שיעור 7 – TCP

**תרגיל: מתיידדים עם TCP**

בתרגיל זה נכיר לעומק את מנגנון הפעולה של TCP. שימו לב – מנגנון הפעולה הוא ברמה תיאורטית. עדיין לא דיברנו כיצד הוא ממומש במחשבים במציאות – זאת נעשה בשבוע הבא.

**1. נכון או לא נכון?**על כל היגד, כתבו האם הוא נכון או לא נכון, **והסבירו את עצמכם.**

1. TCP ו-UDP הם פרוטוקולים בשכבת הרשת.

|  |
| --- |
| נכון / לא נכון: לא נכון  נימוק: הם פרוטוקולים בשכבת התעבורה |
| תשובה |

1. שיחת TCP מתחילה בלחיצת יד משולשת

|  |
| --- |
| נכון / לא נכון: נכון  נימוק: הלקוח שולח SYN, השרת שולח SYN + Ack, והלקוח שולח ACK |
| תשובה |

1. בפרוטוקול TCP כל חבילה נשלחת פעם אחת בלבד, גם אם לא התקבל עבורה אישור קבלה (ACK)

|  |
| --- |
| נכון / לא נכון: לא נכון  נימוק: אם לא נשלח אישור קבלה החבילה תשלח שוב |
| תשובה |

1. מטרת לחיצת היד המשולשת היא סנכרון מספרי ה-Seq איתם כל צד יתחיל, ווידוא כי החיבור הוא תקין ואכן השרת האזין בפורט בו פנו אליו.

|  |
| --- |
| נכון / לא נכון: נכון  נימוק: כשנותנים את הACK זה מסנכרן את הSEQ, ואם מקבלים SYN החיבור תקין והשרת מאזין... |
| תשובה |

1. בחבילה בודדת שנשלחת ב-TCP, ה-seq הוא המספר של הבית הראשון בחבילה, ואילו ה-len הוא מספר הבתים בחבילה

|  |
| --- |
| נכון / לא נכון: נכון  נימוק: הSEQ שומר על המספר הבייט הבא שצריך להשלח, כלומר הבייט הראשון. והLEN זה קיצור לlength כלומר אורך- אורך החבילה |
| תשובה |

1. בפרוטוקול UDP אפשר לקיים שיחה דו-כיוונית, כלומר ששני הצדדים שולחים מידע, בעוד ב-TCP המצב אינו אפשרי

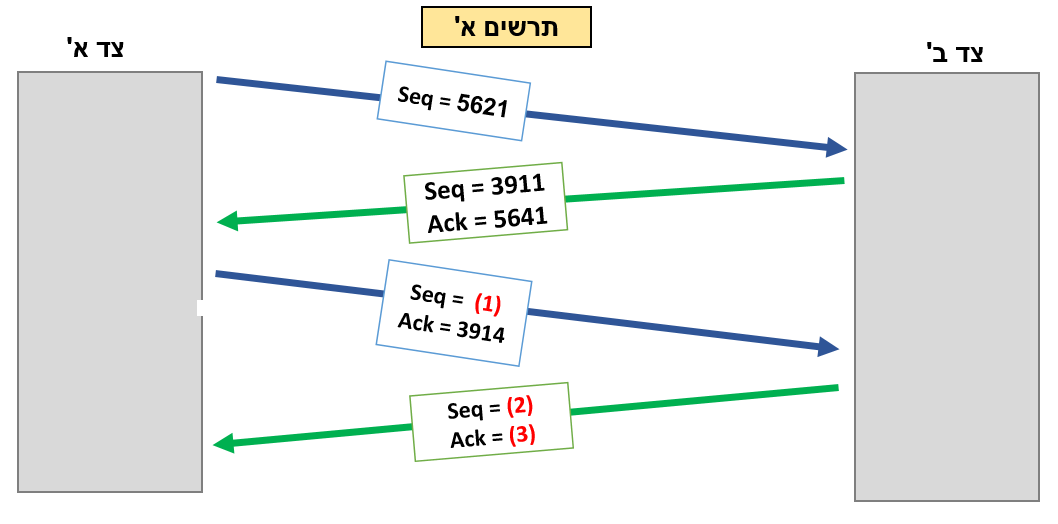
|  |
| --- |
| נכון / לא נכון: לא נכון  נימוק: בשני הפרוטוקולים שני הצדדים יכולים יכולים לשלוח מידע |
| תשובה |

1. ב-TCP נעשה שימוש ב-ISN כדי לשפר את קצב העברת החבילות.

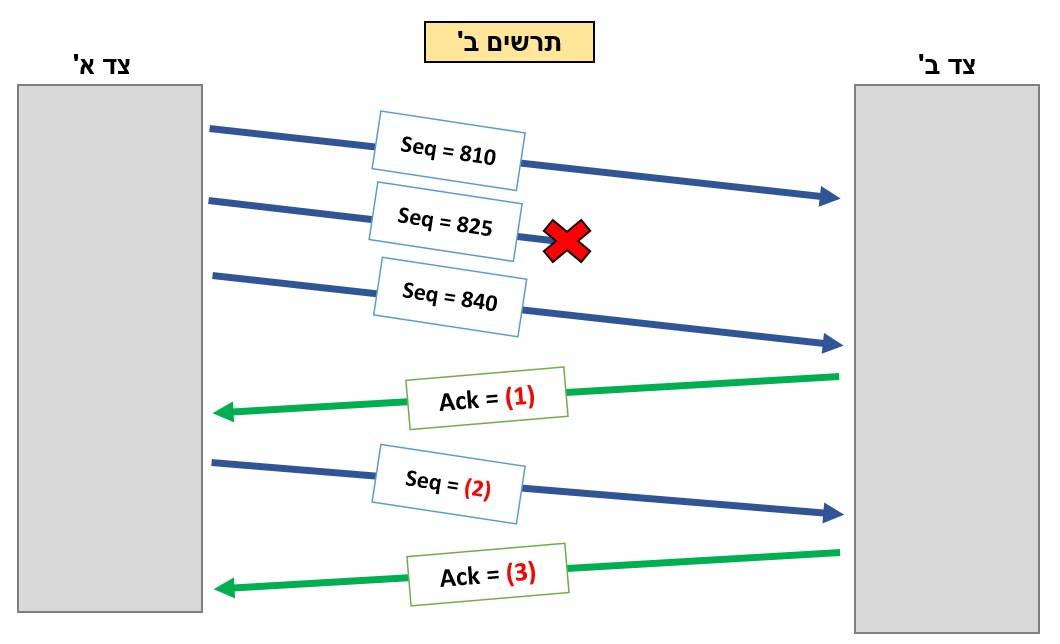
|  |
| --- |
| נכון / לא נכון: לא נכון  נימוק: ISN זה המספר שבו מתחילים את השיחה, לא קשור לקצב |
| תשובה |

1. ב-TCP אם הסדר של החבילות שנשלחו השתבש (לדוגמה, חבילה מס' 2 הגיעה ליעד לפני חבילה מס' 1), הצד המקבל ידאג לסדר אותן.

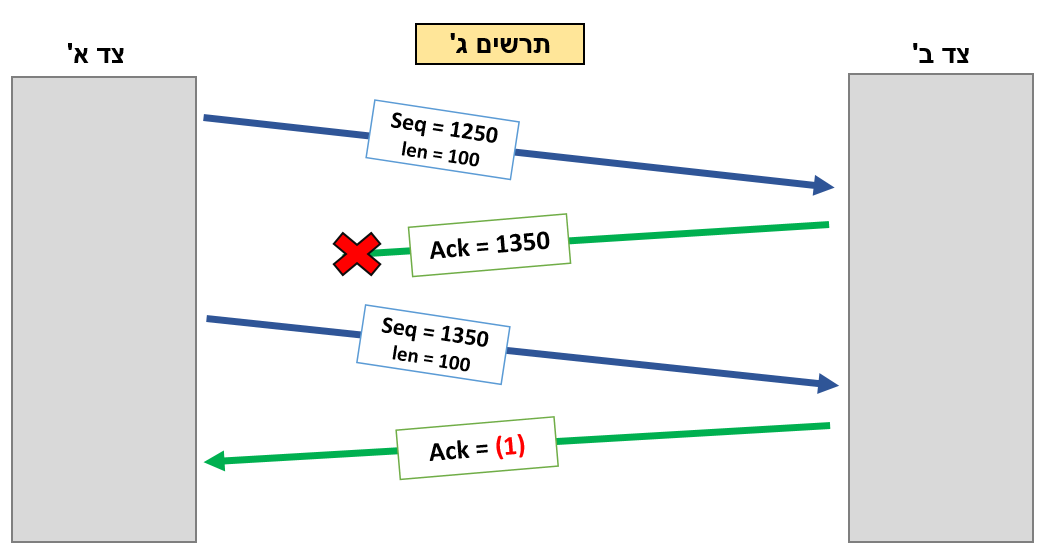
|  |
| --- |
| נכון / לא נכון: נכון  נימוק: בשביל זה קיים הSEQ והLEN |
| תשובה |

**2. השלם את החסר**מלאו את הפרטים החסרים (צבועים באדום) בתרשימים הבאים, והסבירו את תשובתכם:

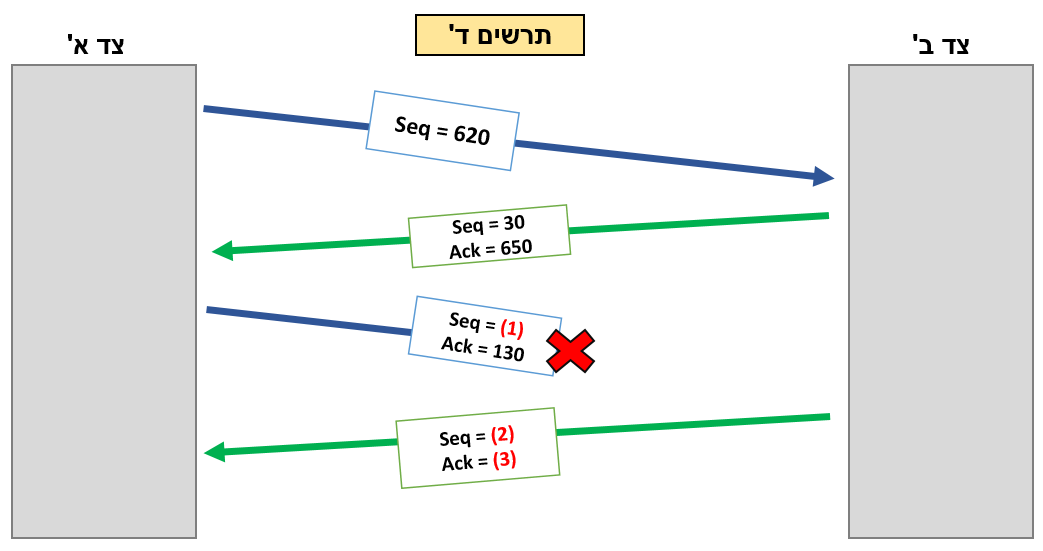
|  |
| --- |
| **(1) 5641 זה הACK של החבילה הקודמת**  **(2) 3914 כי זה ה-ACK של החבילה הקודמת**  **(3) 5661 – the len is 5621-5641 (the first SEQ minus the second SEQ) so we add the blue LEN to the previous blue SEQ and we get the green ACK** |
| תשובה |



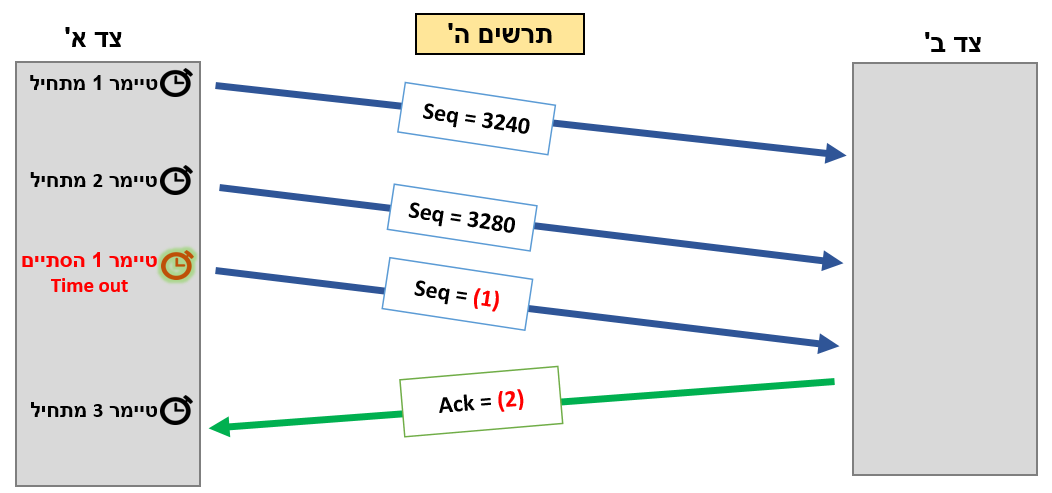
|  |
| --- |
| **(1) 825**  **(2) 825**  **(3) 855**  **הסבר:** חבילה שמתחילה ב825 נאבדה, ולכן האישור אומר להמשיך לשלוח חבילות מ825 כי הוא לא קיבל את החבילה הזו, 840 כבר נשלחה אבל בגלל האישור שלא קיבל את 825 825 תשלח שוב. ואז יגיע האישור שיגיד לנו למשיך מ855 כי 840 כבר הגיע |
| תשובה |



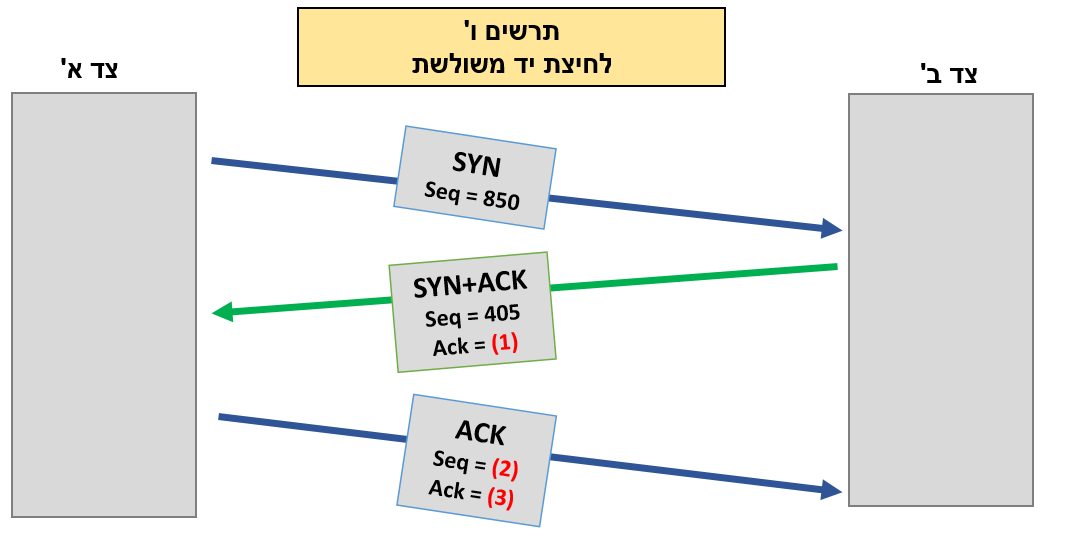
|  |
| --- |
| **(1) 1450**  **הסבר:** כי צד ב' כבר קיבל את 1350 ויכול להיות שהוא אפילו לא שם לב ולא אכפת לו שהאישור לא הגיע. לכן צד ב' יאשר להמשיך מהחבילה הבאה- 1450 |
| תשובה |



|  |
| --- |
| **(1) 650**  **(2) 130**  **(3) 650**  **הסבר: צד ב' יודע להמשיך לשלוח כמה חבילות גם בלי אישור אז למרות שהוא לא קיבל ACK הוא יודע להמשיך ל130, אבל הוא קיבל רק עד 649 אז הוא מבקש להמשיך מ650 למרות שצד א' חושב ש650 כבר נשלח** |
| תשובה |

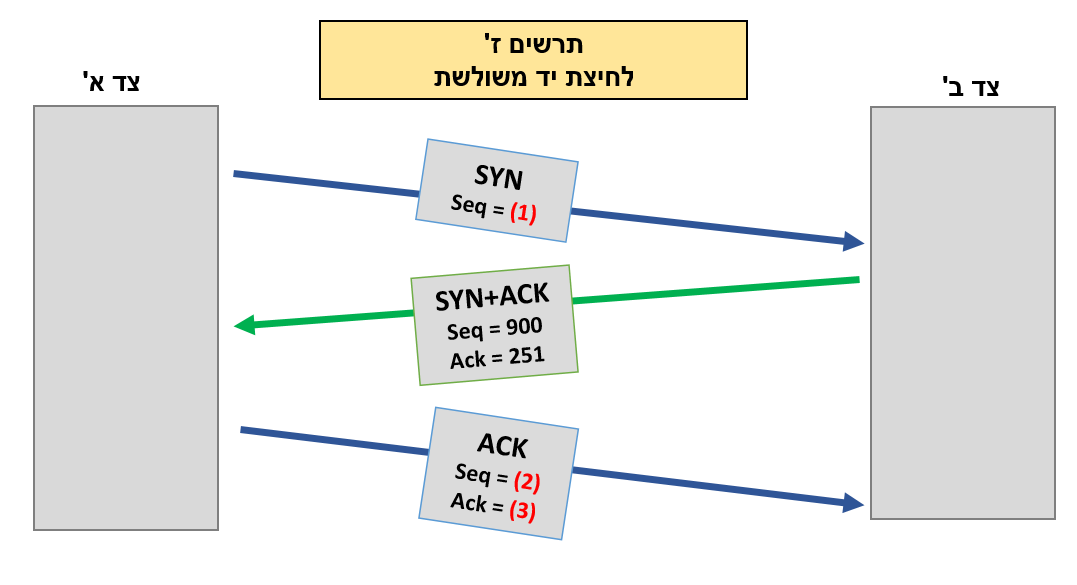


|  |
| --- |
| **(1) 3240**  **(2) 3320**  **הסבר:** בגלל שהטיימר של 1 הסתיים בלי ACK הוא שולח אותו שוב. הACK שקיבל גם את 3240 וגם את 3280 צריך לבקש להמשיך מ3320 |
| תשובה |



|  |
| --- |
| **(1) 851**  **(2) 851**  **(3) 406**  **הסבר:** הLEN של SYN הוא 1. לכן- אם הSEQ בSYN הוא 850, הACK של האישור למשיך יהיה 850 + 1 שזה 851 (!!!) השאר זה אותו עיקרון, (2)פשוט מתחיל מהACK הקודם, ו(3) הוא כמו (1) |
| תשובה |

|  |
| --- |
| **(1) 250**  **(2) 251**  **(3) 901**  **הסבר:** דומה לשאלה הקודמת אבל השגתי את הSEQ הראשון ע"י זה שאני יודע את הLEN של (1) זה 1 כי זה SYN ואני יודע את הACK הבא כי כתוב – 251. 251 – 1 =250 (!!!) אחרי זה זה פשוט כמו קודם. |
| תשובה |



**3. שאלות פתוחות**כתבו תשובה מפורטת לכל שאלה.

1. השלימו, כיצד נראית לחיצת יד משולשת בפרוטוקול TCP?

|  |
| --- |
| **1. צד א' שולח לצד ב': SYN עם SEQ שנקבע ע"י ISN**  **2. צד ב' שולח לצד א': SYN + ACK עם SEQ שנקבע ע"י ISN וACK שהוא הSEQ של צד א' פלוס 1**  **3. צד א' שולח לצד ב': ACK עם הSEQ של צד ב' פלוס אחד בגלל הSYN** |
| תשובה |

1. באותו אופן, כתבו כיצד נראה סיום תקשורת מסודר בפרוטוקול TCP (כתבו בעצמכם מה כל צד שולח לצד השני):

|  |
| --- |
| **FIN**  **FIN+ACK**  **ACK** |
| תשובה |

1. סהר שינה את ההגדרות בפרוטוקול ה-TCP שלו, כך שהוא לא מגריל ISN אקראי. מה הבעיה שעלולה לצוץ בעקבות זאת?

|  |
| --- |
| **בעיקר שהחבילות יתערבבו וגם שהאקרים צריכים רק לקבל את הISN הקבוע ולא את המספר שמוגרל...** |
| תשובה |

1. הדר משתמשת בגרסת TCP שעובדת ללא טיימר על החבילות שנשלחות. מה עלול להתרחש ומדוע?

|  |
| --- |
| **אם לא נשלח אישור על חבילה מסוימת אחרי הרבה זמן, היא לא תשלח עוד הפעם כי אין לה טיימר שיקבע כמה זמן לחכות** |
| תשובה |

1. נגה איחרה לשיעור, והחמיצה את ההסבר על Sequence Number. הסבירו לה מה זה, ולמה הוא משמש?

|  |
| --- |
| Sequence Number **זה המספר של הבייט הראשון בחבילה כולל החבילות הקודמות כשזה מתחיל ממספר רנדומלי. בעצם זה דרך לשמור על איזה חבילה נשלחה מתי, נגיד שהמספר שהוגרל הוא 82. הSEQ של החבילה הראשונה הוא 82, ונגיד שבחבילה נשלחו 10 בייטים (למרות שזה אמור להיות SYN), אז הSEQ של החבילה הבאה יהיה 92 כי 82 פלוס האורך של החבילה הקודמת.** |
| תשובה |

1. ציין שתי דוגמאות למקרים בהם נעדיף להשתמש בפרוטוקול TCP ושתי דוגמאות למקרים בהם נעדיף UDP (לדוגמה, במה נעדיף להשתמש כאשר נשלח מייל?)

|  |
| --- |
| **באימייל נעדיף TCP כי אנחנו רוצים לקבל את האימייל בצורה אמינה, נכונה, שלמה ומדויקת. ולא עם חורים ושיבושים.**  **באתר אינטרנט אנחנו גם נרצה להשתמש בTCP כי אפילו עם נאבד > אחד- האתר לא יעבוד.**  **בוידאו צ'אט אנחנו נרצה להשתמש בUDP, כי גם אם איזה פיקסל או שניים או פקטה בקול יאבדו או ישבתשו- אנחנו נשרוד. וגם המהירות היא מאוד חשובה גם אם מאבדים משהו פה ושם.**  **בדי אן אס נרצה גם UDP כדי שמהירות הגלישה באינטרנט לא תפגע יותר מדי אם אין לנו כתובת שמורה בCache** |
| תשובה |

1. מדוע TCP נחשב פרוטוקול אמין יותר מ-UDP? שלב בתשובתך דוגמאות מהפרוטוקולים.

|  |
| --- |
| **כי TCP ממש בודק אם חבילה הגיעה ואם צריך לשלוח שוב- הוא ישלח שוב(retransmission). אם חבילה הגיעה לפני החבילה הקודמת הצד יסדר את זה (SEQ), יש אישורים על חבילות (ACK), ויש בדיקה אם הקו מוכן לקבל חבילות (הלחיצת יד המשולשת)** |
| תשובה |

1. מדוע UDP נחשב פרוטוקול מהיר יותר מ-TCP? מה ב-UDP יותר מהיר?

|  |
| --- |
| **כי הוא פשוט יותר. הוא לא מתקן טעויות, הוא מחבר פחות חבילות לשכבת האפליקציה, אין לו טיימרים, הוא בכללי פשוט שולח את החבילות ומקווה לטוב** |
| תשובה |