רשתות – תרגילים – **Transport Layer**

אין קשר בין התרגילים

יותר ממוזמן לחפש חומרים נוספים באינטרנט כדי לוודא הבנה!

1. ענה על השאלות הבאות:

* מה המטרה של שכבה 4?
* מה זה session?
* מה זה 3 way Handshake?
* מה ההבדל בין tcp ל udp בקצרה?
* מה הפקודה ב linux וגם בwindows שאפשר איתה לראות את כל החיבורים הקיימים על המחשב שלך?

1. פתח את ה vm linux שלך.

* פתח terminal ומשם תפתח wireshark וסנן את התעבורה בחלון סינון והכנס שם tcp.
* תפתח 2 terminalים נוספים שונים אחד ליד שני.
* באמצעות הפקודה nc עליך ליצור ב terminal אחד שרת מאזין ב port: 4567 וב terminal השני להתחבר אליו.
* ב wireshark תוודא שאתה רואה ה ways handshake 3
* שחק עם זה טיפה ותשלח מידע מהשרת ללקוח מהלקוח לשרת.
* תוודא שב wireshark אתה רואה את המידע המועבר הזה (push ack)
* סגור את הלקוח.
* תוודא שב wireshark אתה רואה את סיום התקשורת (fin - fin ack)

1. עליך לעשות בדיוק את מה שעשית בסעיף 2 רק הפעם באמצעות udp. בwireshark סנן את התעבורה ולכן רשום udp בחלון סינון.

* תוודא שב wireshark אתה רואה אך ורק את המידע המועבר מהשרת ללקוח ללא תהליך התחברות כמו שהיה ב tcp.

1. עליך ליצור באמצעות פייתון 2 תוכנות:
   1. לקוח TCP
   2. שרת TCP

* הלקוח צריך לקבל מהמשתמש מספר:



* הוא שולח את המספר הזה לשרת (לא לשכוח לעשות .encode() לפני על מנת להעביר מידע בינארי)
* השרת מקבל את זה (משתמש בrecv()) והוא מחזיר ללקוח את התוצאה של המכפלה של המספר.
* הלקוח מקבל את התשובה של השרת ומדפיס את מה שהשרת שלח לו:



1. עליך ליצור באמצעות פייתון 2 תוכנות:
   1. לקוח UDP
   2. שרת UDP (אני יודע לא לא הדגמנו את זה בסרטון אבל זה אחלה דרך ללמוד דברים לבד בעצמך!)

* הפעם עליך להחליט מה לעשות, כלומר תהיה יצירתי (לקוח צריך לבקש משאב ולבקש מהשרת לעשות פעולה כלשהיא)
* שים לב: שעליך להשתמש בספריית struct (קרא באינרטנט) על מנת להמיר מידע לבינארית. **חובה להשתמש בstruct.**