
אלגוריתם עץ פורש, פרוטוקול האינטרנט IP

9 יוני 2026
הרצאה 8

Some Slides Credits: Steve Zdancewic (UPenn)

נושאים להיום

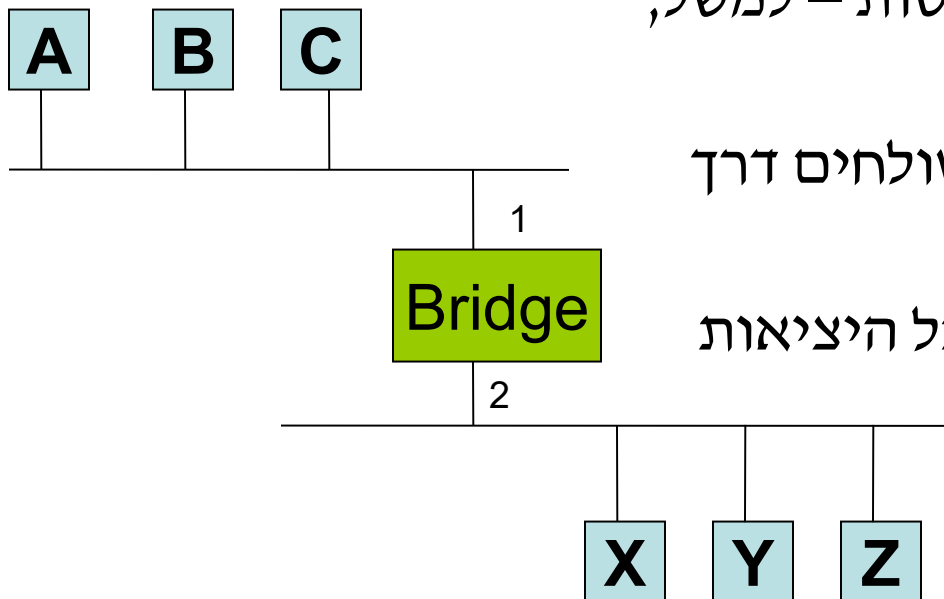
- עץ פורש לגשרים
- פרוטוקול האינטרנט (IP)
 - עקרונות
 - כתובות

גשרים ומתגי LAN

- גשר (Bridge) מקבל מסגרות LAN ביציאה אחת ומוציא אותן ביציאה אחרת.
- אופטימיזציה: מעביר רק מסגרות מתאימות

גשר לומד

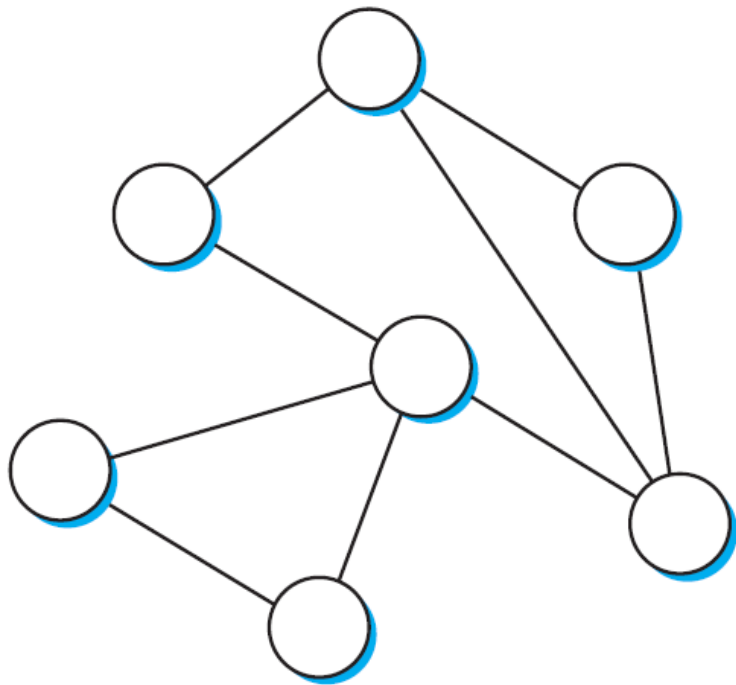
- רושם כתובת "שולח" ממסגרות נכנסות – למשל, ממקור S מיציאה מספר P
- מוסיף רשומה שלהגיע לכתובת S, שולחים דרך P
- אם אין רשומה בטבלה, משדרים לכל היציאות



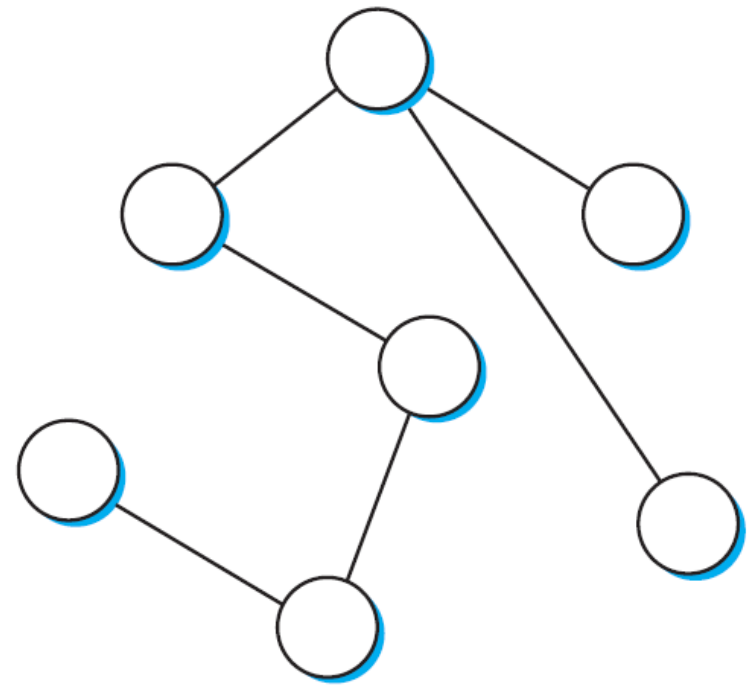
בעיה : לולאות

- מסגרת עלולה להיות משודרת מחדש לנצח
- ניתן למנוע על ידי בנייה מקפידה, אבל:
- קשה, במיוחד ניהול
- לעתים קרובות רוצים חיבורים נוספים כגיבוי
- פתרון:
- הגבלת יציאות לאלה ששייכות לעץ פורש
- אלגוריתם בסיסי : רדיה פרלמן מחברת Digital
- תקן 802.1 לגשרי LAN מבוסס על האלגוריתם
- 2004 : שיפורים כדי להאיץ את קצב ההתכנסות

מהו עץ פורש?



(a)



(b)

Algorhyme

I think that I shall never see
a graph more lovely than a tree.
A tree whose crucial property
is loop-free connectivity.
A tree that must be sure to span
so packets can reach every LAN.
First, the root must be selected.
By ID, it is elected.
Least-cost paths from root are
traced.
In the tree, these paths are placed.
A mesh is made by folks like me,
then bridges find a spanning tree.

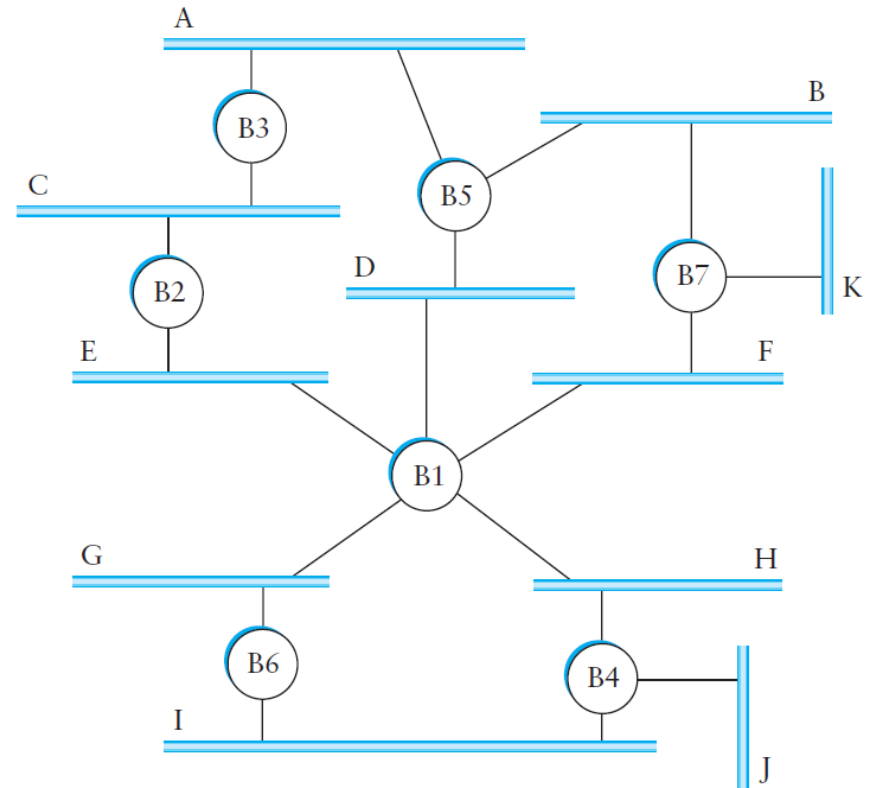
Radia Perlman



מושגי עץ פורש

מושגי יסוד :

- נבחר **גשר שורש אחד**
 - לכל תת-רשת חייב להיות נתיב אחד להגיע לגשר השורש
- כל גשר עשוי להיות מחובר **לתת-רשתות מרובות** (ולקבל מהן מנות)
 - רק **הגשר האחראי** יעביר מנות לכיוון השורש
- כל גשר יודע איזה מהיציאות **הינה הקרובה ביותר** לשורש
 - נקרא **יציאת השורש**



אלגוריתם עץ פורש

פרסומת ($ROOT, dist, SENDER$)

- $ROOT$ מספר זהות של השורש
- $dist$ כמות הצעדים אל השורש $ROOT$
- $SENDER$ מזהה של השולח

כל גשר מתחיל עם ההנחה שהוא השורש ומתחיל לפרסם את עצמו

אם גשר מקבל פרסומת יותר טובה, הוא מפסיק לפרסם את עצמו

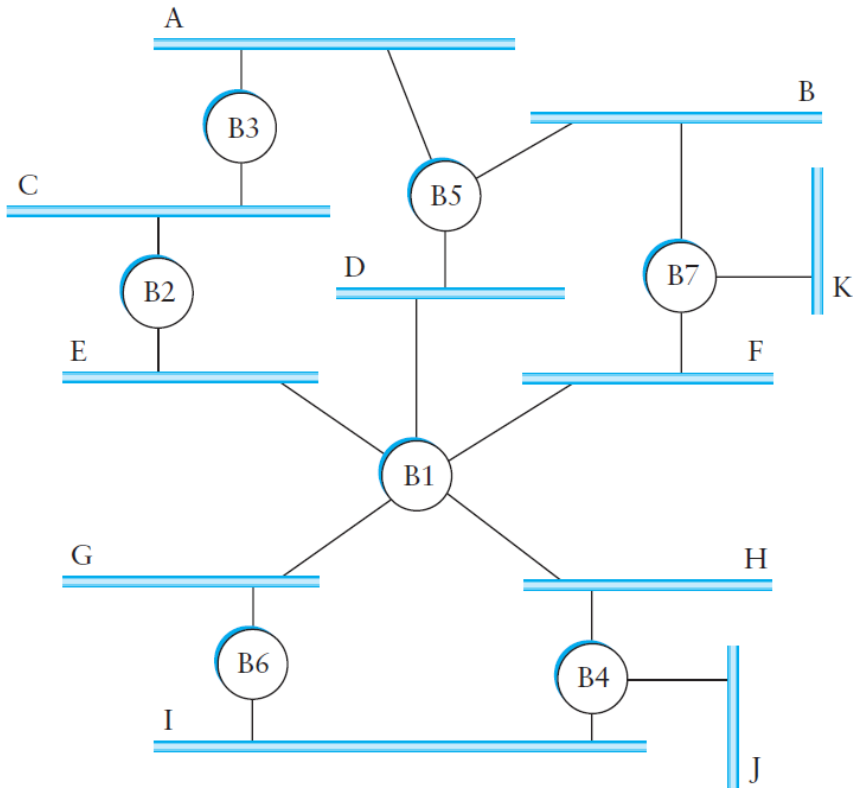
- יותר טובה כלומר $ROOT$ ID יותר קטן או $ROOT$ ID זהה ו- $dist$ יותר קטן
- הגשר זוכר מאיפה הפרסומת הטובה ביותר הגיעה - **Root Port**

בחירות **לגשרים אחראים** בוזמני

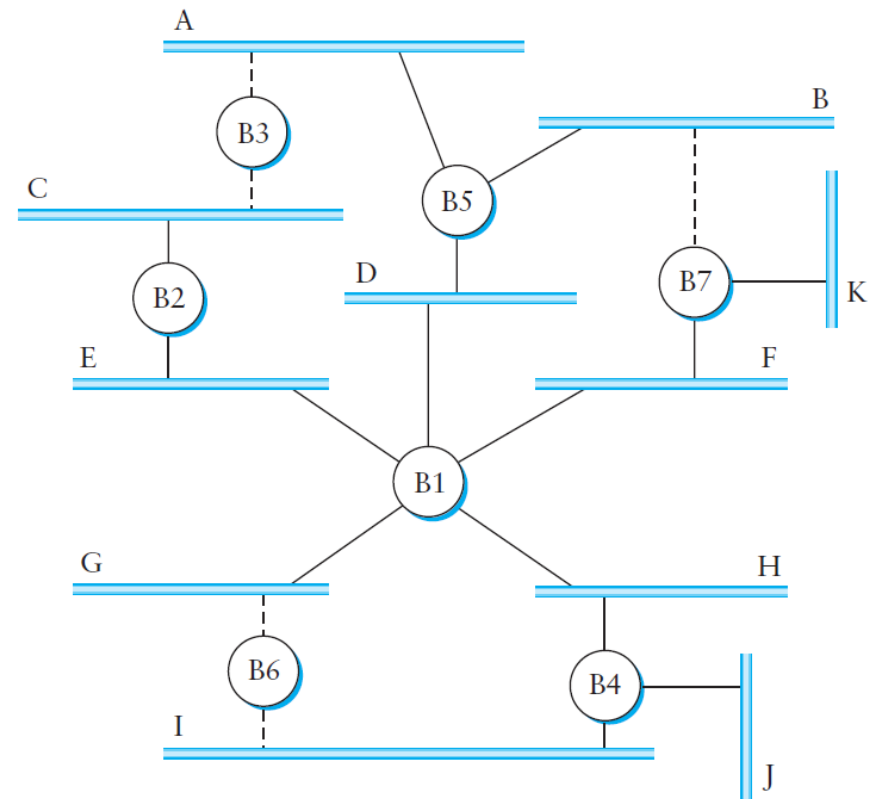
- מרחק הקטן ביותר ל- $ROOT$ או מרחק זהה ו-ID יותר קטן
- אם גשר שומע דרך יותר קצרה אל השורש על יציאה מסויימת, הוא יודע שהוא לא **הגשר האחראי** על אותו תת-רשת

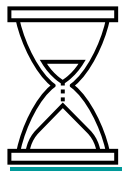
דוגמה לעץ פורש

לפני



אחרי

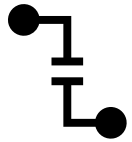




תחזוקת עצים פורשים

גשר השורש הוא האחרון שמייצר פרסומות

- הוא שולח פרסומות מפעם לפעם



- אם גשר מבחין שהוא לא שמע פרסומת במשך זמן מה (טיימר), הוא מפעיל את האלגוריתם שוב

- זיהוי אוטומטי של כשלים ושינויים בטופולוגית הרשת

פרוטוקול עץ פורש מהיר

• עדכון לפרוטוקול משנת 2004

תפקיד יציאת גיבוי

- דרך נוספת להגיע מגשר לאותו תת-רשת
- אותו גשר, רק יציאה שונה

תפקיד יציאה חלופית

- דרך נוספת להגיע לשורש
- חלופה ליציאת השורש

מצבי גשר אפשריים

- Discarding
- Learning
- Forwarding

סנכרון בין גשרים יותר מהיר

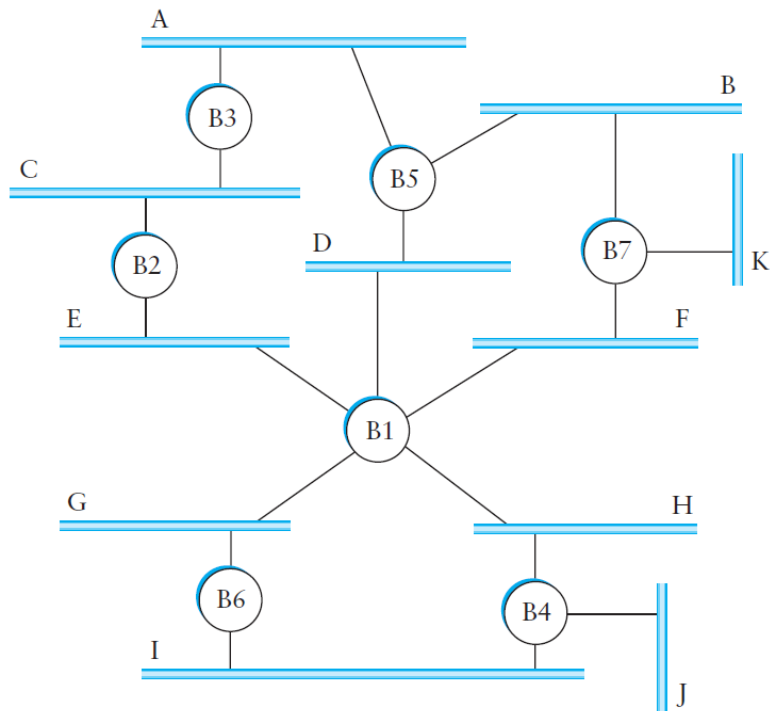
- משא ומתן, לא רק הקשבה להודעות
- מסיים את הכול תוך $3X$ "Hello Time"

כל גשר שולח מנות כל "Hello Time"

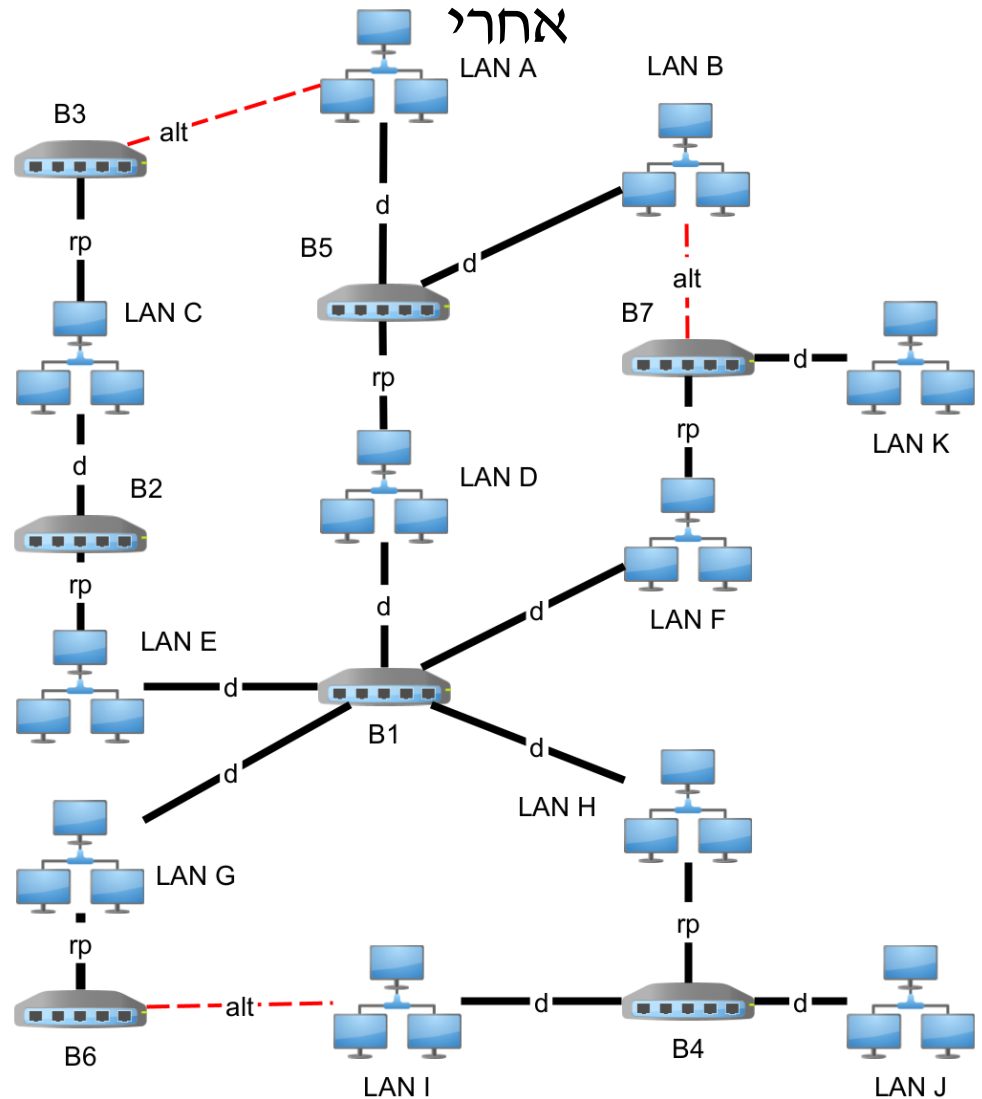
- לא רק בתגובה להודעות השורש

דוגמה לעץ פורש מהיר

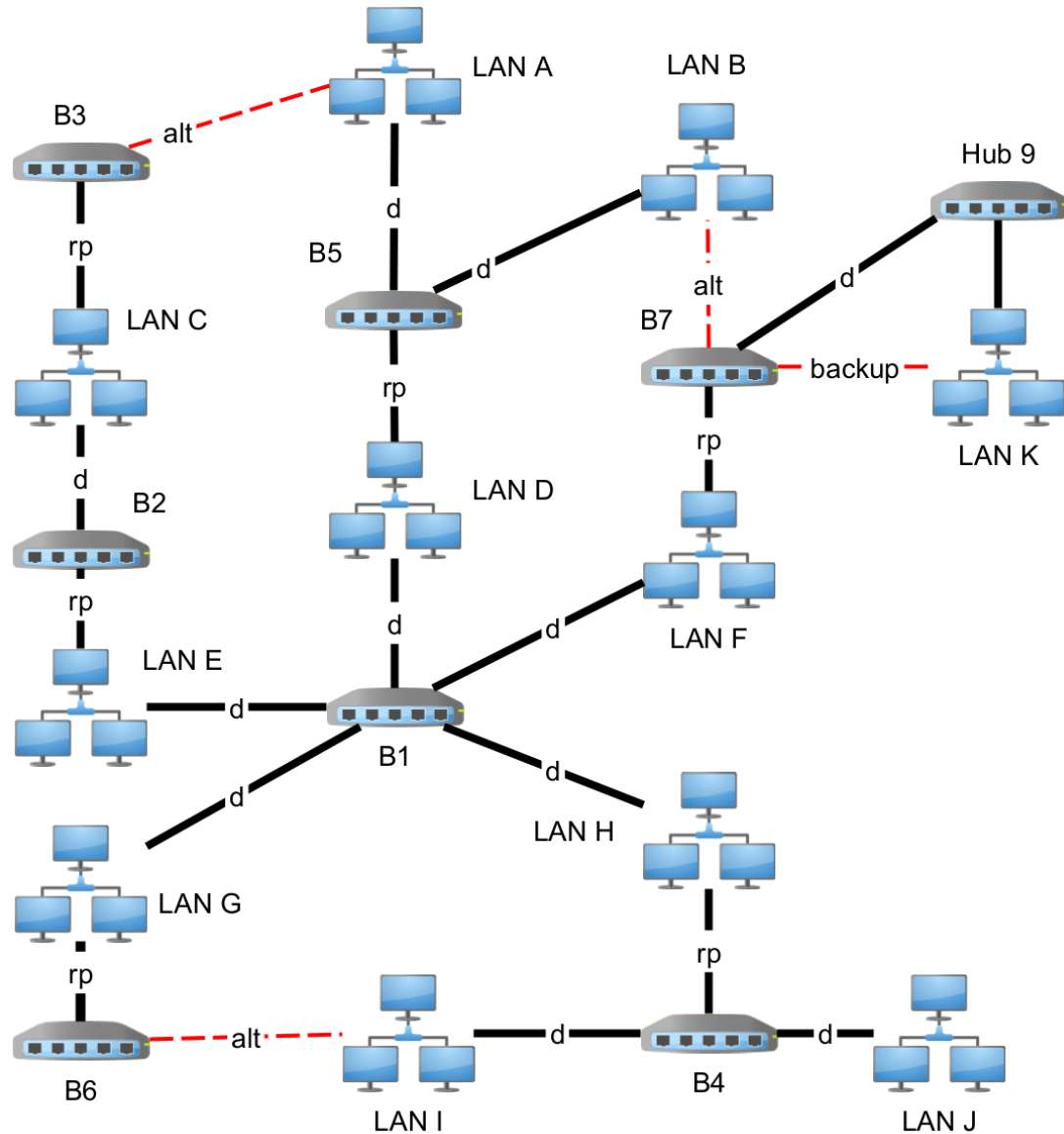
לפני



אחרי



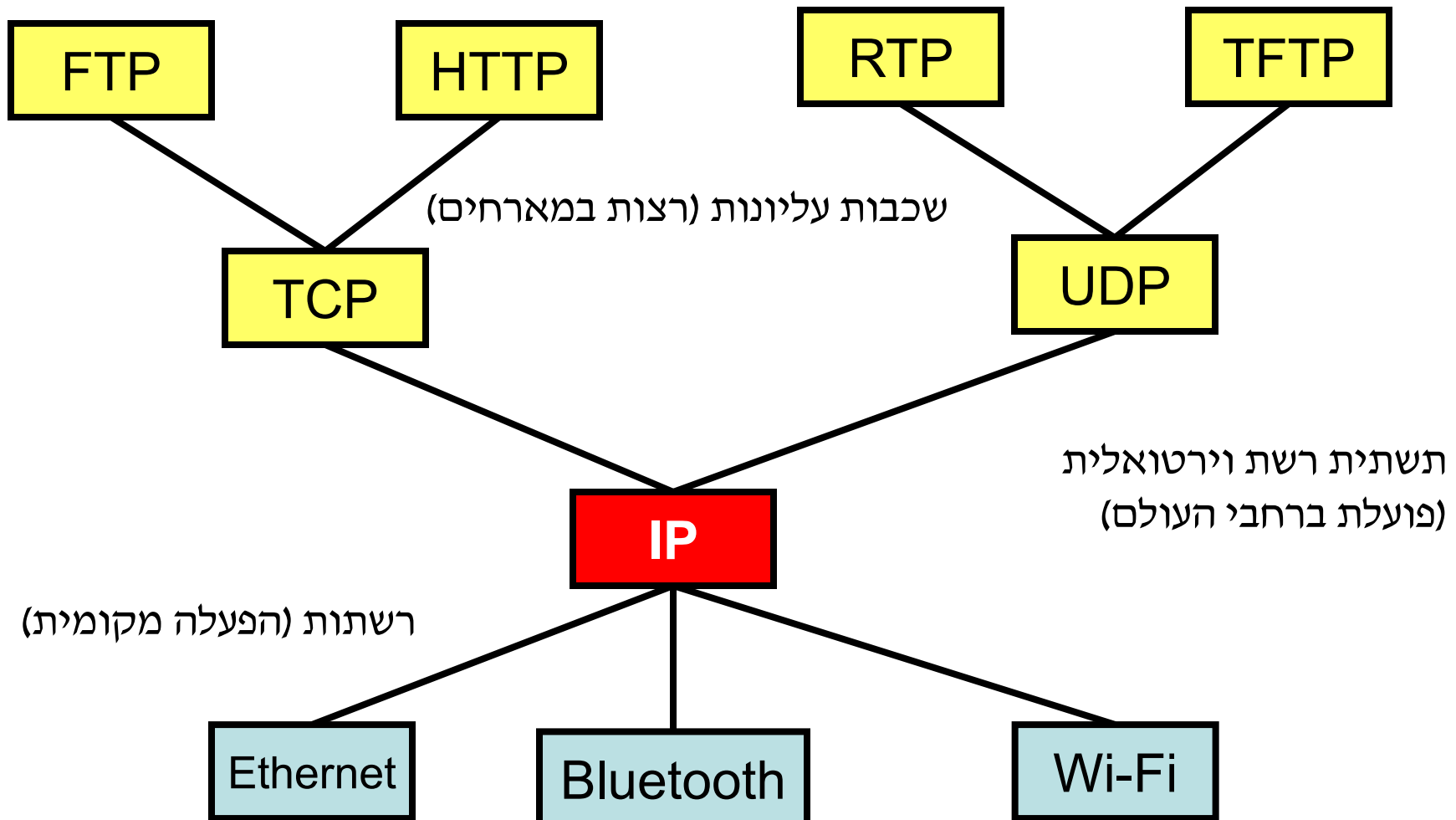
דוגמה עם יציאת גיבוי



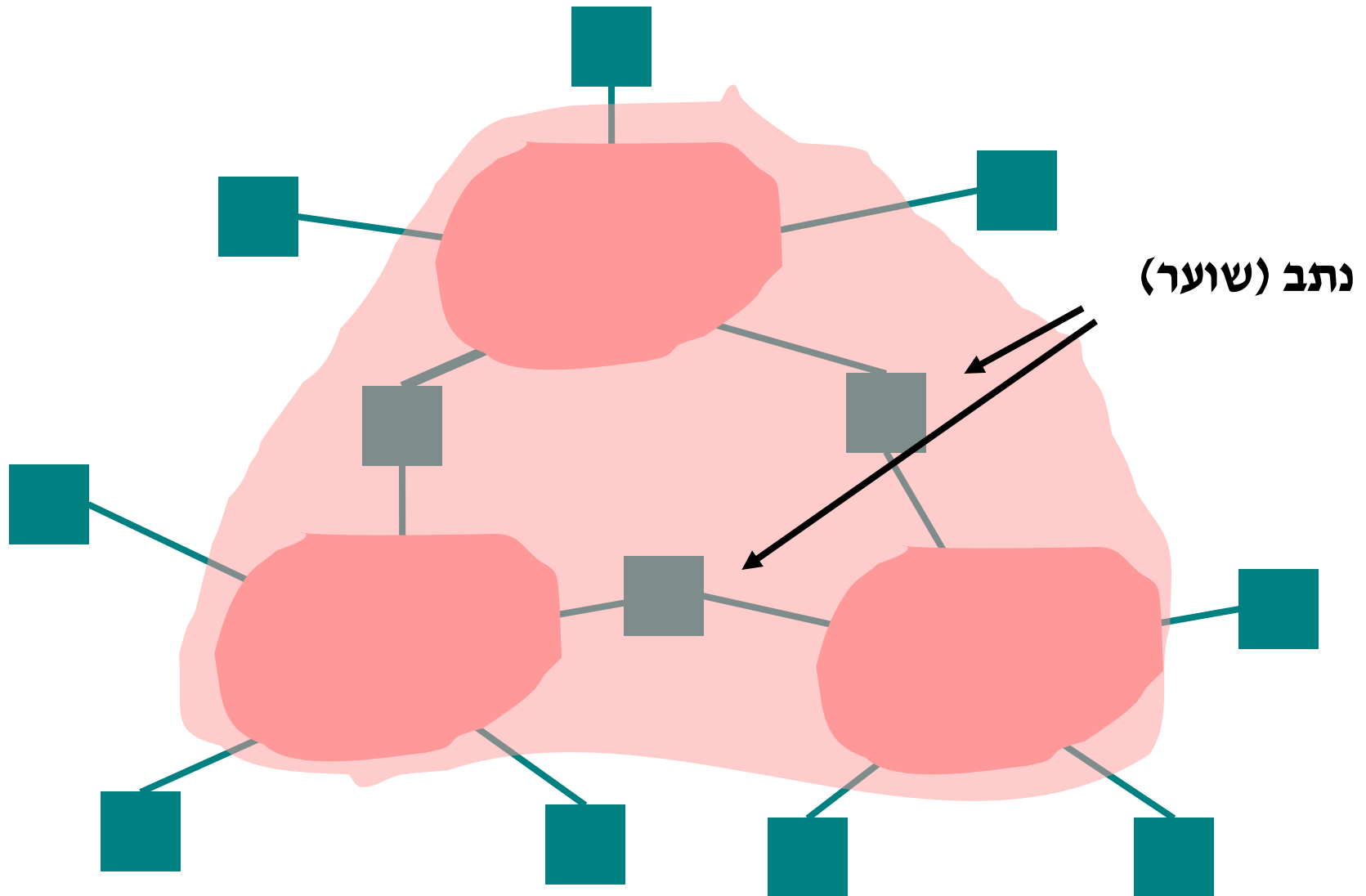
שימו לב
לרכזת!

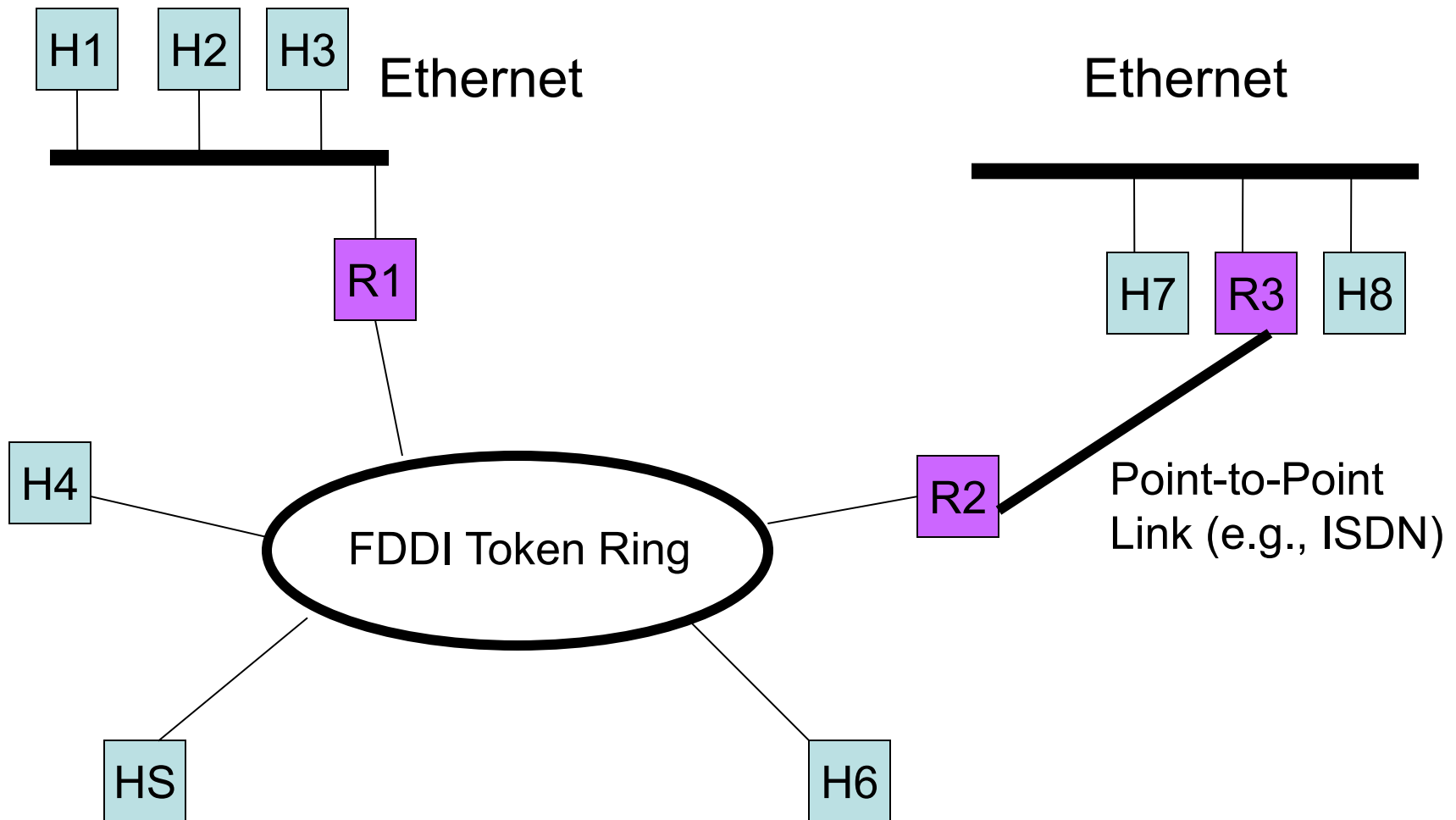
- עץ פורש לגשרים
- פרוטוקול האינטרנט (IP)
 - עקרונות
 - כתובות

שילוב של פרוטוקול אינטרנט



רשת של רשתות (מרשתת)





כתובת



נתב



העברה



פיצול



כתובות IP

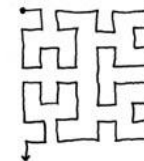
MAP OF THE INTERNET
THE IPv4 SPACE, 2006

- כתובות עם 4 בתים
- נכתבות
- 255.255.255.255
- היררכיות

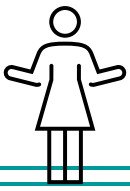


THIS CHART SHOWS THE IP ADDRESS SPACE ON A PLANE USING A FRACTAL MAPPING WHICH PRESERVES GROUPING--ANY CONSECUTIVE STRING OF IP_s WILL TRANSLATE TO A SINGLE COMPACT, CONTIGUOUS REGION ON THE MAP. EACH OF THE 256 NUMBERED BLOCKS REPRESENTS ONE /8 SUBNET (CONTAINING ALL IP_s THAT START WITH THAT NUMBER). THE UPPER LEFT SECTION SHOWS THE BLOCKS SOLD DIRECTLY TO CORPORATIONS AND GOVERNMENTS IN THE 1990's BEFORE THE RIRs TOOK OVER ALLOCATION.

0	1	14	15	16	19
3	2	13	12	17	18
4	7	8	11		
5	6	9	10		



 = UNALLOCATED BLOCK



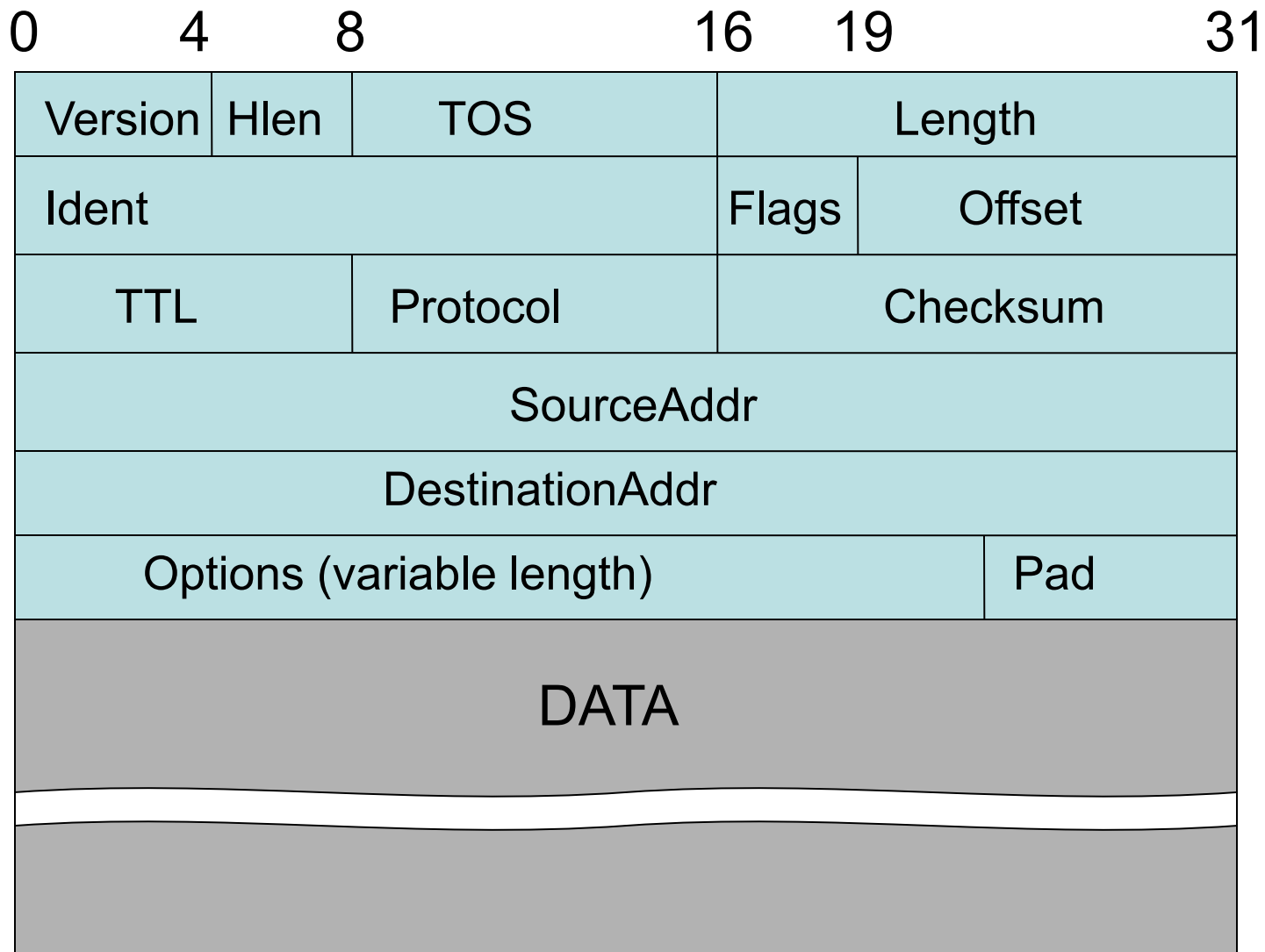
מודל שירות מינימלי

- כל רשת יכולה ליישם
- קיצון – יעבוד גם עם "פחיות וחוט"

מאפיינים

- מבנה מנה אחידה
- ניסיון הטוב ביותר להוביל מנות נתונים (לא אמין)
- רץ על כל תווד

תבנית המנה של IPv4



שדות של כותרת IPv4

TOS - Type of Service

- סוג השירות (שימוש לא נפוץ)
- עדיפות, עיכוב, תפוקה, אמינות

Hlen

- אורך הכותרת, במילים של 32 סיביות

גרסה

- גרסת IP
- בגרסה זו, 4
- השדה הראשון לכן קל לממש קוד סינון

שדות לפיצול והרכבה מחדש

- מזהה
- דגלים
- היסט

אורך

- אורך מנת הנתונים בבתיים
- שדה בגודל 16 סיביות, ולכן גודל מרבי 65,535 בתיים

שדות של כותרת IPv4

Checksum

- תמצית ביקורת של **הכותרת** (לא CRC)
- אם הכותרת לא שלמה, מתעלמים מכל החבילה

פרוטוקול

- דוגמאות: TCP (6), UDP (17)

TTL – Time to Live

- תוחלת חיים
- באמת, ספירת צעדים
- ברירת מחדל - 64.
- גם משתמשים ב-128

תוספות

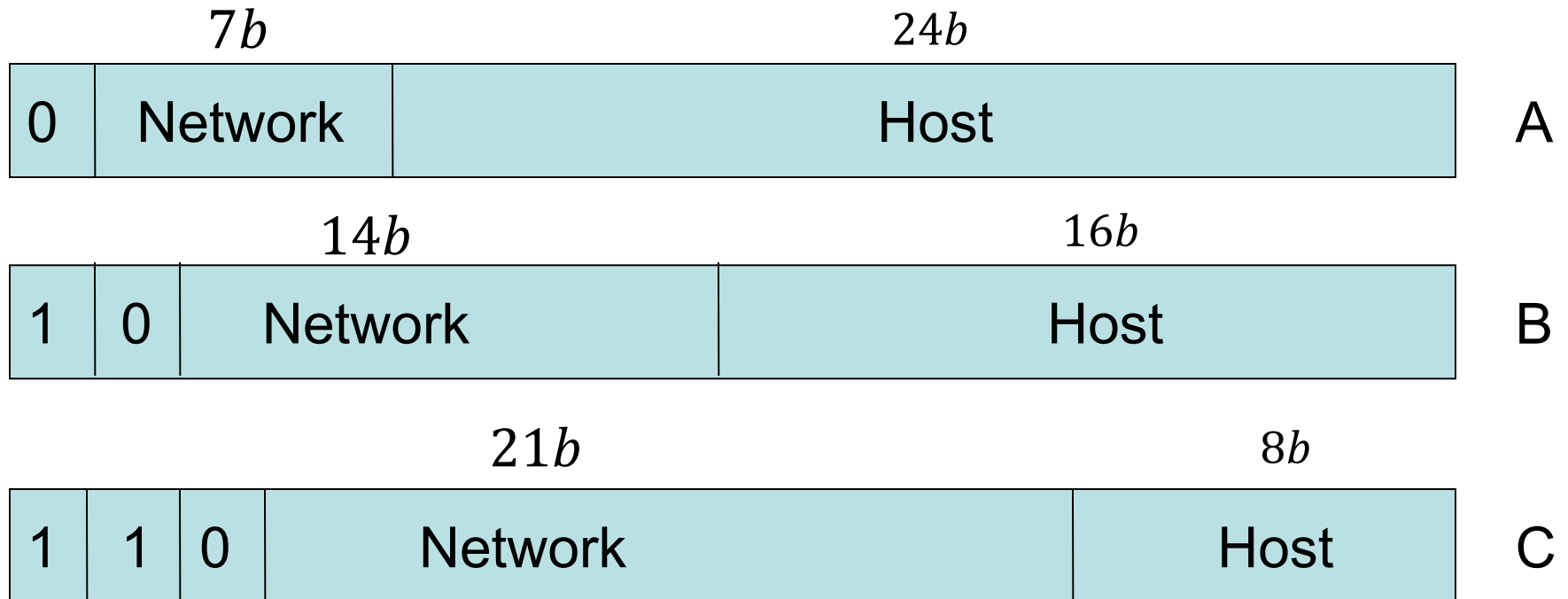
- שדות נוספים אפשריים
- ניתן לחשב אורך באמצעות Hlen

כתובות מקור ויעד

- SourceAddr,
- DestinationAddr
- כתובות בנות 32 סיביות (4 בתים)
- כתובות עולמיות לפי IP

כתובות IP

• היררכיות, לא שטוחות כמו ב-Ethernet



• נכתבות עד ידי ארבעה מספרים עשרוניים המופרדים בנקודות:

158.130.14.2

טווחי כתובות IP

0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0.0.0.0	Any IP
0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	1.0.0.0	Class A
0 0 0 0 0 0 1 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	2.0.0.0	
0 1 1 1 1 1 1 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	126.0.0.0	
0 1 1 1 1 1 1 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	127.0.0.0	Localhost
1 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	128.0.0.0	Class B
1 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	128.1.0.0	
1 0 0 0 0 0 0 0	0 0 0 0 0 0 1 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	128.2.0.0	
1 0 0 0 0 0 0 0	0 0 0 0 0 0 1 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	128.3.0.0	
1 0 1 1 1 1 1 1	1 1 1 1 1 1 1 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	191.254.0.0	
1 0 1 1 1 1 1 1	1 1 1 1 1 1 1 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	191.255.0.0	
1 1 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	192.0.0.0	Class C
1 1 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0	192.0.1.0	
1 1 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 0	0 0 0 0 0 0 0 0	192.0.2.0	
1 1 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 1	0 0 0 0 0 0 0 0	192.0.3.0	
1 1 0 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 0	0 0 0 0 0 0 0 0	223.255.254.0	
1 1 0 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	0 0 0 0 0 0 0 0	223.255.255.0	
1 1 1 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	224.0.0.0	Class D
1 1 1 0 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	239.255.255.255	
1 1 1 1 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	240.0.0.0	Class E
1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	255.255.255.255	

מחלקות כתובות IP

Class מחלקה	# of nets # רשתות	# of hosts per net # מארחים לרשת
A	128	<i>~16 million</i>
B	16,384	65,534
C	<i>~2 million</i>	254

כתובות IP ורשתות

כל מנת IP (מנת נתונים)
מכילה כתובת IP בתור
יעד

לכל מכשיר ברשת יש
כתובת IP

נתבים מחוברים
לממשקי רשת מרובים

- לנתב יש מתאמי רשת מרובים
- נתב יכול לקבל ולשלוח מנות לכל רשת שאליה הוא מחובר

חלק ה-"רשת" של
כתובת IP מזהה באופן
ייחודי רשת פיזית אחת

- חלק מהאינטרנט הגדול

אלגוריתם העברה ל-IP

מבוצע על ידי הנתב:

אם אני מחובר לאותה רשת שעליה נמצא היעד :

שלח את המנה ליעד (ARP)

אחרת : חפש בתוך טבלת הניתוב :

אם רשת היעד נמצאת בטבלת ההעברה :

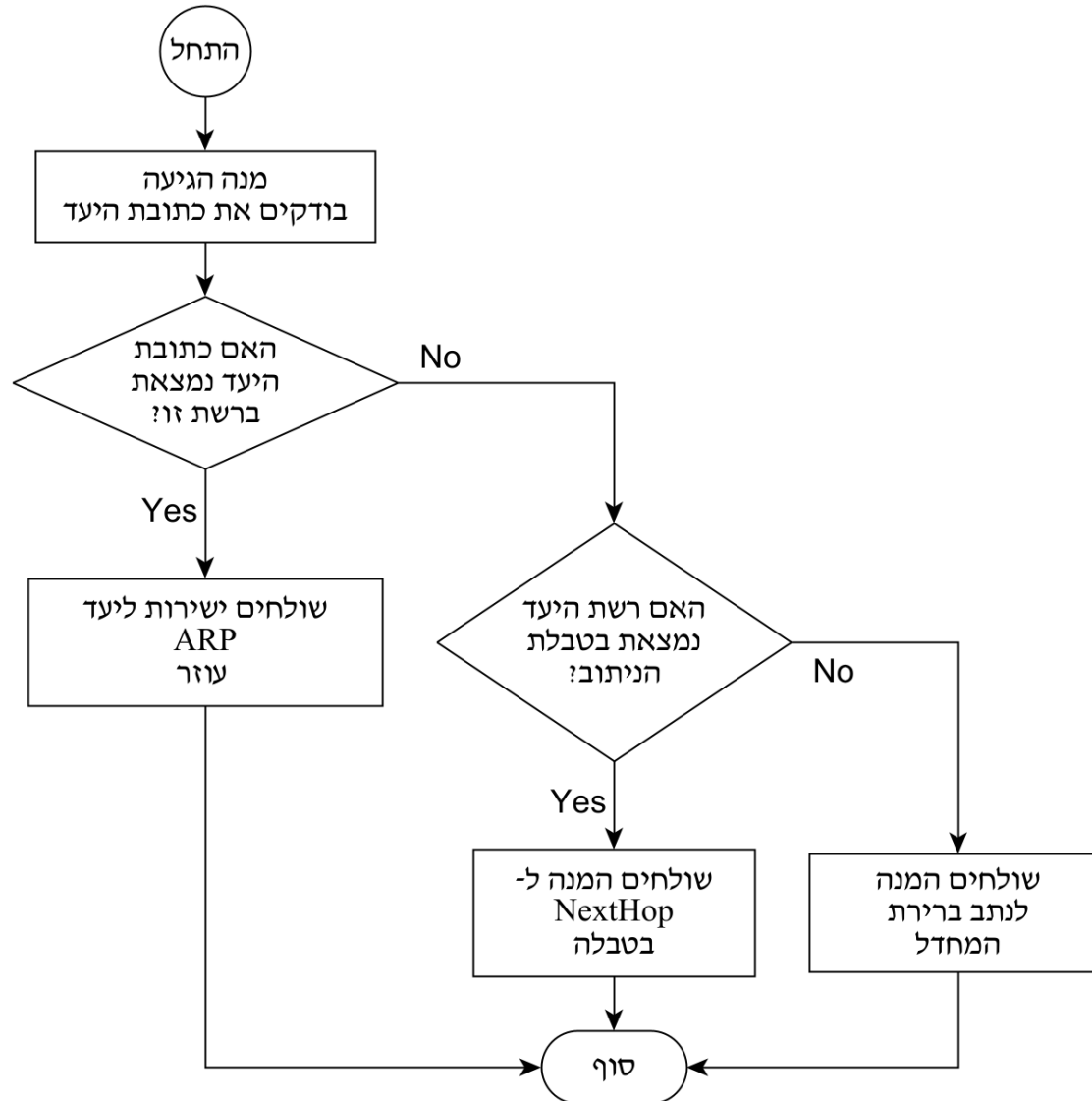
שלח את המנה לנתב NextHop

אחרת : שלח המנה לנתב ברירת המחדל

טבלת ניתוב

- מכילה רשומות (מספר רשת, צעד הבא NextHop)
- מידע נוסף
- נבנה על ידי פרוטוקול ניתוב

אלגוריתם העברה ל-IP



- עץ פורש לגשרים
- פרוטוקול האינטרנט (IP)
 - עקרונות
 - כתובות