רשתות תקשורת – חזרה כללית

אריק וינשטיין

מודל 5 השכבות- הגדרות ותפקידים



- *שכבת Physical*: תפקידיה להעביר את המידע (ביטים) בין המחשבים השונים על גבי התווך.
- סיבים אופטיים, כבל Ethernet, אנטנה + גלים אל"מ ב WiFi)

מודל 5 השכבות- הגדרות ותפקידים



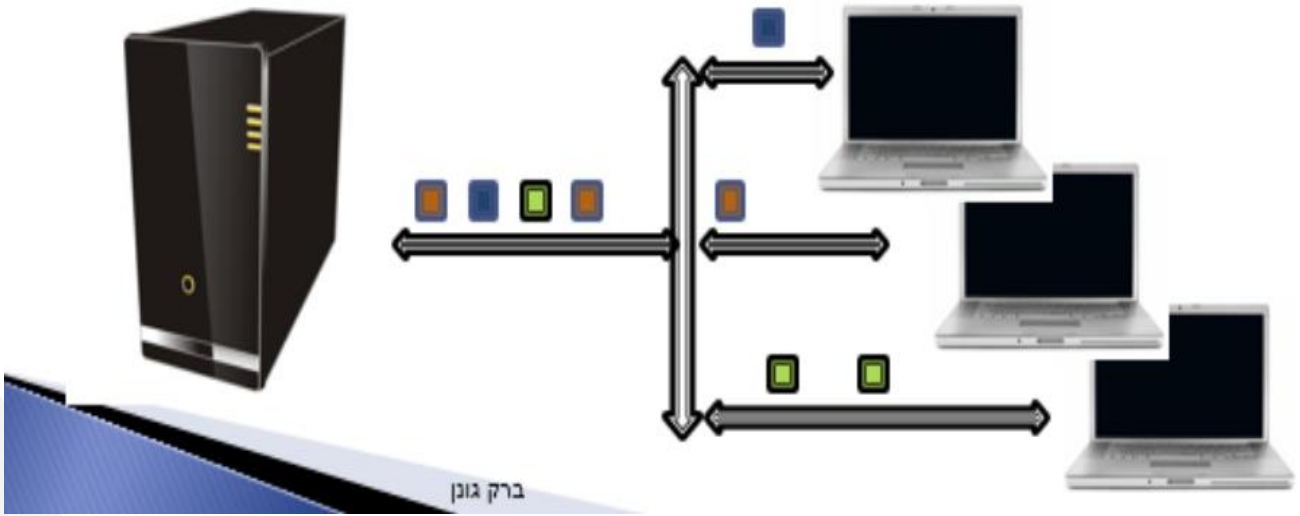
שכבת DataLink:

- העברת גוש מידע (Frames להבדיל מביטים בודדים)
- ריבוב (interleave) של מסגרות שונות על אותו ערוץ פיזי (חלוקה בזמן של הערוץ בין המחשבים)

מודל 5 השכבות- הגדרות ותפקידים



שכבת DataLink:



מודל 5 השכבות- הגדרות ותפקידים



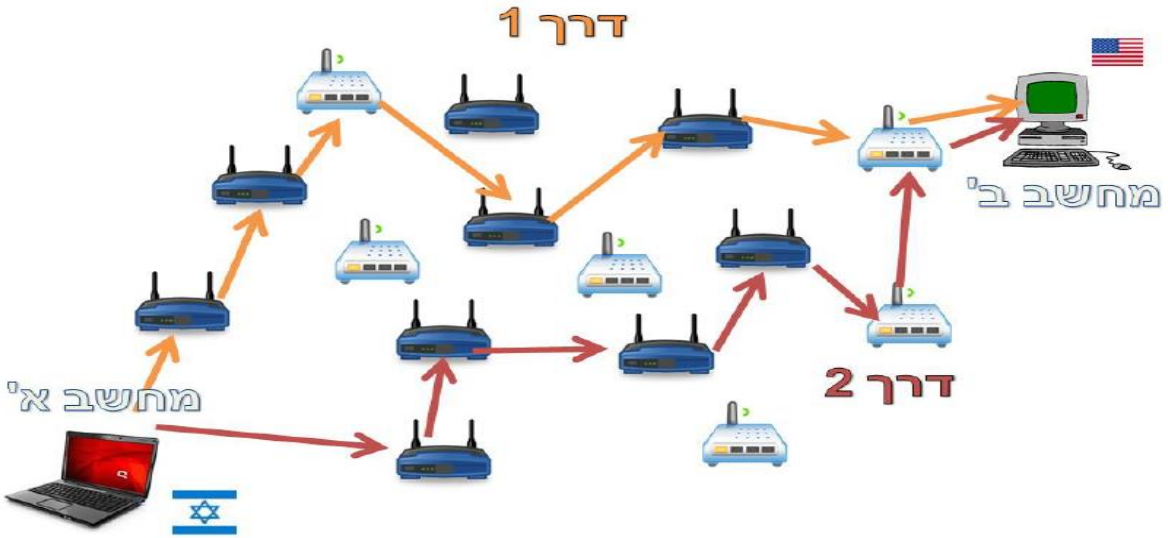
שכבת Network:

- מציאת המסלול ברשת בין מחשב המקור למחשב היעד (שימוש בנתבים)
- ניתוב של חבילות (packets)
- Routing

מודל 5 השכבות- הגדרות ותפקידים

- Application
- Transport
- **Network**
- Data Link
- Physical

שכבת *Network*:



מודל 5 השכבות- הגדרות ותפקידים



שכבת התעבורה Transportation:

- מאפשרת תקשורת למספר יישומים במחשב (port)
- מאפשרת טיפול בשגיאות (שידור חוזר) ובסידור של המידע על פי הסדר

מודל 5 השכבות- הגדרות ותפקידים



שכבת Application:

- מאפשרת ליישומים להתחבר לשרתים\תוכנות מרוחקים
- לכל אפליקציה יש את הפרוטוקול שלה(נזכור את ה- header של http)

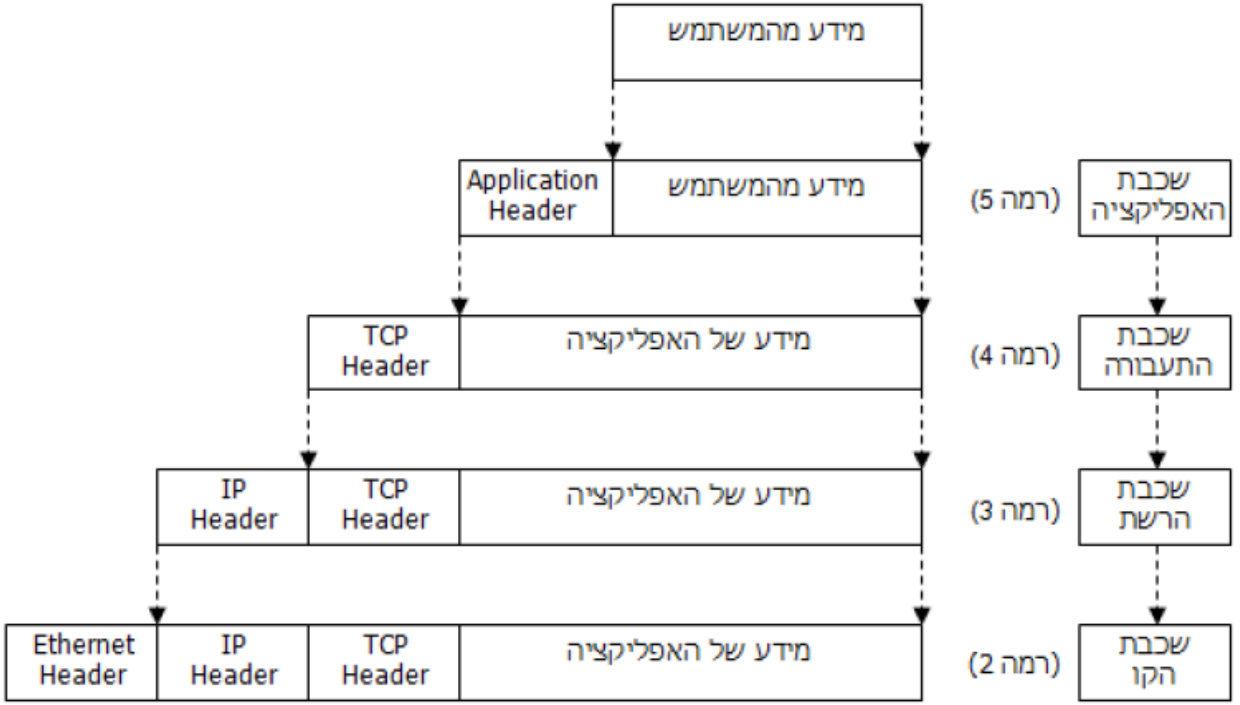
מודל 5 השכבות- הגדרות ותפקידים



שכבת Application:

- מאפשרת ליישומים להתחבר לשרתים\תוכנות מרוחקים
- לכל אפליקציה יש את הפרוטוקול שלה(נזכור את ה- header של http)
- החיבור של שכבה 5 לשכבות התחתונות נעשה ע"י שימוש ב-socket .

מודל 5 השכבות- כימוס (encapsulation)



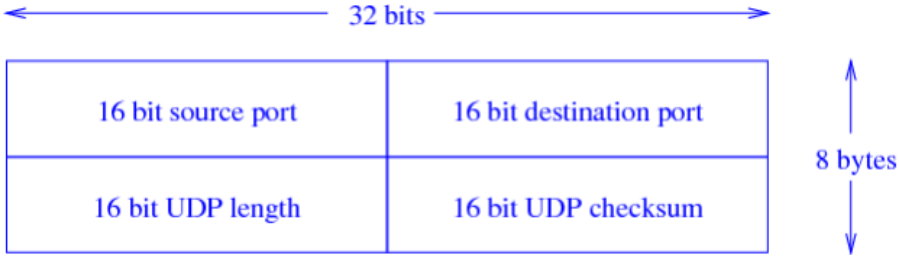
שכבה 4: UDP מול TCP



2 הפרוטוקולים בשכבת התעבורה:

- UDP פרוטוקול שמאפשר שליחה זריזה, header קטן יחסית. מאפשר זיהוי שגיאות אך ללא אפשרות לשידור חוזר ותיקונן.
- TCP יותר כבד, דורש הקמת קישור (3-way handshake) , מאפשר זיהוי שגיאות ובקשה לשידור חוזר של המידע השגוי. כולל מנגנון של flow-control ו-

שכבה 4: UDP מול TCP



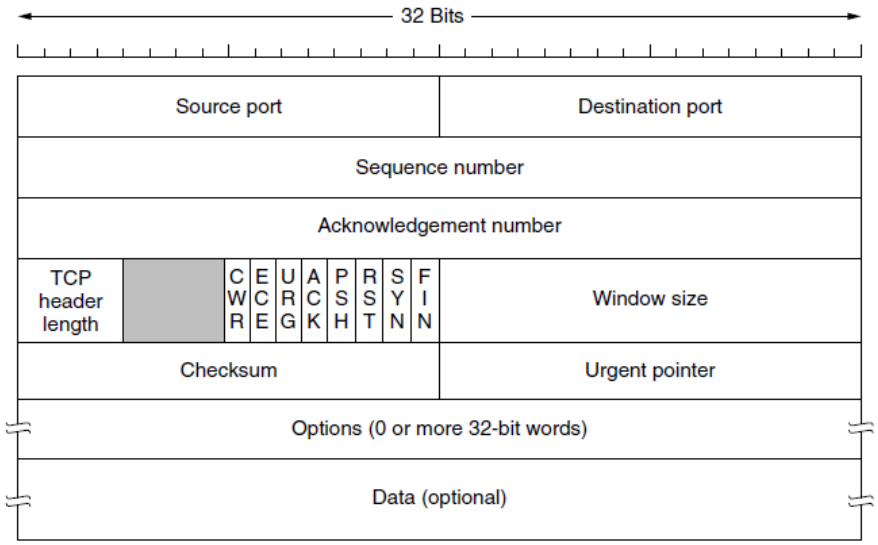


Figure 6-36. The TCP header.

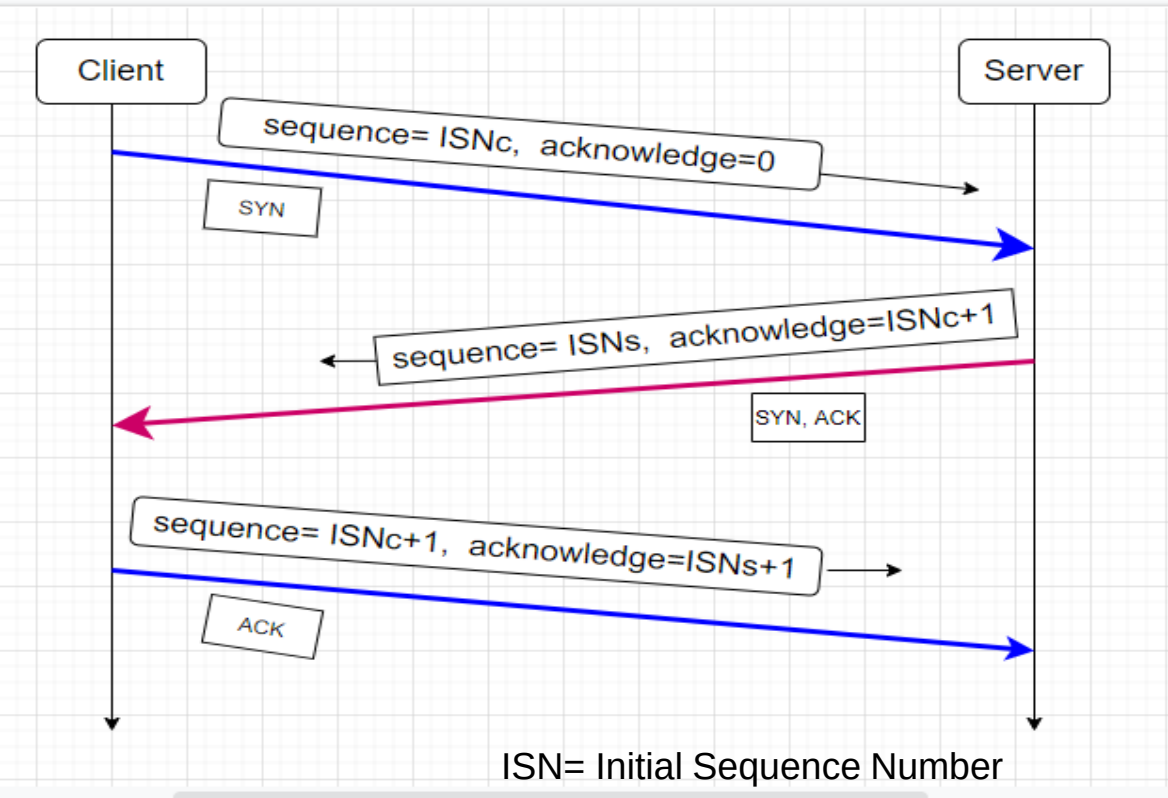
שכבה 4: UDP מול TCP



2 הפרוטוקולים בשכבת התעבורה:

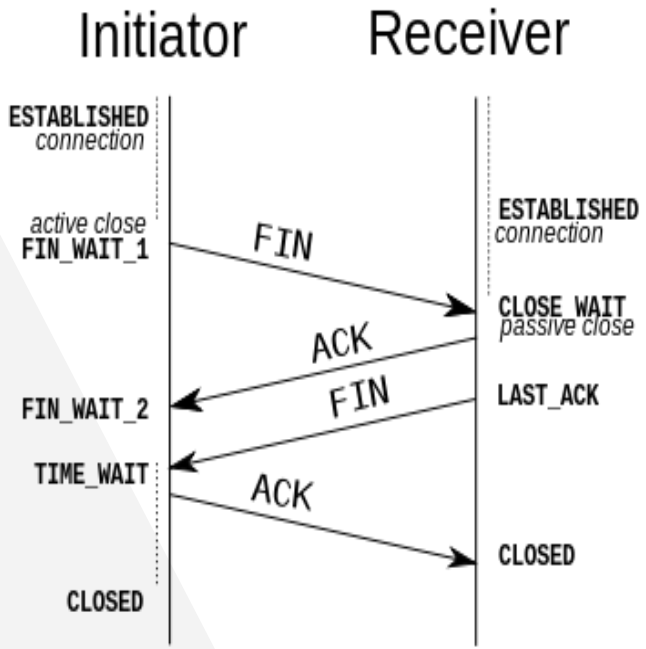
- UDP פרוטוקול שמאפשר שליחה זריזה, header קטן יחסית. מאפשר זיהוי שגיאות אך ללא אפשרות לשידור חוזר ותיקונן.
- TCP יותר כבד, דורש הקמת קישור (3-way handshake) , מאפשר זיהוי שגיאות ובקשה לשידור חוזר של המידע השגוי. מאפשר stream

TCP 3-Way Handshake



ISN= Initial Sequence Number

TCP 3-Way Handshake



כמו שמרימים קישור כך ניתן להוריד את הקישור (ראה בספר)

שכבה 4: UDP מול TCP



מנגנונים מתקדמים ב TCP :

- *Flow-Control*: מאפשר בקרה של הצד הקולט בקצב הנתונים של "המשדר", ע"י שדה Window
- *Fast Retransmit* : כאשר הצד המשדר מקבל 3 ACKs עם אותו ערך בשדה acknowledge , הוא מניח שהחבילה הבאה הלכה לאיבוד וישדר אותה שוב .

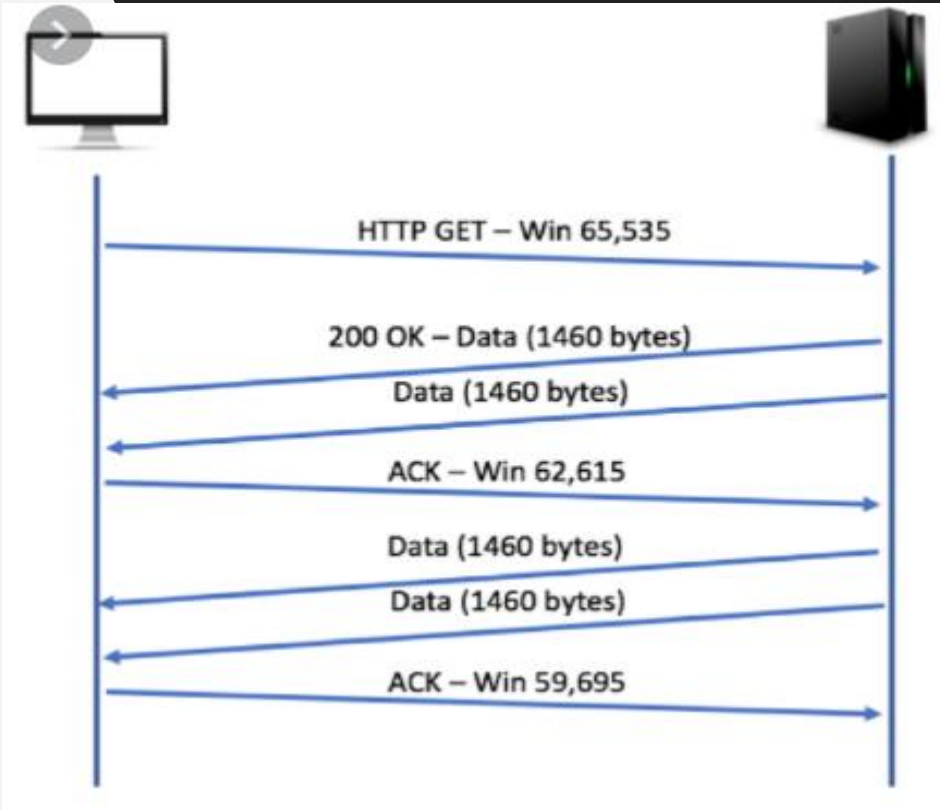
שכבה 4: UDP מול TCP



מנגנונים מתקדמים ב TCP :

- *Flow-Control*: מאפשר בקרה של הצד הקולט בקצב הנתונים של "המשדר", ע"י שדה Size Window
- *Fast Retransmit* : כאשר הצד המשדר מקבל 3 ACKs עם אותו ערך בשדה acknowledge , הוא מניח שהחבילה הבאה הלכה לאיבוד וישדר אותה שוב .

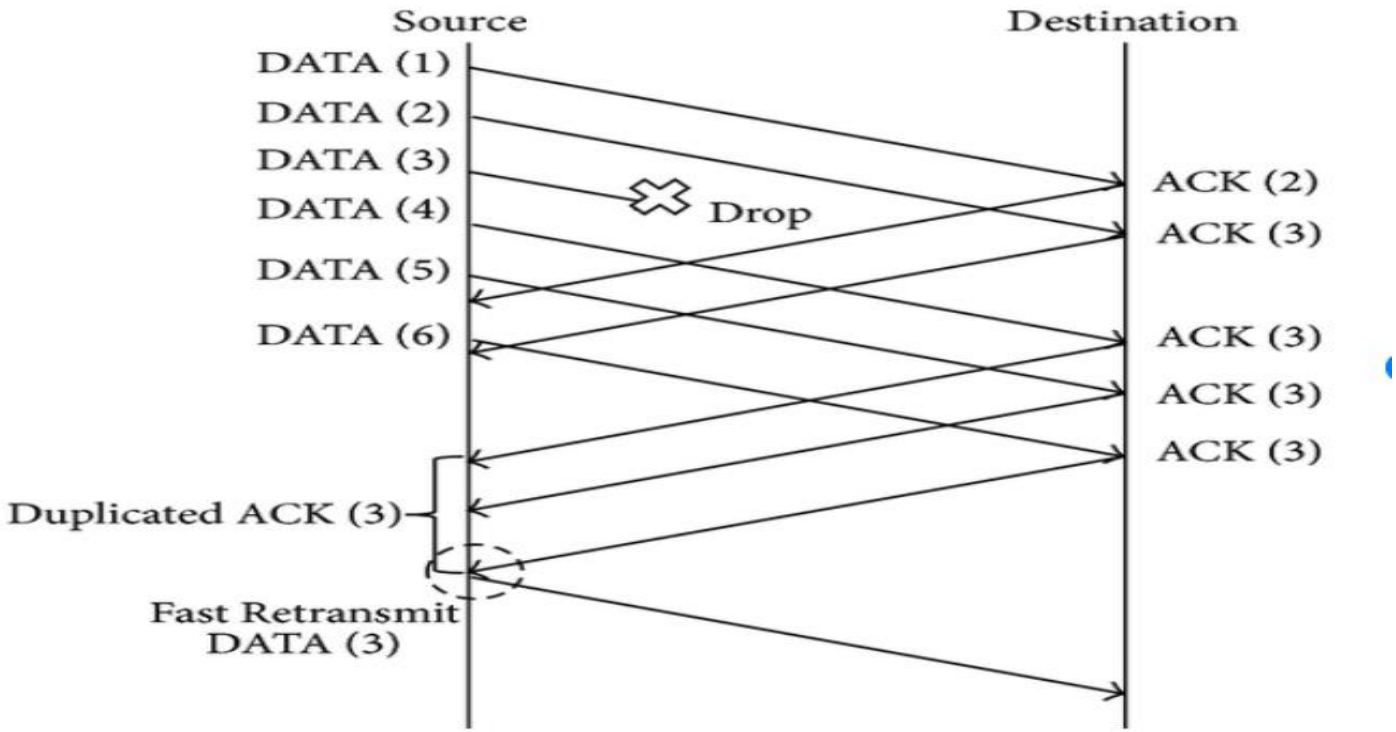
שכבה 4: TCP Flow Control



- מאפשר בקרה של הצד הקולט בקצב הנתונים של "המשדר", ע"י שדה **Window size**

TCP Fast Retransmit :4 שכבה

- Application
- Transport
- Network
- Data Link
- Physical



שכבה 3: IP

פרוטוקול IP

- פקודת Tracert, ipconfig
- מבנה כתובת IPv4 :
  - מזהה רשת (network id)
  - מזהה מחשב\ישות (host id)

- מחשבים בעלי network id זהים שייכים לאותה רשת

שכבה 3: IP

פרוטוקול IP

- דוגמה 1: 10.38.83.15
- Subnet mask: 255.255.255.0 ==> 10.38.83.15
- דוגמה 2: 192.168.85.217
- Subnet mask: 255.255.0.0 => 192.168.85.217
-

שכבה 3: IP

IP Classes

Class	Class A 0. 0. 0. 0 = 00000000.00000000.00000000.00000000 127.255.255.255 = 01111111.11111111.11111111.11111111 0nnnnnnn.HHHHHHHH.HHHHHHHH.HHHHHHHH Class B 128. 0. 0. 0 = 10000000.00000000.00000000.00000000 191.255.255.255 = 10111111.11111111.11111111.11111111 10nnnnnn.nnnnnnnn.HHHHHHHH.HHHHHHHH Class C 192. 0. 0. 0 = 11000000.00000000.00000000.00000000 223.255.255.255 = 11011111.11111111.11111111.11111111 110nnnnn.nnnnnnnn.nnnnnnnn.HHHHHHHH	ess	End address
Class A		.0	127.255.255.255
Class B		0.0.0	191.255.255.255
Class C		0.0.0	223.255.255.255
Class D (multicast)		0.0.0	239.255.255.255
Class E (reserved)	defined defined defined defined (2^{28})	0.0.0	255.255.255.255

שכבה 3: IP

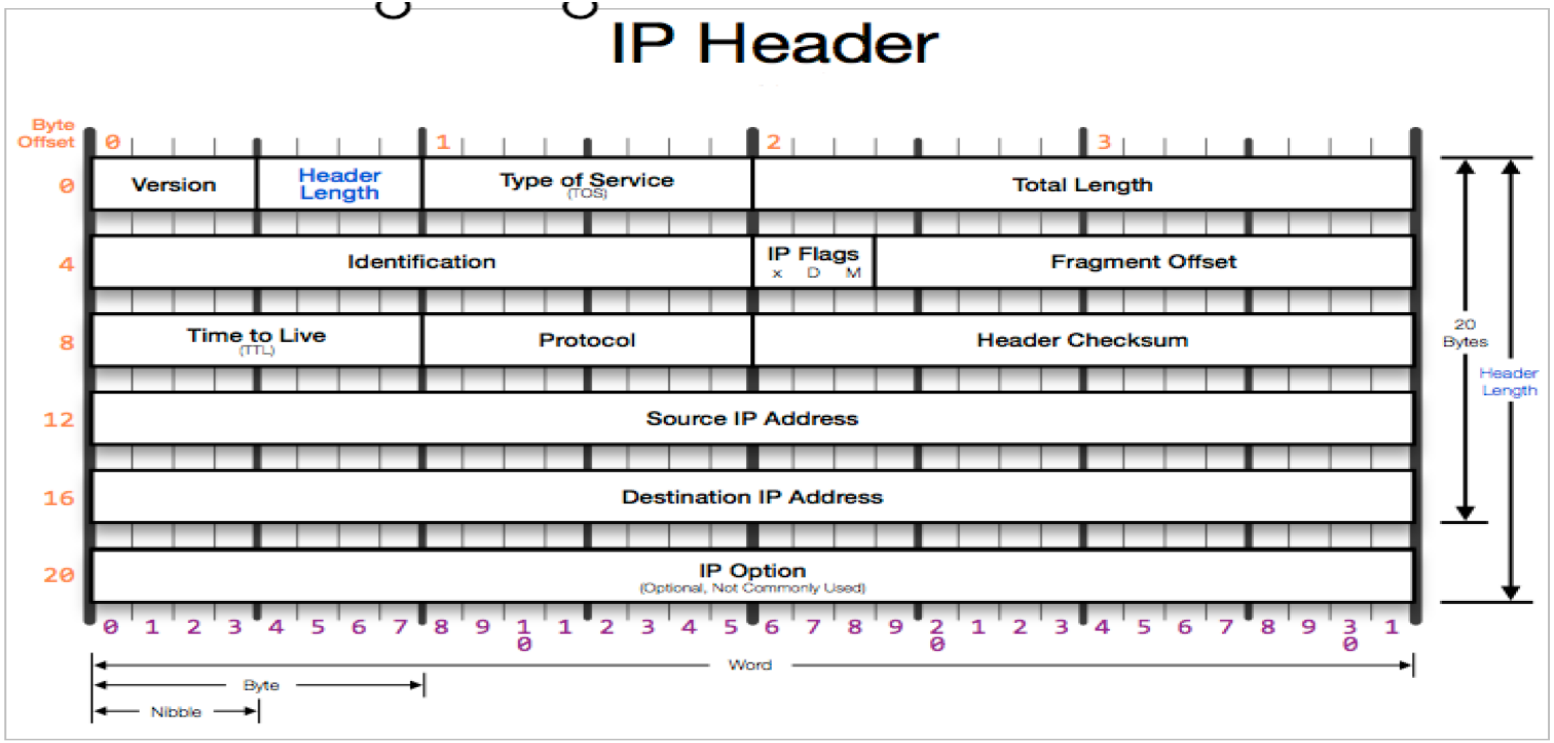
כתובות IP מיוחדות :

כתובות פרטיות (NAT):

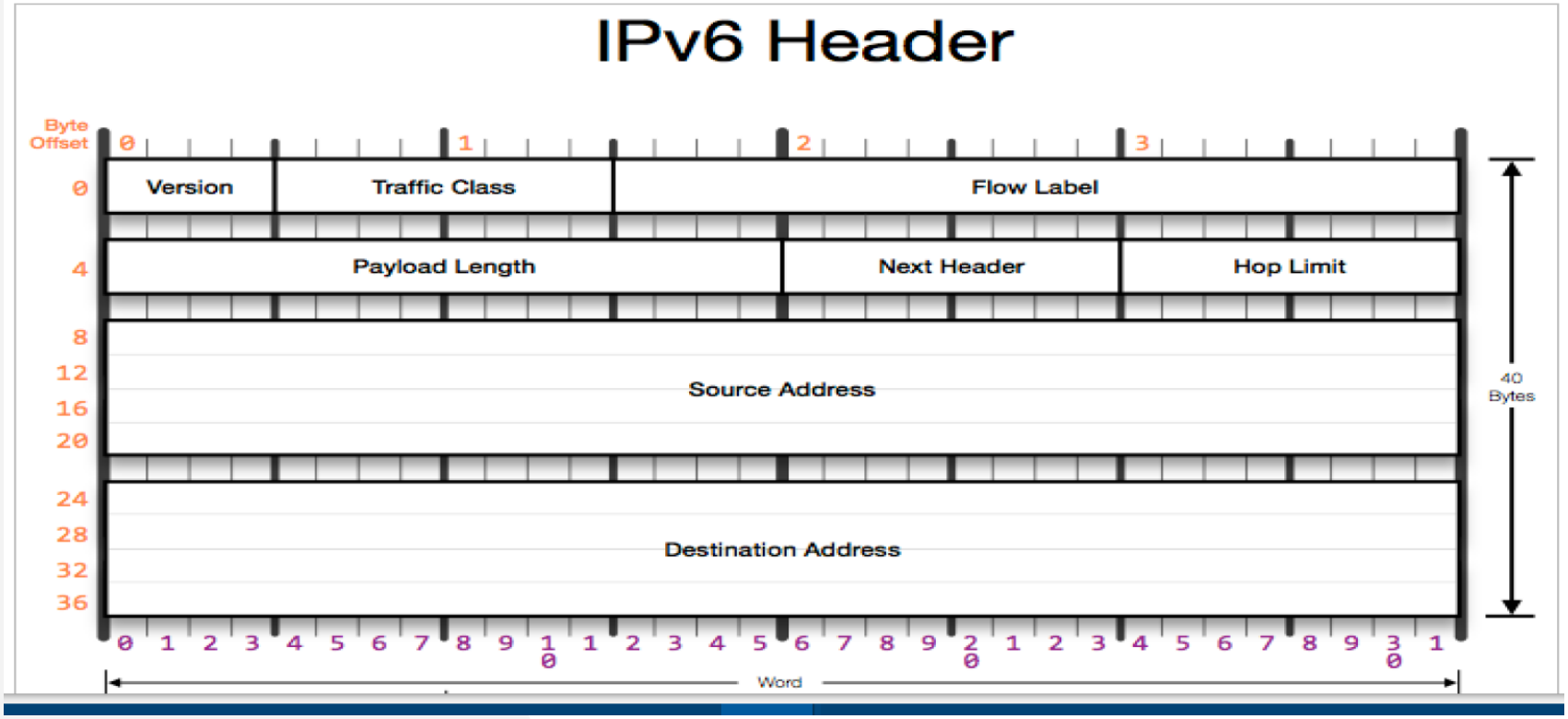
10.0.0.1 - 10.255.255.255
172.16.0.0 - 172.31.255.255
192.168.0.0 - 192.168.255.255

127.0.0.0	Loopback address (the host's own interface)
224.0.0.0	IP Multicast
255.255.255.255	Broadcast (sent to all interfaces on network)

שכבה 3: IP


IP Header
Byte Offset
0
4
8
12
16
20
0 1 2 3
Version
Header Length
Type of Service (TOS)
Total Length
Identification
IP Flags x D M
Fragment Offset
Time to Live (TTL)
Protocol
Header Checksum
Source IP Address
Destination IP Address
IP Option (Optional, Not Commonly Used)
20 Bytes
Header Length
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Word
Byte
Nibble

שכבה 3: IPv6


IPv6 Header
Byte Offset
0
1
2
3
0
Version
Traffic Class
Flow Label
4
Payload Length
Next Header
Hop Limit
8
12
16
20
Source Address
24
28
32
36
Destination Address
40 Bytes
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Word

DHCP

Application
Transport
Network
Data Link
Physical

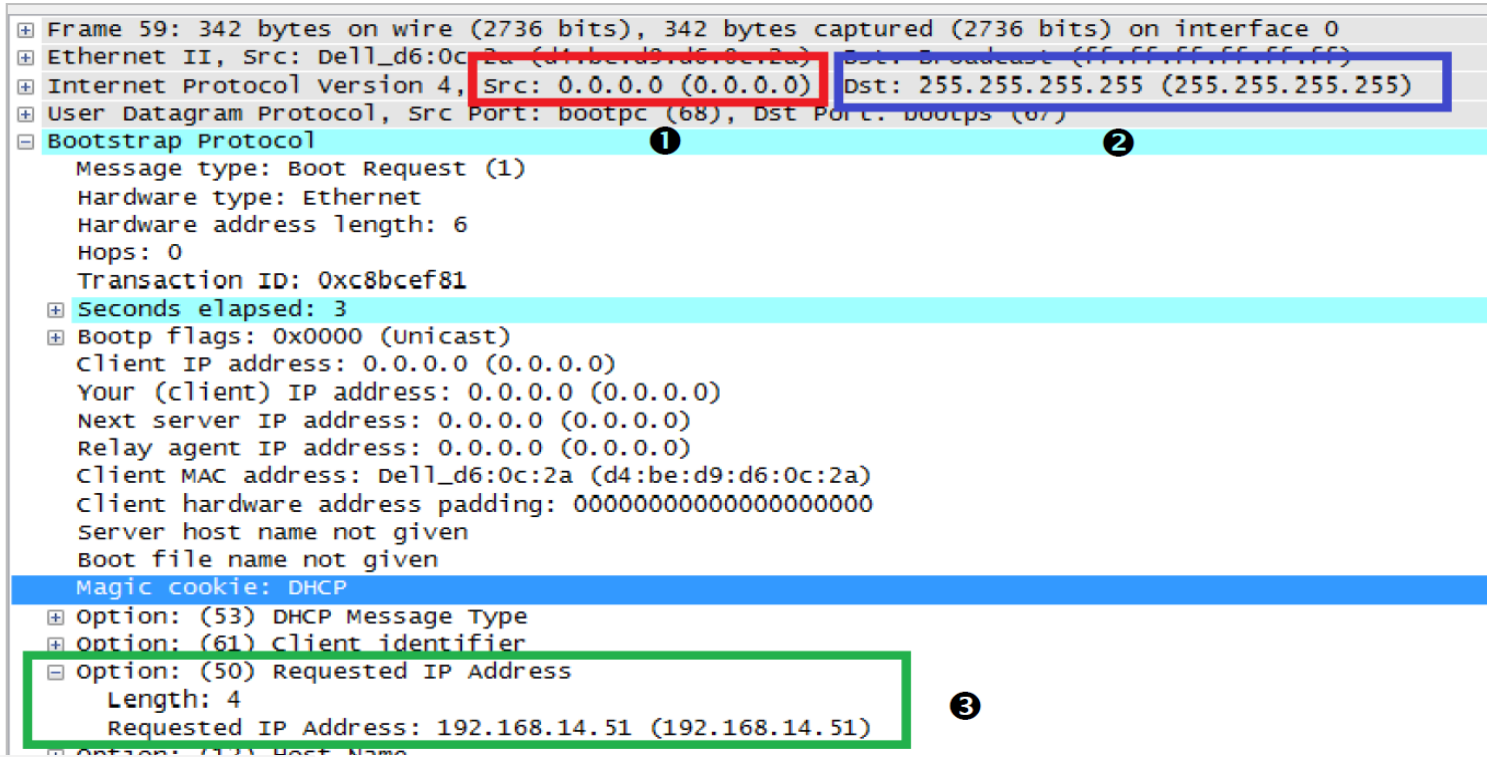
הקצאת דינאמית של כתובת IP למחשב:

- הפרוטוקול מכיל 4 שלבים (הודעות) :



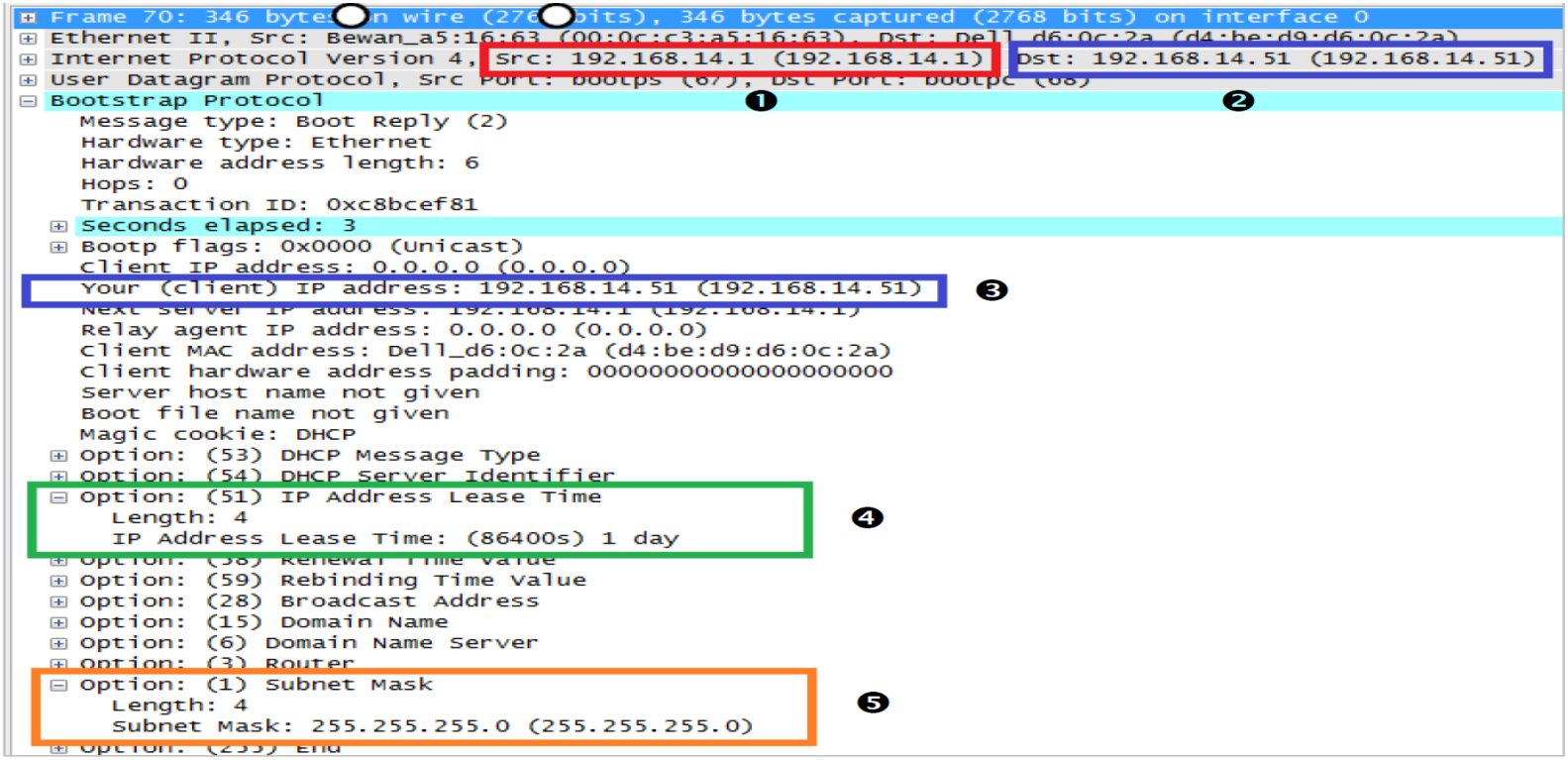
DHCP Discover

- Application
- Transport
- Network
- Data Link
- Physical



DHCP Offer

- Application
- Transport
- Network
- Data Link
- Physical



שכבה 2: Ethernet



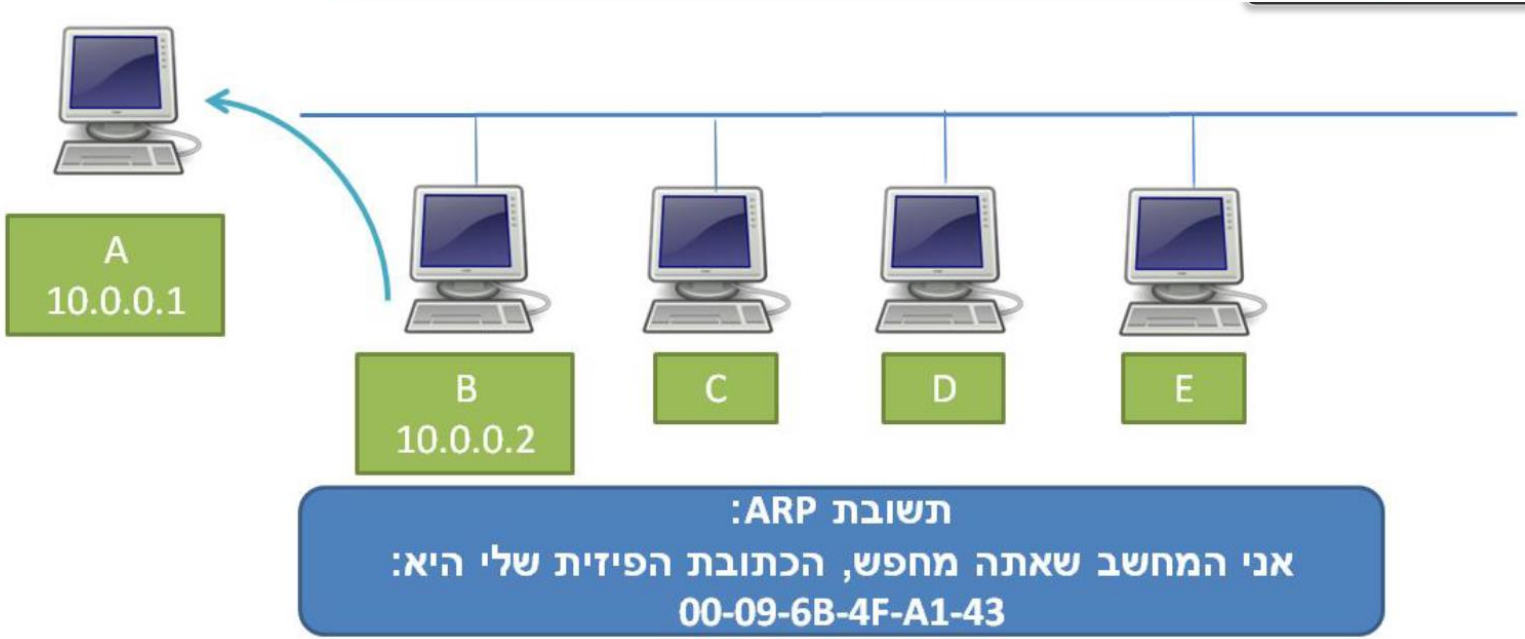
- שכבת הקו – אחראית להעביר את המידע למחשב הבא במבלול
- עושה שימוש בכתובת MAC בפורמט הקסה-דצימלי: AA:BB:CC:DD:EE:FF
- לדוגמא D4-BE-D9-D6-0C-2A
- לצורך ההעברת המידע יש להמיר בין כתובות IP לכתובות MAC

שכבה 2: פרוטוקול ARP



- כדי להעביר פקטה למחשב עם כתובת IP מסוימת , יש לבדוק מה הכתובת הפיסית שלו – MAC כדי לשלוח לכרטיס רשת שלו.
- את זה עושה ARP (Address Resolution Protocol)

שכבה 2: פרוטוקול ARP



מערכות הפעלה- תהליכים

תהליכים (Processes)

- רוב התוכנות שאנחנו מכירים הן בעצם תהליך (Process)
- "תהליך במדעי המחשב, הוא מופע של תוכנית מחשב שמופעל על ידי מערכת מחשב שיש לה היכולת להפעיל מספר תהליכים בו זמנית. *תוכנית מחשב היא בעצמה רק אוסף פקודות, בעוד שתהליך הוא ההפעלה של אותן פקודות.*" (מתוך ויקיפדיה)
- תהליך זה המעטפת של מערכת ההפעלה לכל מה שקשור בריצה של תוכנה.
  - אבל מה בעצם המערכת הפעלה צריכה לעטוף?

מערכות הפעלה- תהליכים-תהליכונים

- Thread היא "יחידת ריצה" עצמאית עבור מערכת ההפעלה
- מה זה אומר ש-Thread צריך?
  - מחסנית
  - Context
  - כל מיני מונים
- אז מה ההבדל בין Process ל-Thread?
  - Process אחראי על משאבים, Thread אחראי על ריצה
  - ב-Process אחד יכולים להיות כמה Thread-ים, אבל חייב להיות לפחות אחד
  - כל ה-Thread-ים באותו ה-Process משתפים משאבים (זיכרון, קבצים פתוחים וכו')

מערכות הפעלה- תקשורת בין תהליכים

- במ"ה יש הרבה Process-ים ו-Thread-ים
- אבל כל אחד מהם עושה משהו פשוט ויוצא. איך מדברים ביניהם- מעבירים מידע בינהם?
  - Socket
  - Shared Memory
  - Pipe
  - קבצים
  - סינכרון: Mutex, Semaphore, Event
- תקשורת ותזמונים – **Inter Process Communication**

מערכות הפעלה- סנכרון בין תהליכים IPC

- במ"ה יש הרבה Process-ים ו-Thread-ים
- אבל כל אחד מהם עושה משהו פשוט ויוצא. איך מדברים ביניהם- מעבירים מידע בינהם?
  - Socket
  - Shared Memory
  - Pipe
  - סינכרון: Mutex, Semaphore, Event
- תקשורת ותזמונים – **Inter Process Communication**

מערכות הפעלה- סנכרון בין תהליכים IPC

- נזכר בדוגמא של הבנק :

```
balance = get_balance(account);
balance -= amount;  // 50K-25K
```

```
balance = get_balance(account);
balance -= amount;  // 50K-25K
put_balance(account, balance);
return balance;     // = 25K
```

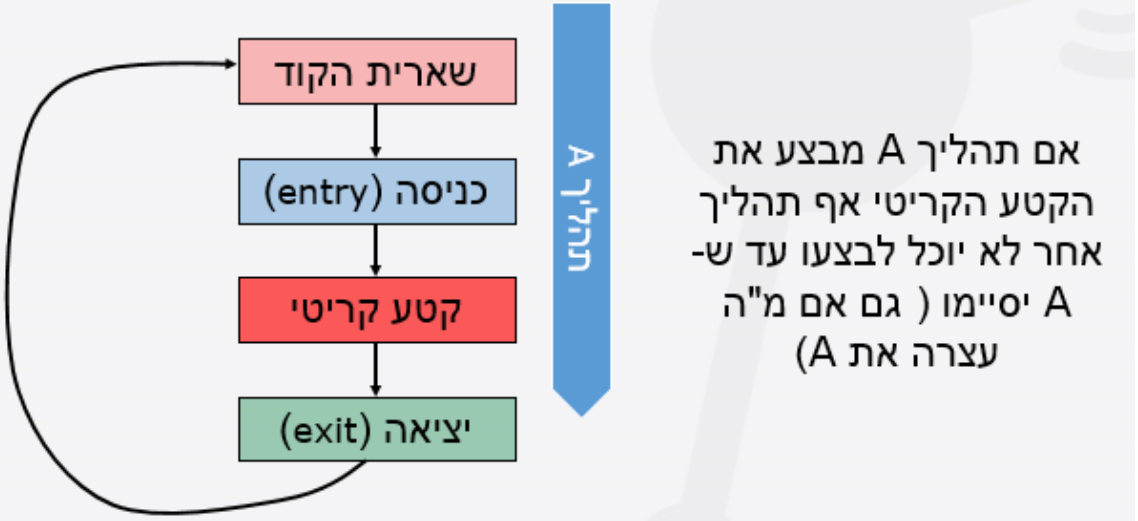
```
put_balance(account, balance);
return balance;      // = 25K
```

מי שמח עכשיו?



מערכות הפעלה- סנכרון בין תהליכים IPC

- ראינו כי יש להגן על קטעי קוד קריטיים (יתרה בחשבון) ואת זה עושים בעזרת Semaphore (Mutex)



מערכות הפעלה- סנכרון בין תהליכים IPC

- ראינו כי יש להגן על קטעי קוד קריטיים (יתרה בחשבון) ואת זה עושים בעזרת Semaphore (Mutex)

- על הסמפור מוגדרות שתי פעולות אטומיות:
  - אם ערך הסמפור גדול מ-0 הקטן אותו ב-1 והמשך, אחרת המתן עד שיהיה גדול מ-0, ואז הקטן אותו ב-1 והמשך.
  - הגדל את ערך הסמפור ב-1.

```
WAIT ( S ):
    while   ( S <= 0 );
    S = S - 1;
```

```
SIGNAL ( S ):
    S = S + 1;
```

37

מערכות הפעלה- סנכרון בין תהליכים IPC

- ראינו כי יש להגן על קטעי קוד קריטיים (יתרה בחשבון) ואת זה עושים בעזרת Semaphore (Mutex)
- Counting Semaphore:

- על הסמפור מוגדרות שתי פעולות אטומיות:
- אם ערך הסמפור גדול מ-0 הקטן אותו ב-1 והמשך, אחרת המתן עד שיהיה גדול מ-0, ואז הקטן אותו ב-1 והמשך.
- הגדל את ערך הסמפור ב-1.

```
WAIT ( S ):
    while   ( S <= 0 );
    S = S - 1;
```

```
SIGNAL ( S ):
    S = S + 1;
```

37

מערכות הפעלה- סנכרון בין תהליכים IPC

- שימוש לסינכרון תהליכים – Binary Semaphore
- התהליך הראשון מבצע שחרור של מנעול בסוף התהליך ותהליך הבא מבצע המתנה בתחילתו.

```
P1:
    Code Section 1;
    signal(synch)
```

```
P2:
    wait(sync)
    Code Section;
```

- לאיזה ערך צריך לאתחל את S? 0

תודה

אריק וינשטיין