

שנקר ביי"ס גבוה להנדסה ולעישוב
החוג להנדסת תוכנה
קורס 'תקשורת ותקשוב 1' (קורס מספר 3503834)
תשע"ג – סמסטר א'

מרצה: יצחק נודלר

תרגיל בית מס' 4

מועד הגשה: 24.01.2013

הנחיות:

יש לענות על השאלות בצורה עניינית, תמציתית וברורה. נא להימנע מ-"סיפורים".
על כל סטודנט לפתור את התרגיל לבדו.
יש לצרף לכל תרגיל בצורה ברורה ובמקום בולט את שם מגיש התרגיל וכן את מס' תעודת הזהות.
את התרגיל יש להגיש דרך אתר הקורס המנוהל ב- moodle.

שאלה מס' 1

נתון משתמש קצה U אשר עובד מול מחשב C.
המשתמש מפעיל (מריץ) ב-C את הדפדפן ומקליד בשורת הכתובת של הדפדפן את ה-URL הבא:
www.google.com
נתון כי בדפדפן ממומש פרוטוקול HTTP גרסא, וכי הפרמטרים של הדפדפן מוגדרים כך
שהדפדפן עובד במצב של pipelining (= יכול לשול מספר הודעות בקשה בזו אחר זו, במידה ואכן
יש לו הודעות כאלו לשלוח, מבלי שעדיין התקבלו עבורן הודעות התשובה).
נתון כי דף הבית של google מורכב מ-5 קבצים בסה"כ: index.html, fig1.gif, fig2.gif, fig3.gif, fig4.gif.
נסמן ב-S את ה-web server של google.

א. בהינתן כי ה-MSS של TCP-C הוא 500 בתים ובהינתן כי גודל הודעת http request הראשונה
הוא 1300 בתים, האם ניתן לדעת במדויק כמה סגמנטים ייצור TCP-C לצורך העברת הודעת ה-
http request הראשונה?

ב. האם בהכרח ה-MSS של TCP-S שווה בגודלו ל-MSS של TCP-C?

ג. תחת נתוני השאלה, תארו תסריט שבו TCP-C יוצר סגמנט אשר מכיל data ששייכת ל-2
הודעות http request שונות.

ד. תחת נתוני השאלה, תארו תסריט שבו TCP-S שולח סגמנט אשר בכוותרת שלו $A=1$ והוא מכיל
MSS בתים בשדה data.

ה. תחת נתוני השאלה, תארו תסריט שבו ACK segment שמגיע מ-TCP-S גורם להוצאת 3 סגמנטים מחלון השליחה של TCP-C

ו. תארו תסריט שבו חלון השליחה של TCP-C מלא לחלוטין וכמו כן יש data ב-outBuf שלו.

ז. תארו תסריט שלפיו 5 סגמנטים מגיעים בזה אחר זה ל-TCP-C ומתוכם 2 סגמנטים ממוקמים בחלון הקבלה והסגמנט האחרון מבין ה-5 גורם להוצאת 2 סגמנטים אלו מחלון הקבלה והעברת התוכן שלהם (ה-data) ל-inBuf.

ח. תארו תסריט אשר בו גם חלון השליחה וגם חלון הקבלה של TCP-S מלאים לחלוטין.

ט. תארו תסריט אשר לפיו חלון השליחה של TCP-C מלא לחלוטין וכן ה-outBuf שלו מלא לחלוטין ולמרות זאת סגמנטים לא מגיעים ל-TCP-S.

שאלה מס' 2

בשאלה שלהלן הסימונים C ו-S יסמנו :

C = צד הלקוח של פרוטוקול שכבה 5 (לדוגמא, http client)

S = צד השרת של פרוטוקול שכבה 5 (לדוגמא, http server)

נתון כי C מקים קישור TCP עם S.

C כותב לסוקט סך כולל של 1000 בתים (בין ה- connect שחוזר בהצלחה לבין ה-close).

א. האם ניתן לדעת, בהסתמך על הנתונים לעיל, כמה פעולות write ביצע C ?

ב. איך C יודע שבתים מסוימים שכתב לסוקט באמצעות write אכן הגיעו ל-S ?

ג. כמה סגמנטים TCP יצר בסה"כ כדי להעביר את כל 1000 הבתים ?

ד. כמה פעמים TCP בצע retransmit ?

שאלה מס' 3

- בספרות שעוסקת בפרוטוקול TCP, מתואר מנגנון האישורים של TCP (ACKs) כמנגנון מסוג *cumulative ACK*.
- משמעות המושג הזה : אם מגיע ACK segment עם $ACK\ number = X$ אזי ה-ACK הזה מאשר את קבלתם של כל ה- DATA segments עד (ולא כולל) לסגמנט שמספרו הסידורי X.
- א. הסבירו מדוע אכן, לפי ההגדרה הנ"ל, מנגנון האישורים של TCP הוא מנגנון קומולטיבי.
- ב. תארו תסריט שבו מגיע ACK segment והוא 'מאשר' הגעתם התקינה של ארבעה DATA segments (גם אם מנגנון ה- delayed ACK שמתואר בהמשך 'מושבת'. ז"א, נשלח מיד ACK segment על כל DATA segment שמגיע)

שאלה מס' 4

- אחד ממנגנוני האופטימיזציה של TCP הוא מנגנון שנקרא delayed ACK.
- עקרון המנגנון הזה הוא כדלקמן :
- כאשר מתקבל DATA segment המקבל 'לא מזדרז' לשלוח ACK segment, אלא משהה את שליחת ה- ACK segment למשך פרק זמן של עד 500 מילישניות. אם תוך פרק זמן זה מגיע DATA segment נוסף, ישלח מיד ACK segment.
- אם תוך פרק זמן זה של 500 מילישניות לא מגיע עוד DATA segment, ישלח ACK segment (עבור הסגמנט שהגיע 500 מילישניות קודם לכן).
- א. מה 'ההגיון' מאחורי מנגנון זה ?
- ב. באלו נסיבות מנגנון זה עשוי להיות בעייתי ?