
UDP ,RIP

2026 יוני 30
הרצאה 11

Some Slides Credits: Steve Zdancewic (UPenn)

נושאים להיום

- RIP •
- UDP •

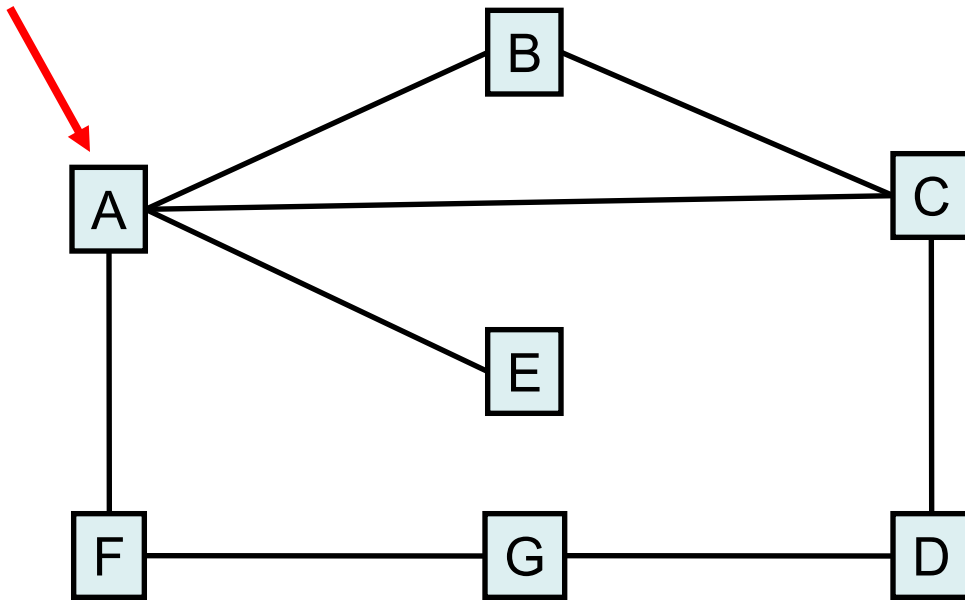
אלגוריתם וקטור מרחק (RIP)

- דומה לאלגוריתם עץ פורש לגשרים
- שונה שהמידע על המרחק מועבר לכל הצמתים (לא רק מידע על השורש).
- גם נקרא "אלגוריתם בלמן-פורד"

כל צומת בונה וקטור מרחקים

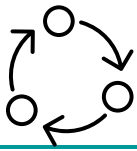
- מכיל המרחק (עלות) להגיע לכל צומת אחר
- בהתחלה:
- הרחק לשכנים הישירים (הפשטה לעכשיו) $= 1$
- המרחק לאחרים $= \infty$
- טבלת הניתוב משקפת את מה שידוע לצומת

גרף רשת לדוגמה



המידע ההתחלתי של A

Dest יעד	Cost מרחק	NextHop צעד הבא
B	1	B
C	1	C
D	∞	-
E	1	E
F	1	F
G	∞	-



שלבי עבודה

לאחר כמה
סבבי עבודה,
מידע והניתוב
מתכנס

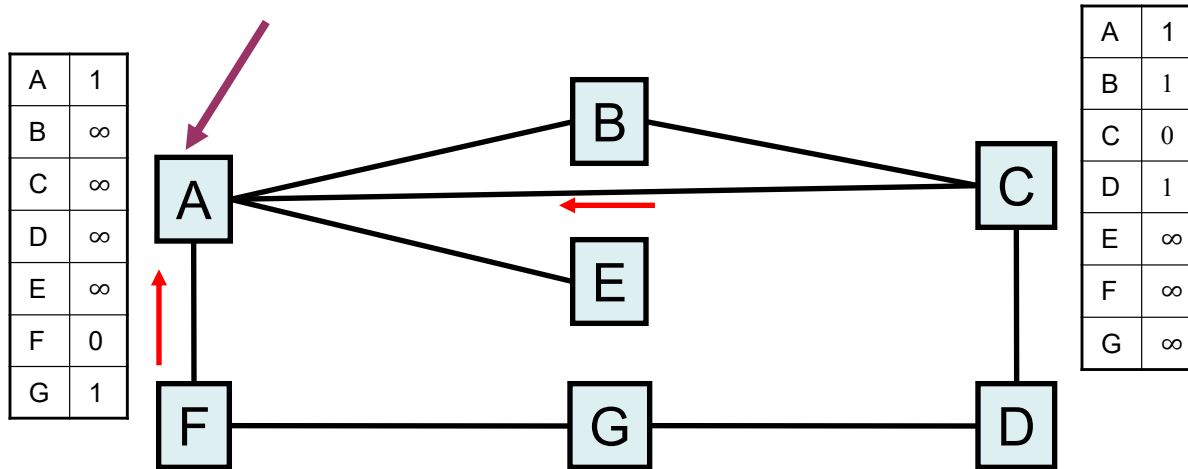
אם
המחשב
שינה את
הטבלאות,
שלח את
הווקטור
החדש
לשכנים

השכנים
מעדכנים את
וקטורי
המרחק
בהתאם.

- מתעלם ממידע
שלא משפר
- מעדכן
מסלולים
שהשתפרו

כל מחשב
שולח את
הווקטור
שלו לשכנים

שלבי עבודה



- F שולח את הווקטור ל-A
- A מגלה שניתן להגיע ל-G בשני צעדים דרך F
- C שולח את הווקטור ל-A
- A מגלה שניתן להגיע ל-D בשני צעדים דרך C

Dest	Cost	NextHop
B	1	B
C	1	C
D	∞	-
E	1	E
F	1	F
G	∞	-

Dest	Cost	NextHop
B	1	B
C	1	C
D	∞	-
E	1	E
F	1	F
G	2	F

Dest	Cost	NextHop
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	2	F

- אין לאף מחשב כל המידע על כל הרשת
- מתי לשלוח וקטורי מרחקים?
- כאשר טבלת הניתוב שלך משתנה (פעולה יזומה או תגובה)
- מדי פעם באופן קבוע ("אני חי!")

- זיהוי כשל בחיבור/צומת:
- (1) החליפו מדי פעם הודעות "אני חי!".
- (2) מנגנון גילוי אי שליחה תוך פרק זמן

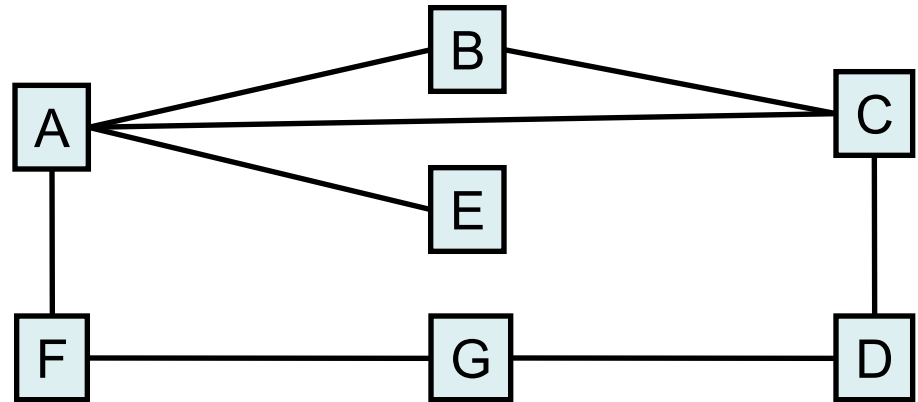
- ברשת סטטית**, אם כל המשקלים הם 1, ברגע ש-A מגלה מסלול ל-B (עלות ∞), אף סבב עתידי לא יחשוף מסלול יותר טוב ל-B
- מצב בו כל המשקלים 1 נקרא "ספירת צעדים" (פעולת ברירת המחדל של (RIP)

רשתות דינמיות

מה לגבי רשת דינמית?

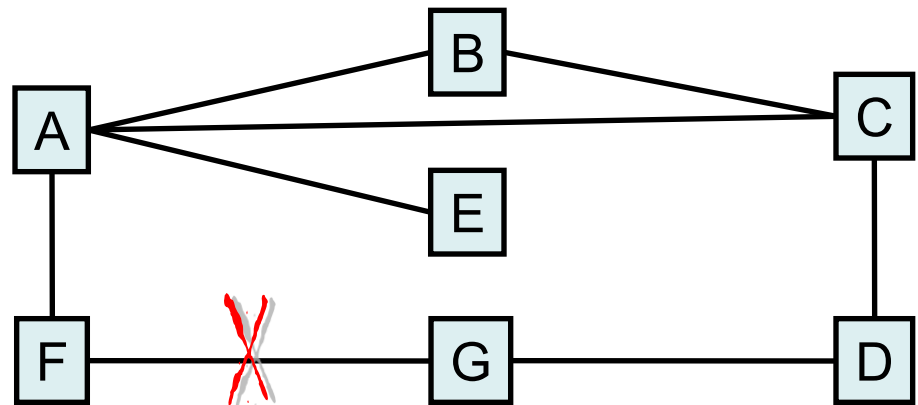
• צמתים חדשים נכנסים

– השכנים החדשים בסופו של דבר יודיעו את כולם (קל)



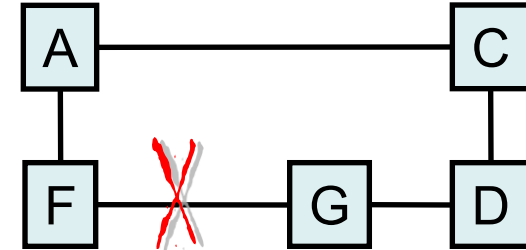
• צומת/חיבור נפל

– האם הרשת מחולקת לשתים?
– איך צמתים מגלים שצומת/קישור נעלם?



סיפור 1: כשל בחיבור רשת

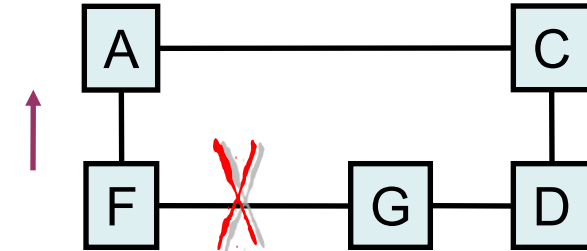
A			C			D			F		
Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop
A	-	-	A	1	A	A	2	C	A	1	A
C	1	C	C	-	-	C	1	C	C	2	A
D	2	C	D	1	D	D	-	-	D	2	G
F	1	F	F	2	A	F	2	G	F	-	-
G	2	F	G	2	D	G	1	G	G	1	G



A			C			D			F		
Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop
A	-	-	A	1	A	A	2	C	A	1	A
C	1	C	C	-	-	C	1	C	C	2	A
D	2	C	D	1	D	D	-	-	D	2	G
F	1	F	F	2	A	F	2	G	F	-	-
G	2	F	G	2	D	G	1	G	G	∞	-

סיפור 1 : כשל בחיבור רשת

A			C			D			F		
Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop
A	-	-	A	1	A	A	2	C	A	1	A
C	1	C	C	-	-	C	1	C	C	2	A
D	2	C	D	1	D	D	-	-	D	2	G
F	1	F	F	2	A	F	2	G	F	-	-
G	2	F	G	2	D	G	1	G	G	∞	-

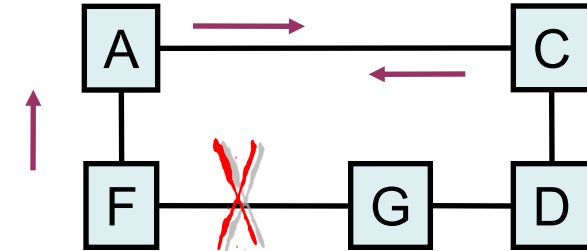


מקרה 1 :
1. F מעדכן את A

A			C			D			F		
Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop
A	-	-	A	1	A	A	2	C	A	1	A
C	1	C	C	-	-	C	1	C	C	2	A
D	2	C	D	1	D	D	-	-	D	2	G
F	1	F	F	2	A	F	2	G	F	-	-
G	∞	-	G	2	D	G	1	G	G	∞	-

סיפור 1: כשל בחיבור רשת

A			C			D			F		
Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop
A	-	-	A	1	A	A	2	C	A	1	A
C	1	C	C	-	-	C	1	C	C	2	A
D	2	C	D	1	D	D	-	-	D	2	G
F	1	F	F	2	A	F	2	G	F	-	-
G	∞	-	G	2	D	G	1	G	G	∞	-



מקרה 1:

1. F מעדכן את A

2. A מעדכן את C

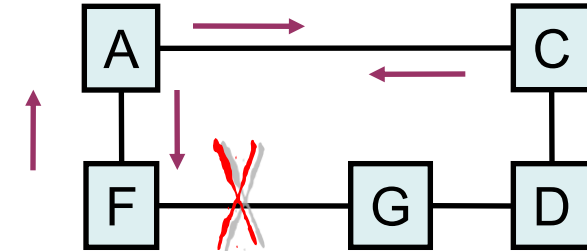
• (כלום)

3. C מעדכן את A

A			C			D			F		
Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop
A	-	-	A	1	A	A	2	C	A	1	A
C	1	C	C	-	-	C	1	C	C	2	A
D	2	C	D	1	D	D	-	-	D	2	G
F	1	F	F	2	A	F	2	G	F	-	-
G	3	C	G	2	D	G	1	G	G	∞	-

סיפור 1: כשל בחיבור רשת

A			C			D			F		
Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop
A	-	-	A	1	A	A	2	C	A	1	A
C	1	C	C	-	-	C	1	C	C	2	A
D	2	C	D	1	D	D	-	-	D	2	G
F	1	F	F	2	A	F	2	G	F	-	-
G	3	C	G	2	D	G	1	G	G	∞	-



מקרה 1:

1. F מעדכן את A

2. A מעדכן את C

• (כלום)

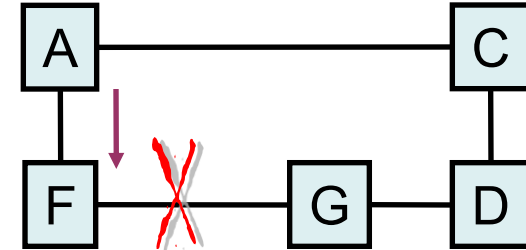
3. C מעדכן את A

4. A מעדכן את F

A			C			D			F		
Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop
A	-	-	A	1	A	A	2	C	A	1	A
C	1	C	C	-	-	C	1	C	C	2	A
D	2	C	D	1	D	D	-	-	D	2	G
F	1	F	F	2	A	F	2	G	F	-	-
G	3	C	G	2	D	G	1	G	G	4	A

סיפור 2: כשל בחיבור רשת

A			C			D			F		
Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop
A	-	-	A	1	A	A	2	C	A	1	A
C	1	C	C	-	-	C	1	C	C	2	A
D	2	C	D	1	D	D	-	-	D	2	G
F	1	F	F	2	A	F	2	G	F	-	-
G	2	F	G	2	D	G	1	G	G	∞	-



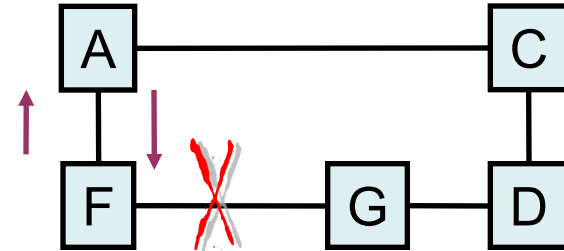
מקרה 2:

1. A מעדכן את F

A			C			D			F		
Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop
A	-	-	A	1	A	A	2	C	A	1	A
C	1	C	C	-	-	C	1	C	C	2	A
D	2	C	D	1	D	D	-	-	D	2	G
F	1	F	F	2	A	F	2	G	F	-	-
G	2	F	G	2	D	G	1	G	G	3	A

סיפור 2: כשל בחיבור רשת

A			C			D			F		
Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop
A	-	-	A	1	A	A	2	C	A	1	A
C	1	C	C	-	-	C	1	C	C	2	A
D	2	C	D	1	D	D	-	-	D	2	G
F	1	F	F	2	A	F	2	G	F	-	-
G	2	F	G	2	D	G	1	G	G	3	A



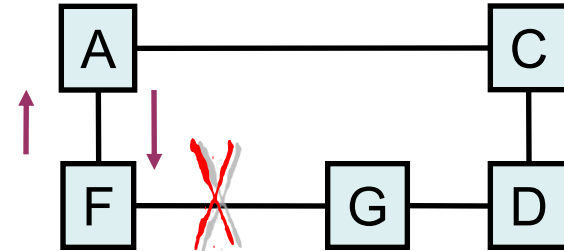
מקרה 2:

1. A מעדכן את F
2. F מעדכן את A

A			C			D			F		
Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop
A	-	-	A	1	A	A	2	C	A	1	A
C	1	C	C	-	-	C	1	C	C	2	A
D	2	C	D	1	D	D	-	-	D	2	G
F	1	F	F	2	A	F	2	G	F	-	-
G	4	F	G	2	D	G	1	G	G	3	A

סיפור 2: כשל בחיבור רשת

A			C			D			F		
Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop
A	-	-	A	1	A	A	2	C	A	1	A
C	1	C	C	-	-	C	1	C	C	2	A
D	2	C	D	1	D	D	-	-	D	2	G
F	1	F	F	2	A	F	2	G	F	-	-
G	4	F	G	2	D	G	1	G	G	3	A



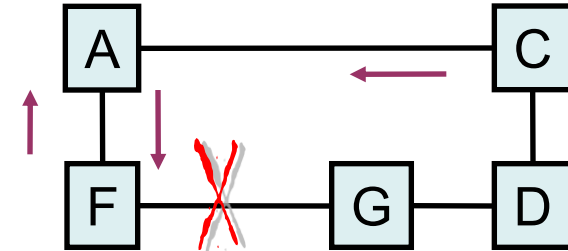
מקרה 2:

1. A מעדכן את F
2. F מעדכן את A
3. A מעדכן את F

A			C			D			F		
Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop
A	-	-	A	1	A	A	2	C	A	1	A
C	1	C	C	-	-	C	1	C	C	2	A
D	2	C	D	1	D	D	-	-	D	2	G
F	1	F	F	2	A	F	2	G	F	-	-
G	4	F	G	2	D	G	1	G	G	5	A

סיפור 2 : כשל בחיבור רשת

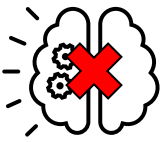
A			C			D			F		
Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop
A	-	-	A	1	A	A	2	C	A	1	A
C	1	C	C	-	-	C	1	C	C	2	A
D	2	C	D	1	D	D	-	-	D	2	G
F	1	F	F	2	A	F	2	G	F	-	-
G	4	F	G	2	D	G	1	G	G	5	A



מקרה 2 :

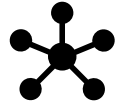
1. A מעדכן את F
2. F מעדכן את A
3. A מעדכן את F
4. ...
5. C מעדכן את A
6. A מעדכן את F

A			C			D			F		
Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop	Dest	C	Next hop
A	-	-	A	1	A	A	2	C	A	1	A
C	1	C	C	-	-	C	1	C	C	2	A
D	2	C	D	1	D	D	-	-	D	2	G
F	1	F	F	2	A	F	2	G	F	-	-
G	3	C	G	2	D	G	1	G	G	4	A



- הלולאה בין A ל-F נשברת רק על ידי העדכון של C
- ל-C יש מידע אמיתי על מסלול אחר ל-G
- מה אם העדכון של C לא מגיע? מה אם המידע של C שגוי?
 - הרשת מחולקת – או –
 - G לגמרי ירד מהרשת
- הצמתים האחרים ימשיכו לספור עד האינסוף (או עד שהשדה "מרחק" יגיע לערך המרבי)

שלושה פתרונות ל-RIP



רעיון 1: אינסוף קטן

- לא ניתן לשדה "מרחק" לעלות מעל רמה מוגדרת מראש
 - נניח 15 או 20 וכדומה (15 הוא הערך האמיתי)
 - הערך המרבי חייב להיות גדול מקוטר הרשת



רעיון 2: אל תציע מסלול למי ששלח לך אותו

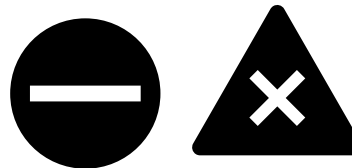
דוגמה: F מקבל עדכון מ-A לגבי דרך להגיע ל-G

- בדוגמה הקודמת, F לא "ידע" שהמסלול ש-A הציע עובר דרך F
- A לא ישלח ל-F הצעה להגיע ל-G בגלל שהיא עוברת דרך F
- נקרא "אופק מחולק"
- "אופק מחולק עם רעל בכיוון הפוך" – A ישלח הצעה ל-F עבור G אך עם מרחק אינסופי

שלושה פתרונות ל-RIP

אפשרות 3 : הרעלת מסלולים ו-Holddown

- במקום לפסיק רק להודיע על יעד לא נגיש, הנתב ישלח הודעת "רעל" עבור היעד
 - לדוגמה, הנתב מודיע על מרחק 16 (∞) עבור היעד
- לתקופה קבועה (נניח 3 דקות) כל הצעה חדשה ליעד דרך השולח ההוא תידחה
 - למשל, לחכות שכולם ישמעו את הודעה הפסילה ואחר כך קבל הצעות חדשות
- זה נותן הזדמנות לכולם למחוק את המסלול הישן ולא קיים
- קיים רק בציווד של חברת סיסקו



פרטים על RIP גרסה 1 (1986)

פקודות

- בקשה: בקש שמישהו ישלח את פרטיו
- תגובה: מגיב עם טבלת ניתוב

תומך במספר משפחות כתובות

ניתוב מבוסס מחלקות בלבד

- אין מסכות תת-רשת
- מניח שלכולם יש את אותה מסכה

בוחר את הדרך הקצרה ביותר

- אם יש תיקו לקטן, מחלק את התנועה עליהם באופן שווה

אופק מחולק, הרעלת מסלולים, אינסוף קטן

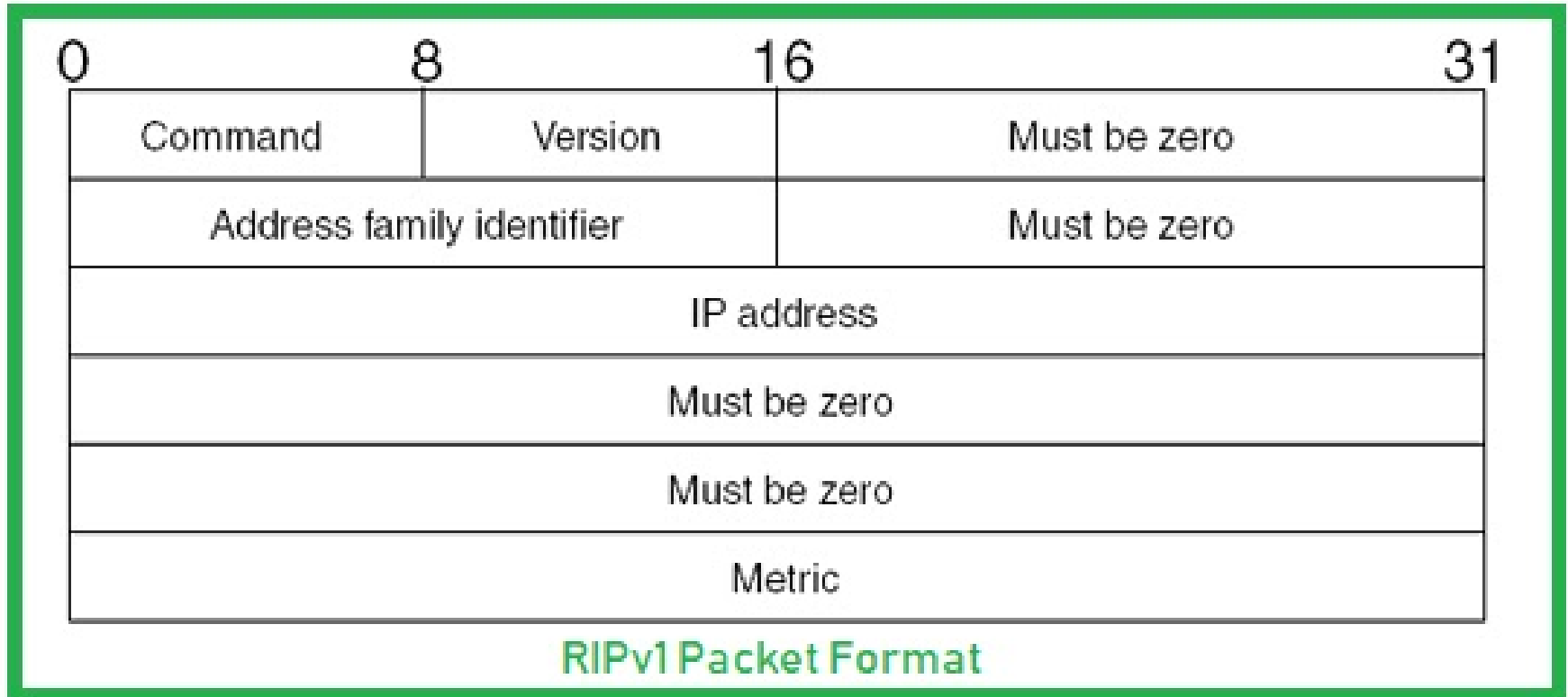
שולח עדכונים

- טיימר בערך כל 30 שניות בין העדכונים
- עקב שינויים

מחיקת מסלול

- אם אין עדכון תוך 180 שניות או דרך מגיע לאינסוף
- מסמן כמחוק
- חי עוד 120 שניות כדי להודיע לאחרים

פרטים על RIP גרסה 1 (1986)



<https://www.networkurge.com/2020/05/ripv1.html>

פרטים על RIP גרסה 2 (1998)


RIP •

UDP •

Open Systems Interconnection (OSI)

מחשב קצה

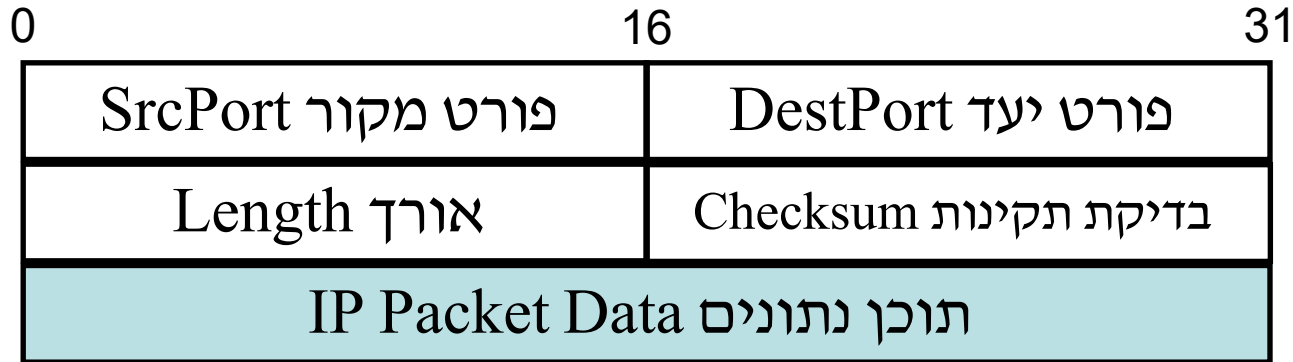
מודל פורמלי – לא מימש אמיתי

Application אפליקציה	שולח הודעות (למשל, HTTP או FTP)
Presentation תצוגה	מטפל בפורמט נתונים (למשל, big- vs. little-endian)
Session שיחה	מנהל זרימות נתונים
Transport תעבורה	פרוטוקולי תהליך לתהליך 
Network רשת	מנתב מנות בין צמתים ברשת
Link עורק	אורז סיביות בתוך מסגרות
Physical פיזי	מעביר סיביות פשוטות על גבי הלינק

צרכי אפליקציות לעומת יכולות הרשת

מאפייני הרשת	צרכי אפליקציות
מוחקת, משכפלת, מסדרת מחדש הודעות	שליחה אמינה, לפי סדר, הגעה פעם אחת בלבד
גודל הודעה קבוע וסופי (MTU)	הודעות גדולות באופן שרירותי
עיכוב שרירותי	קבלת הודעות בזמן
מנגנוני שליחה מבוססי כתובת פיזית ומסירה ברמת המארח (ללא יכולת להבחין בין שיחות)	תמיכה ביישומים מרובים לכל מארח
אין מנגנוני בקרת זרימה מקצה לקצה	בקרת זרימה על ידי המקבל

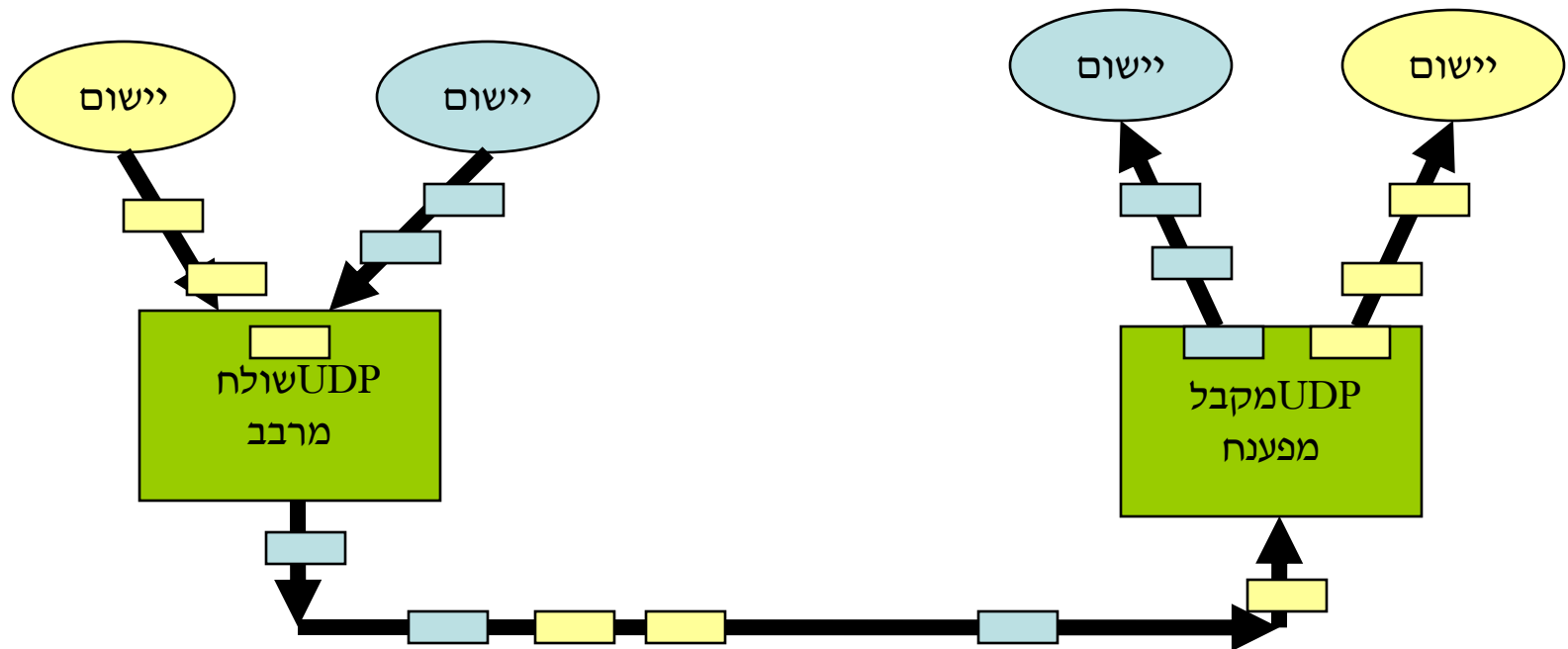
User Datagram Protocol (UDP)



- פרוטוקול למנות ברמת המשתמש
 - פרוטוקול שכבת התעבורה הפשוט ביותר
 - פשוט חושף את הפונקציונליות של מנות IP לרמת התעבורה
 - פורט (Port) מזהה את תהליך השולח/המקבל
 - מפענח ריבוב הנתונים (מספר פורט הוא מפתח פענוח)
 - צמד (Port, כתובת מחשב (IP)) מזהה תהליך ברשת
- שאלה:** מדוע יש שדה Length בכותרת?

מודל UDP מקצה לקצה

- ריבוב/פענוח עם מספר פורט



שימוש בפורטים

- לקוחות שולחים הודעות לשירותים בפורטים ידועים מראש (Well Known Ports)
 - SMTP (שליחת הודעות אימייל) : פורט 25
 - DNS (תרגום שמות ווביות) : פורט 53
 - POP3 (הורדת אימיילים) : פורט 110
 - HTTP (גלישה בווב) : פורט 80
 - UNIX Talk (כלי צ'אט ישן) : פורט 517
 - ביוניקס ולינוקס, הפורטים מופיעים ב-`/etc/services`
 - בחלונות, הפורטים מופיעים ב-`C:\Windows\System32\drivers\etc\services`
- פורטים הינם מושגים מופשטים (הפשטה) שמערכת ההפעלה מייצרת
 - מיושם בצורה שונה במערכות הפעלה שונות
 - בדרך כלל תור הודעות

RIP •

UDP •