רשתות תקשורת – תרגיל תיאורטי 4

TCP

1. הגודל המקסימלי הוא 232 בתים - גודל זה מבטיח לכל סגמנט שנשלח מזהה (seq no.) ייחודי.
2. נחשב באופן הבא:

* כמות החבילות (בגודל מלא, כלומר, ניצול מלא של ה-MSS המותר) שנצטרך לשלוח:
* גודל החבילה האחרונה:
* הגודל של כל חבילה בתוספת ה-header (מלבד האחרונה) הוא 1514 + 90 = 1604 bytes.
* הגודל של החבילה האחרונה בתוספת ה-header: 620+90 = 710 bytes.
* סך כל כמות ה**בתים** שנעביר (כולל headers): 2836834\*1604 + 710 = 4550282446
* זמן שליחת ההודעה:

נצילות השליחה בערוץ היא:

Tcp – congestion control

1. המצב האפשרי היחיד הוא Slow start – לאחר כל ACK החלון גדל בדיוק ב-MSS (2000 בתים).

נסביר מדוע לא ייתכן שאנחנו במצבים אחרים:

* Fast recovery – נניח בשלילה שאנחנו ב-FR, אז יוצא שלכל המאוחר לפני שלב A ביצענו עדכון של ssthresh למחצית מגודל החלון ואת הגדול החלון ל-ssthresh + 3\*MSS (הכוונה כאן היא ל-ssthresh המעודכן). מכיוון ש-3\*MSS = 6000 יוצא כי גודל החלון לפני השינוי היה 0 (וכך גם גודל ssthresh החדש), וזה לא הגיוני – גודל חלון כזה לא מאפשר לשלוח מידע כלל.
* Congestion Avoidance – לו היינו ב-CA אז היינו מקבלים בשורה B גודל חלון של 6000 + 2000/3 ≈ 6667 (באופן כללי, ב-CA על כל ACK מתקבלת תוספת שהיא שבר של MSS ולא MSS שלם, בניגוד לנתון בשאלה).

1. אין מספיק נתונים ולכן לא ניתן לקבוע ערך ssthresh מקסימלי. כן ניתן לקבוע כי הוא לפחות 10000 – אם היה פחות מזה היינו רואים ירידה בגודל החלון.
2. מכיוון שאנחנו ב-SS, לאחר 3 dupACKs (הכוונה ל-3 dupACKs שהגיעו לאחר מצב C) נכנסים למצב FR ולכן גודל החלון הופך ל- congWin/2 + 3\*MMS, כלומר, מחכים במשך 3 dupACKs מבלי לשנות את גודל החלון, ולאחר מכן מעדכנים כך: **congWin** = 10000/2 + 3\*2000 = **11000**. למען השלמות נציין כי לאחר קבלת ה-ACK על החבילה האבודה (בהנחה שאכן יגיע לפני timeout) גודל החלון יעודכן לבסוף למחצית מגודלו לפני קבלת ה-dupACKs, כלומר, **congWin = 5000**.