

# רשתות מחשבים

פרק 6ד' - שכבת התעבורה, פרוטוקול TCP

ברק גונן

מבוסס על ספר הלימוד "רשתות מחשבים" מאת

עומר רוזנבוים

## מטרות הפרק

---

- ▶ בפרק זה נעמיק בפרוטוקול TCP
- ▶ נבין מהו sequence number של TCP
- ▶ נבין מהו מנגנון ה-ACK של TCP
- ▶ נבין את תהליך הקמת קשר TCP שבין שרת ולקוח
- ▶ נכתוב כלי שבודק אילו פורטים פתוחים

## חזרה קצרה

▶ למדנו ששכבת התעבורה יכולה (אופציונלית) לספק

שירות אמין

◦ שירות לא אמין - UDP

◦ שירות אמין - TCP

▶ מהו שירות אמין?

◦ כל הפקטות הגיעו

◦ הסדר לא התבלבל

◦ אין שגיאות במידע

# איך מספקים שירות אמין?

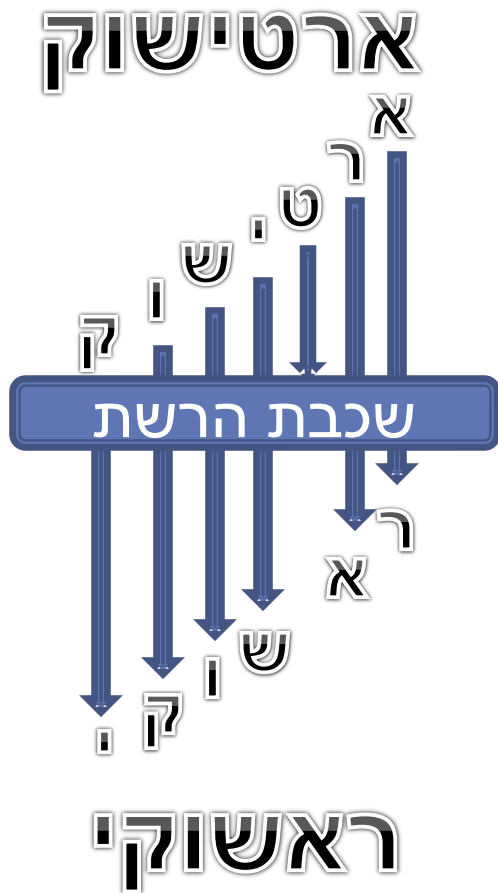
ארטישוק



ראשוקי

▶ היזכרו ב"משחק התעבורה". איך הצלחתם להעביר את המסר באופן תקין?

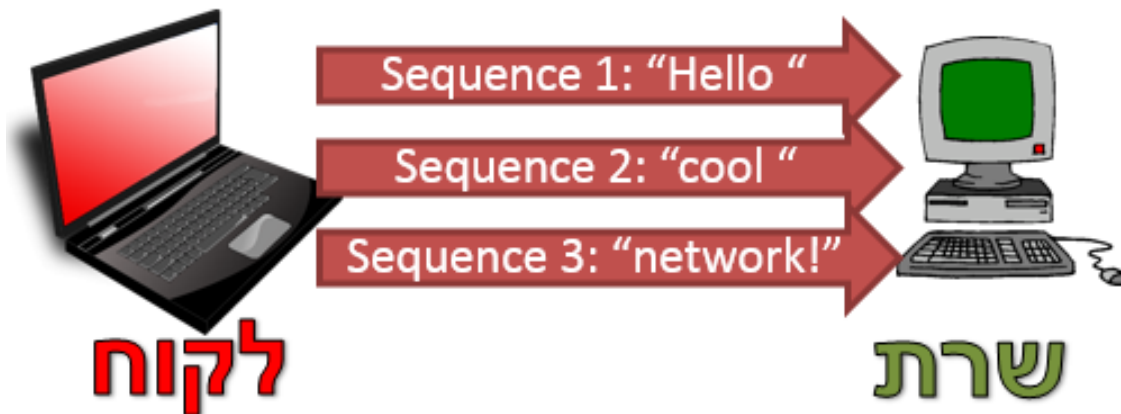
# איך מספקים שירות אמין?



- ▶ סביר שהשתמשתם בשילוב הבא:
  - שיטה, שמאפשרת לצד המקבל לבדוק אם הוא קיבל את כל המידע, ולתקן בלבול בסדר ההגעה
  - TCP: Sequence Numbers
  - שיטה, שמאפשרת לצד המקבל לבקש משלוח חוזר של מידע (אם משהו השתבש)
  - TCP: ACK

# Sequence Numbers

- ▶ נפרק את המשפט 'Hello cool network!' למקטעים, שנקראים \*Segments
- ▶ כל סגמנט ישודר עם sequence number

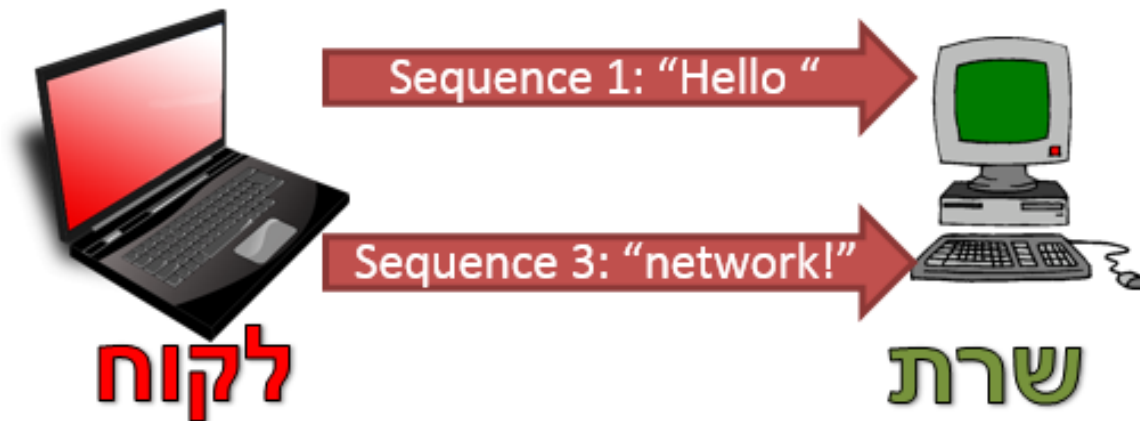


- ▶ חישוב: בשביל מה זה טוב?

\* סגמנט- גוש מידע של שכבת התעבורה. עקב מודל השכבות, כל סגמנט נעטף בפקטה של שכבת הרשת, לכן נהוג לקרוא לסגמנט שעובר ברשת "פקטה".

# Sequence Numbers - המשך

▶ כעת, אם נופל סגמנט בדרך, או אם הסדר מתבלבל, לצד השני יש דרך לדעת זאת





- ▶ ACK - קיצור של Acknowledgement, אישור
- ▶ לא מספיק שהצד המקבל יידע שמהו חסר, הצד השולח צריך לדעת זאת, ואם אין ACK - לשלוח שוב





# ACK- הרחבה

- ▶ הצגנו מנגנון פשוט: אם צד אחד לא מקבל ACK, הוא שולח שוב את המידע
- ▶ חישובו, אילו בעיות יכולות להיות במנגנון זה?
- ▶ נציג באופן כללי מספר בעיות ודרכי הפתרון

# ACK - נושאים מתקדמים (העשרה)



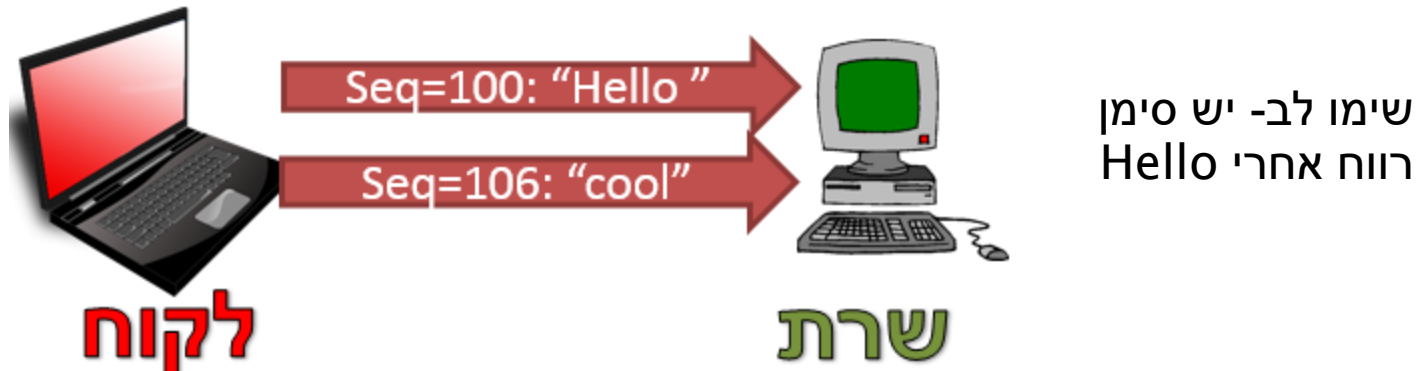
- ▶ בעיה מס' 1: איך השולח יודע מתי לשלוח חבילה חוזרת? (אולי המקבל שלח ACK, שנמצא בדרכו לשולח)
- ▶ פתרון: השולח מחכה פרק זמן מוגדר ל-ACK, ולאחר מכן שולח שוב
- ▶ חישוב: האם עדיף שפרק הזמן יהיה ארוך? או קצר?
- חסרון של פרק זמן קצר: סיכוי גבוה יותר שה-ACK לא יתקבל ואז צריך לשלוח שוב
- חסרון של פרק זמן ארוך: המתנה ארוכה בין משלוח פקטות מורידה את קצב התעבורה

# ACK - נושאים מתקדמים (העשרה)

- ▶ בעיה מס' 2: איך מעבירים קצב גבוה של מידע בערוץ?  
▶ המחשה:
  - ערוץ יכול להעביר 1 Gb מידע לשניה. למידע לוקח 25msec להגיע מהשולח ליעד. השולח משתמש בפקטות בגודל 1024 בתים. מה קצב העברת המידע?
  - השולח יצטרך לחכות לפחות 50msec עד שיגיע ACK על פקטה (זמן הגעת הפקטה + חזרת ה-ACK). לכן יישלחו 20 פקטות בשניה - 20KB בלבד על ערוץ של 1 Gb.
- ▶ פתרון: השולח לא מחכה ל-ACK, ממשיך לשלוח פקטות ואוסף ACKים בדיעבד (צפו: [Go-Back-N](#))

# TCP Sequential Numbers

- ▶ נעבור מתיאוריה כללית למימוש ב-TCP
- ▶ כל בית של מידע מקבל מספר
- ▶ שדה Seq מקבל את הערך של הבית הראשון



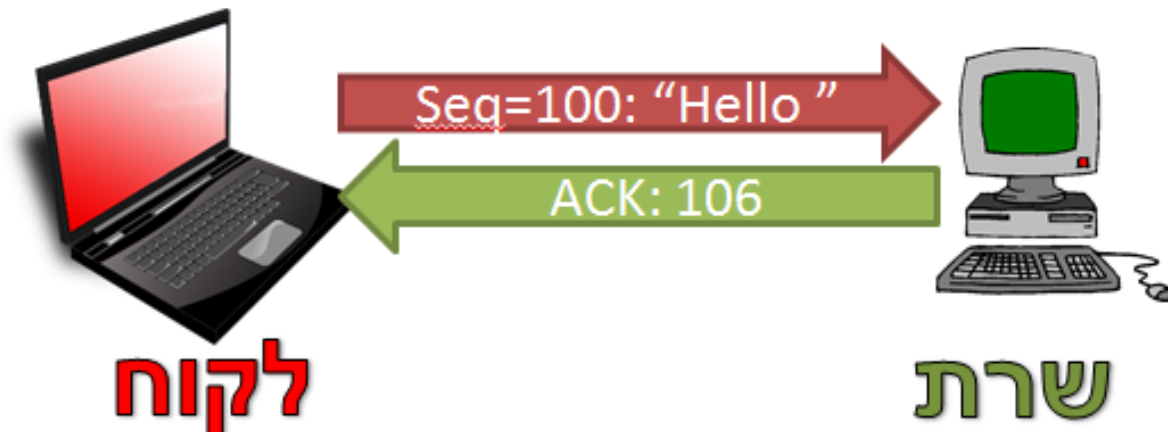
- ▶ מה יהיה ה-Seq בסגמנט הבא?
  - 110 (106+4)

# TCP Sequence Numbers

▶ בצעו את תרגיל 6.4 מודרך, צפיה ב-TCP Seq



- ▶ כמו שה-Seq מתייחס לבתים, גם ה-ACK
  - לדוגמה: ACK 106 = "קיבלתי עד בית 105, כולל. הבית הבא שאני מצפה לקבל הוא 106"

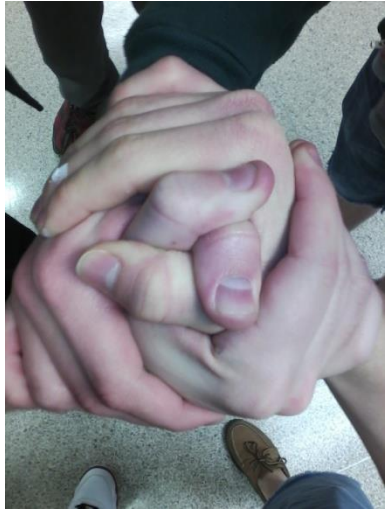


▶ בצעו את תרגיל 6.5 מודרך, צפיה ב-TCP ACK

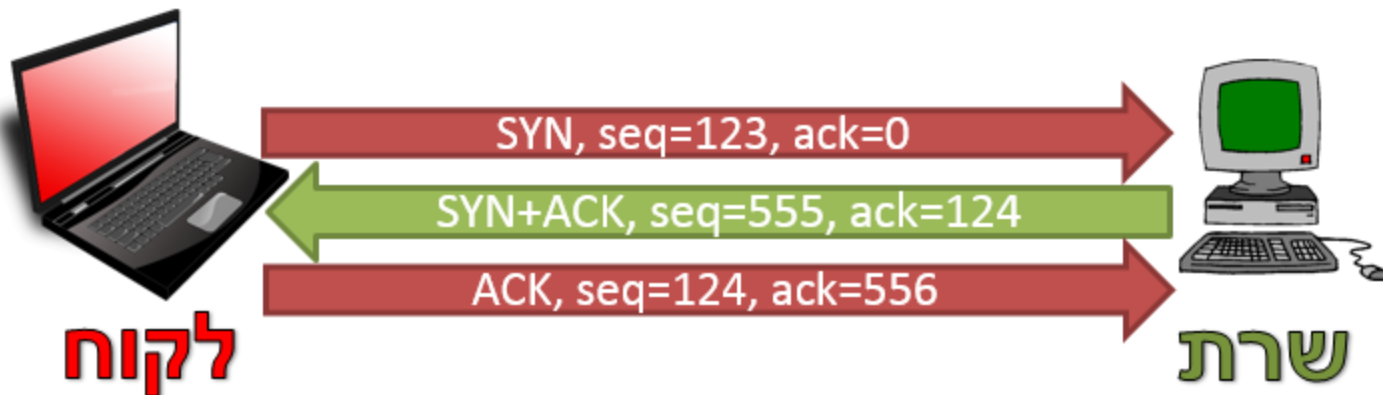




# TCP: Three Way Handshake

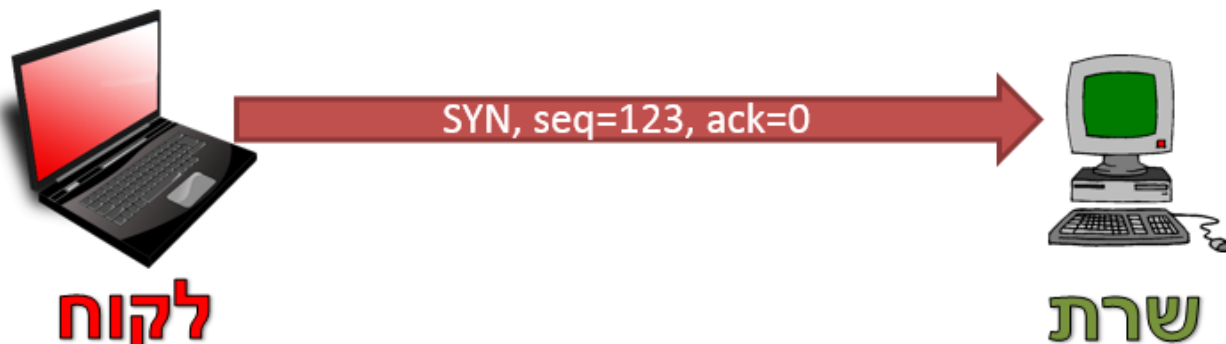


- ▶ מנגנוני ה-Sequential Number וה-ACK מחייב תהליך של הקמת קישור
  - שני הצדדים מודיעים שמוכנים לקלוט ולשדר
  - אם אחד הצדדים אינו מוכן, מנגנון ה-ACK אינו יכול לעבוד
- ▶ התהליך נקרא Three Way Handshake ומורכב מ-3 פקטות- פירוט בהמשך:



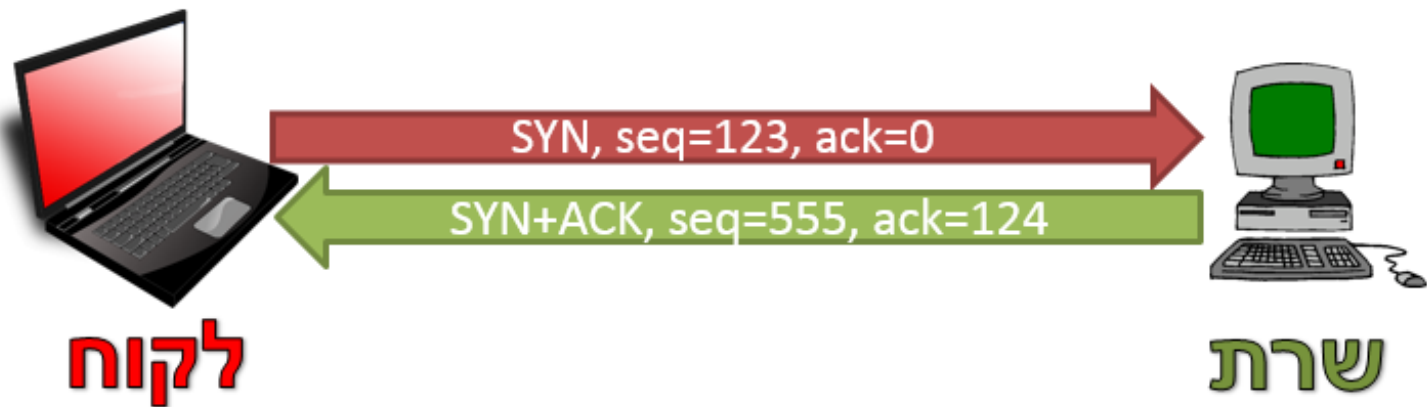
# פקטה א'- SYN

- ▶ משמעות: "אני רוצה להקים קישור"
- ▶ דגל ה-SYN דולק
- ▶ פקטת SYN לא נושאת מידע, אך סופרים אותה באורך 1
- ▶ בחירת Seq התחלתי אקראי
  - חישוב: מדוע לא להתחיל מ- Seq = 0?
  - תשובה: נניח שקישור מתנתק ונוצר חדש, חבילה מהקישור הקודם עלולה להגיע באיחור, עם Seq של הקישור הישן
- ▶ ערך ה-ACK תמיד יהיה 0



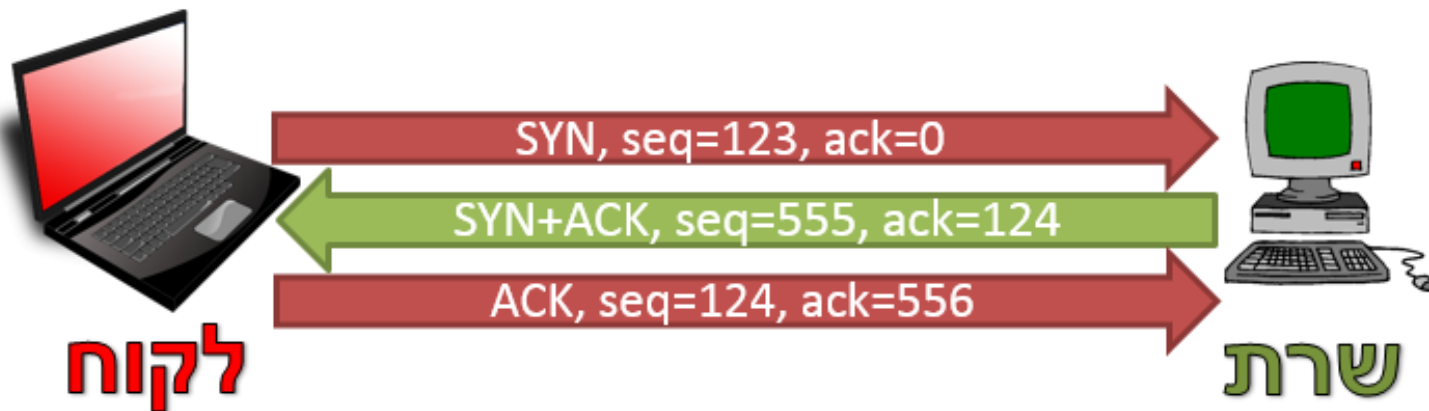
# פקטה ב'- SYN ACK

- ▶ משמעות: "אני מסכים להרים את הקישור"
- ▶ פקטה באורך בית אחד, דגלי ה-SYN וה-ACK דולקים
- ▶ בחירת Seq אקראי (לא קשור ל-Seq של פקטת ה-SYN)
- ▶ ערך ה-ACK שווה ל-Seq של פקטת ה-SYN + 1
  - היזכרו- 1 הוא אורך פקטת ה-SYN



# פקטה ג' - ACK

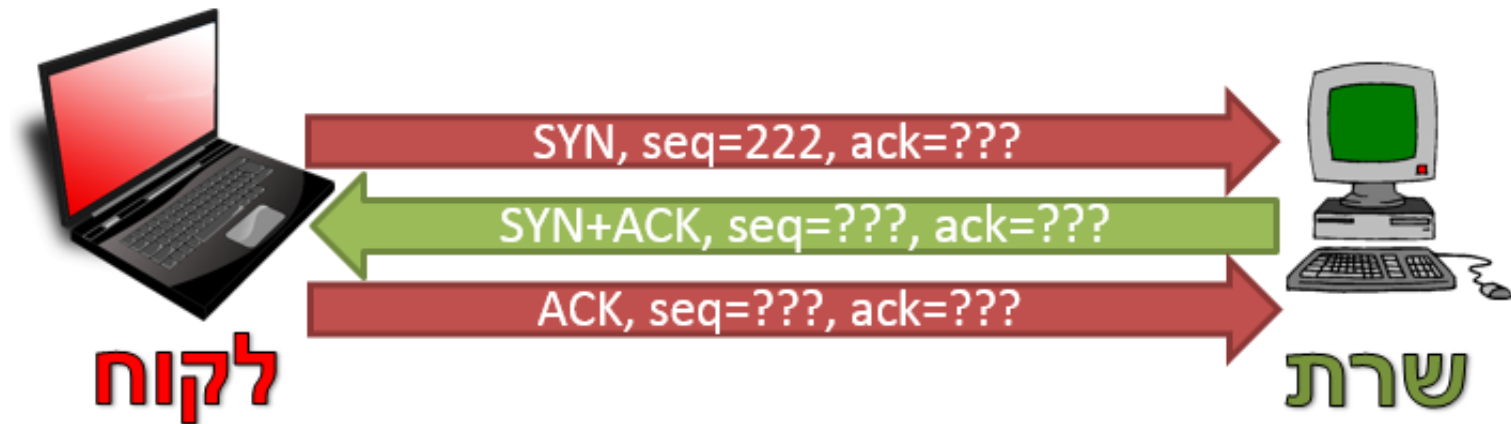
- ▶ משמעות: "קיבלתי את ה-SYN ACK ואנחנו מסונכרנים. אפשר להתחיל בתקשורת"
- ▶ דגל ה-ACK דולק (דגל ה-SYN כבוי)
- ▶ ערך ה-Seq הוא מספר הבית האחרון שנשלח
- ▶ ערך ה-ACK הוא ה-Seq של פקטת ה-SYN ACK ועוד 1



# סיכום – חישובי Seq, ACK

▶ חשבו את ערכי ה-Seq וה-ACK של תהליך הקמת הקשר  
הבא

◦ הכניסו ערך אקראי במקום הנכון



# צפיה ב- Three Way Handshake

▶ בצעו את תרגיל מודרך 6.16, השוו את הערכים למה שציפיתם לקבל



## תרגיל: יצירת Three Way Handshake

▶ השתמשו ב-scapy ליצירת Three Way Handshake ומיצאו את התהליך ב-Wireshark

▶ צד השרת:

- מאזין לפקטות ומחפש פקטות מסוג SYN
- מחזיר עליהן SYN+ACK

▶ צד הלקוח:

- שולח פקטת SYN לפורט כלשהו
- מצפה לתשובת SYN+ACK
- אם אין תשובה אחרי שניה, שולח שוב
- שולח פקטת ACK



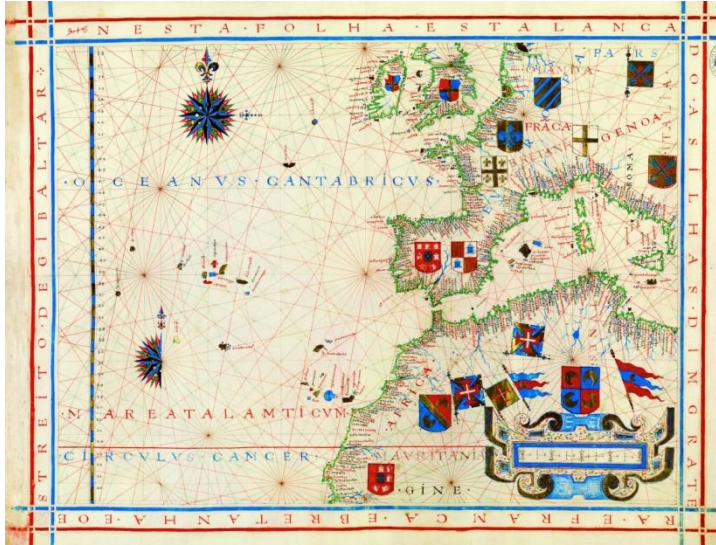
# תרגיל: יצירת Three Way Handshake

דוגמאות - wireshark בשרת ובלקוח ▶

| No. | Time       | Source   | Destination | Protocol | Length | Info                                           |
|-----|------------|----------|-------------|----------|--------|------------------------------------------------|
| 14  | 4.96560800 | 10.0.0.4 | 10.0.0.8    | TCP      | 54     | 2222->80 [SYN] Seq=0 win=8192 Len=0            |
| 17  | 5.78265300 | 10.0.0.8 | 10.0.0.4    | TCP      | 60     | 80->2222 [SYN, ACK] Seq=0 Ack=1 win=8192 Len=0 |
| 18  | 5.96839100 | 10.0.0.4 | 10.0.0.8    | TCP      | 54     | 2222->80 [ACK] Seq=1 Ack=1 win=8192 Len=0      |

| No. | Time       | Source   | Destination | Protocol | Length | Info                                           |
|-----|------------|----------|-------------|----------|--------|------------------------------------------------|
| 561 | 13.0079590 | 10.0.0.4 | 10.0.0.8    | TCP      | 60     | 2222->80 [SYN] Seq=0 win=8192 Len=0            |
| 597 | 13.8201310 | 10.0.0.8 | 10.0.0.4    | TCP      | 54     | 80->2222 [SYN, ACK] Seq=0 Ack=1 win=8192 Len=0 |
| 607 | 14.0082450 | 10.0.0.4 | 10.0.0.8    | TCP      | 60     | 2222->80 [ACK] Seq=1 Ack=1 win=8192 Len=0      |

# תרגיל מסכם TCP- מיפוי פורטים פתוחים



▶ בצעו את תרגיל 6.19

▶ שימו לב:

- בידקו אילו פורטים פתוחים רק על הרשת בכיתה, או על רשת ביתית
- אסור להריץ סריקת פורטים על שרת כלשהו ברשת:
- בחלק מהמדינות הפעולה אסורה על פי חוק
- לעיתים מדובר בהפרת תנאי השימוש מול ה-ISP שלכם

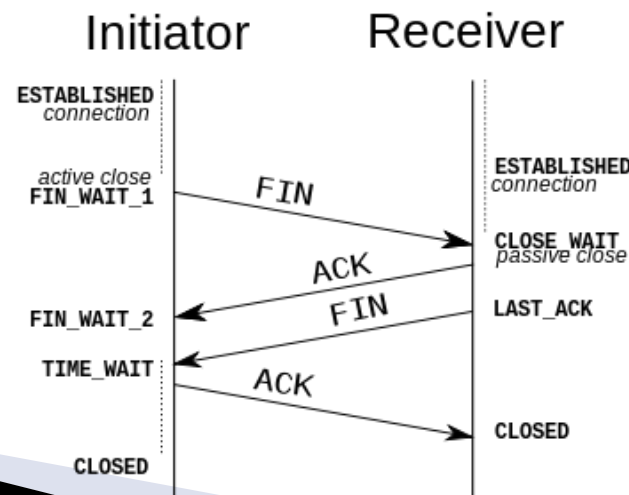
# הרחבה: סגירת קישור TCP



▶ סגירת קישור TCP מתבצעת לפי

הסדר הבא:

- 1. צד א' (שמבקש לסגור) שולח פקטה עם דגל FIN
- 2. צד ב' עונה בפקטה עם דגל ACK
- 3. צד ב' שולח פקטה עם דגל FIN
- 4. צד א' מאשר על ידי ACK



אם שלבים 3-4 לא מתבצעים הקישור נותר פתוח לשליחת מידע רק מצד אחד - Half Close



- ▶ מהם Sequence Numbers וכיצד הם נקבעים?
- ▶ למה משמשות פקטות ACK?
- ▶ תארו את שלבי ה-Three Way Handshake
- ▶ בכמה מעלה פקטת SYN את ה-Seq?
- ▶ בכמה מעלה פקטת ACK את ה-Seq?
- ▶ במהלך שיחה, צד א' שלח,  $seq=1200$ ,  $ack=580$ ,  
 $data='Jon Snow'$ . מה ערך ה-ack שהצד השני אמור להחזיר לו?