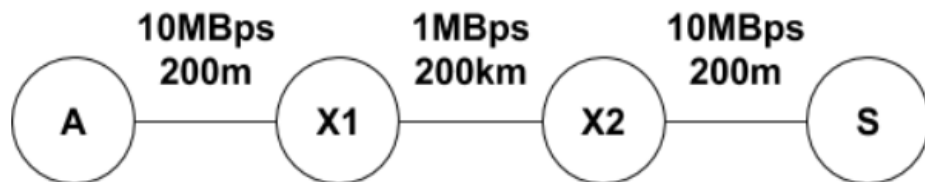


שאלה 1:

מחשב A מחובר לשרת web שכתובתו 9.42.41.3 דרך שני נתבים X1, X2 כמתואר בתרשים. A שולח בקשה לדף שמכיל רק אובייקט אחד (בנוסף לטקסט עצמו), ועם קבלתו מבקש את אותו אובייקט. גודל הבקשה 100 בתים, גודל הדף 900 בתים וגודל האובייקט 1800 בתים. שני הנתבים מקושרים זה לזה בערוץ פיבראופטי, עם גודל מסגרת מרבית של 1000 בתים, ואילו המחשבים מחוברים לנתבים ב-Ethernet עם גודל מסגרת מרבית (MTU) של 1500 בתים. קצב ההתפשטות 200,000 ק"מ לשניה ומשתמשים ב-HTTP 1.1. אין תקשורת אחרת.



- כמה זמן יקח עד שהדף והתמונה יגיעו למחשב A?
- מה (אם בכלל) יהיה השינוי בסעיף הראשון, אם נשתמש ב-HTTP 1.0?
- הניחו כי ל-X1 מחוברים עוד 3 לקוחות בדומה ל-A, וכל לקוח מבקש את הדף בהסתברות 20% מדי יום לפני ש-A מבקש (הדף משתנה מדי יום). כמה זמן, בממוצע, ייקח עד קבלת הדף והתמונה ב-A, אם מותקן web-proxy בנתב X1? אם מותקן web-proxy בנתב X2?

א. ראשית, נציין שכיוון שלא מתייחסים בשאלה לנושא DNS אזי אין צורך להתייחס לכך גם בתשובה. כמו כן, כיוון שלא נאמר לנו בשאלה שניתן להתעלם מתחיליות נצטרך להתייחס אליהן במהלך חישוב הזמנים.

שנית, נחשב כמה זמן ייקח עד שהדף והתמונה יגיעו למחשב A –

מכיוון שמחשב A עובד ב-http1.1 אזי מדובר על חיבור שהוא ברירת מחדל persistent. כלומר, כל הבקשות והתשובות נשלחות תחת אותו חיבור.

בקשת http נשלחת תחת פרוטוקול TCP. לכן, היא מתחילה בפתחת חיבור. בפתחת החיבור יש שתי חבילות שהן תחיליות, אחת syn ואחת syn+ack. גודל כל תחילית הוא 40B (20 עבור TCP ו-20 עבור IP).

השהיית השידור של תחילית היא:

$$\frac{40B}{10MBps} + \frac{40B}{1MBps} + \frac{40B}{10MBps} = \dots = 0.048ms$$

השהיית ההתפשטות של חבילה היא:

$$\frac{200m + 200km + 200m}{200,000km/s} = \dots = 1.002ms$$

כלומר, השהיית שידור + התפשטות עבור תחילית אחת זה סה"כ 1.05ms ועבור שתי תחיליות 2.1ms.

לאחר מכן, A שולח לשרת ack יחד עם הבקשה שלו לדף. גודל הבקשה הוא 100B ובתוספת התחיליות 140B.

השהיית השידור של הבקשה+ack היא:

$$\frac{140B}{10MBps} + \frac{140B}{1MBps} + \frac{140B}{10MBps} = \dots = 0.168ms$$

כלומר, השהיית שידור + התפשטות עבור הבקשה לדף+ack זה סה"כ 1.17ms.

לאחר מכן, השרת שולח לא את הדף (ack+). גודל הדף הוא 900B ובתוספת התחיליות 940B.

השהיית השידור של הדף היא:

$$\frac{940B}{10MBps} + \frac{940B}{1MBps} + \frac{940B}{10MBps} = \dots = 1.128ms$$

כלומר, השהיית שידור + התפשטות עבור הדף (ack+) זה סה"כ 2.13ms.

לאחר מכן, A שולח לשרת בקשה לאובייקט (ack+) וזה לוקח בדיוק אותו זמן כמו שליחת בקשה לדף+ack <- 1.17ms.

לאחר מכן, השרת שולח לא את האובייקט (ack+). גודל הדף הוא 1800B ובתוספת התחיליות 1840B.

TCP יודע את ה MTU של הערוץ שלו. הוא גם יודע את ה MTU של הערוץ שמחובר ליעד כי זה חלק מהמידע שהועבר לו בתהליך החיבור. TCP יודע שאין טעם לשלוח יותר מהמינימום של שני הערכים האלה אבל הוא עוד לא יודע את המינימום לאורך המסלול כי הוא לא יודע את ה MTU של הראוטרים שבדרך. לכן, הוא מבצע את האלגוריתם Path MTU Discovery.

הוא שולח חבילה בגודל 1500B (1460B מידע + 40B תחיליות) עם דגל DF דלוק. החבילה עוברת בערוץ בין השרת לנתב x2 כיוון שערוץ זה עם MTU=1500B, מגיעה לנתב x2 ונזרקת כיוון שהערוץ בין x2 ל x1 הוא עם MTU=1000B. לכן, הנתב x2 שולח הודעת ICMP לשרת שגודלה 40B.

השהיית השידור של חבילה בגודל 1500B מהשרת ל x2 היא:

$$\frac{1500B}{10MBps} = \dots = 0.15ms$$

השהיית השידור של הודעת ICMP מהשרת ל x2 היא:

$$\frac{40B}{10MBps} = \dots = 0.004ms$$

השהיית ההתפשטות של חבילה מהשרת ל x2 היא:

$$\frac{200m}{200,000kmps} = \dots = 0.001ms$$

לאחר מכן, TCP מפצל את החבילה המקורית ל 2 סגמנטים: הראשון בגודל 1000B (960B מידע + 40B תחיליות) והשני בגודל 880B (840B מידע + 40B תחיליות).

נשים לב – גודל החלון כרגע הוא 2 ולכן השרת יכול לשלוח את 2 הסגמנטים אחד אחרי השני מבלי לחכות ל ack על הסגמנט הראשון.

נחשב את השהיית השידור וההתפשטות של החבילה הראשונה עד שהיא מגיעה לערוץ האיטי, את השהיית השידור וההתפשטות של כל החבילות בערוץ האיטי ואת השהיית השידור וההתפשטות של החבילה האחרונה מהערוץ הכי איטי.

הערוץ האיטי הוא הערוץ שבין הנתב x2 ל x1.

השהיית השידור של החבילה הראשונה עד שהיא מגיעה לערוץ האיטי:

$$\frac{1000B}{10MBps} = \dots = 0.1ms$$

השהיית השידור של כל החבילות בערוץ האיטי:

$$\frac{1000B + 880B}{1MBps} = \dots = 1.88ms$$

השהיית השידור של החבילה האחרונה מהערוץ הכי איטי:

$$\frac{880B}{10MBps} = \dots = 0.088ms$$

נסכום הכל:

$$0.15ms + 0.001ms + 0.004ms + 0.001ms + 0.1ms + 1.88ms + 0.088ms + 1.002ms = 3.226ms$$

כלומר, השהיית שידור + התפשטות עבור האובייקט (ack+) זה סה"כ 3.226ms.

$$\text{סה"כ הזמן שייקח עד שהדף והתמונה יגיעו למחשב A הוא } 3.226ms + 1.17ms + 2.13ms + 1.17ms + 2.1ms = 9.796ms$$

ב. אם נשתמש ב-http1.0 אזי מדובר על חיבור שהוא ברירת מחדל no persistent. כלומר, כל בקשה נשלחת בחיבור נפרד וגודל החלון הופך שוב ל-1 בין חיבור לחיבור. לכן:

- נצטרך לנתק את החיבור ולפתוח חיבור חדש עבור בקשת האובייקט
- בחיבור החדש גודל החלון הופך ל-1 ולכן השרת צריך לחכות ל-ack על הסגמנט הראשון לפני שהוא שולח את הסגמנט השני במקום לשלוח את הסגמנטים אחד אחרי השני

ג. המשמעות של הימצאות web-proxy בנתב היא שאם אחד הלקוחות האחרים מבקש את הדף לפני A אזי הדף נשמר ב-cache של proxy כך שכאשר A יבקש את הדף – proxy יביא לו אותו מה-cache.

נחשב כמה זמן ייקח עד שהדף והתמונה יגיעו למחשב A אם מותקן web-proxy בנתב x1 –

כמה זמן דבר זה ייקח לו ?

בפתיחת החיבור בין A ל-x1 יש שתי חבילות שהן תחיליות, אחת syn ואחת syn+ack. גודל כל תחילית הוא 40B (20 עבור TCP ו-20 עבור IP).

השהיית השידור של תחילית היא:

$$\frac{40B}{10MBps} = \dots = 0.004ms$$

השהיית ההתפשטות של חבילה היא:

$$\frac{200m}{200,000kmps} = \dots = 0.001ms$$

כלומר, השהיית שידור + התפשטות עבור תחילית אחת זה סה"כ 0.005ms ועבור שתי תחיליות 0.01ms.

לאחר מכן, A שולח ל-x1 ack **יחד עם הבקשה שלו לדף**. גודל הבקשה הוא 100B ובתוספת התחיליות 140B.

השהיית השידור של הבקשה+ack היא:

$$\frac{140B}{10MBps} = \dots = 0.014ms$$

כלומר, השהיית שידור + התפשטות עבור הבקשה לדף+ack זה סה"כ 0.015ms.

לאחר מכן, web-proxy של x1 שולח ל-A את הדף (ack+). גודל הדף הוא 900B ובתוספת התחיליות 940B.

השהיית השידור של הדף היא:

$$\frac{940B}{10MBps} = \dots = 0.094ms$$

כלומר, השהיית שידור + התפשטות עבור הדף (ack+) זה סה"כ 0.095ms.

לאחר מכן, A שולח לx1 בקשה לאובייקט (ack+) וזה לוקח בדיוק אותו זמן כמו שליחת בקשה לדף+ack < 0.015ms.

לאחר מכן, web-proxy של x1 שולח לA את האובייקט (ack+). גודל הדף הוא 1800B ובתוספת התחיליות 1840B.

TCP מפצל את החבילה המקורית ל2 סגמנטים: הראשון בגודל 1500B (1460B מידע + 40B תחיליות) והשני בגודל 380B (340B מידע + 40B תחיליות).

נשים לב – גודל החלון כרגע הוא 2 ולכן web-proxy של x1 יכול לשלוח את 2 הסגמנטים אחד אחרי השני מבלי לחכות לack על הסגמנט הראשון.

השהיית השידור של שתי החבילות:

$$\frac{1500B + 380B}{10MBps} = \dots = 0.188ms$$

נסכום הכל:

$$0.188ms + 0.001ms = 0.189ms$$

כלומר, השהיית שידור + התפשטות עבור האובייקט (ack+) זה סה"כ 0.189ms.

סה"כ הזמן שייקח עד שהדף והתמונה יגיעו למחשב A הוא $0.324ms = 0.189ms + 0.015ms + 0.095ms + 0.015ms + 0.01ms$

נתון לנו שכל לקוח מבקשת את הדף בהסתברות 20% מידי יום לפני שA מבקש. לכן, ההסתברות שלקוח לא ביקש את הדף לפני A היא 0.8.

ההסתברות שגם הלקוח הראשון וגם הלקוח השני וגם הלקוח השלישי לא ביקשו את הדף לפני A היא: 0.8^3 .

לפיכך, הזמן הממוצע שייקח עד לקבלת הדף והאובייקט בA הוא:

$$0.8^3 * 9.796ms + (1 - 0.8^3) * 0.324ms = 5.173664ms$$

כאשר $9.796ms$ זה הזמן שלקוח לקבל את הדף והאובייקט כשהוא לא נמצא בcache של web-proxy (חישבנו בסעיף א) ו- $0.324ms$ זה הזמן שלקוח לקבל את הדף והאובייקט כשהוא כן נמצא שם.

נחשב כמה זמן ייקח עד שהדף והתמונה יגיעו למחשב A אם מותקן web-proxy בנתב x2 –

כמה זמן דבר זה ייקח לו ?

בפתיחת החיבור בין A לx2 יש שתי חבילות שהן תחיליות, אחת syn ואחת syn+ack. גודל כל תחילית הוא 40B (20 עבור TCP ו-20 עבור IP).

השהיית השידור של תחילית היא:

$$\frac{40B}{10MBps} + \frac{40B}{1MBps} = \dots = 0.044ms$$

השהיית ההתפשטות של חבילה היא:

$$\frac{200m + 200km}{200,000km/s} = \dots = 1.001ms$$

כלומר, השהיית שידור + התפשטות עבור תחילית אחת זה סה"כ $1.045ms$ ועבור שתי תחיליות $2.09ms$.

לאחר מכן, A שולח לx2 ack יחד עם הבקשה שלו לדף. גודל הבקשה הוא 100B ובתוספת התחיליות 140B.

השהיית השידור של הבקשה+ack היא:

$$\frac{140B}{10MBps} + \frac{140B}{1MBps} = \dots = 1.154ms$$

כלומר, השהיית שידור + התפשטות עבור הבקשה לדף+ack זה סה"כ $1.155ms$.

לאחר מכן, הweb-proxy של x2 שולח A את הדף (+ack). גודל הדף הוא 900B ובתוספת התחיליות 940B.

השהיית השידור של הדף היא:

$$\frac{940B}{10MBps} + \frac{940B}{1MBps} = \dots = 1.034ms$$

כלומר, השהיית שידור + התפשטות עבור הדף (+ack) זה סה"כ 2.035ms.

לאחר מכן, A שולח לx2 בקשה לאובייקט (+ack) וזה לוקח בדיוק אותו זמן כמו שליחת בקשה לדף+ack <- 1.155ms.

לאחר מכן, הweb-proxy של x2 שולח A את האובייקט (+ack). גודל הדף הוא 1800B ובתוספת התחיליות 1840B.

TCP מפצל את החבילה המקורית ל2 סגמנטים: הראשון בגודל 1000B (960B מידע + 40B תחיליות) והשני בגודל 880B (840B מידע + 40B תחיליות).

נשים לב – גודל החלון כרגע הוא 2 ולכן הweb-proxy של x2 יכול לשלוח את 2 הסגמנטים אחד אחרי השני מבלי לחכות לack על הסגמנט הראשון.

נחשב את השהיית השידור וההתפשטות של כל החבילות בערוץ האיטי ואת השהיית השידור וההתפשטות של החבילה האחרונה מהערוץ הכי איטי.

הערוץ האיטי הוא הערוץ שבין הנתב x2 לx1.

השהיית השידור של כל החבילות בערוץ האיטי:

$$\frac{1000B + 880B}{1MBps} = \dots = 1.88ms$$

השהיית השידור של החבילה האחרונה מהערוץ הכי איטי:

$$\frac{880B}{10MBps} = \dots = 0.088ms$$

נסכום הכל:

$$1.88ms + 0.088ms + 1.001ms = 2.969ms$$

כלומר, השהיית שידור + התפשטות עבור האובייקט (+ack) זה סה"כ 2.969ms.

סה"כ הזמן שייקח עד שהדף והתמונה יגיעו למחשב A הוא 9.404ms = 2.969ms + 1.155ms + 2.035ms + 1.155ms + 2.09ms.

באופן דומה לחישוב הקודם, הזמן הממוצע שייקח עד לקבלת הדף והאובייקט בA הוא:

$$0.8^3 * 9.796ms + (1 - 0.8^3) * 9.404ms = 9.604704ms$$

כאשר 9.796ms זה הזמן שלוקח לקבל את הדף והאובייקט כשהוא לא נמצא בcache של הweb-proxy (חישובנו בסעיף א) ו9.404ms זה הזמן שלוקח לקבל את הדף והאובייקט כשהוא כן נמצא שם.

שאלה 2:

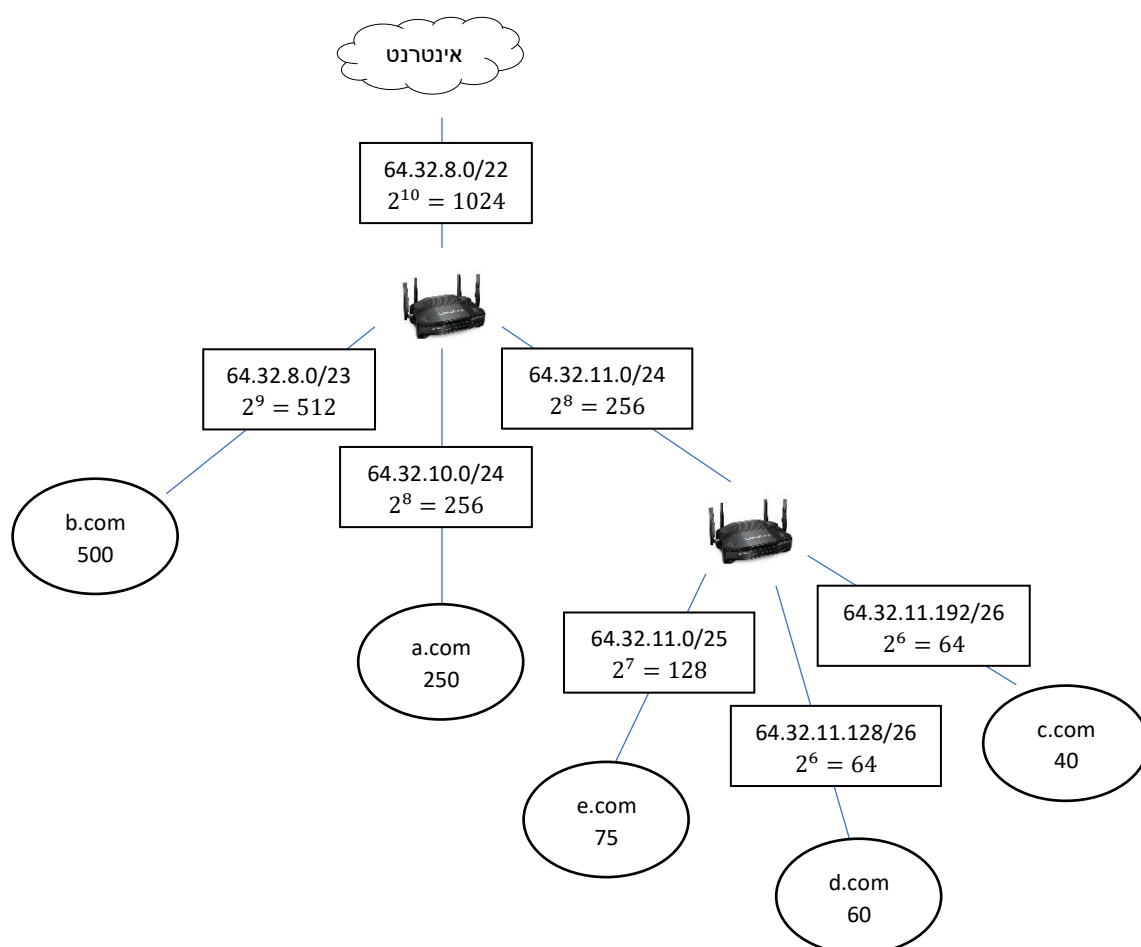
ספק אינטרנט קיבל בלוק כתובות 64.32.8.0/22 שמתוכו עליו להקצות כתובות לחמישה לקוחות (להלן) באופן יעיל ביותר כאשר הוא משתמש במספר מינימלי של נתבים. לכל נתב לכל היותר ארבעה ממשקי רשת (שיזוהו ע"י המספרים 1 עד 4).

לקוח	a.com	b.com	c.com	d.com	e.com
מספר מחשבים	250	500	40	60	75

ציירו את מבנה תת-הרשת שהקצה ספק האינטרנט (החל מהחיבור לאינטרנט שאותו יש לצייר בתרשים). על כל ערוץ ציינו את כתובת ה-IP (ואת ה-mask לפי CIDR) וציינו את מספר כתובת ה-IP המקסימלי בכל תת הרשת.

אם subnet mask הוא 22 זה אומר שיש לנו 2 בחזקת 10 (1024) כתובות לחלק.

נשים לב שכשלקוח רוצה 500 כתובות למשל אנחנו לא יכולים לתת לו 500 כתובות אלא את הקפיצה הבינארית הכי קרובה לזה והוא פשוט לא ישתמש בעודף. למשל, 512.



מתחילים מהראוטר של ספק האינטרנט.

נתון בשאלה שלכל נתב יש לכל היותר 4 ממשקים ושנחנו רוצים להשתמש במינימום נתבים ככל האפשר.

זה אומר שלראוטר של ספק האינטרנט יש 4 ממשקים. אחד משמש אותו כדי להתחבר לאינטרנט. הראוטר מפרסם לאינטרנט שהוא אחראי על כל הsubnet 64.32.8.0/22 ואם יש לו חבילות שמיועדות לאחת מהכתובות ברשת הזאת שיעביר אליו והוא ידאג שהיא תגיע ליעדה.

נותרו עוד 3 ממשקים. לראוטר יש כל מיני לקוחות ולכל אחד היו דרישות שונות.

b.com רצה 500 כתובות. נותנים לו 512. כדי לדעת איזה כתובות נתנו לו אנחנו רושמים בטבלה שנתנו לו את כל הכתובות 64.32.8.0/23 subnet mask הוא 23 כי נתנו לו 2 בחזקת 9 (512) כתובות. הכתובות הן - 64.32.8.0 עד 64.32.8.255 + 64.32.9.0 עד 64.32.9.255.

a.com רצה 250 כתובות. נותנים לו 256. כדי לדעת איזה כתובות נתנו לו אנחנו רושמים בטבלה שנתנו לו את כל הכתובות 64.32.10.0/24 subnet mask הוא 24 כי נתנו לו 2 בחזקת 8 (256) כתובות. הכתובות הן - 64.32.10.0 עד 64.32.10.255.

נותר עוד ממשק אחד פנוי ו3 לקוחות. לכן, הממשק שנותר מנוצל לצורך חיבור בין הנתב הראשון לנתב השני.

לנתב השני נשארו 3 ממשקים פנויים.

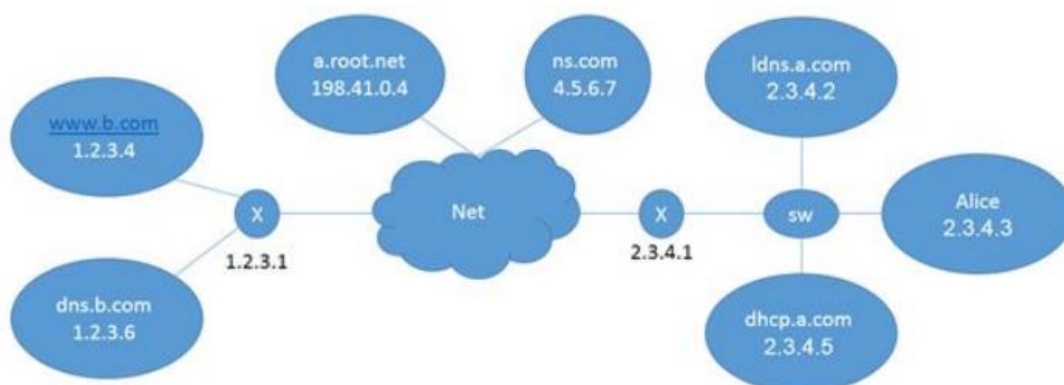
e.com רצה 75 כתובות. נותנים לו 128. כדי לדעת איזה כתובות נתנו לו אנחנו רושמים בטבלה שנתנו לו את כל הכתובות 64.32.11.0/25 subnet mask הוא 25 כי נתנו לו 2 בחזקת 7 (128) כתובות. הכתובות הן - 64.32.11.0 עד 64.32.11.127.

d.com רצה 60 כתובות. נותנים לו 64. כדי לדעת איזה כתובות נתנו לו אנחנו רושמים בטבלה שנתנו לו את כל הכתובות 64.32.11.128/26 subnet mask הוא 26 כי נתנו לו 2 בחזקת 6 (64) כתובות. הכתובות הן - 64.32.11.128 עד 64.32.11.191.

c.com רצה 40 כתובות. נותנים לו 64. כדי לדעת איזה כתובות נתנו לו אנחנו רושמים בטבלה שנתנו לו את כל הכתובות 64.32.11.192/26 subnet mask הוא 26 כי נתנו לו 2 בחזקת 6 (64) כתובות. הכתובות הן - 64.32.11.192 עד 64.32.11.255.

שאלה 3:

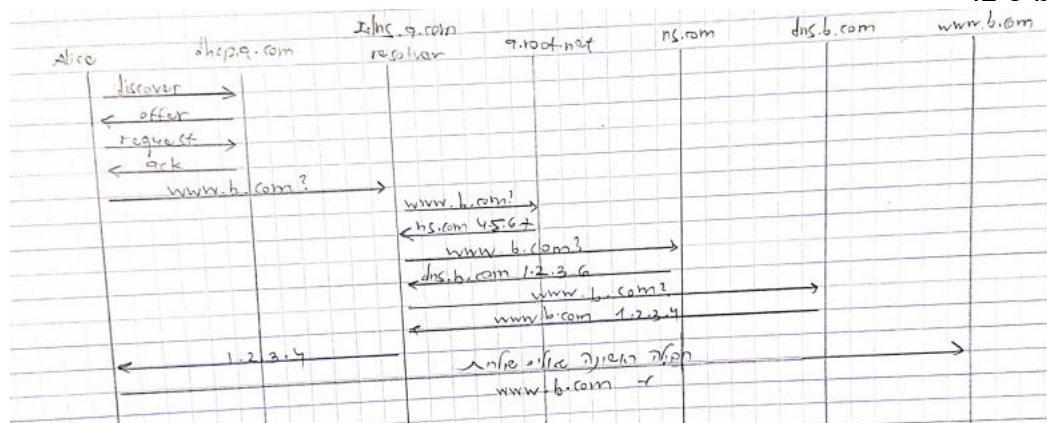
נתונה רשת כבתרשים, כאשר ns.com הוא שרת השמות של com, ו a.root.net הוא שרת השמות של השורש, ו dns.b.com שרת השמות של b.com, ו ldns.a.com הוא ה-resolver (שרת השמות המקומי) של הרשת 2.3.4.0/24.



א. הניחו כי הדפדפן של אליס פונה עם ההדלקה לבקשת הדף www.b.com, וכי לא הייתה כל פעילות ברשת לפני כן (ובפרט המטמון של ה-resolver ריק, והמתג עוד לא למד איזה התקן מחובר באיזו כתובת). הציגו תרשים העברת הודעות (ללא זמנים) עד שמחשב Alice שולח חבילה ראשונה לשרת www.b.com. התרשים צריך לכלול הליך קבלת הכתובת ע"י אליס וקבלת/למידת התצורה (לרבות כתובת הנתב ושרת ה-DNS המקומי).

ב. הניחו כי השהיית ההתפשטות ברשתות המקומיות זניחה, ובערוצים המחוברים לרשת היא 0.1 שניה. קצב השידור בכל הערוצים הוא 1.5Mbyte/sec. מותר להתעלם מזמן שידור של הודעות קצרות מ-100 בתים. הציגו את המשך תרשים העברת ההודעות, עד לקבלת הדף והצגתו במחשב של אליס, הפעם בציון זמנים. הניחו שגודל הדף הוא 4KB וה-MSS בכל הערוצים הוא של 1500 בתים.

א. תרשים:

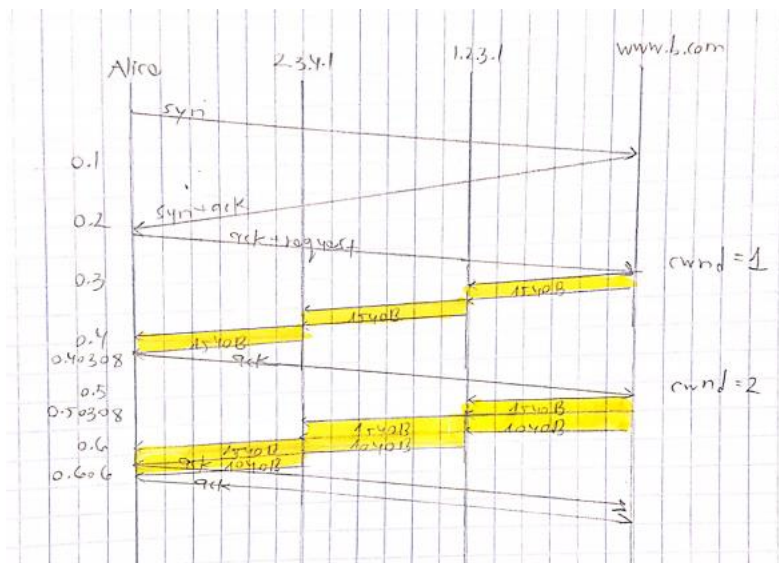


אליס שולחת הודעה בbroadcast על מנת למצוא את שרת DHCP שיביא למחשב שלה כתובת IP. שרת ה-DHCP של הרשת שאליס נמצאת בה (dhcp.a.com) עונה לה ומציע לה כתובת IP. אליס מבקשת לקבל את כתובת ה-IP שהוא הציע לה והשרת מאשר לה ע"י החזרת ACK. כעת לאליס יש כתובת IP 2.3.4.3, כתובת של שרת ה-DNS המקומי (idns.a.com) 2.3.4.2 וכתובת של הראוטר 2.3.4.1.

אליס פונה לשרת ה-DNS המקומי ושואלת אותו מהי כתובת ה-IP של הדומיין www.b.com. על פי הנתון המטמון של ה-resolver ריק ולכן הוא פונה לשרת השמות של root (a.root.net) ושואל אותו את אותה שאלה. שרת השמות של Root מחזיר לו שהוא צריך לשאול את שרת השמות של .com. שהוא ns.com וכתובת ה-IP שלו היא 4.5.6.7. ה-resolver פונה לשרת השמות של .com ושואל אותו את אותה שאלה. שרת השמות של .com מחזיר לו שהוא צריך לשאול את שרת השמות של .b.com. שהוא dns.b.com וכתובת ה-IP שלו היא 1.2.3.6. ה-resolver פונה לשרת השמות של .b.com ושואל אותו את אותה שאלה. שרת השמות של .b.com מחזיר את כתובת ה-IP של www.b.com שהוא 1.2.3.4. ה-resolver לומד את התשובה ומחזיר אותה לאליס.

אליס כעת יכולה לשלוח את החבילה הראשונה שלה ל-www.b.com.

ב. ציור להמחשה:



נחשב את הזמנים:

בפתיחת החיבור יש שתי חבילות שהן תחיליות, אחת syn ואחת syn+ack. נתון שאפשר להתעלם מזמני השידור של הודעות שקצרות מ-100B ותחילית היא כזו ולכן ניקח בחשבון רק את זמן ההתפשטות שהוא 0.1s לכל תחילית. סה"כ 0.2s.

אליס שולחת את הבקשה שלה + ack האחרון של פתיחת החיבור. נניח שגודל הבקשה + ack הוא לכל היותר 100B ולכן שוב ניקח בחשבון רק את זמן ההתפשטות שהוא 0.1s.

ב-TCP אנחנו מנסים להימנע מפרגמנטציה ע"י שימוש בסגמנטציה מראש.

TCP יודע את ה-MTU של הערוץ כי זה חלק מהמידע שהועבר לו בתהליך החיבור והוא יודע שאין טעם לשלוח יותר מזה בכל סגמנט. נשים לב שבנתונים מופיע ה-MSS שהוא המספר המקסימלי של בתים שאפשר להעביר ללא תחיליות, MTU כן כולל את התחיליות. לכן, החבילה המקורית מתחלקת ל-3 סגמנטים – הראשון בגודל 1540B (1500 בתים של מידע + 40 בתים תחיליות), השני בגודל 1540B והשלישי בגודל 1040B (1000 בתים של מידע + 40 בתים תחיליות).

השליחה מתבצעת בslow start ולכן מתחילים עם חלון בגודל 1 MSS. השרת שולח את סגמנט 1.

השהיית השידור של חבילה בגודל 1540B בכל ערוץ:

$$\frac{1540B}{1.5MBps} = \frac{1540B}{1.5 \times 10^6 Bps} = 0.0010266667s$$

לכן, סך השהיית השידור של סגמנט 1 בכל הערוצים הוא 3 כפול השהיית השידור של הסגמנט בערוץ. כלומר, 0.0030800001s.

נוסיף לזה את השהיית ההתפשטות ונקבל שסגמנט 1 מגיע לאליס לאחר 0.1030800001s.

אליס מחזירה ack שהוא תחילית ולכן לוקח זמן התפשטות שהוא 0.1s.

החלון גדל ל-MSS 2. השרת שולח את סגמנט 2 ומיד לאחר מכן את סגמנט 3. קצב השידור בכל הערוצים הוא אותו דבר ולכן נתייחס לערוץ האמצעי כערוץ הכי איטי. נחשב את ההשהיות עבור הסגמנט הראשון בערוץ שבין www.b.com לנתב, את ההשהיות עבור שני הסגמנטים בערוץ האיטי ואת ההשהיות עבור הסגמנט השני בערוץ שבין הנתב לאליס.

השהיית השידור של חבילה בגודל 1040B בכל ערוץ:

$$\frac{1040B}{1.5Mbps} = \frac{1040B}{1.5 * 10^6Bps} = 0.0006933333s$$

סה"כ נקבל –

$$0.0010266667s + 0.0010266667s + 0.0006933333s + 0.0006933333s + 0.1s = 0.606s$$

שאלה 4:

Network Address	Next Router	hops	Interface
196.80.0.0/12	-	0	eth0
196.94.16.0/20	-	0	eth1
196.96.0.0/12	128.3.3.1	1	C
196.104.0.0/14	10.0.0.1	2	D
128.0.0.0/1	-	0	E
64.0.0.0/2	-	0	F
0.0.0.0/2	-	0	G

*בתא בטבלה בו מופיע - הכוונה לכך שאין נתב הבא, וה-host הקרוב ביותר הוא היעד הסופי.

לעיל מוצגת טבלת ניתוב מבוססת CIDR. לאיזה Interface תשלח חבילה עם כתובת IP הבאה (ענו לכל כתובת יעד ברשימה):

195.65.127.3 .1

196.94.100.13 .2

196.96.34.9 .3

94.100.100.18 .4

196.106.49.46 .5

196.0.0.1 .6

ראשית, נהפוך את הכתובות שבטבלה לבינארית ונסמן את subnet mask:

Network address	Interface
11000100.01010000.00000000.00000000	Eth0
11000100.01011110.00010000.00000000	Eth1
11000100.01100000.00000000.00000000	C
11000100.01101000.00000000.00000000	D
10000000.00000000.00000000.00000000	E
01000000.00000000.00000000.00000000	F
00000000.00000000.00000000.00000000	G

שנית, נהפוך את הכתובות שברשימה לבינארית ונחליט לאן כל אחת תישלח:

1. 11000011.01000001.01111111.00000011 – מתאימה רק לממשק E ולכן תישלח אליו.
2. 11000100.01011110.01100100.00001101 – מתאימה רק לממשק eth0 ולכן תישלח אליו.
3. 11000100.01100000.00100010.00001001 – מתאימה רק לממשק C ולכן תישלח אליו.
4. 01011110.01100100.01100100.00010010 – מתאימה רק לממשק F ולכן תישלח אליו.
5. 11000100.01101010.00110001.00101110 – מתאימה גם לממשק C וגם לממשק D אבל D יותר ספציפי ולכן תישלח אליו.
6. 11000100.00000000.00000000.00000001 – מתאימה רק לממשק E ולכן תישלח אליו.

שאלה 5:

משה לוחץ על קישור כלשהו בדפדפן. נתון שכתובת ה IP של הדומיין שמופיע בקישור אינה שמורה ב cache של משה, ולכן משתמשים ב DNS. הניחו שתהליך ה DNS הצריך פניה ל n שרתי DNS, וכל פניה שכזאת (שליחת שאלה לשרת וקבלת התשובה) לקחה RTT_i זמן. כלומר, שאלה + תשובה משרת ה DNS הראשון (מספר 1) לקחה RTT_1 שניות, שאלה+תשובה משרת ה DNS השני (מספר 2) לקחה RTT_2 שניות וכן הלאה.

הקישור שמשה לחץ עליו מחזיר עמוד html קטן (נכנס בחבילה אחת) על גבי HTTP ללא הפניות למשאבים נוספים. זמן ה RTT בין המחשב של משה לשרת מוגדר בתור RTT_0 .

חבילות בקרה כמו syn ו ack הינן חבילות לכל דבר ואין להתעלם מהן.

שימו לב שאין השהיות נוספות מעבר למתואר בשאלה, וזמני השידור וההתפשטות מבוטאים באמצעות ערכי ה RTT שהוגדרו בשאלה.

נניח כעת שעמוד ה html מכיל 8 משאבים קטנים (כל משאב הינו בגודל שקטן מחבילה אחת) וכל המשאבים נמצאים על אותו השרת שנמצא עליו עמוד ה html. כמה זמן לוקח מהרגע שמשה לחץ על הקישור ועד שמשה קיבל את העמוד וכל המשאבים כאשר מדובר ב:

1. פרוטוקול HTTP non-persistent ללא חיבורים במקביל?
2. פרוטוקול HTTP non-persistent עם 5 חיבורים במקביל?
3. פרוטוקול HTTP persistent ללא חיבורים במקביל וללא pipelining?

1. בפרוטוקול HTTP non-persistent עבור כל בקשה פותחים חיבור נפרד. כיוון שהדבר נעשה על פי הנתון ללא חיבורים במקביל אזי בכל רגע חיבור אחד בלבד פתוח.

פניה ל n שרתי DNS לוקחת $\sum_{i=1}^n RTT_i$ זמן. כמו כן, בכל חיבור יש:

- פתיחת חיבור שכוללת שליחת syn וקבלת syn+ack. לוקח RTT_0 זמן.
- המשך פתיחת חיבור שכולל שליחת ack + שליחת בקשה לשרת וקבלת תשובה מהשרת. לוקח RTT_0 זמן.
- סגירת חיבור. לוקח RTT_0 זמן.

משה פונה לשרת 9 פעמים – פעם אחת בשביל עמוד ה HTML ועוד 8 פעמים בשביל המשאבים שבעמוד.

נשים לב שהשאלה מבקשת לחשב את הזמן שלוקח מהרגע שמשה לוחץ על הקישור ועד שהוא מקבל את העמוד וכל המשאבים. לכן, אין צורך לכלול בחישוב את הזמן שלוקחת סגירת החיבור האחרון.

סה"כ הזמן שלוקח הוא:

$$\sum_{i=1}^n RTT_i + 8 * 3RTT_0 + 1 * 2RTT_0 = \sum_{i=1}^n RTT_i + 26 * RTT_0$$

2. הפעם הדבר נעשה על פי הנתון עם 5 חיבורים במקביל.

בהתחלה משה לא יודע שבעמוד ה HTML יש עוד משאבים שצריך לבקש מהשרת. לכן, הוא קודם כל פותח חיבור אחד בלבד לצורך בקשת העמוד. לאחר מכן, את 8 המשאבים הוא מבקש כך שבכל רגע נתון פתוחים 5 חיבורים.

עבור 5 הבקשות הבאות ניקח בחשבון זמן שלוקחת בקשה אחת ועבור 3 הבקשות הנותרות ניקח בחשבון זמן שלוקחת בקשה אחת.

נשים לב שהשאלה מבקשת לחשב את הזמן שלוקח מהרגע שמשה לוחץ על הקישור ועד שהוא מקבל את העמוד וכל המשאבים. לכן, אין צורך לכלול בחישוב את הזמן שלוקחת סגירת החיבור של 3 הבקשות האחרונות.

סה"כ הזמן שלוקח הוא:

$$\sum_{i=1}^n RTT_i + 1 * 3RTT_0 + 1 * 3RTT_0 + 1 * 2RTT_0 = \sum_{i=1}^n RTT_i + 8 * RTT_0$$

3. בפרוטוקול HTTP persistent אין צורך לפתוח חיבור נפרד עבור כל בקשה. כיוון שהדבר נעשה על פי הנתון ללא חיבורים במקביל אזי בכל רגע חיבור אחד בלבד פתוח וכיוון שהדבר נעשה ללא pipelining אזי אי אפשר לשלוח חבילה לפני שמקבלים ack עליה.

הפעם ניקח בחשבון זמן שלוקחת פתיחת חיבור אחד, שליחה של 9 בקשות לשרת (הבקשה הראשונה כוללת ack של פתיחת החיבור) וקבלה של 9 תשובות מהשרת.

נשים לב שהשאלה מבקשת לחשב את הזמן שלוקח מהרגע שמשה לוחץ על הקישור ועד שהוא מקבל את העמוד וכל המשאבים. לכן, אין צורך לכלול בחישוב את הזמן שלוקחת סגירת החיבור.

סה"כ הזמן שלוקח הוא:

$$\sum_{i=1}^n RTT_i + RTT_0 + 9 * RTT_0 = \sum_{i=1}^n RTT_i + 10 * RTT_0$$

שאלה 6:

לקוח ושרת אינטרנט מחוברים בערוץ ישיר בקצב 1Mbps. הלקוח רוצה להוריד מהשרת קובץ בגודל 15KB כאשר MSS=1KB. התעלמו מזמן השידור של תחיליות. כמו כן, נתון שגודל הבקשה שהלקוח שולח לשרת להוריד את הקובץ זניח ולכן ניתן להתעלם מזמן השידור שלה.

מהירות ההתפשטות היא 10^6 mps.

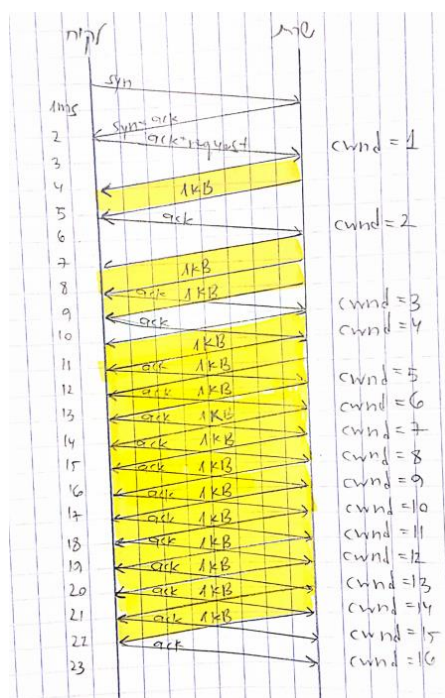
כמה זמן יקח להוריד את הקובץ החל מהרגע שהלקוח הקליד בדפדפן את פרטי הבקשה אם נתון שהמרחק בין השרת ללקוח הוא:

1. 1km

2. 10km

3. 0.1km

1. ציור להמחשה:



כאשר המרחק בין השרת ללקוח הוא 1km, ההיית ההתפשטות היא: $\frac{1km}{10^6mps} = 1ms$

ההיית השידור של חבילה בגודל 1KB היא: $\frac{1KB}{1Mbps} = 1ms$

נחשב את הזמנים:

בפתיחת החיבור יש שתי חבילות שהן תחיליות, אחת syn ואחת syn+ack. נתון שאפשר להתעלם מזמני השידור של תחיליות ולכן ניקח בחשבון רק את זמן ההתפשטות שהוא 1ms לכל תחילית. סה"כ 2ms.

הלקוח שולח את הבקשה שלו + ack האחרון של פתיחת החיבור. נתון שאפשר להתעלם מזמן השידור של הבקשה (וגם של תחיליות) ולכן שוב ניקח בחשבון רק את זמן ההתפשטות שהוא 1ms.

ב TCP אנחנו מנסים להימנע מפרגמנטציה ע"י שימוש בסגמנטציה מראש.

TCP יודע את ה MTU של הערוץ כי זה חלק מהמידע שהועבר לו בתהליך החיבור והוא יודע שאין טעם לשלוח יותר מזה בכל סגמנט. לכן, החבילה המקורית מתחלקת ל 15 סגמנטים בגודל MSS כל אחד (סה"כ יחד 15KB). נסמן את הסגמנטים הללו ב i כאשר i נע בין 1 ל 15.

השליחה מתבצעת slow start ולכן מתחילים עם חלון בגודל MSS 1. השרת שולח את חבילה 1 ב 2ms (השהיית שידור + התפשטות). הלקוח מחזיר ack שהוא תחילת חלון לוקח זמן התפשטות שהוא 1ms.

החלון גדל ל MSS 2. השרת שולח את חבילה 2 ומיד לאחר מכן את חבילה 3. בזמן שחבילה 2 מתפשטת חבילה 3 משודרת ולכן צריך לקחת בחשבון 2 השהיות שידור + התפשטות אחת. סה"כ 3ms. בזמן שחבילה 3 מתפשטת ה ack על חבילה 2 מתפשט ולכן אין צורך לקחת אותו בחשבון.

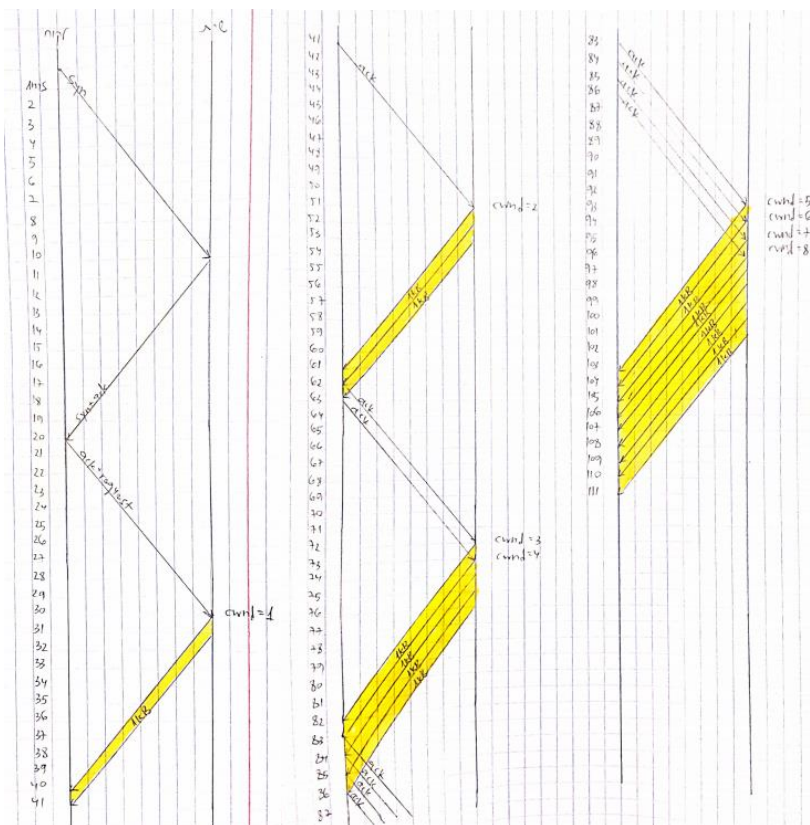
החלון גדל ל MSS 3.

משלב זה, נשים לב שבגלל שהשהיית השידור והשהיית ההתפשטות זהות ACK על חבילה מגיע לאחר שידור חבילה שלישית. למשל, ACK על חבילה 4 מגיע לאחר שידור חבילה 6. ז"א שהחלון גדל בכל פעם לפחות ב 1 ולכן השרת שולח חבילה אחרי חבילה ללא הפסקה (full pipe) עד שהוא מסיים לשלוח את החבילה 15, סה"כ 12 חבילות שלוקחות $12 \times 1\text{ms} + 1\text{ms} = 13\text{ms}$ (השהיות שידור והשהיית התפשטות אחת).

נסכום הכל יחד ונקבל –

$$2\text{ms} + 1\text{ms} + 2\text{ms} + 1\text{ms} + 3\text{ms} + 13\text{ms} = 22\text{ms}$$

2. ציור להמחשה:



$$\frac{10\text{km}}{10^6\text{mps}} = 10\text{ms}$$

כאשר המרחק בין השרת ללקוח הוא 10km השהיית ההתפשטות היא:

נחשב את הזמנים:

בפתיחת החיבור יש שתי חבילות שהן תחיליות, אחת syn ואחת syn+ack. נתון שאפשר להתעלם מזמני השידור של תחיליות ולכן ניקח בחשבון רק את זמן ההתפשטות שהוא 10ms לכל תחילית. סה"כ 20ms.

הלקוח שולח את הבקשה שלו + ack האחרון של פתיחת החיבור. נתון שאפשר להתעלם מזמני השידור של הבקשה (וגם של תחיליות) ולכן שוב ניקח בחשבון רק את זמן ההתפשטות שהוא 10ms.

ב TCP אנחנו מנסים להימנע מפרגמנטציה ע"י שימוש בסגמנטציה מראש.

TCP יודע את ה MTU של הערוץ כי זה חלק מהמידע שהועבר לו בתהליך החיבור והוא יודע שאין טעם לשלוח יותר מזה בכל סגמנט. לכן, החבילה המקורית מתחלקת ל 15 סגמנטים בגודל MSS כל אחד (סה"כ יחד 15KB). נסמן את הסגמנטים הללו ב i כאשר i נע בין 1 ל 15.

הפעם, בגלל שהשהיית ההתפשטות יותר גדולה לוקח יותר זמן עד שמגיעים למצב של full pipe. נגיע למצב כזה כש-

$$W * d_t = RTT$$

$$W * 1ms = RTT$$

נחשב מהו RTT:

$$RTT = d_t + d_p + d_p = 1ms + 10ms + 10ms = 21ms$$

נציב ונקבל-

$$W * 1ms = 21ms$$

$$W = 21$$

קיבלנו שמגודל חלון 21 השרת מתחיל לשלוח חבילות אחת אחרי השנייה ללא הפסקה (full pipe). אצלנו יש רק 15 חבילות ולכן לא נגיע למצב כזה. כדי לשלוח 15 חבילות אנחנו צריכים 4 חלונות – הראשון של חבילה אחת, השני של 2, השלישי של 4 והרביעי של 8 (כמו בציור).

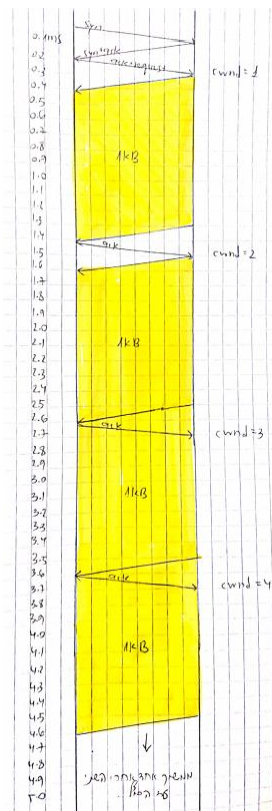
3 החלונות הראשונים לוקחים 3RTT זמן. בחלון הרביעי אנחנו מעוניינים בזמני שידור והתפשטות של המידע בלבד ולא של הack's עליהם. סה"כ-

$$3RTT + 8d_t + d_p = 3 * 21ms + 8 * 1ms + 10ms = 81ms$$

נסכום הכל יחד ונקבל –

$$20ms + 10ms + 81ms = 111ms$$

3. ציור להמחשה:



כאשר המרחק בין השרת ללקוח הוא 0.1km השהיית ההתפשטות היא: $\frac{0.1km}{10^6mps} = 0.1ms$

נחשב את הזמנים:

בפתיחת החיבור יש שתי חבילות שהן תחיליות, אחת syn ואחת syn+ack. נתון שאפשר להתעלם מזמני השידור של תחיליות ולכן ניקח בחשבון רק את זמן ההתפשטות שהוא 0.1ms לכל תחילית. סה"כ 0.2ms.

הלקוח שולח את הבקשה שלו + ack האחרון של פתיחת החיבור. נתון שאפשר להתעלם מזמן השידור של הבקשה (וגם של תחיליות) ולכן שוב ניקח בחשבון רק את זמן ההתפשטות שהוא 0.1ms.

ב TCP אנחנו מנסים להימנע מפרגמנטציה ע"י שימוש בסגמנטציה מראש.

TCP יודע את ה MTU של הערוץ כי זה חלק מהמידע שהועבר לו בתהליך החיבור והוא יודע שאין טעם לשלוח יותר מזה בכל סגמנט. לכן, החבילה המקורית מתחלקת ל 15 סגמנטים בגודל MSS כל אחד (סה"כ יחד 15KB). נסמן את הסגמנטים הללו ב i כאשר i נע בין 1 ל 15.

השליחה מתבצעת בslow start ולכן מתחילים עם חלון בגודל MSS 1. השרת שולח את חבילה 1 ב 1.1ms (השהיית שידור + התפשטות). הלקוח מחזיר ack שהוא תחילית ולכן לוקח זמן התפשטות שהוא 0.1ms.

החלון גדל ל MSS 2.

משלב זה, נשים לב שבגלל שהשהיית ההתפשטות היא עשירית מהשהיית השידור מתקיים ש ACK על חבילה מגיע לפני שמסיימים לשדר את החבילה הבאה. למשל, בזמן שחבילה 3 משודרת ה ack על חבילה 2 כבר מספיק להגיע בחזרה לשרת. ז"א שהחלון גדל בכל פעם לפחות ב 1 ולכן השרת שולח חבילה אחרי חבילה ללא הפסקה (full pipe) עד שהוא מסיים לשלוח את החבילה ה 15, סה"כ 14 חבילות שלוקחות $14 * 1ms + 0.1ms$ (14 השהיות שידור והשהיית התפשטות אחת).

נסכום הכל יחד ונקבל –

$$0.2ms + 0.1ms + 1.1ms + 0.1ms + 14.1ms = 15.6ms$$