**פתרון תרגיל בית 1**

B

R4

R3

R2

R1

A

Tp

Tp

Tt

Tt

Td

Td

Tp

Tp

Tt

Tt

Td

Td

Tp

Tp

Tt

Tt

Td

Td

Td

Td

Tt

Tt

Tt

Tp

Tp

Tp

[Mbps]

B

R4

R3

R2

R1

A

Tp

Tp

Tt

Tt

Td

Td

Tp

**תחנה A מסיימת לשלוח 5 חבילות**

Tp

Tt

Tt

Td

Td

Tp

Tp

Tt

Tt

Td

Td

**תחנה B מקבלת את הביט הראשון**

Tt

Tt

Tt

Tt

Td

Td

Tt

Tt

Tt

Tp

Tp

Tp

א. ניתוב חבילות: כדי לשלוח את כל המידע צריך F/L חבילות.

המקור משדר בכל פעם קבוצה של Q חבילות יחד. זמן השידור של הקבוצה הראשונה (Q החבילות הראשונות) הוא (L+H)/R, כי לכל חבילה צריך להוסיף רישא בגודל H.

**זמן העיבוד**. מאחר שהנתב מטפל בכל פעם ב-Q חבילות במקביל, ומאידך – לא יכול לשדר באותו זמן, נדרש זמן עיבוד כולל של D כפול מס' החבילות חלקי Q, כלומר: *DF/(LQ)*.

**זמן השידור הכולל.** יש שני מקטעים (מהמקור אל הנתב; ומהנתב אל היעד). אבל המקור יכול לשלוח את כל קבוצות החבילות, מהשנייה והלאה, במקביל לזמן שבו הנתב משדר את קבוצות החבילות שהוא כבר קיבל, מהראשונה והלאה, אל היעד. לפיכך, לשם קבלת זמן השידור הכולל, יש לכפול את זמן השידור של קבוצה אחת במספר הקבוצות + 1. כלומר: [F/(LQ) + 1] [(L+H)/R]

לפיכך, הזמן הכולל הוא: [F/(LQ) + 1] [(L+H)/R] + *DF/(LQ).*

ב. יש להקים את הרשת (לוקח זמן *C*), ולאחר מכן לשדר את כל המידע ברצף, ללא תקורה של רישות (*headers*) – אך על ערוץ יחיד. מכיוון שהמעגל מתפקד כ"צינור" יחיד, מספיק זמן שידור יחיד של הקובץ כולו, ללא כל השהייה של שידור מחדש ו/או של עיבוד בנתב בדרך.

לפיכך, הזמן הכולל הוא: *C +F/R*.

ג. ניתן לראות אריתמטית, כי הגדלה של: *D,H* תגדיל את העדיפות של *circ. switch.*. ניתן להבין זאת אינטואיטיבית, שכן ככל שלכל חבילה יש תקורה של רישא יותר גדולה או זמן עיבוד ארוך יותר בנתב, יורדת המוטיבציה לשימוש בחבילות.

ניתן לראות אריתמטית, כי הגדלה של *L,C* תגדיל את העדיפות של "מיתוג חבילות". ניתן להבין זאת אינטואיטיבית, שכן ככל שמגדילים את כמות המידע בחבילה, התקורה היחסית של הרישא של החבילה הופכת קטנה יותר. כמו כן, ככל שזמן הקמת המעגל ב-*circ.switch* גדול יותר, שיטה זו הופכת פחות יעילה – מה שמגדיל את העדיפות ל"מיתוג חבילות".

ההשפעה של *R,F* מורכבת יותר, ותלויה ביחסי הפרמטרים האחרים. באופן כללי, שינוי גודל הקובץ ו/או קצב השידור בערוץ לא אמור לשנות דרמטית את העדיפות שיש לאחת השיטות על פני רעותה.

**פתרון תרגיל בית 2**

**1.**

א. עבור כל אחד מהסעיפים תחילה יש 2RTT לחיבור + לבקשה הראשונה(GET) + זמן הורדת הדף. ההבדל בין הסעיפים הוא באופן שבו התמונות יורדות.

1. *Ta* = 2RTT + + *n*\*(2RTT +).
2. *Tb*= 2RTT + + *n*\*(RTT +).
3. *Tc* = 2RTT + + RTT + *n*\*().

ב.

A. גדול יותר. הסבר: במקרה הראשון, שבו הזמן הלו"ש גדול מאוד ביחס לזמן השידור, השימוש בקשר עקבי עם צינור הרבה יותר יעיל; אם גם *n* גדול מאוד, זמן ההורדה בשיטה זו קטן פי *2n/3* בערך מאשר בשיטה הנאיבית, של קשר לא עקבי.

B. קטן יותר. הסבר: כאשר זמן ההתפשטות הלו"ש זניח ביחס לזמן שידור הקובץ, התועלת היחסית שבשימוש בקשר עקבי עם צינור היא זניחה, כי אין תקורה משמעותית לבקשה חדשה של הלקוח מהשרת עבור כל אובייקט נוסף: מה שחשוב הוא זמן שידור הקובץ, שהוא זהה בשתי השיטות.

**2.**

א.

**שרת-לקוח:**

T = MAX{ 1G \* 30 / 100M , 1G/4M } = MAX { 300 sec, 250 sec} = 300sec

**P2P:**

T = MAX{ 1G \* 30 / (10M + 50M + 50M + 20M ), 1G/4M , 1G/10M } = MAX {231 sec, 250 sec , 100sec } = 250 sec

ב.

**שרת-לקוח:** צוואר הבקבוק הוא קצב ההעלאה של השרת. שיפור הפרמטר הנ"ל בלבד יקצר את זמן ההורדה הכולל ל-250 שניות – זמן הנובע מקצב ההורדה של הלקוחות האיטיים ביותר.

**P2P**: צוואר הבקבוק הוא קצב ההורדה של הלקוחות האיטי ביותר. שיפור מהירות ההורדה של לקוחות אלה תקצר את זמן ההורדה ל-230sec – זמן הנובע מקצבי העלאת הנתונים ברשת.

**4.**

1. המשתמש ייגש רק אל זיכרון מס' 0, שהוא הזול ביותר. הוא ישלם עלות גישה מינימלית של 1, אך לא ימצא את הקובץ הדרוש.
2. המשתמש ייגש אל כל הזיכרונות 0 עד n. הוא ימצא את הקובץ, אך ישלם על כך בעלות גישה כוללת גבוהה של 2n+1.
3. המשתמש ייגש אל אחד הזכרונות שבהם הקובץ נמצא באמת, ויימצא את הקובץ המבוקש, בעלות נמוכה של 2.

**5.**

א. עלות הגישה דרך המטמון היא: *C* לכל גישה; ובנוסף, *M* עבור כל נתון שלא נמצא במטמון. אם נסמן את ה-*hit rate* ב-*h*, תוחלת העלות של גישה דרך המטמון היא *C+(1-h)M*.

עלות גישה ישירות לשרת המרוחק היא *M*.

כדי שיהיה יעיל יותר לחפש קודם כל במטמון נדרוש: *C+(1-h)M < M*, שניתן לבטא גם כ:

*h > C/M.*

ב. מסנן בלום הוא מבנה נתונים, שעונה על שאילתות של שייכות לקבוצה. בהקשר של מערכת עם זיכרון מטמון, מסנן בלום יכול לענות במהירות גבוהה וברמת דיוק גבוהה מאוד על השאלה האם נתון מסויים נמצא במטמון, כך שקודם כל נשאל את מסנן הבלום, ורק על פי תשובתו נחליט האם לגשת קודם לזיכרון המטמון, או שמא לגשת ישירות לשרת המרוחק.

ג. ללא ההנחה הנ"ל, ניאלץ לגשת לשרת המרוחק בכל מקרה, כדי לוודא שהנתון שנמצא במטמון הוא המעודכן ביותר – דבר שמשנה את החישוב, ובאופן כללי מקטין את התועלת שבשימוש בזיכרון מטמון.

**6.**

א. הגדלת *RTT* תקטין את יעילות השימוש ב-*Cond. Get*, כי היא גורמת לכך שהחלק היחסי של זמן ההתפשטות בהבאת הנתון (שאותו יש לשלם בכל מקרה – הן ב-*Cond. Get* והן ב-*Get* רגיל)נהיה משמעותי יותר.

ב. הגדלת הקובץ שצריך להעביר מגדילה את יעילות השימוש ב-*Cond Get* יחסית ל-*Get* רגיל, כי זה הופך את זמן השידור של הקובץ – שנחסך ע"י *Cond* Get במקרה שהקובץ לא השתנה מאז שהובא למטמון - למשמעותי יותר.

ג. ככל שקצב השינויים של הקבצים **איטי** יותר, קיים סיכוי גדול יותר, שהקובץ לא השתנה מאז שהורד אל המטמון, ואז התועלת שבשימוש ב-*Cond Get*, שחוסך את הורדת הקובץ מחדש במקרים אלה, תהיה גדולה יותר.

**פתרון תרגיל בית 3**

**3.**

א.

1+2+ … + N = N (1 + ½ + ¼ + … + 1/N) = 2N-1

ב.

1 + 2 + … + N/2 = N-1

גודל החלון בפעם ה-m הוא 2^m. במשלוח האחרון מתקיים :

2^m = N/2 🡪 m = lg(N) – 1

כאשר לאחר כל "מחזור" עם משלוח ממתינים לקבלת החיוויים.

הספירה מתחילה מ- m=0, כי בפעם הראשונה גודל החלון הוא 1 = 2^0, ומסתיימת ב-lg(N)-1. לפיכך, יש בסה"כ lg(N)-1 המתנות של , כלומר: הזמן המבוקש הוא lg(N)-1] ].

ג. מס' הסגמנטים שנשלחים סה"כ הוא:

1 + 2 + 4 + … + N + (N+1) + (N+2) + … + 2N = ?

1 + 2 + 4 + … + N/2 = N-1

1 + 2 + 3 + 4 + … + 2N = 2N (2N+1) / 2 = N(2N+1) = 2N^2 + N

1 + 2 + 3 + 4 + … + (N-1) = N(N-1)/2 = ½ N^2 - ½ N

* N + (N+1) + (N+2) + … + 2N = 1.5N^2 + 1.5 N
* 1 + 2 + 4 + … + N + (N+1) + (N+2) + … + 2N = **1.5N^2 + 2.5N -1**

זמן המשלוח עד שגודל החלון הוא N/2 הוא lg(N)-1.

זמן המשלוח עד שגודל החלון הוא N הוא lg(N).

זמן המשלוח לאחר מכן הוא כמס' האיברים בסדרה N, (N+1), (N+2), …, 2N, פחות אחד (לא סופרים את ההמתנה לאחר המשלוח האחרון) - כלומר, N.

לפיכך, זמן המשלוח הכולל הוא N + lg(N).

ד. בסעיף ב' נשלח בתחילה N/4 סגמנטים; אחרי עוד N/2 סגמנטים; ואחרי עוד את היתר. בסה"כ

המתנו 2.

הגדלת החלון ההתחלתי הקטינה את זמן המשלוח של הקובץ של סעיף ב' פי *O(lg(N)).*

בסעיף ג' שולחים:

N/4 + N/2 + N + (N+1) + (N+2) + … + 2N

זה לוקח כמעט אותו מס' מחזורים כמו קודם, פרט לכך שהפעם הורדנו מההתחלה *lg(N/8) = lg(N)-3* מחזורים, שבהם נשלחו *N/4 – 1* הסגמנטים הראשונים; וכדי להשלים את המשלוח החסר הנ"ל, יש להוסיף בסוף, אחרי שליחת 2N סגמנטים, שליחה נוספת, יחידה (גודל החלון הוא 2N+1, וזה די והותר עבור שליחת כל המידע החסר הנ"ל).

הגדלת החלון ההתחלתי הקטינה את זמן המשלוח של הקובץ של סעיף ג' ב *lg(N) -2*. הקיטון בזמן המשלוח במקרה זה הוא זניח – הקטנו מ-*O(N + lg(N))* ל-*O(N)*.

ה. כדאי להסתכן ולהתחיל עם חלון גדול יותר רק במקרה של משלוח מידע קטן למשל, מחרוזת חיפוש, או מייל קצר, ללא צרופות. במקרה של משלוח נתונים גדולים מאוד – למשל, הורדת קובץ גדול - השיפור הנובע מהגדלת החלון ההתחלתי הוא זניח.

**4.**

א. בכ"א משני הכיוונים ישלח מידע בגודל בית יחיד כשהוא עטוף ברישות / תחיליות בגודל 100 – כלומר, תקורה של כ-10,000%!

ב. במקרה זה ישלחו 1kByte נתונים עם 100 Byte רישות – כלומר, התקורה היא בערך 10%.

ג. בתקשורת אינטראקטיבית כגון telnet המשתמש מעוניין לקבל חיווי מהיר. עיכוב העברת הנתונים ייצור הצטברות של נתונים רבים אצל השולח לפני שליחתם – בדומה לתופעה ה"מעצבנת" של הקלדת מספר אותיות מבלי שהן מופיעות על הצג – שלאחר מכן "מדפיס" בבת אחת את כולן יחד.

**Question 8**

**A.**

The total *Tprop* is: 20,000 [km] / 2\*10^8[m/sec] = 0.1 sec.

Server 🡪 router *Dxmt*: 100kb/1mbps = 0.1s.

The rest of the values are negligible.

* *D~= 0.2 s*

**B.**

The total *Tprop* is: 20,000 [km] / 2\*10^8[m/sec] = 0.1 sec.

Router 🡪 Client *Dxmt*: 100kb / 1Gbps = 0.1ms

The rest of the values are negligible.

* *D=0.1 s*

**C.**

~50%.

**3. A.**

Server 🡪 router *Dxmt*: 200kb/1mbps = 0.2s.

The rest of the calculation is unchanged 🡪 *D=0.3s.*

**B**

No change from the baseline: *D=0.1s.*

**C.**

(0.3-0.1)/0.3 = 67%.