יעל אביעוז 207237421

מבוא לרשתות תקשורת – תרגיל 3

שאלה 1

מחשב A שולח הודעה למחשב B ובמסלול ביניהם יש 2 נתבים.
ה-MTU בין A ל- R1, וכן בין R1 ל- R2 הוא 1500B.
ה-MTU בין R2 לבין B הוא 660B.
מחשב A שולח למחשב B כמות של 1980 בתים של מידע (שכבת אפליקציה).

נתון: גודל תחילית UDP 10B, תחילית TCP של 20B ותחילית IP של 20B.
תארו כיצד ישלח המידע באמצעות UDP ובאמצעות TCP.
בתשובתכם יש לציין את ערכי השדות: IP-ID, Length, MF, DF, Offset

פתרון

UDP

מ- A ל- R1 נתונה חבילה שמורכבת באופן הבא:

- IP Header = 20B
- UDP Header = 10B
- Data = 1980B

לכן נקבל כי ערכי השדות הם:

- Offset = 0 – משום שעדיין לא ביצענו פרגמנטציה הערך הוא 0
- DF = 0
- MF = 0 - משום שעדיין לא ביצענו פרגמנטציה הערך הוא 0.
- ID = x - לכל חבילה יש מזהה אחר שהיא מקבלת ב- Header IP , נסמנו ב-x.
- Length = 2010B – חיבור של גודל ה- Data עם גודל ה- UDP Header וה- IP Header

נתון שה-MTU בין A ל- R1 הוא 1500B, לכן החבילה גדולה מידי והיא מתפרקת לשני פרגמנטים:

Frag1

- IP Header = 20B
- UDP Header = 10B
- Data = 1470B כי נתון שה-MTU הוא 1500B ולכן יחד עם ה-IP Header וה-UDP Header נקבל גודל של 1500B (שזהו הגודל המקסימלי שניתן לשלוח)

- Offset = 0 – משום שזהו הפרגמנט הראשון.
- DF = 0
- MF = 1 – דולק בכדי לציין שזהו לא הפרגמנט האחרון בקבוצה.
- ID = x – שייך לחבילה המקורית שהמזהה שלה הוא x.
- Length = 1500B – חיבור של גודל ה- Data עם גודל ה- UDP Header וה- IP Header

Frag2

- IP Header = 20B
- Data = 510B – כל המידע שנותר להעביר
- Offset = 185 – $\frac{1480}{8} = 185$
- DF = 0

- MF = 0 – כי זה הפרגמנט האחרון בקבוצה ואין עוד פרגמנטים אחריו.
- ID = x - שייך לחבילה המקורית שהמזהה שלה הוא x.
- Length = 530B – חיבור של גודל ה- Data שנותר לשלוח עם ה- IP Header.

בין R1 ל- R2 אין צורך בפרגמנטציה משום שה- MTU זהה.

מ- R2 ל- B מתבצעת פרגמנטציה עבור Frag1 משום שהMTU הנתון כעת הוא 660 והחבילה גדולה מידי, נבצע פרגמנטציה באופן הבא :

<u>Frag1A</u>

- IP Header = 20B
- UDP Header = 10B
- Data = 630B כי נתון שה-MTU הוא 660B ולכן יחד עם ה-IP Header וה-UDP Header נקבל 660B

- Offset = 0 – משום שזהו הפרגמנט הראשון.
- DF = 0
- MF = 1 – דולק בכדי לציין שזהו לא הפרגמנט האחרון בקבוצה.
- ID = x - שייך לחבילה המקורית שהמזהה שלה הוא x.
- Length = 660B – חיבור של גודל ה- Data עם גודל ה- UDP Header וה- IP Header
- Data = 630B

<u>Frag1B</u>

- Offset = 80 – $\frac{640}{8} = 80$
- DF = 0
- MF = 1 – דולק בכדי לציין שזהו לא הפרגמנט האחרון בקבוצה.
- ID = x - שייך לחבילה המקורית שהמזהה שלה הוא x.
- Length = 660B – חיבור של גודל ה- Data עם גודל ה- IP Header
- Data = 640B

<u>Frag1C</u>

- Offset = 160 – $\frac{1280}{8} = 160$
- DF = 0
- MF = 1 – דולק בכדי לציין שזהו לא הפרגמנט האחרון בקבוצה.
- ID = x - שייך לחבילה המקורית שהמזהה שלה הוא x.
- Length = 230B – חיבור של גודל ה- Data עם גודל ה- IP Header
- Data = 210

Frag2 מהפרגמנטציה הקודמת תישלח כך:

- Offset = 185 – $\frac{1480}{8} = 185$
- DF = 0
- MF = 0 – כי זה הפרגמנט האחרון בקבוצה ואין עוד פרגמנטים אחריו.
- ID = x - שייך לחבילה המקורית שהמזהה שלה הוא x.
- Length = 530B – חיבור של גודל ה- Data עם גודל ה- IP Header
- Data = 510B

TCP

פרוטוקול TCP שואף למזער פרגמנטציה ולכן ישתמש בתהליך שנקרא Discovery MTU Path .
בתהליך זה מגלים את ה-MTU המינימלי לאורך המסלול כולו.

A ישלח ל- B חבילה על פי מגבלות הגודל הידועות כאשר דגל ה-DF דולק, כשהחבילה תגיע לערוץ בו ה-MTU קטן מהחבילה , במקום פרגמנטציה, תישלח בחזרה הודעת שגיאה בעזרת פרוטוקול ICMP ,שאומרת שהחבילה גדולה מדי ומה צריך להיות גודל החבילה המקסימלי. A יעדכן בהתאם את גודל ה-MTU, MSS וישלח בהתאם.

החבילה שA ישלח ל-B מורכבת כך:

- IP Header = 20B
- TCP Header = 20B
- Data = 1980

- Offset = 0
- DF = 1
- MF = 0
- ID = x – לכל חבילה יש מזהה אחר שהיא מקבלת ב- Header IP , נסמנו ב-x.
- Length = 2020B

בעזרת התהליך שתיארנו קודם נגלה מהו ה-MTU המינימלי לאורך המסלול כולו, שהרי הוא 660B ונשלח את החבילות באופן הבא:

Frag1

- IP Header = 20B
- TCP Header = 20B
- Data = 620B

- Offset = 0
- DF = 1
- MF = 0
- ID = x
- Length = 660B – גודל Data, TCP Header וה-IP Header

Frag2

- IP Header = 20B
- TCP Header = 20B
- Data = 620B

- Offset = 0
- DF = 1
- MF = 0
- ID = y
- Length = 660B – גודל Data, TCP Header וה-IP Header

Frag3

- IP Header = 20B
- TCP Header = 20B
- Data = 620B

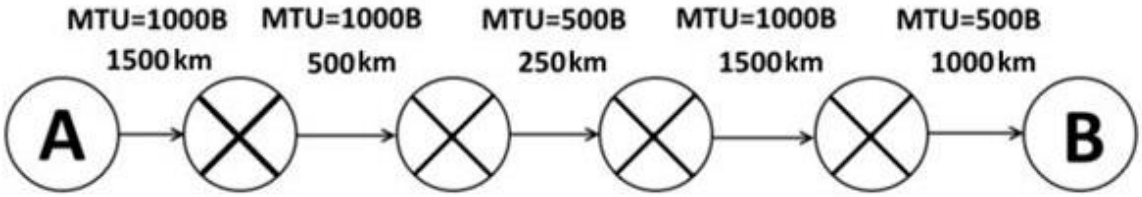
- Offset = 0
- DF = 1
- MF = 0
- ID = z
- Length = 660B – גודל Data, TCP Header וה-IP Header

Frag4

- IP Header = 20B
- TCP Header = 20B
- Data = 120B

- Offset = 0
- DF = 1
- MF = 0
- ID = w
- Length = 160B – גודל Data, TCP Header וה-IP Header

שאלה 2



במסלול בין A ל- B יש 4 נתבים, המרחקים של הערוצים וכמו כן ה MTU בערוצים מופיעים באיור לעיל ונניח ש-A שולח חבילה ל- B בגודל 2000 בתים בפרוטוקול UDP.

נתון:

- שידור FIFO בכולם הוא 1000KBps. מהירות ההתפשטות 250,000 ק"מ בשניה.
- במסלול בין A ל- B החבילה פוגשת ב- 4 חבילות (בגודל זהה 500 בתים) נוספות בכל נתב אי זוגי (כלומר הנתב הראשון והשלישי)
- ה MTU בערוץ הראשון , השני והרביעי (משמאל לימין) הוא 1000 בתים, ובערוץ השלישי והחמישי 500 בתים.
- הניחו כי גודל תחילית IP הוא 20 בתים, גודל תחילית UDP הוא 10 בתים (מקורב לשם נוחות), וגודל תחילית TCP הוא 20 בתים.

א. חשבו כמה זמן עד ש- B מקבל את החבילה.
ב. כמה בתים מקבל B כאשר נשלח בצורה זו קובץ בגודל 1,000,000 בתים (שמחולק ל- 500 חבילות באורך 2000 בתים)?
ג. כמה בתים מקבל B כאשר נשלח בצורה זו קובץ בגודל 1,000,000 בתים בפרוטוקול TCP?

פתרון

א. נתון ש-A שולח חבילה ל- B בגודל 2000 בתים בפרוטוקול UDP. כלומר נרצה לשלוח 2000B ועוד תחילית UDP שהיא 10B בסה"כ 2010B. ה- MTU בערוץ הראשון הוא 1000B ולכן עלינו לעשות פרגמנטציה.
נחלק באופן הבא:

<u>Frag1</u>
Data = 970B
UDP Header = 10B
IP Header = 20B

<u>Frag2</u>
Data = 980B
IP Header = 20B

<u>Frag3</u>
Data = 50B
IP Header = 20B

בעת הגעתנו לערוץ שבו ה- MTU הוא 500 נצטרך לבצע פרגמנטציה ל- Frag1 ו- Frag2,
Frag3 – מספיק קטן ואין צורך לבצע לו פירוק.

נפרק את Frag1 ל:

<u>Frag1A</u>
Data = 470B
UDP Header = 10B
IP Header = 20B

<u>Frag1B</u>
Data = 480B
IP Header = 20B

<u>Frag1C</u>
Data = 20B
IP Header = 20B

ואת Frag2 נפרק ל:

<u>Frag2A</u>
Data = 470B
UDP Header = 10B
IP Header = 20B

<u>Frag2B</u>
Data = 480B
IP Header = 20B

<u>Frag2C</u>
Data = 20B
IP Header = 20B

קיבלנו 7 פרגמנטים שגודלם כולל תחיליות הוא 2150 בתים.
נחשב את המרחק לפי התרשים המצורף ונקבל:

$$d = 1500 + 500 + 250 + 1500 + 1000 = 4750km$$

נתון שקצה ההתפשטות הוא 250,000KMps

לכן נקבל שהשהיית ההתפשטות היא:

$$d_{prop} = \frac{d}{s} = \frac{4750}{250000} = 0.019\ sec$$

$$d_{tran} + d_q = \frac{8(1000 + 3000 + 500 + 2500 + 2150)}{100 \cdot 10^3 \cdot 8} = \frac{9150}{100000} = 0.0915$$

$$d_{E2E} = 0.1105$$

ב. כאשר נשלח 500 חבילות בגודל 2000 בתים כל אחת נבצע את אותה פרגמנטציה שביצענו בעבור חבילה בגודל 2000 מהסעיף הקודם ולכן בסה"כ נשלח 2150*500 בתים שזה 1075000 בתים.

ג. כאשר נשלח קובץ בגודל 1,000,000 בפרוטוקול TCP, כל חבילה תהיה בגודל של הערוץ הקטן ביותר, כלומר בגודל 500 בתים
כל חבילה תכיל IP Header בגודל 20 בתים ו TCP Header בגודל 20 ולכן ה Data בכל חבילה יהיה 460 בתים לכל היותר.

$$\frac{1000000}{460} = 2173 + \frac{420}{460}$$

לכן בסה"כ נקבל 2173 חבילות שכל אחת מהן תכיל 460 בתים של מידע ועוד חבילה אחת שתכיל 420 בתים של מידע. לכל חבילה יש בנוסף 40 בתים של תחיליות ולכן נקבל 2173 חבילות של 500 בתים וחבילה אחת של 460 בתים.
בסה"כ נקבל: $2173 \cdot 500 + 460 = 1086960B$

שאלה 3

לקוח רוצה לשלוח הרבה מידע לשרת.

נתון:

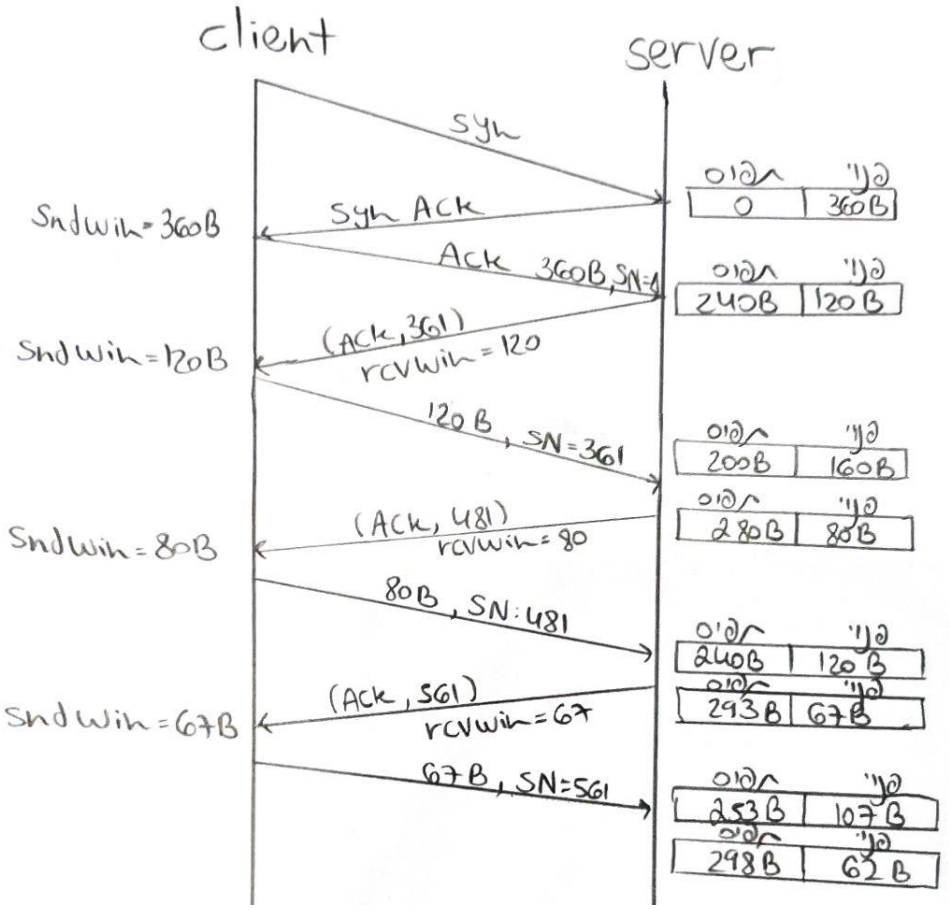
- הבאפר אצל השרת הוא 360B.
- משתמשים במספרים סידוריים יחסיים המתחילים מ- 0.
- ה- MSS=360B
- אין delayed Acks.
- האפליקציה קוראת בקצב שליש כאשר היא מקבלת מידע (על כל 3 בתים שהיא מקבלת היא מספיקה לקרוא 1) ,וכאשר היא לא מקבלת מידע (כלומר ,בזמן שלוקח לack להתפשט ול byte הראשון של החבילה הבאה להגיע), היא מספיקה לקרוא 40 בתים.
- אין שימוש במנגנונים שנועדו להתמודד עם syndrome window silly.

א. הראו באמצעות דיאגרמת חבילות (ללא חישוב זמנים) את הקמת החיבור ואת החבילות הנשלחות עד שהלקוח יודע בוודאות שהשרת קיבל לפחות 560 בתים. עבור כל החבילות יש לציין את השדות:
seq# , ack# , syn -ו ack דגלי , ReceiveWindow , Data length
כמו-כן, יש לצייר את באפר השליחה של הלקוח ואת באפר הקבלה של השרת לאורך הדיאגרמה.

ב. הסבירו את בעיית ה silly window syndrome כפי שמתבטאת בחלק א' ,והסבירו כיצד המנגנון המתאים היה פותר אותה.

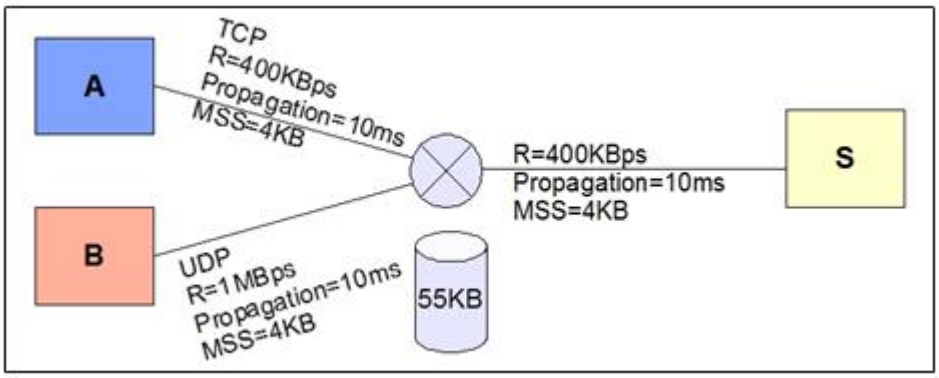
פתרון

א.



ב. בעיית ה silly window syndrome שמתבטאת בחלק א' היא שהאפליקציה קוראות בקצב איטי יותר מהקצב בו מגיע המידע. נתון לנו שהאפליקציה קוראת בקצב שליש כאשר היא מקבלת מידע, כלומר על כל שלושה בתים שהיא מקבלת היא קוראת אחד ולכן עלולה להיווצר תופעה בה כמות המידע ששולחים בכל חבילה הולך וקטן – מה שגורם לשליחת הרבה מאוד תחיליות.
בנוסף עד הרגע בו השרת שולח ACK ללקוח השרת קורא מהבאפר עוד 10 בתים ולכן בבאפר יש יותר מקום שאינו מנוצל לקבלת חבילה גדולה יותר.
המשמעות של זה היא הוספת זמן התפשטות הפוגע ביעילות העברת המידע.
על מנת למנוע זאת על השרת לא לפרסם חלון קטן מידי.
אבל, אם גודל החלון קטן מכך, הלקוח יפרסם חלון בגודל 0 ואז השרת יקבל את החלון בגודל 0, מה שימנע שליחה של חבילות קטנות מידי. כאשר השרת יפנה מקום בבאפר, הוא ישלח ACK עם גודל חלון עדכני, מה שיאפשר ללקוח להמשיך לשלוח.

שאלה 4

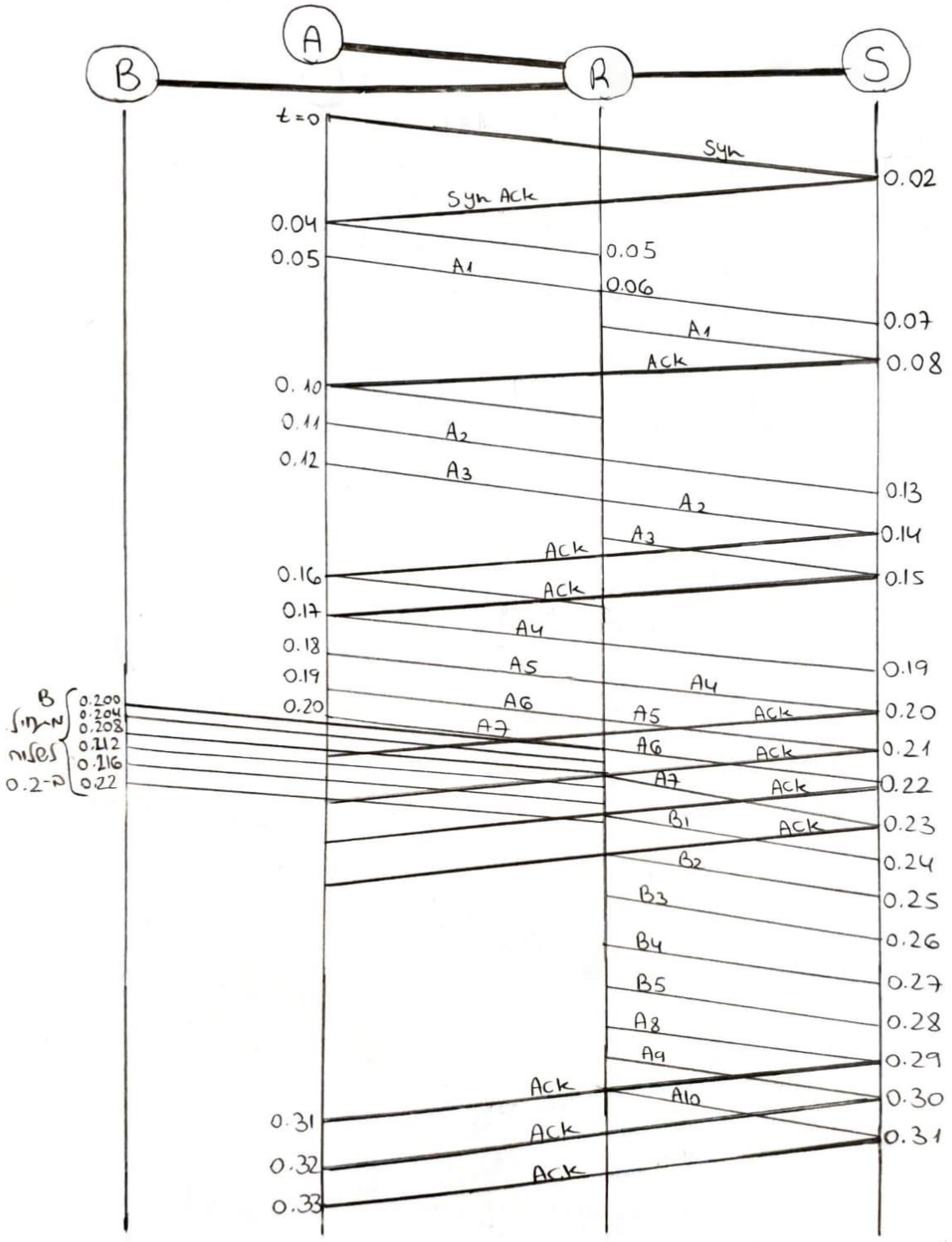


נתון:

- לקוח A מעלה קובץ בגודל 40KB לשרת S מעל TCP
- לקוח B מעלה קובץ בגודל 20KB לשרת S מעל UDP
- בזמן 0 לקוח A פונה לשרת S לצורך העלאת הקובץ בגודל 40KB מעל TCP, וכעבור 0.2 שניות הלקוח B מתחיל בהעלאת הקובץ שלו, קובץ בגודל 20KB לשרת S מעל UDP.
- התור בנתב בגודל 55KB
- הבאפר בשרת S בגודל 100KB
- ה- MSS בגודל 4KB
- אין delayed ACK
- קצב שידור של לקוח A, שרת S והנתב הוא 4000KBps, וקצב שידור של B הוא 1MBps
- Timeout = 0.1 seconds
- השהיית ההתפשטות בכל הערוצים 10 מילישניות.
- התעלמו מ- Headers ומהשהיית השידור של הודעות בקרה ACK.

הדגימו באמצעות תרשים חבילות וזמנים מזמן 0 ועד שהלקוח A יודע בוודאות שהשרת S קיבל את הקובץ בשלמותו. הציגו חישוב זמנים מפורט וחשבו במדויק את הזמן עד הרגע ש A-יודע בוודאות שהקובץ התקבל אצל השרת בהצלחה

פתרון



בשלב הראשון נראה כמה מידע A מספיק לשדר על 0.2 שניות שזהו הזמן בו B מתחיל לשדר.
A פותח חיבור (זמן של 4 השהיות התפשטות) A משדר לפי slow start ולכן בפעם הראשונה שולח חבילה אחת לאחר מכן מכפיל את גודל החלון ושולח שתי חבילות ושוב מכפיל את החלון ושולח 4 חבילות, כלומר הוא מספיק לשדר שלושה חלונות ראשונים ובעצם משדר 7 מתוך ה10 חבילות אותן הוא רוצה לשדר.
אנו מקבלים כי זמן פתיחת החיבור + פעמיים RTT + זמן שידור של 4 חבילות הוא 0.2

בזמן 0.2 B מתחיל לשדר , הוא שולח 5 חבילות. בזמן 0.214 החבילה הראשונה ש- B שלח מגיעה לנתב, הנתב R קיבל את כל החבילות שנשלחו מ- B בזמן 0.23 ובמקביל החבילה הרביעית בחלון השלישי של A , כלומר חבילה A7 מגיעה לנתב ב0.21 והנתב מסיים לשדר אותה ב0.22 ולאחר מכן החבילה מ-B משודרת מהנתב ב0.22. בזמן 0.22 A מקבל ACK על החבילה הראשונה מהחלון השלישי, כלומר חבילה A4 , אנו מקבלים כי עבור A זמן פתיחת חיבור יחד עם 3RTT הוא 0.22 מכאן שהחבילה השמינית ש- A רוצה מגיעה לנתב ב0.24 ולכן החבילה הזו יחד עם החבילות שנמצאות אחריה צריכות להמתין עד סיום שליחת החבילות מ-B . R התחיל לשדר את החבילות מ-B ב0.22 ולכן ב0.24 נותרו לו שלוש חבילות לשדר ורק בזמן 0.27 R יהיה פנוי לשדר חבילות מ-A והוא מסיים לשדר אותן ב0.3

נוסיף זמן של שלוש השהיות התפשטות כאשר אחת היא עבור החבילות שצריכות להגיע לשרת ושתיים נוספות עבור ה-ACK

מכאן שסה"כ הזמן הוא 0.33 שניות.

לא מתרחש timeout משום שהחבילה נשלחת מ-A בזמן 0.22 וה-ACK עליה מתקבל ב0.31 , כלומר לאחר 0.09 שניות ולפי הנתון timeout מתרחש רק לאחר 0.1 שניות.

שאלה 5

לקוח ושרת מחוברים ביניהם בעזרת ראוטר. לקוח פותח חיבור לשרת ושולח בקשה (קצרה) להורדת קובץ בגודל 16 KBמהשרת (הניחו זמן שידור הבקשה זניח, ולכן ניתן להתעלם ממנו).

נתון:

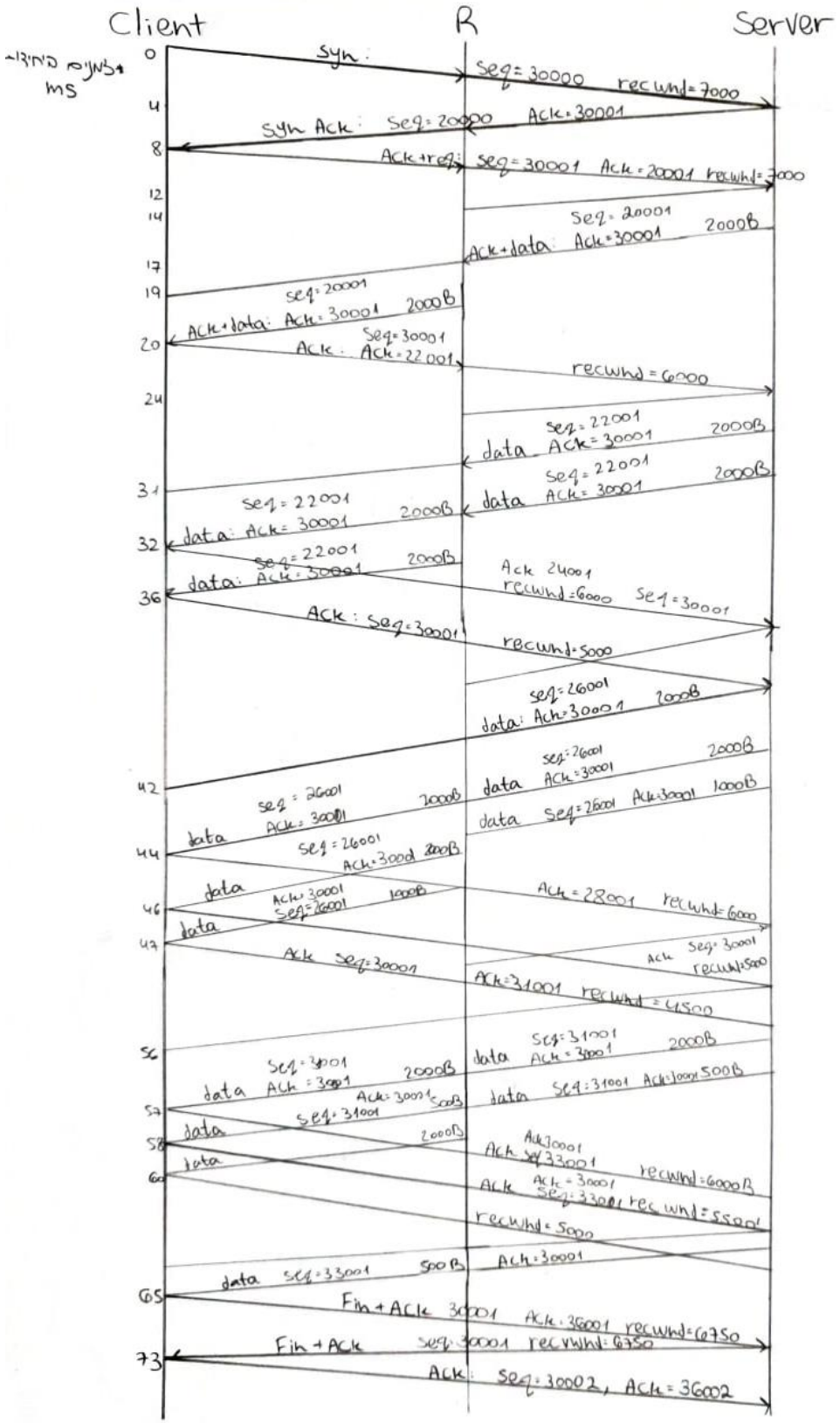
- קצב שידור בשני הערוצים (בשני כיוונים) הוא 1MBps
- מהירות התפשטות 250000 ק"מ לשניה.
- מרחק בין הלקוח לנתב הוא 250 ק"מ ובין נתב לשרת 750 ק"מ.
- ה- MSS הוא 2KB (התעלמו מ headers), ועם סיום קבלת החבילה תמיד נשלח ACK.
- הודעות ה- ACK קטנות וקצב שידור שלהן זניח (לכן ניתן להתעלם מזמן השידור שלהן)
- התור בנתב בגודל 4KB ואין תעבורה אחרת ברשת.
- הניחו שהאפליקציה בלקוח קוראת מה- buffer בקצב קבוע של 0.5MBps. גודל ה- receiveBuffer הוא 7KB

א. הראו דיאגרמת זמנים וחבילות בין השרת והלקוח כולל הקמת וסגירת החיבור (המספר הסידורי ההתחלתי בשרת הוא 20000 ובלקוח הוא 30000). על כל חבילה שנשלחת יש לציין את המספרים הסידוריים, גודל ה ,Window receive-דגלי, TCP זמן וחלון בקרת עומס.

ב. חשבו כמה זמן יקח להעביר את הקובץ.

פתרון

א.


Client
R
Server
זמנים מיוחדים ms
0
4
8
12
14
17
19
20
24
31
32
36
42
44
46
47
56
57
58
60
65
73
syn: seq=30000 recwnd=7000
syn Ack: seq=20000 Ack=30001
Ack+req: seq=30001 Ack=20001 recwnd=7000
Ack+data: seq=20001 Ack=30001 2000B
Ack+data: seq=20001 Ack=30001 2000B
Ack: seq=30001 Ack=22001
recwnd=6000
data seq=22001 Ack=30001 2000B
data seq=22001 Ack=30001 2000B
data: seq=22001 Ack=30001 2000B
data: seq=22001 Ack=30001 2000B
Ack 24001 recwnd=6000 seq=30001
Ack: seq=30001
recwnd=5000
data: seq=26001 Ack=30001 2000B
data seq=26001 Ack=30001 2000B
data seq=26001 Ack=30001 1000B
data seq=26001 Ack=30001 2000B
data Ack=30001 seq=26001 1000B
Ack=28001 recwnd=6000
data
Ack seq=30001
Ack seq=30001 recwnd=5000
Ack seq=30001
Ack=31001 recwnd=4500
data seq=31001 Ack=30001 2000B
data seq=31001 Ack=30001 500B
data seq=31001 Ack=30001 500B
data seq=31001 2000B
data
Ack=30001 seq=33001 recwnd=6000B
Ack=30001 seq=33001 recwnd=5500
Ack
recwnd=5000
data seq=33001 500B
Ack=30001
Fin+Ack 30001 Ack=36001 recwnd=6750
Fin+Ack seq=30001 recwnd=6750
Ack: seq=30002, Ack=36002

ב. נחשב כמה זמן ייקח להעביר את הקובץ:

$$d_{p,client-R} = \frac{250Km}{250000Kmps} = 1ms$$

$$d_{p,server-R} = \frac{750Km}{250000Kmps} = 3ms$$

$$d_t = \frac{2KB}{1MBps} = 2ms$$

הקמת החיבור ושליחת הבקשה מתרחש בזמן של שלוש פעמים התפשטות מקצה לקצה, כלומר 12ms.
שידור החלון הראשון בשני הערוצים לוקח 4ms בשני הערוצים ובנוסף התפשטות וגם התפשטות ACK לוקח עוד 12ms וסה"כ כעת 24ms. נתון לנו שהאפליקציה בלקוח קוראת מה- buffer בקצב קבוע של 0.5MBps כלומר היא קוראת בקצב של חצי ממה שהיא מקבלת ומאחר שחישבנו שלוקח לה לקבל 2KB ב 2ms אז לוקח לה 2ms על מנת לקרוא 1KB. לכן מרגע שידור החלון הראשון ועד זמן של 24ms האפליקציה סיימה לקרוא את כל המידע מה- buffer.
בתחילת שידור החלון השני (כעת בחלון יש שתי חבילות כי הוא הוכפל) נשים לב כי תוך כדי קבלת הACK על החבילה הראשונה בחלון (חבילה מספר 2 בסה"כ) חבילה השנייה בחלון (חבילה מספר 3) מגיעה ולכן אין חבילות בבאפר של הנתב וגם עד הגעת החבילה השלישית ללקוח האפליקציה תסיים לקרוא את המידע מהבאפר.
בשידור החלון השלישי, ה ACK על החבילה השנייה מגיע עם חלון בגודל של 6KB וה-ACK על החבילה שאחריה מגיע עם חלון בגודל של 5KB, מכאן שניתן לשדר 5KB. נתון שקצב השידור זהה בשני הערוצים ולכן בזמן שהחבילה השנייה בחלון הגיעה לנתב, הראשונה שודרה מהנתב ולכן לא תהיה בעיית מקום בבאפר של הנתב. ה-ACK על החבילה הראשונה בחלון מגיע לאחר 48ms ומשום שלאפליקציה היה מספיק זמן לרוקן את הבאפר נקבל חלון בגודל של 6KB ה- ACK הבא מתקבל ב 50ms עם חלון של 5KB, ה-ACK השלישי מגיע כבר בזמן של 51ms עם חלון של 4500B.
בזמן 48 מילישניות משדרים שני סגמנטים מלאים עד זמן 52, בסהכ השרת שלח ללקוח עד עכשיו 15 קילו אבל נשים שניתן לשדר רק עוד 500B בגלל גודל החלון של בקרת הזרימה. נקבל Ack על החבילה בזמן 60 ונשדר את כל המידע שנשאר השידור ועוד ההתפשטות וההתפשטות של ack יסתיימו לאחר 69 מילישניות.

שאלה 6

לארגון A ישנו ראוטר R1 אשר מחובר בערוץ ישיר לראוטר R2 של ארגון B. קצב השידור של הערוץ הינו 100MBps בתוך ארגון A ישנם 10 לקוחות ובתוך ארגון B ישנו שרת שיכול לטפל בלקוחות בו זמנית .
בשאלה זו נתעלם מכל ההשהיות בתוך הרשתות המקומיות. המרחק בין R1 לבין R2 הוא 2500m ומהירות ההתפשטות היא $2.5 * 10^8$ מטר לשנייה.
בשרת יש 10 קבצים שונים, כל אחד בגודל 10KB
נניח MSS=1KB
בשאלה זו ניתן להזניח את זמן השידור של התחיליות והודעות בקרה.
כמו כן נניח שה timeout = 1s
כל הלקוחות מתחילים בפניה בו-זמנית אל השרת להורדה של קובץ אחד כל אחד על גבי TCP.

א. כמה זמן עובר עד שהשרת יודע שהמחשב האחרון קיבל את הקובץ?

ב. חשבו את גודל החוצצים המינימלים בנתבים R_1, R_2 על מנת שלא יהיה אובדן חבילות.

פתרון

א. הלקוחות מקימים חיבורים נפרדים מול השרת בו זמנית, נתון כי ניתן להזניח את זמן השידור של התחיליות ולכן נחשב את זמן ההתפשטות שלהן בין הנתבים.

$$d_p = \frac{2500m}{2.5 \cdot 10^8 mps} = 10^{-5}s$$

מאחר ונתון שגם זמן השידור זניח אז ההתפשטויות מתרחשות במקביל ולכן ניתן לחשב רק עבור חיבור אחד.
נחשב כמה זמן לוקח לשרת לשדר חבילה אחת:

$$\frac{1KB}{100MBps} = 10^{-5}s$$

נשים לב, שכאשר החבילה השלישית משודרת החבילה הראשונה הספיקה להתפשט והאישור עליה מתפשט בחזרה, לכן שתי חבילות חדשות יגיעו מהשרת לנתב ולכן הנתב ישדר ב full pipe כל הדרך.
נחשב כמה זמן לוקח לשדר 100KB:

$$\frac{100 \cdot 1KB}{100MBps} = 1ms$$

נוסיף את השהיית ההתפשטות של החבילה האחרונה ושל ה- ack שלה, נקבל:

$$3 \cdot 10^{-5} + 1 + 2 \cdot 10^{-5} = 1.05\, ms$$

ב. לפי נתוני השאלה R1 לא צריך חוצץ גדול, אבל R2 כן צריך משום שבהתחלה יש בחוצץ שלו 10KB , לאחר שידור של 3 חבילות מתקבל אישור על החבילה הראשונה ולכן מרגע זה בכל פעם נוספות שתי חבילות חדשות לחוצץ, אבל רק אחת תהיה משודרת עד האישור הבא.
לכן אם ישנן 7 חבילות בחוצץ, ו 45 ובמשך 45 האישורים הבאים יתווספו כל פעם עוד חבילה לחוצץ ויש עוד 90 חבילות לשדר, אז מספר החבילות המקסימלי שימתינו הוא 52 חבילות ולכן גודל הבאפר הוא 52KB.

שאלה 7

אפליקציה מסויימת מתקשרת בעזרת ערוץ לוויני על גבי מרחק של 60,000Km כאשר באמצע אין נתבים. מהירות ההתפשטות היא 300,000Km/s. כלומר השהיית ההתפשטות היא $d_p = \frac{60000Km}{300000Km/s} = 0.2\ sec$

נניח שלמקבל יש באפר בגודל 200,000B

א. נניח שקצב השידור הוא 10MBps והאפליקציה קוראת מידע מהבאפר בקצב קבוע של 8MBps כיצד ניתן לשפר את קצב העברת המידע?

ב. מהו החסם העליון על גודל הבאפר שיכול להיות מפורסם?

פתרון

א. המקבל יכול לפרסם חלון גדול יותר מגודל הבאפר שיש לו באמת וכך המידע יוכל להגיע מהר יותר ולמרות שאין בפועל גודל באפר שכזה האפליקציה מרוקנת כל הזמן את הבאפר ומפנה מקום.

ב. נסמן ב – R את הקצב שבו מגיע המידע
נסמן ב – O את הקצב שהאפליקציה קוראת מהבאפר
נסמן ב – W את גודל הבאפר
ונקבל כי גודל החלון שניתן לפרסם הוא: $\frac{W}{1-\frac{O}{R}}$
נשים לב שאם השולח ישלח מידע בגודל הבאפר ויחכה לתשובה נקבל כי קצב העברת המידע הוא:
$\frac{200000B}{RTT} = \frac{200000B}{0.4s} = 500KBps$

נציב בנוסחה ונקבל $\frac{200000B}{1-\frac{8Mbps}{10Mbps}} = 10^6 B$ ולכן נקבל קצב של $\frac{10^6}{0.4} = 2.5MBps$