

## פתרונות חלקיים רשתות תקשורת – טופס 1

### פתרון שאלה 1

החסם המדוייק הוא קוטר הגרף פחות 1. מאחר ובשאלה אין נתונים על הקוטר, אפשר גם לומר שהחסם הוא V-2 (מפני שקוטר הגרף הוא לכל היותר V-1).

הערה: התשובה הזו מניחה, בדומה למה שראיתם בהרצאות, שבתחילת הריצה כל הצמתים מכירים כבר את השכנים המיידיים שלהם. אם נניח שבתחילת הריצה כל הטבלאות ריקות, אזי התשובה שנקבל תהיה קוטר הגרף (או V-1). בבחינה קיבלנו את שתי האפשרויות.

זה אכן חסם עליון: בהנתן שני צמתים בגרף  $s, t$ , נתבונן במק"ב ביניהם  $s, a_1, a_2, \dots, a(n), t$ .

$a(n)$  מכיר את  $t$  והמרחק אליו כבר בקונפיגורציה התחילית. כעבור טיק אחד גם  $a(n-1)$  מכיר את  $t$ , כעבור שני טיקים גם  $a(n-2)$  מכיר את המרחק ל  $t$  על פני המק"ב שלנו וכן הלאה עד שכעבור V-2 צעדים גם  $s$  מכיר את המרחק ל  $t$  על המק"ב הזה. לכן, כעבור V-2 צעדים  $s$  ידע להגיע ל  $t$  דרך קשת שמהווה חלק ממק"ב ולכן לא ישנה יותר את הקשת שדרכה הוא בוחר להגיע ל  $t$ .

החסם אכן מינימלי: ניקח פשוט גרף לינארי (שרשרת) של  $V$  צמתים. שני הקצוות אינם מתכנסים למצבם הסופי לפני V-2 צעדים.

### פתרון שאלה 9

א  $a$ ) נניח ש  $R$  יוצא מייד אחרי ה  $ACK$  האחרון. אם  $S$  לא קיבל את ה  $ACK$  האחרון (למשל כי הפקטה אבדה ברשת) הוא לא ידע שכל המידע הגיע בצורה תקינה ל  $R$ . במצב כזה  $S$  יעבור timeout וינסה לשלוח שוב את ה  $DATA$  האחרון. אבל  $R$  כבר סגר את הצד שלו ולכן  $S$  לא יוכל כבר לקבל את האישור על ה  $DATA$  הזה. כלומר, העברת המידע הסתיימה בהצלחה, אבל ל  $S$  אין דרך לדעת את זה בוודאות.

אם לעומת זאת  $R$  ימתין מספיק זמן אחרי ה  $ACK$  האחרון הוא יתפוס את ה  $DATA$  האחרון שנשלח שוב מצד  $S$  וישלח שוב את ה  $ACK$ .

ב) כי העברת הקובץ ל  $R$  בכל מקרה מסתיימת בהצלחה. הבעיה היא רק ש  $S$  לא יכול לדעת אם ההעברה הצליחה.

ב) הפרוטוקול כבר לא היה תקין כי אם ה  $DATA$  האחרון אובד ברשת, בשלב כלשהו תגיע מצד  $R$  פקטת  $ACK$  שתסמן שיש צורך ב  $retransmission$ , אבל  $S$  כבר יצא ולא יוכל לקבל אותה כי סיים את הצד שלו של הקישור.

ג) התהליך:

- $\frac{8 \cdot 560}{10^9} = 4.48 \mu s$ : 48+512 בגודל מידע פקטת מידע
- $\frac{12000 \cdot 10^3}{3 \cdot 10^8} = 0.04 s$ : צריכה לעבור לצד השני: 0.04s
- $\frac{8 \cdot 48}{10^9} = 0.384 \mu s$ : 48 בגודל ACK שידור
- $\frac{12000 \cdot 10^3}{3 \cdot 10^8} = 0.04 s$ : צריך לעבור לצד השני: 0.04s
- $\frac{8 \cdot 536}{10^9} = 4.288 \mu s$ : 48+488 בגודל מידע פקטת מידע
- $\frac{12000 \cdot 10^3}{3 \cdot 10^8} = 0.04 s$ : צריכה לעבור לצד השני: 0.04s
- $\frac{8 \cdot 48}{10^9} = 0.384 \mu s$ : 48 בגודל ACK שידור
- $\frac{12000 \cdot 10^3}{3 \cdot 10^8} = 0.04 s$ : צריך לעבור לצד השני: 0.04s

סה"כ 0.160009536 שניות

ד) השולח שלח מידע בפועל רק במשך  $8.768 \mu s$ . הנצילות היא לפיכך בערך 0.0055%.

ה) pipelining אומר שאפשר להעביר כמה הודעות במקביל מבלי להמתין ל – ACK. זה ישפר את הנצילות כי רוב הזמן בתרחיש המתואר מתבזבז על ההמתנה ל – ack ול – data שיעברו מצד לצד. זה מקרה קלאסי של high bandwidth delay product.

ו) מאחר והזיכרון של R מוגבל, עדיף להשתמש ב – GBN שדורש buffer בגודל פקטה אחת בצד המקבל שמיועד רק לפקטה הבאה, ו – counter שאומר מהו מספר הפקטה הבא שמצפים לו. SR לעומת זאת, דורש buffers עבור פקטות שהגיעו שלא לפי הסדר. (SR גם תובעני יותר בזכרון כי דורש יותר state variables כמו טיימר נפרד לכל פקטה שנשלחה).