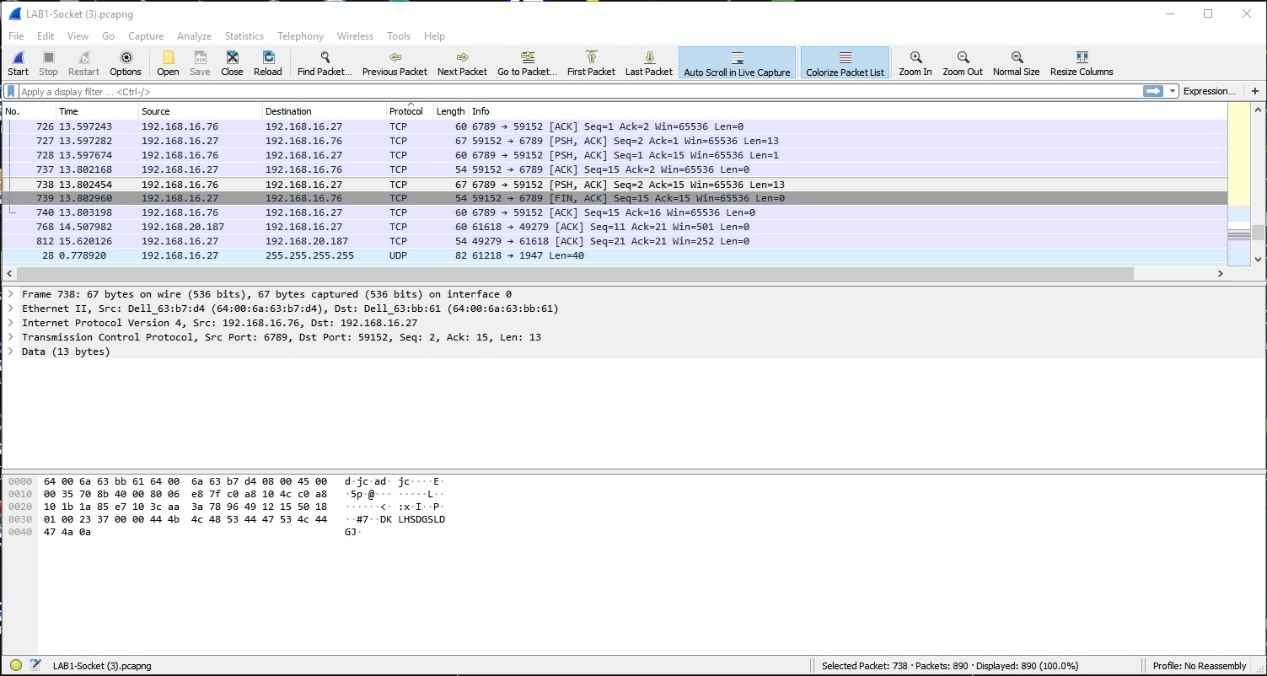
12:   
Server IP: 192.168.16.76  
Client IP: 192.168.16.27

14. הקליינט יזם את החיבור.

15.



61. Data (13 bytes)



17. הפקודה שהקליינט שולח לשרת ברגע שקיבל את מה שרצה היא FIN כפי שמופיע ב-wireshark וזאת בקשת ניתוק אל מול השרת.