
TCP, בקרת גודש, פרוטוקולי דבק

24 יוני 2025
הרצאה 12

Some Slides Credits: Steve Zdancewic (UPenn)

- TCP

- לחיצת יד ויסודות

- בקרת גודש

- פרוטוקולי דבק

- ICMP

Transmission Control Protocol (TCP)

- הפרוטוקול בקרת שידור

- הפרוטוקול הנפוץ ביותר לשליחת זרימת נתונים בצורה אמינה

- אמינה, מספקת שליחת זרימת בתים בסדר

- דו-כיווני (Full Duplex): זוג זרמים, אחד לכל כיוון

- מנגנונים לבקרת הזרימה ולבקרת הגודש

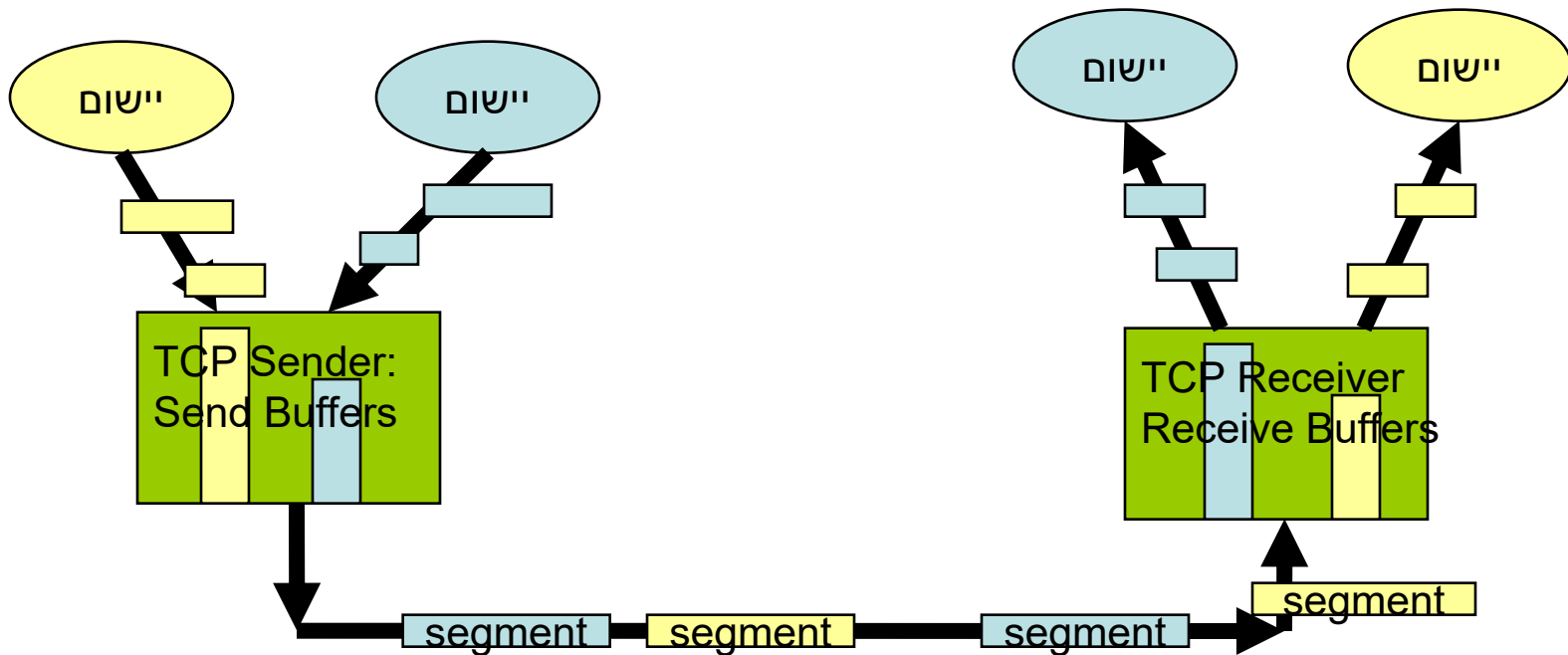
- כמו UDP, עובד עם פורטים (ports)

- בנוי על גבי IP בלבד, לכן נקרא TCP/IP

- תקן יצא בשנת 1980

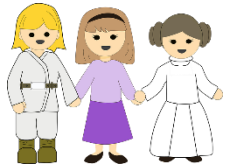
מודל TCP מקצה לקצה

- השימוש במחסניות מתקנת שגיאות אך עלולה לגרום לעיכובים
- Segment – מקטע ולא מנה



תבנית המקטע

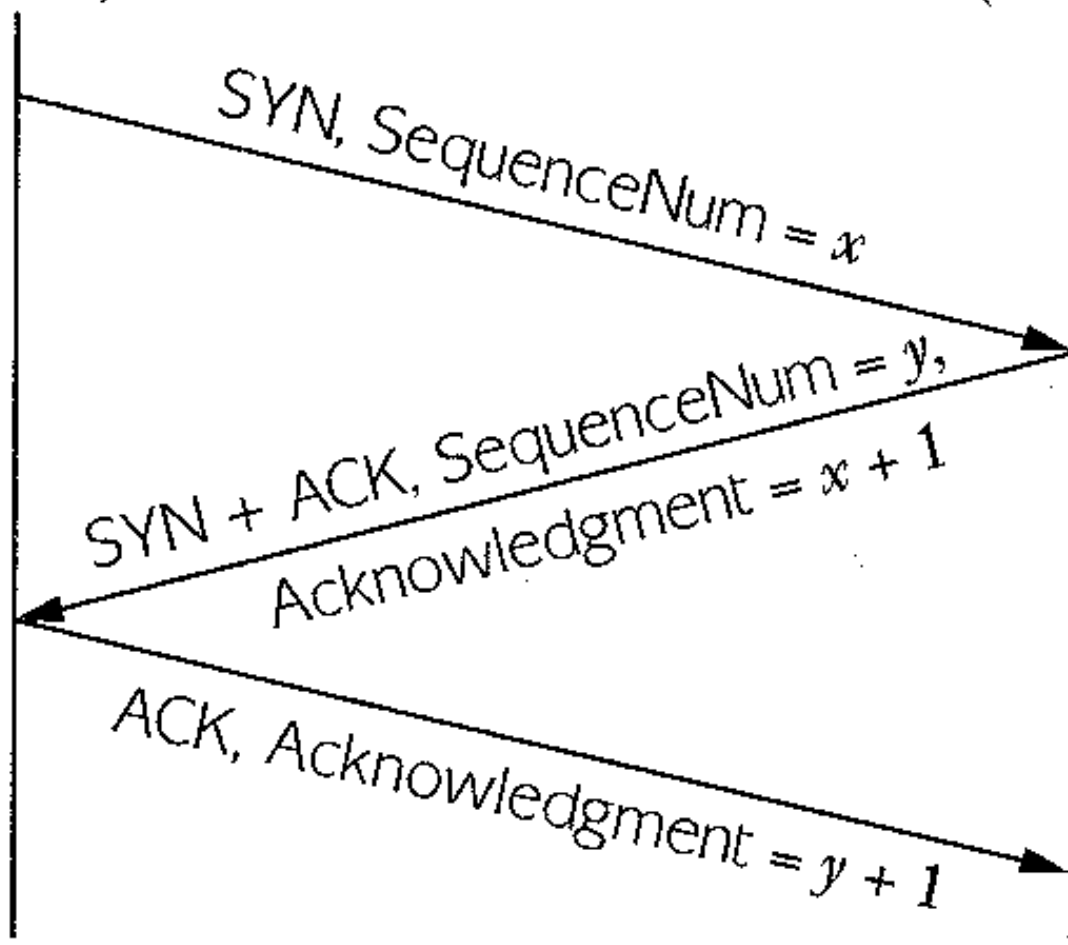
דגלים •		0	15	31
SYN –	Source Port		Destination Port	
FIN –	פורט מקור		פורט יעד	
RESET –	Sequence Number מספר רץ			
PUSH –	Acknowledgement אישור קבלה			
URG –	HL	0	Flags	Advertised Window
ACK –			דגלים	חלון המפורסם
	Checksum		Urgent Pointer	
			מצביע דחוף	
	Options (variable) תוספות			
	Data נתונים			



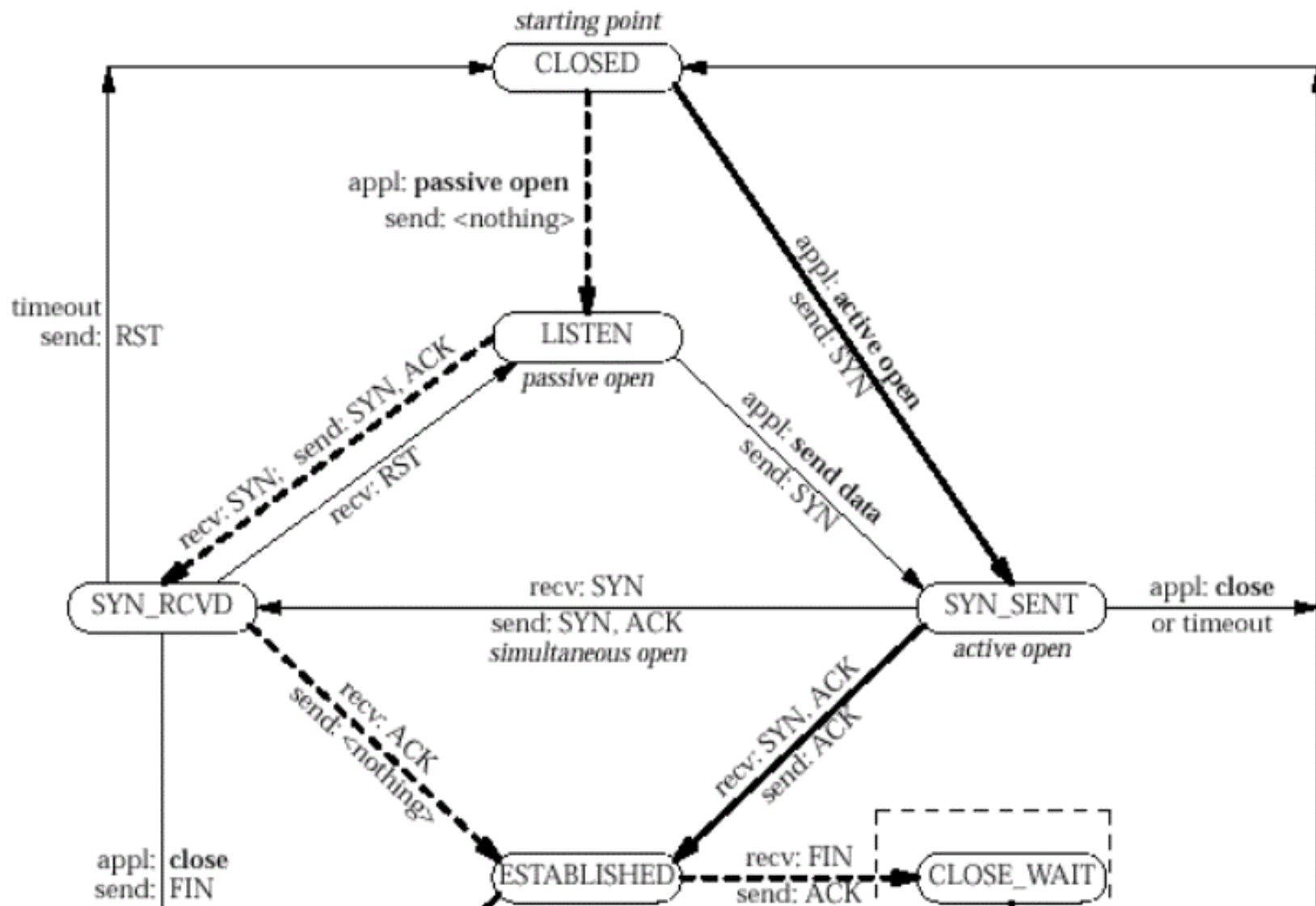
לחיצת יד תלת-שלבית

Active participant
(client)

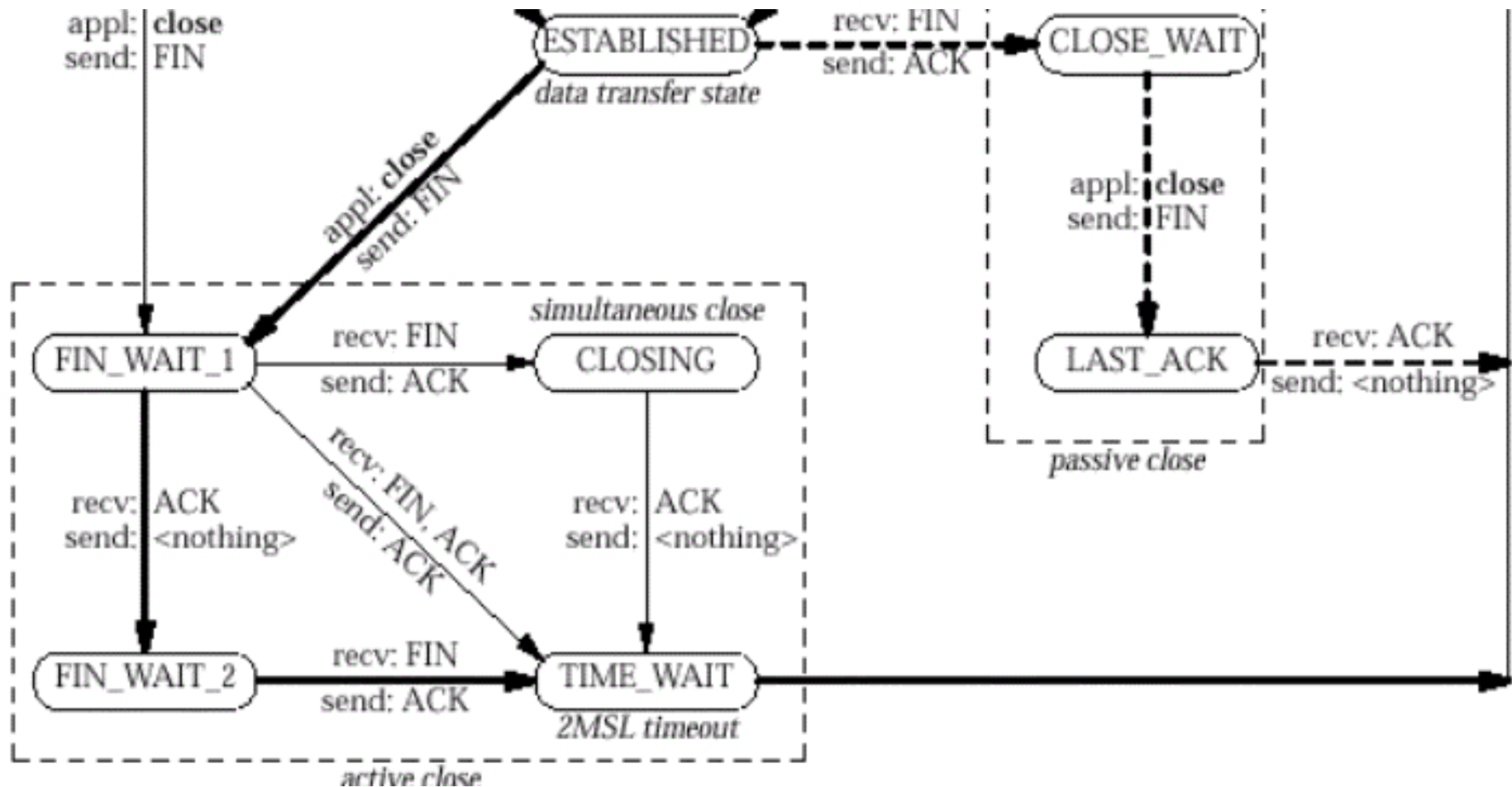
Passive participant
(server)



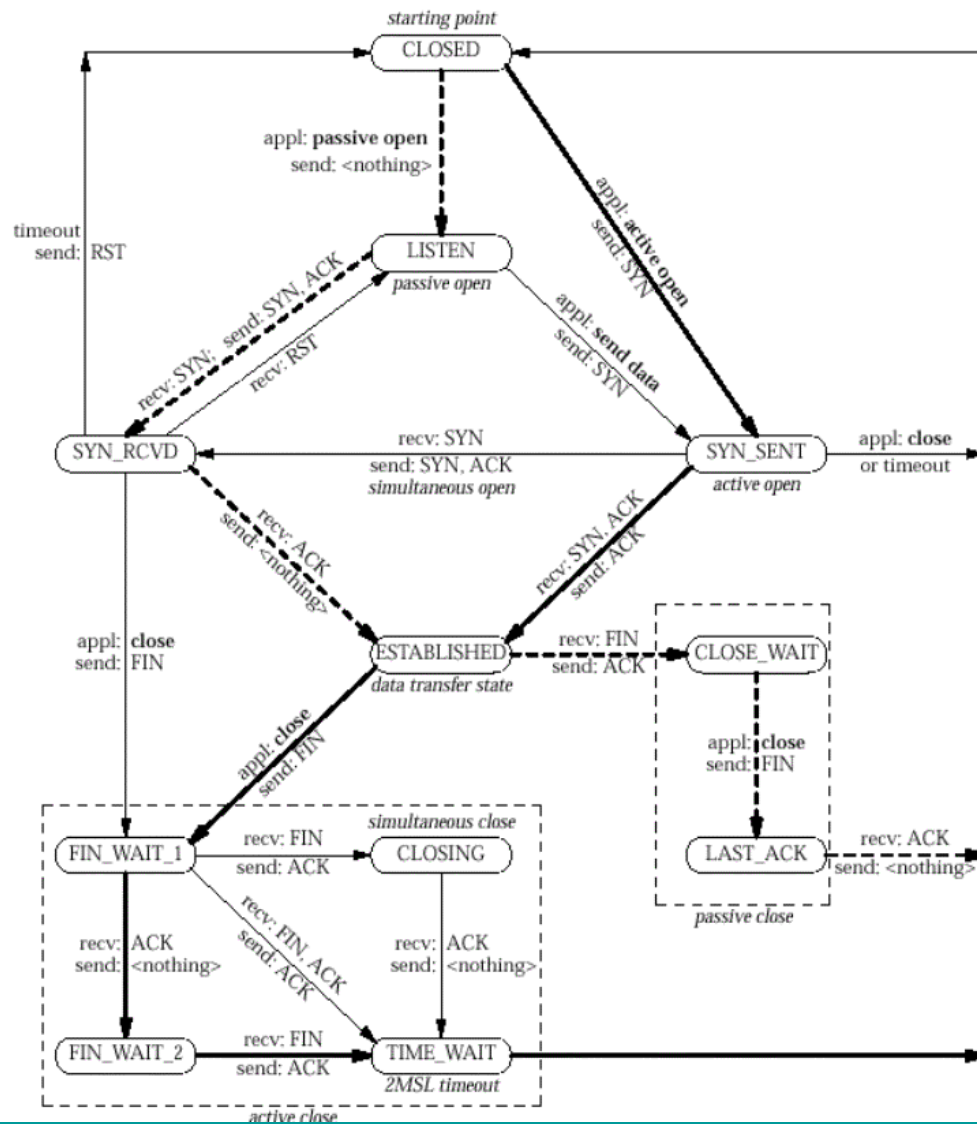
מכונת המצבים של TCP



מכונת המצבים של TCP



מכונת המצבים של TCP



שני דגלים

URG 

- לשולח יש נתונים דחופים

SYN	FIN	RST	PSH	URG	ACK
-	-	-	-	1	-

Urgent Pointer = 100

Byte #	0	100	101	500
	Urgent Data		נתונים אחרים	

- ה-TCP של המקבל יעביר את הנתונים לאפליקציה בעדיפות גבוהה

 PUSH

- פיסת נתונים קריטית לשולח
- השולח מגדיר סיבית PUSH במקטע התוצאה:

- ה-TCP של השולח ישלח את הנתונים מייד (לא תשמור במחסנית עד שהוא קרוב ל-MTU)

- ה-TCP של המקבל ישלח את הנתונים לאפליקציה מייד, ללא שמירה במחסנית

שולח ומקבל TCP

שולח TCP

- מחזיק מחסנית שליחה
- היישום השולח חסום עד שיש מקום במחסנית לקלוט את הבתים
- חלון ההזזה מאחסן את הנתונים עד שהם יאושרו על ידי המקבל
- חלון ההזזה גדל וקטן בצורה דינמית
- aw משפיע על הגודל

מקבל TCP

- מחזיק מחסנית קלט עבור היישום
- היישום רואה רק בתים נכונים בסדר
- מפרסם $aw = (bsize - filled)$ כחלון מפורסם עבור חלון הזזה
- עוקב אחרי הבתים שהגיעו בסדר ($ackBytes$)
- מגיב עם $ackBytes$ ו- aw בכל שליחה
- אם $aw=0$, אין מקום במחסנית הקבלה
- נשלח בדרך כלל עם גורם מכפיל

אינטראקציה פשוטה של חלון מפורסם

Illustration of Window Advertisement

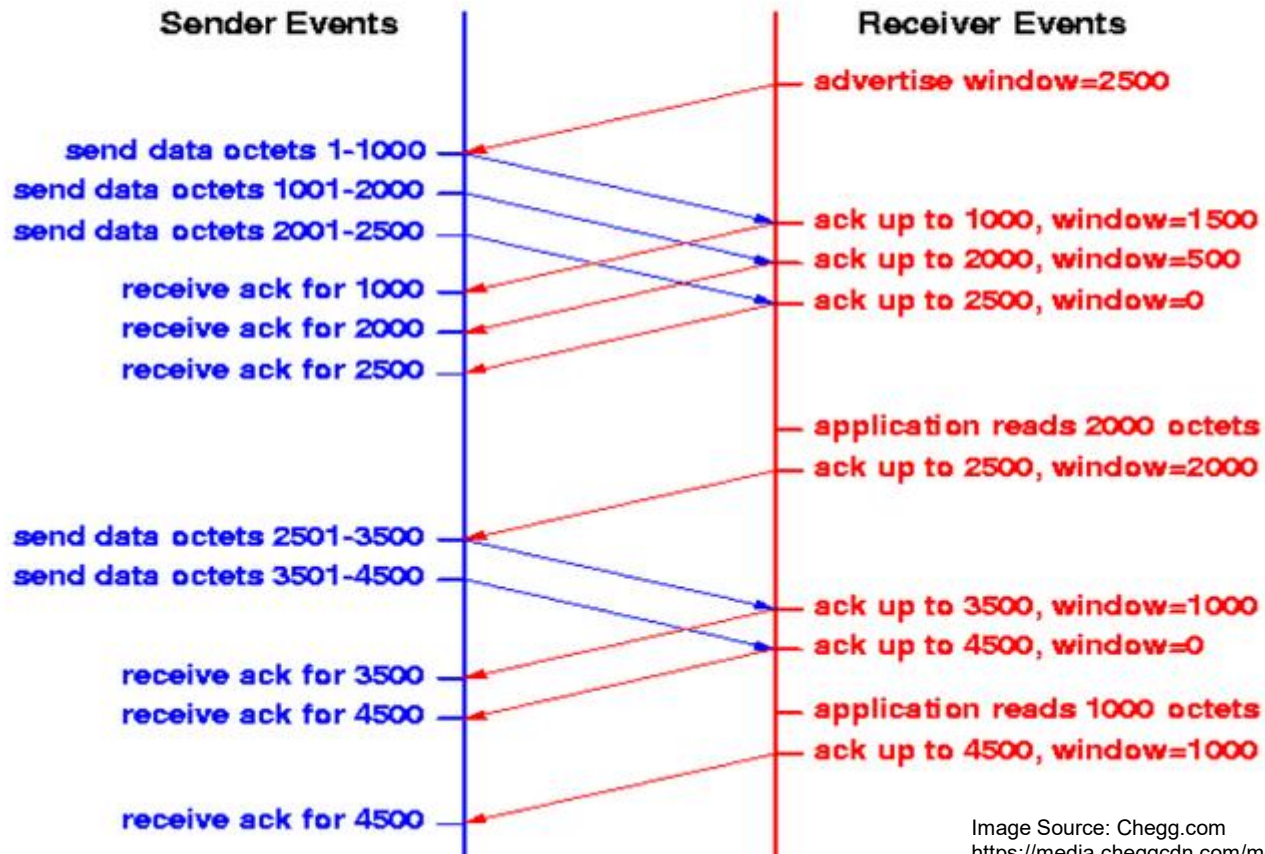


Image Source: Chegg.com
<https://media.cheggcdn.com/media/2c5/2c5cc97c-eb97-4376-b57d-0257161bf650/php3ADfPM.png>

TCP •

– לחיצת יד ויסודות

– בקרת גודש

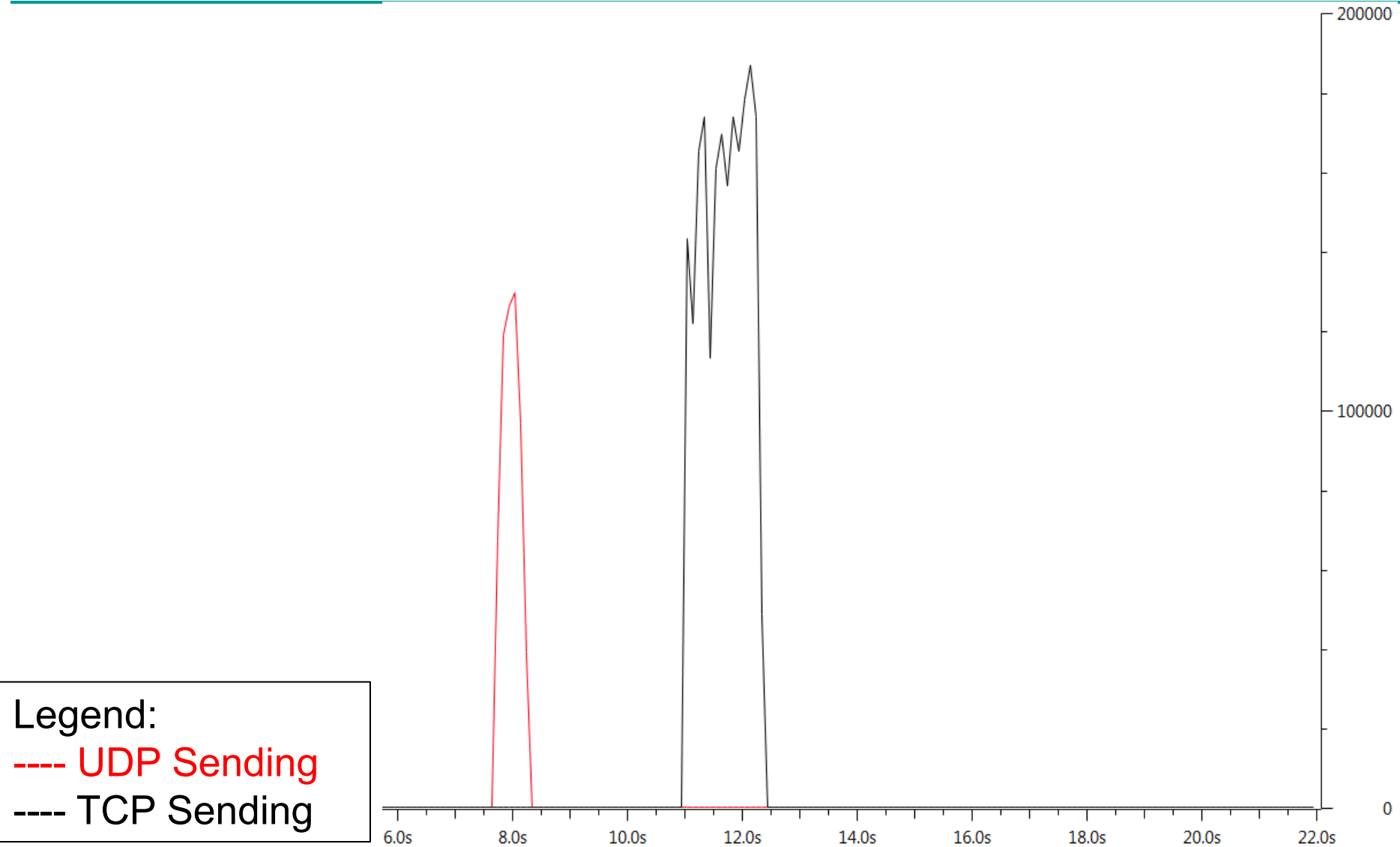
• פרוטוקולי דבק

– ICMP

בקרת הזרימה ובקרת הגודש ב-TCP

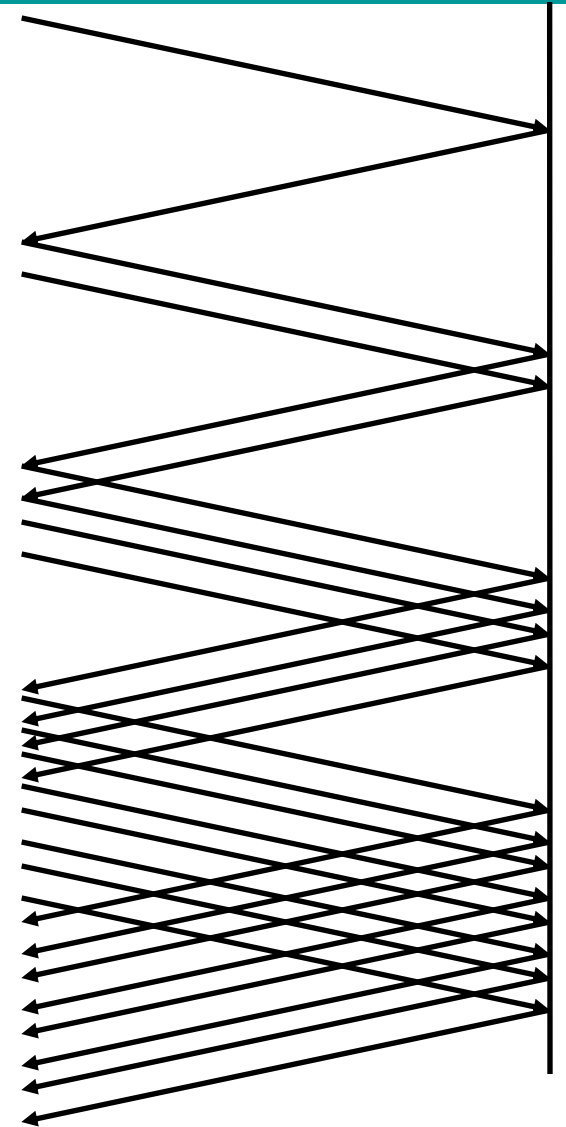
- בקרת הזרימה לעומת בקרת הגודש
 - **בקרת הזרימה** מגינה על **המקבל** מפני הצפה (aw)
 - **בקרת הגודש** מגינה על **הרשת** מפני הצפה.
- בקרת הגודש ב-TCP
 - עלייה תוספתית / ירידה כפולה
 - התחלה איטית
 - שידור חוזר מהיר והתאוששות מהירה

שליחה ב-TCP לעומת שליחה ב-UDP



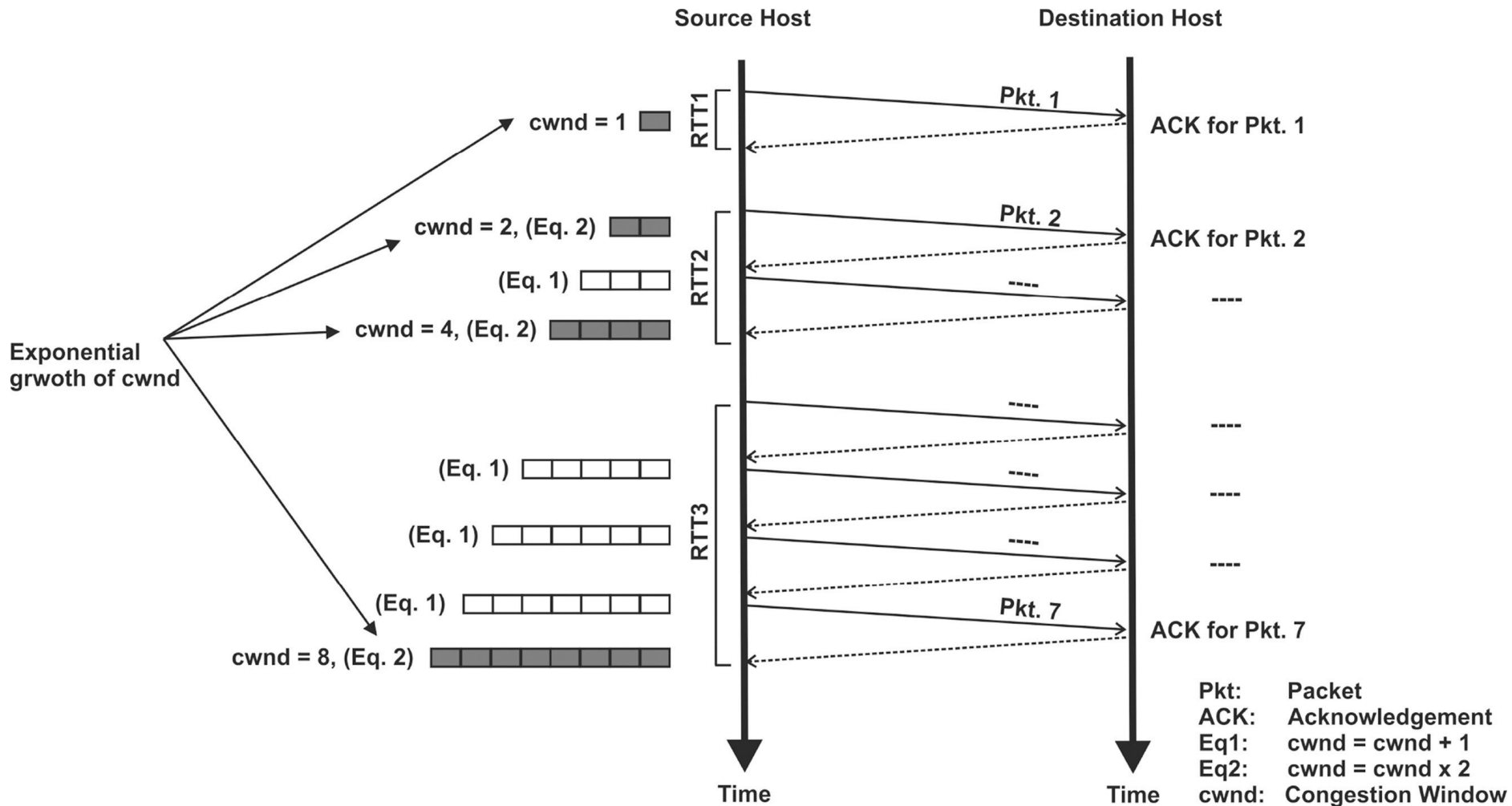
התחלה איטית ב-TCP

- אם נאתחל את cw לגודל החלון המרבי של השולח ונתחיל בשליחת כל החלון בבת אחת, אנחנו עלולים לגרום לעומס על הרשת
- במקום, נתחיל "לאט" על ידי הגדרת $cw=2$ מקטעים (מינימום 576 בתים)
- כאשר מגיע האישור ACK, נגדיל לפי כמות החבילות שאושרו (למעשה cw מכפיל כל RTT)
- המשך כך עד ש-ACKs לא מגיעים או עד שבקרת הזרימה שולטת.
- $SWS = \min(cw, aw)$



התחלה איטית בתמונה

Fig. 2. Exponential growth of congestion window during slow start phase (Wang et al., 2014).



אלגוריתמי בקרת הגודש ב-TCP



- Vegas - שונה מאוד, שרון מעורר לכל מנה



- SACK - שונה מאוד, משתמש באישורי קבלה נקודתיות

– מה הגיע - טווחים של ACKs לא רציפים

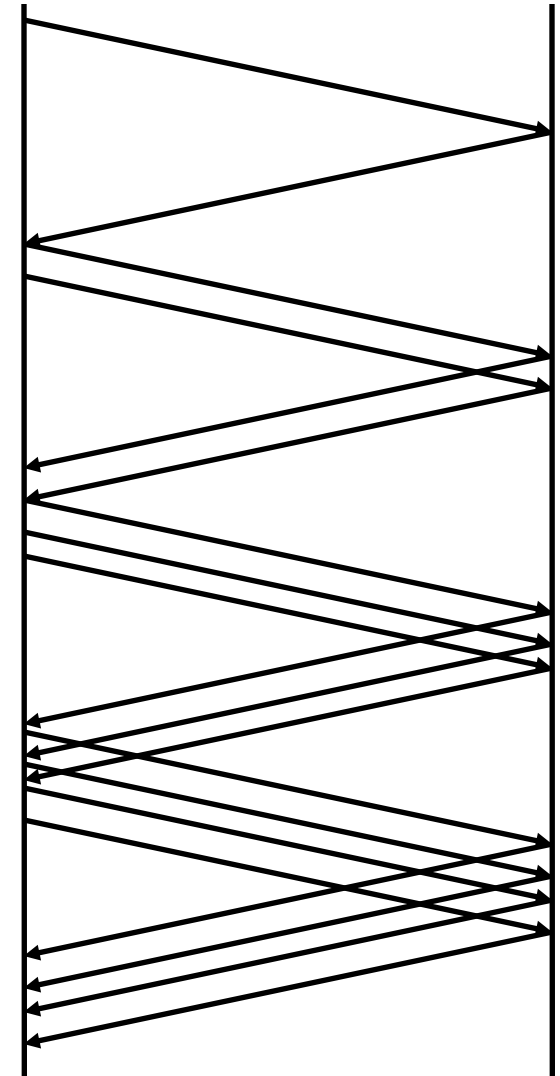
- New Reno - קצת שונה – כיצד להתמודד עם נפילות מרובות והגעה לא לפי הסדר



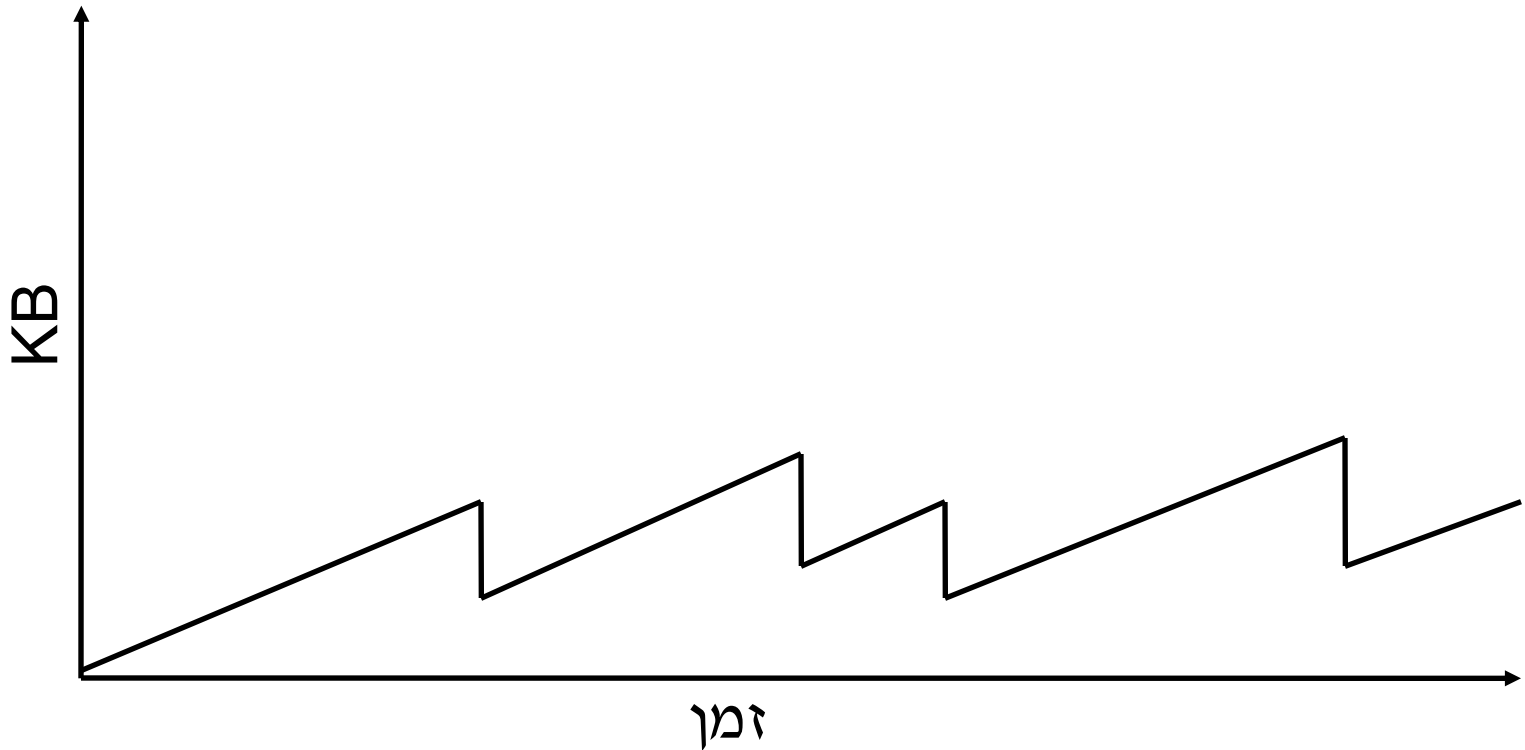


Reno AIMD

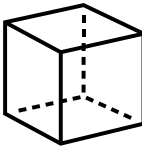
- Congestion Window (cw) **חלון הגודש**
- משתנה שעוזר לשלוט על מספר המקטעים שנשלחו שעוד לא אושרו (עובד יחד עם ה-aw)
- cw גדל באופן **ליניארי** בכל RTT עד שמתחילים לא לקבל אישורי קבלה Acks
- כאשר לא קבלנו ACK עד לפני זמן פג התוקף, cw **יורד בחצי** ($cw = cw \times 0.5$) כדי להוריד את הלחץ על הרשת.
- נקרא "עלייה תוספתית / ירידה בכפל".
- Additive Increase / Multiplicative Decrease



דפוס מסור TCP



TCP CUBIC



$$W(t) = C(t - K)^3 + W_{max} \quad (1)$$

where C is a CUBIC parameter, t is the elapsed time from the last window reduction, and K is the time period that the above function takes to increase W to W_{max} when there is no further loss event and is calculated by using the following equation:

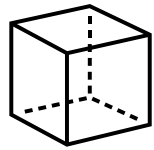
$$K = \sqrt[3]{\frac{W_{max}\beta}{C}} \quad (2)$$

- מיועד לחיבורי רשת עם RTT ארוך והרבה רוחב פס

– תחשבו על ענן

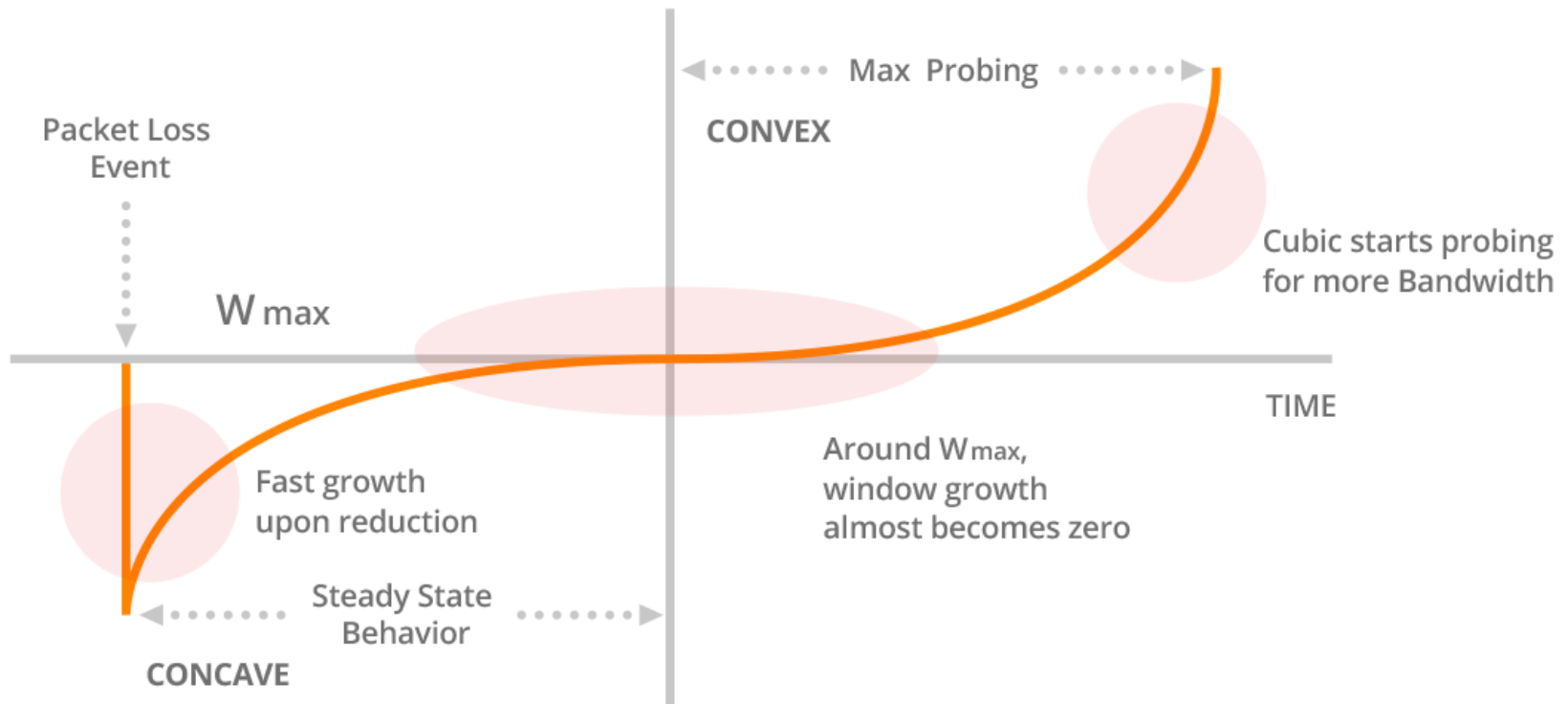
- במקום AIMD שיורד ב-50%, יורדים ב-20%
- גידול החלון מבוסס על:
– $\beta = 0.2, C = 0.4$

Sangtae Ha, Injong Rhee, and Lisong Xu. 2008. CUBIC: a new TCP-friendly high-speed TCP variant. SIGOPS Oper. Syst. Rev. 42, 5 (July 2008), 64–74.
<https://doi.org/10.1145/1400097.1400105>



הגרף של TCP CUBIC

<https://www.noction.com/blog/tcp-transmission-control-protocol-congestion-control>



אלגוריתמים לבקרת הגודש

אלגוריתם ברירת מחדל לבקרת גודש TCP	מערכת הפעלה/מערכת
TCP CUBIC	MacOS
TCP Compound	Microsoft Windows
TCP CUBIC	Linux
TCP Fusion	Sun Solaris
TCP BBR	YouTube (Google)
TCP CUBIC	Android
TCP CUBIC	iOS
TCP BBR	Amazon CloudFront
COPA (?) over QUIC	Facebook

בינה מלאכותית לבקרת הגודש?

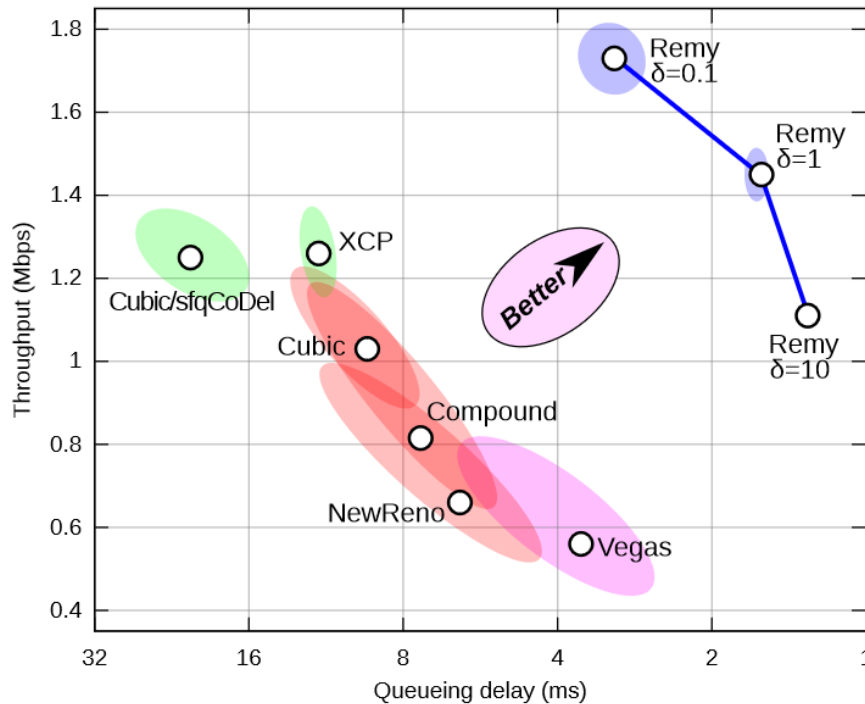


Figure 4: Results for each of the schemes over a 15 Mbps dumb-bell topology with $n = 8$ senders, each alternating between flows of exponentially-distributed byte length (mean 100 kilobytes) and exponentially-distributed off time (mean 0.5 s). Medians and 1- σ ellipses are shown. The blue line represents the efficient frontier, which here is defined entirely by the RemyCCs.

<https://web.mit.edu/remy/TCPexMachina.pdf>

TCP ex Machina: Computer-Generated Congestion Control

by Keith Winstein and Hari Balakrishnan
MIT Computer Science and Artificial Intelligence Laboratory
(SIGCOMM 2013)

An Experimental Study of the Learnability of Congestion Control

by Anirudh Sivaraman, Keith Winstein, Pratiksha Thaker, and Hari Balakrishnan
MIT Computer Science and Artificial Intelligence Laboratory
(SIGCOMM 2014)

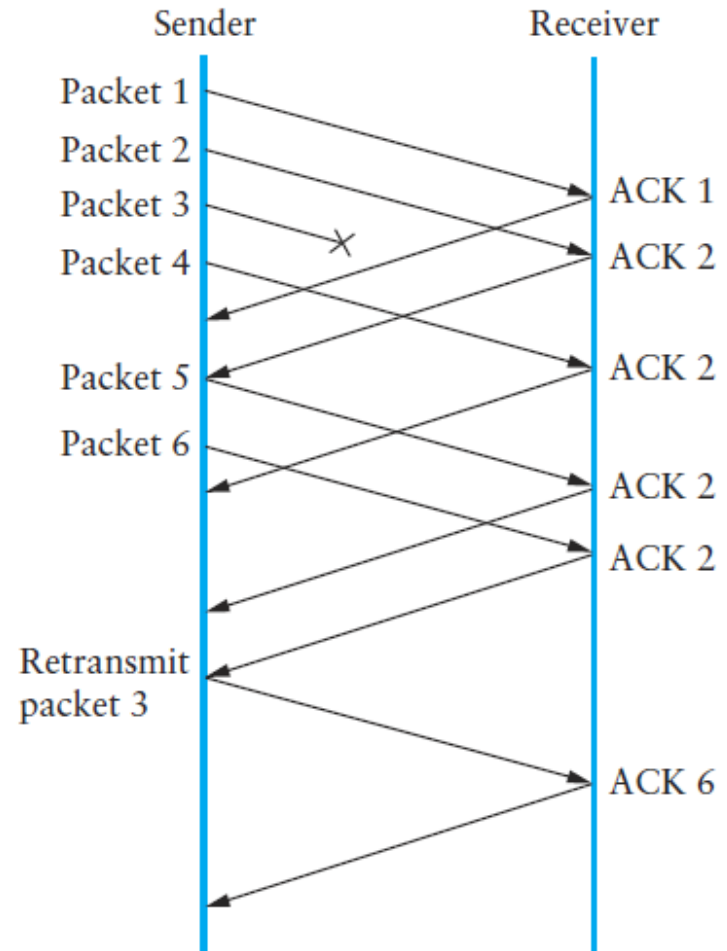
Remy is a computer program that figures out how computers can best cooperate to share a network.

Remy creates end-to-end congestion-control algorithms that plug into the Transmission Control Protocol (TCP). These computer-generated algorithms can achieve **higher performance** and **greater fairness** than the most sophisticated human-designed schemes.

TCP : שליחה חוזרת מהירה

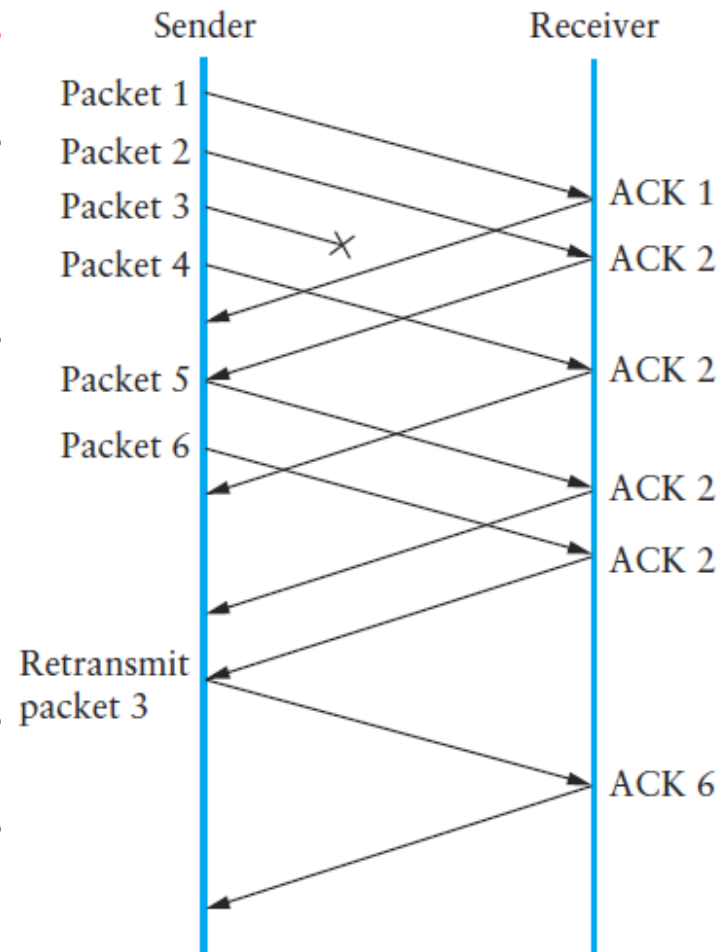
- אובדן מנה בודדת יכול להאט את השליחה של מנות אחרות

- צריכים להגיע לפג תוקף של שרון המעורר וסביר להניח שאם מחכים הרבה החיבור יתנתק או נצטרך להתחיל את ה-cw מחדש



TCP : שליחה חוזרת מהירה

- אלטרנטיבה : שליחה חוזרת מהירה
- שולח ACK-ים מצטברים כפי שראינו בחלון הזזה
- אם מגיעים 3 ACK-ים כפולים, שלח את המקטע הבא מחדש
- ACK כפול פירושו מנות ללא נתונים
- מאשרים משהו שכבר אושר ועם אותו aw
- יכול להגדיל את התפוקה ב-20%
- לא פותר את כל הבעיות (אם SWS קטן או נמצא באמצע התחלה איטית)



TCP •

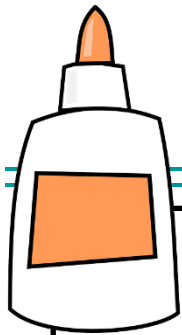
– לחיצת יד ויסודות

– בקרת גודש

• פרוטוקולי דבק

– ICMP

פרוטוקולי "דבק"



צריכים דרך למפות
כתובות ברמה אחת
לכתובות ברמה אחרת.

- למשל: כתובות פיזית (אתרנט)
לכתובת IP

צריכים דרך לטפל
בשגיאות ברשת

צריכים דרך לספק
כתובות שקריאות לבני
אדם

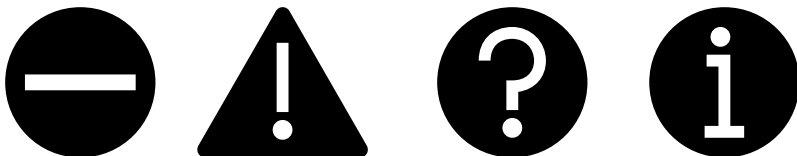
- כדי שוכל לכתוב
במקום www.telhai.ac.il
104.22.27.116

צריכים דרך לחלק
כתובות IP

- מה לגבי מחשבים שנכנסים
ויוצאים?

ICMP: Internet Control Message Protocol

- פרוטוקול להודעות בקרה באינטרנט
- אוסף הודעות שגיאה ובקרה
- נשלח חזרה למקור כאשר הנתב או המארח אינם יכולים לעבד את המנה כראוי
- דוגמאות לשגיאות:
 - לא ניתן להגיע למארח היעד
 - תהליך ההרכבה מחדש נכשל
 - TTL הגיע ל-0
 - Checksum של כותרת ה-IP נכשל
- דוגמה לבקרה:
- הפניה מחדש – מודיע למקור על מסלול טוב יותר



הקלטה לדוגמה של ICMP

No.	Time	Source	Destination	Protocol	Length	Info
155	22.572657000	10.0.0.3	64.233.166.160	ICMP	106	Echo (ping) request id=0x0001, seq=44/11264, ttl=1 (no response found!)
156	22.573156000	10.0.0.138	10.0.0.3	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
157	22.627346000	10.0.0.3	64.233.166.160	ICMP	106	Echo (ping) request id=0x0001, seq=45/11520, ttl=1 (no response found!)
158	22.628191000	10.0.0.138	10.0.0.3	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
159	22.628941000	10.0.0.3	64.233.166.160	ICMP	106	Echo (ping) request id=0x0001, seq=46/11776, ttl=1 (no response found!)

Frame 155: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface 0

Ethernet II, Src: Dell_e6:7f:66 (44:a8:42:e6:7f:66), Dst: D-Link_75:8a:ab (00:22:b0:75:8a:ab)

Internet Protocol Version 4, Src: 10.0.0.3 (10.0.0.3), Dst: 64.233.166.160 (64.233.166.160)

Internet Control Message Protocol

Type: 8 (Echo (ping) request)

Code: 0

Checksum: 0xf7d2 [correct]

Identifier (BE): 1 (0x0001)

Identifier (LE): 256 (0x0100)

Sequence number (BE): 44 (0x002c)

Sequence number (LE): 11264 (0x2c00)

[No response seen]

Data (64 bytes)

• TCP

– לחיצת יד ויסודות

– בקרת גודש

• פרוטוקולי דבק

– ICMP