מגישים: 200878627 יניב לידן
204736961 דן אברהם

תקשורת – עבודה 3

1. אורך הודעת SYN הוא 15B.
גודל הזיכרון במחשב הוא 2GB.
זמן למילוי הזיכרון הוא 30 sec.

$N = \frac{2GB}{32KB} = \frac{2^{31}B}{2^{15}B} = 2^{16}B$, $T = \frac{L}{R} = \frac{15B}{R}$

$T * N = 30sec$ → $\frac{2^{16}*15B}{R} = 30sec$ →

$R = 2^{15}Bps = 2^{18}bps = 256Kbps$

2. א. <u>שימוש מסחרי:</u> מניעת שירותים ספציפיים של שיתוף קבצים בין משתמשים ברשת בכך שמזייפים חבילות TCP הגורמות לשני הצדדים להאמין שהצד השני אינו מעוניין להמשיך ב-session.
<u>שימוש פוליטי:</u> שימוש בפרוטוקול לשם צנזור הרשת של סין ע"י הממשל.

ב. הבודקים של IEE הצליחו לזהות שחברת Comcast "שתלה" חבילות מזויפות בתעבורה שבין שני המשתמשים באמצעות שימוש ב-Wireshark וניסו להוריד קבצי Bittorent ממנויים ב-Comcast. בעזרת Wireshark הם עקבו אחר תעבורת החבילות משני הצדדים. כיוון שהגיעו לאחד הצדדים הודעות עם חתימת השולח אך בפועל השולח מעולם לא שלח אותן, הם הסיקו כי הייתה התערבות מגורם שלישי.

ג. מה שביצעה Comcast פוגע בעקרון ה end-to-end ע"י כך שהוא פוגע בעקרונות יסודיים של חופש השימוש באינטרנט עליהם מושתת האינטרנט. פגיעה זו נעשית בכך שמעשיה מאיימים ליצור סיטואציה שבה ייאלצו להשיג הרשאות מחברת ISP על מנת להבטיח כי הפרוטוקולים שלהם יעבדו כמו שצריך.

ד. יכולה להיווצר בעיה כזו שחברה כמו Comcast יכולה לנתר חבילות של שירותים מתחרים שלה ולפגוע בתעבורת הנתונים שלהם ובכך לחבל בתחרות בין שני הגופים.

ה. האינטרנט של פרטנר עובד על התשתית של בזק, ושתיהן מתחרות בתחום ספקיות האינטרנט. בזק יכולה לנתר ולשבש תעבורה של חבילות של לקוחות פרטנר ובכך להבטיח לעצמה יתרון משמעותי ב"איכות השירות" בין המתחרות.

3. א.

$$SentInfo = 1 + 2 + 4 + 8 \ldots + N = \frac{MSS*(2^{\log_2 N+1}-1)}{2-1} = (2N-1) * MSS$$

ב. השליחות עולות אקספוננציאלית, לכן הזמן יקטן לוגריתמית:

$$Time = \log_2 N * \tau$$

ג.

$$Segments = 1 + 2 + 4 + 8 + \cdots + N + 2N = \frac{1*(2^{\log_2 2N}+1-1)}{2-1} = 2 * 2N - 1 = 4N - 1$$

$$Time = (\log_2 2N) * \tau = (1 + \log_2 N) * \tau$$

מגישים: 200878627 יניב לידן
204736961 דן אברהם

ד. בקשר של סעיף ב': ייקחו 3 איטרציות (3τ) עד להעברת N MSS. כלומר הזמן בסעיף ב' לוקח פי: $\frac{\log_2 N}{3} = \frac{\log_2 N}{\log_2 8} = \log_2(\frac{N}{8})$.
בקשר של סעיף ג': ייקחו 4 איטרציות (4τ) עד להעברת 2N MSS. כלומר הזמן בסעיף ג' לוקח פי: $\frac{1+\log_2 N}{4} = \frac{\log_2 2N}{\log_2 16} = \log_2(2N-16) = 1 + \log_2(\frac{N}{8})$.

ה. כדאי להסתכן ולהשתמש בחלון בגודל גדול מ-1 כשנרצה להעביר מידע בכמה שפחות חלונות. הסיכון יהיה יעיל רק אם נדע שאנחנו לא שולחים יותר סגמנטים בחלון מאשר מה שהצד השני יכול לקבל.
כאשר אין לנו הרבה מידע להעביר, או כאשר אין לנו מידע על כמות הסגמנטים שהצד השני יכול לקבל אין תועלת בסיכון זה.

4. א. <u>נתונים:</u> גודל רישא מקסימלי בTCP – 60 Bytes, גודל רישא של שכבת הרשת – 20 Bytes, גודל הרישא של שכבת הקו – 20 Bytes, גודל ה-Data של המשתמש – 1 Byte.

$segment = 60 + 20 + 20 + 1 = 101\ Bytes \rightarrow$

תקורה = $\frac{1B}{101B} = \frac{1}{101}$

ב. עפ"י האלגוריתם של נייגל יישלח סגמנט ראשון ועד לקבלת ACK נצבור את ההקלדות עד שיגיע לגודל המקסימלי של הסגמנט ורק אז יישלח הסגמנט ובכך התקורה של הרישות תעלה בכל הודעה.
עפ"י האלגוריתם של קלארק מקבל ההודעה יבקש מהשולח שיענה על הנוסחה הבאה:
$window \geq \min\{MSS, \frac{BufferSize}{2}\}$, הנוסחה מבטיחה שהשולח לא יקבל ACK עד שלמקבל לא יהיה מקום פנוי בBuffer לקבל הודעה בגודל זה. הדבר מגדיל את כמות המידע שנשלח בכל הודעה ולכן התקורה תעלה בכל הודעה.

ג. אם ה-RTT גדול יכול להיווצר מצב שהשולח עלול לחכות זמן רב ובכך לגרום לסגמנט הבא לגדול ולעכב את משלוח ההודעה למרות שמקבל ההודעה פנוי לקבלת ההודעה.

מגישים: 200878627 יניב לידן
204736961 דן אברהם

5.

No.	Time	Source	Destination	Protocol	Length	Info
101	5.764499	132.73.214.120	172.217.21.206	TCP	66	51363 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
127	5.841490	172.217.21.206	132.73.214.120	TCP	66	443 → 51363 [SYN, ACK] Seq=0 Ack=1 Win=60720 Len=0 MSS=1380 SACK_PERM=1 WS=256
128	5.841639	132.73.214.120	172.217.21.206	TCP	54	51363 → 443 [ACK] Seq=1 Ack=1 Win=66048 Len=0
129	5.842004	132.73.214.120	172.217.21.206	TLSv1.3	571	Client Hello
155	5.922295	172.217.21.206	132.73.214.120	TCP	54	443 → 51363 [ACK] Seq=1 Ack=518 Win=61952 Len=0
156	5.924341	172.217.21.206	132.73.214.120	TCP	54	[TCP Dup ACK 155#1] 443 → 51363 [ACK] Seq=1 Ack=518 Win=61952 Len=0
158	5.927838	172.217.21.206	132.73.214.120	TLSv1.3	1484	Server Hello, Change Cipher Spec
159	5.929213	172.217.21.206	132.73.214.120	TCP	1484	443 → 51363 [ACK] Seq=1431 Ack=518 Win=61952 Len=1430 [TCP segment of a reassembled PDU]
160	5.929454	132.73.214.120	172.217.21.206	TCP	54	51363 → 443 [ACK] Seq=518 Ack=2861 Win=66048 Len=0
168	5.932397	172.217.21.206	132.73.214.120	TLSv1.3	926	Application Data
170	5.933751	132.73.214.120	172.217.21.206	TLSv1.3	118	Change Cipher Spec, Application Data
206	6.008856	172.217.21.206	132.73.214.120	TLSv1.3	556	Application Data
207	6.009709	172.217.21.206	132.73.214.120	TLSv1.3	116	Application Data
208	6.009925	132.73.214.120	172.217.21.206	TCP	54	51363 → 443 [ACK] Seq=582 Ack=4297 Win=66048 Len=0
5342	51.010283	132.73.214.120	172.217.21.206	TCP	55	[TCP Keep-Alive] 51363 → 443 [ACK] Seq=581 Ack=4297 Win=66048 Len=1
5345	51.088132	172.217.21.206	132.73.214.120	TCP	66	[TCP Keep-Alive ACK] 443 → 51363 [ACK] Seq=4297 Ack=582 Win=61952 Len=0 SLE=581 SRE=582
7640	73.497692	132.73.214.120	172.217.21.206	TCP	54	51363 → 443 [FIN, ACK] Seq=582 Ack=4297 Win=66048 Len=0
7641	73.497738	132.73.214.120	172.217.21.206	TCP	54	51363 → 443 [RST, ACK] Seq=583 Ack=4297 Win=0 Len=0
7652	73.590829	172.217.21.206	132.73.214.120	TCP	54	443 → 51363 [FIN, ACK] Seq=4297 Ack=583 Win=61952 Len=0

א.

No.	Time	Source	Destination	Protocol	Length	Info
101	5.764499	132.73.214.120	172.217.21.206	TCP	66	51363 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
127	5.841490	172.217.21.206	132.73.214.120	TCP	66	443 → 51363 [SYN, ACK] Seq=0 Ack=1 Win=60720 Len=0 MSS=1380 SACK_PERM=1 WS=256
128	5.841639	132.73.214.120	172.217.21.206	TCP	54	51363 → 443 [ACK] Seq=1 Ack=1 Win=66048 Len=0

ניתן לראות את לחיצת הידיים המשולשת:
I. ה-IP 132.73.214.120 שלח פקטה עם SYN ל-IP 172.217.21.206. (בקשה לקשר TCP).
II. ה-IP 172.217.21.206 החזיר פקטה עם SYN ו-ACK חזרה ל-IP 132.73.214.120. (אישור ופתיחה חזרה).
III. ה-IP 132.73.214.120 השיב ACK ל-IP 172.217.21.206 (אישור סופי לפתיחת הסשן).

ב. ניתן לראות כי בבקשה לפתיחת הסשן ובהחזרה של הSYN, ACK (כל פעם שנשלח SYN) נשלח איתו flag ובו מצוין SACK_PERM=1. כלומר גם השרת וגם המחשב תומכים ב-SACK.

ג. ניתן לראות כי בכל שליחת SYN עוד דגל שנשלח הינו WINDOW ואחריו מצוין גודל החלון. גודל החלון המקסימלי במחשב שלי (בקשה ראשונה) הינו 64240 ובשרת (בקשה שניה) 60720.

ד.

1628	13.718611	132.73.214.120	172.217.18.3	TCP	54	51548 → 443 [FIN, ACK] Seq=582 Ack=3733 Win=65280 Len=0
1633	13.799348	172.217.18.3	132.73.214.120	TLSv1.3	556	Application Data
1635	13.799608	132.73.214.120	172.217.18.3	TCP	54	51548 → 443 [RST, ACK] Seq=583 Ack=4235 Win=0 Len=0
1638	13.800292	172.217.18.3	132.73.214.120	TLSv1.3	116	Application Data
1639	13.800292	172.217.18.3	132.73.214.120	TCP	54	443 → 51548 [FIN, ACK] Seq=4297 Ack=583 Win=61952 Len=0
3165	50.719129	132.73.214.120	172.217.18.99	TCP	55	[TCP Keep-Alive] 51542 → 443 [ACK] Seq=998 Ack=4763 Win=65792 Len=1
3166	50.735296	132.73.214.120	172.217.21.206	TCP	55	[TCP Keep-Alive] 51546 → 443 [ACK] Seq=581 Ack=4296 Win=66048 Len=1
3167	50.735298	132.73.214.120	216.58.210.10	TCP	55	[TCP Keep-Alive] 51544 → 443 [ACK] Seq=581 Ack=4485 Win=66048 Len=1
3168	50.743236	132.73.214.120	216.58.206.13	TCP	55	[TCP Keep-Alive] 51545 → 443 [ACK] Seq=581 Ack=3619 Win=65280 Len=1
3169	50.805341	172.217.18.99	132.73.214.120	TCP	66	[TCP Keep-Alive ACK] 443 → 51542 [ACK] Seq=4763 Ack=999 Win=62976 Len=0 SLE=998 SRE=999
3170	50.810738	172.217.21.206	132.73.214.120	TCP	66	[TCP Keep-Alive ACK] 443 → 51546 [ACK] Seq=4296 Ack=582 Win=61952 Len=0 SLE=581 SRE=582
3171	50.811491	216.58.210.10	132.73.214.120	TCP	66	[TCP Keep-Alive ACK] 443 → 51544 [ACK] Seq=4485 Ack=582 Win=61952 Len=0 SLE=581 SRE=582
3172	50.817364	216.58.206.13	132.73.214.120	TCP	66	[TCP Keep-Alive ACK] 443 → 51545 [ACK] Seq=3619 Ack=582 Win=61952 Len=0 SLE=581 SRE=582

היזימה של סגירת הקשר נעשתה ע"י המחשב שלי (IP 132.73.214.120), ששלח FIN, ACK לשרת (דרך הראוטר של האוניברסיטה, IP 172.217.18.3)לסיום הקשר.
השרת החזיר FIN, ACK גם הוא.
ניתן לראות גם כי המחשב השאיר את הקשר חצי פתוח. (בקשות Keep-Alive בין המחשב שלי לשרת)

6. רעיון הצפנת הheader אינו רעיון יעיל כיוון שה-header עובר במספר תחנות שאינן ידועות מראש והן משתמשות ב-header על מנת לדעת כיצד לטפל בחבילה.
הצפנת ה-header תאלץ שליחת מפתח להצפנה עבור כל התחנות בהן החבילה עלולה לעבור ועל כן המפתח יהיה בכל אחת מהתחנות ברשת. לכן רעיון זה אינו יעיל כלל.

מגישים: 200878627 יניב לידן
204736961 דן אברהם

7. א. שלושה יישומים בהם תומכים מרכזי נתונים מודרניים הם:

I. Query traffic
II. Short messages (דורש זמן השהייה נמוך)
III. long flows (דורש תפוקה גדולה)

ב. 1. במרכזי נתונים empty queue RTT לוקחות בעקביות פחות מ-250ms. לעומת רשתות לטווחים גדולים שם הבקשות לוקחות יותר.
2. במרכזי נתונים, לעומת WAN, לעיתים תכופות יש מעט multiplexing סטטיסטי.
3. במרכזי נתונים, לעומת WAN, הסביבה מאוד הומוגנית ותחת יחידה אדמיניסטרטיבית יחידה.

ג. השאילתה תישלח להרבה aggregators ו-workers כך שאם השאילתה תחולק למספר חלקים כל worker יהיה אחראי לחלק אחר של האינדקס. לאחר מכן חוזרת תשובה ל-aggregator האחראי אותה הוא יוכל לתקן ולשלוח מחדש על מנת לשפר את התשובות או להעבירן הלאה. כלל התשובות מתאחדות לתשובה אחת.
במידה ושרת לא מחזיר תשובה מתעלמים ממנו בשקלול ואיחוד התשובות לתשובה סופית.
לא ניתן לעכב מעט את התשובה לשאילתה כיוון שעיכוב כזה יכול לגרור תשובה פחות איכותית לשאילתה או לגרור עיכוב גדול בהחזרת התשובה למשתמש הקצה, מה שיוריד מאיכות השירות של המנוע ויגרום לאובדן משתמשים. על מנת למנוע מצב זה קובעים מראש זמן להחזרת תשובות של ה-aggregators וה-workers.

ד. הפתרון בו השתמשה Facebook כדי להתגבר על הקושי הרב לעמוד ב"דדליינים" קפדניים בקשרי TCP היה פיתוח UDP-based congestion control משל עצמה.

8. 1. חוק אמדל הוא חוק למציאת חסם עליון לשיפור הצפוי במערכת מחשב שחלקה מקבילי וחלקה סדרתי. חוק זה משמש בעיקר לחיזוי החסם העליון להאצה המקסימלית שהרצת מספר מעבדים על המערכת תספק. הנוסחה לחוק אמדל:

$$S_{latency}(s) = \frac{1}{(1-p)+\frac{p}{s}}$$

2. א. <u>נתונים:</u>

$$fileSize = 99Kb, headerSize = 1KB, t_{proc}(router) = 1ms, t_{proc}(server) = 1ms,$$

$$t_{queue} = 0, distance = 10{,}000km, bandwidth_{client2ro} = 1Gbps,$$

$$bandwidth_{router2server} = 1Mbps$$

למשתמש ישנו קשר TCP פתוח עם השרת. נזניח את הזמן בין הנתב ללקוח כיוון שהוא קטן מ1.5%.

$$D = 3ms + \frac{2*10^7 m\sec}{2*10^8} + \frac{100Kb\ sec}{1Mb} = 3ms + 100ms + 100ms = 203ms$$

ב.

$$D = 3ms + 100ms + \frac{2Kb\ sec}{1Mb} + \frac{10^5\ sec}{10^9} = 105.1ms$$

מגישים: 200878627 יניב לידן
204736961 דן אברהם

ג. החסכון היחסי:

$\frac{(203+1 \quad .1)ms}{203} = 1.5177 \rightarrow 0.5177$

.3

$$D = 3ms + \frac{2*10^7 m\sec}{2*10^8} + \frac{200Kb\ sec}{1Mb} = 3ms + 100ms + 200ms = 303ms$$

.4

$$D = 3ms + \frac{2*10^7 m\sec}{2*10^8} + \frac{109Kb\ sec}{1Mb} = 3ms + 100ms + 109ms = 212ms$$

.5

$$D = 201ms + \frac{2*10^7 m\sec}{2*10^8} + \frac{100Kb\ sec}{1Mb} = 201ms + 100ms + 100ms = 401ms$$

.6

$$D = 3ms + \frac{2*20^7 m\sec}{2*10^8} + \frac{100Kb\ sec}{1Mb} = 3ms + 200ms + 100ms = 303ms$$

.7

$$D = 3ms + \frac{2*10^7 m\sec}{2*10^8} + \frac{100Kb\ sec}{10Mb} = 3ms + 100ms + 10ms = 113ms$$