

רשתות תקשורת מחשבים

תרגיל 10 :

TCP Flow Control
and Congestion Control

תזכורת - RDT

- TCP משתמש ב-Sliding Window.
- שדה Sequence number – מיקום בחלון השליחה.
- שדה Acknowledgment number – מיקום חלון הקבלה.
- הפרוטוקול דו-צדדי, ושני המספרים בלתי תלויים.
- ISN מוחלף בין הצדדים ב-three-way handshake.

Flow Control

הגבלת השולח, כדי לא להציף את המקבל.

חלון הקבלה

- בצד המקבל קיים חוצץ קבלה.
- חלק מהחוצץ מכיל מידע שהתקבל מהרשת.
- בתוך החלק הזה, חלק מהמידע הועבר לשכבה למעלה.
- שאר המידע, שנקרא מהרשת אך טרם הועבר, תופס מקום בחוצץ.
- גודל חלון הקבלה = גודל השטח הפנוי בחוצץ.

הגבלת התקשורת

- כל צד שולח את חלון הקבלה שלו בשדה window.
- נשלח עם כל חבילת TCP.
- כך הוא מסמן כמה מידע הוא מוכן לקבל.
- הצד השני דואג שהפרש בין המידע ששודר למידע שאושר יכנס לגודל החלון של המקבל.
- חלון קטן אצל צד אחד – מאט את השליחה מהצד השני.

חלון בגודל 0

- כשחלון הקבלה בגודל 0, לא נשאר מקום פנוי בחוצץ.
- אישורים (ACK) יכולים להתקבל.
- בכל זאת, הצד השולח ימשיך לשלוח מדי פעם.

Congestion Control

הגבלת השולח, כדי לא להציף את הרשת.

מהו Congestion?

- צפיפות ברשת.
- קורה כאשר צינור אחד או יותר מועברים לצינור צר יותר.
- המקום בחוצץ של הנתב (מתג, או אחר) אוזל, והמידע מושמט.
- בדרך כלל אין בעיה בתוך LAN.

TCP-ב Congestion

- אמנם מקור הבעיה הוא בשכבת הרשת, אבל IP לא מספק אף שירות שיכול לעזור.
- צריך לגלות congestion.
- צריך להגביל את השידור.
- צריך לשלוט בהגבלת השידור.

הגבלת השידור

- הצד השולח מגדיר חלון congestion.
- מכונה cwnd.
- כמות המידע הנשלחת מוגבלת לקטן מבין cwnd שקבע השולח ו-window שקבע המקבל.
- בהמשך הדיון נניח שחלון המקבל הוא גדול מספיק.

גילוי Congestion

- כמות החבילות המשובשות בדרך כלל קרובה מאד ל-0%.
- לכן, אפשר להניח שכל החבילות שאבדו, הושמטו בגלל congestion.
- לא התקבל ACK אחרי timeout.
- אחרי שהתקבל ACK התקבלו עוד שלושה ACK-ים זהים.
- שידור חוזר לאחר מקרה כזה נקרא fast retransmit.

שליטה בהגבלת השידור

- Reno
 - RFC 2001
- NewReno
 - RFC 2581
 - Windows XP
 - Linux 2.6.?
- BIC, CUBIC
 - Internet draft
 - Default on current Linux
- CTCP
 - Internet draft
 - Current Windows
- ה-congestion control הוא הרחבה של TCP.
- קיימות שיטות שונות.
 - בהתאם למאפייני הערוץ והרשת.
- העקרונות שנוציג בהמשך הם כללים, אך שילוב שלהם מתאר את שיטת Reno/NewReno.
- זוהי השיטה התקנית היחידה, והאגרסיבית ביותר המותרת.

Slow Start

- **נבטא את cwnd ביחידות של MSS.**
- **cwnd מתחיל בערך כלשהו.**
- **1 לפי RFC 2001 (Reno).**
- **עד 2 לפי RFC 2581 (NewReno).**
- **תלוי ב-MSS לפי RFC 3390.**
- **על כל ACK שמתקבל, cwnd גדל ב-1.**
- **קצב גידול cwnd הוא אקספוננציאלי ביחס ל-RTT.**
- **ממשיכים כך עד להגעה לסף ssthresh או לגילוי congestion.**
- **בהתחלה הסף גבוה.**
- **שלב ה-slow start פעיל כאשר $cwnd < ssthresh$.**

Congestion Avoidance

- **שלב זה מתרחש כאשר $.\text{cwnd} > \text{ssthresh}$**
- **במקרה של שוויון שני השלבים מותרים.**
- **בשלב זה cwnd גדל ב-1 כל RTT .**
- **קירוב על ידי הנוסחה $\text{cwnd} += 1/\text{cwnd}$ לכל ACK תקין.**
- **קצב גידול cwnd כעת הוא לינארי.**
- **ממשיך עד לגילוי congestion .**

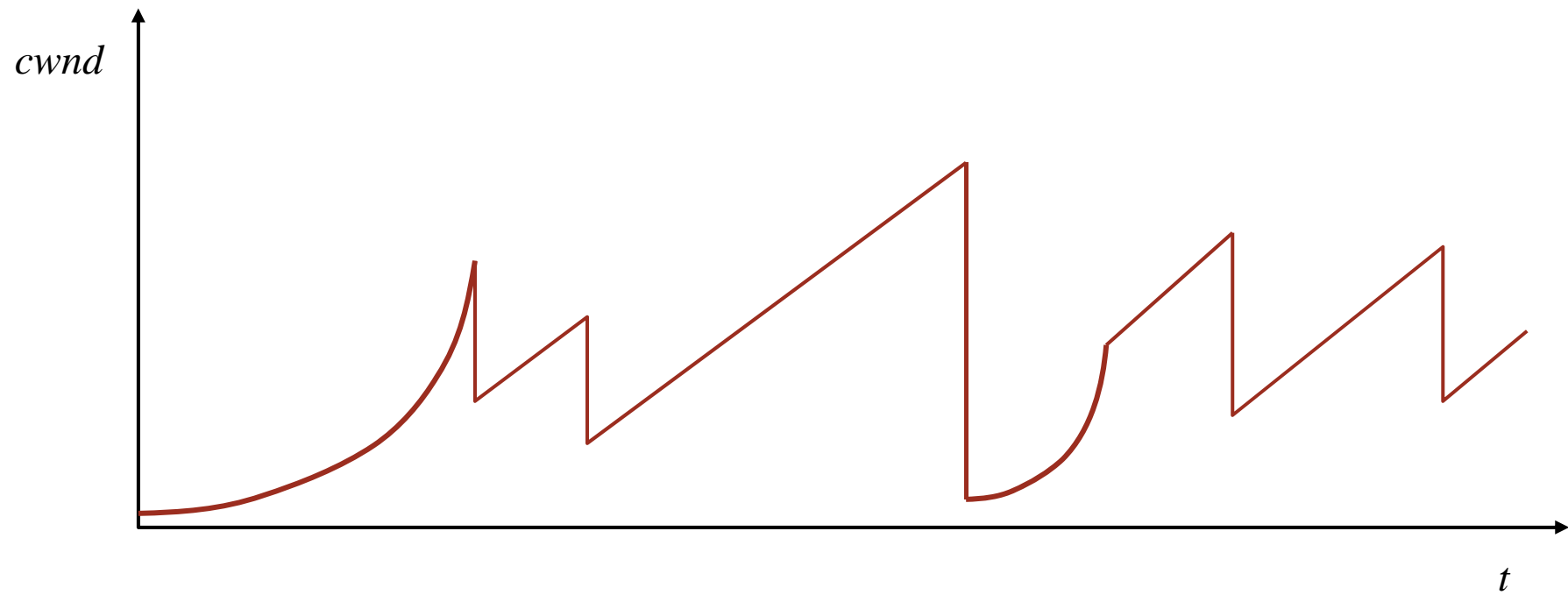
במקרה של Congestion (Timeout)

- $ssthreshold$ מקבל מחצית הערך הנוכחי של $cwnd$, ולכל הפחות
2.
- $cwnd$ מקבל ערך של לכל היותר 1.
 - ללא תלות בערך התחילי.
- לכן, חייבים לחזור לשלב ה- $slow\ start$.

במקרה של Fast Retransmit

- Fast Recovery
- שנה את ssthresh כמקודם.
- $cwnd$ הוא ssthresh ועוד 1 לכל ACK כפול שהתקבל (לפחות 3 שגרמו ל-fast retransmit).
- כשמגיע ACK חדש $cwnd = ssthresh$.
- אין מעבר לשלב slow start.

גודל cwnd לאורך זמן



שיטות אחרות

- כדי לגלות מתי יוצר congestion, Reno/NewReno מייצרות congestion.
- ברגע שנוצר congestion, השימוש בערוץ יורד.
- שיטות אחרות מנסות למנוע את הירידה הזו, על ידי הימנעות מ-congestion.

הגינות

- Congestion control מבטיח הגינות כלפי חיבורי TCP.
- אבל -
- אין אף דרך להכריח שימוש ב-congestion control (למרות שהתקן מחייב).
- ההגינות היא כלפי חיבור בודד ולא כלפי תחנה או אפליקציה (אין מניעה שאפליקציה אחת תשתמש במספר חיבורים לאותו יעד).
- שיטה אגרסיבית יותר (NewReno) תקבל יותר משיטות אגרסיביות פחות.
- השכבה מתחת יכולה לשרת גם פרוטוקולים אחרים (בפרט, UDP).