**סקירה וניתוח של פרוטוקול QUIC**

*מטלת גמר ברשתות תקשורת*

2024

יוכבד אופשטיין, מור ברגר, לינוי קאופמן, נועה שלום.

**חלק יבש**

**שאלה 1:**

נתאר 5 חסרונות ומגבלות של פרוטוקול TCP לפי מה שמתואר במאמר:

1. זיווג בין בקרת העומס (Congestion Control) לבקרת הזרימה (Flow Control):

מטרת בקרת העומס היא לשלוט על כמות החבילות אשר יכולות להיות ברשת. בקרת העומס משנה את גודל החלון לפי מצב הרשת, במצב של איבוד חבילות תהיה הגבלה על גודל החלון עד קבלת החבילות האבודות. TCP משתמש בחלון משותף עבור מנגנון זה וגם עבור מנגנוני בקרת הזרימה שנועדה להבטיח זרימה אמינה. מכיוון ששני המנגנונים משתמשים בחלון משותף יכול להיווצר מצב שבו יש המתנה ארוכה וניצול לא יעיל של הרשת.

1. חסימת ראש התור (HLB):

מכיוון שב-TCP יש מסירה מסודרת של נתונים ע"י העברת נתונים בסדר שבו הם נשלחו, אובדן של חבילה אחת תוביל לעיקוב של קבלת כל החבילות שבאות אחריה בסידור.

1. עיכוב עקב התהליך יצירת קשר:

כדי לייצר קשר בין שני נקודות קצה בעזרת TCP יש צורך בלחיצת יד משולשת. תהליך זה מאוד הכרחי לסדר ואמינות העברת נתונים אך גורמת לעיכוב משמעותי. בנוסף, אם יש הוספת אבטחה בעזרת TLS אז נדרש עוד סבב נוסף שלהחלפת אישורי אבטחה מה שמוסיף עוד RTT.

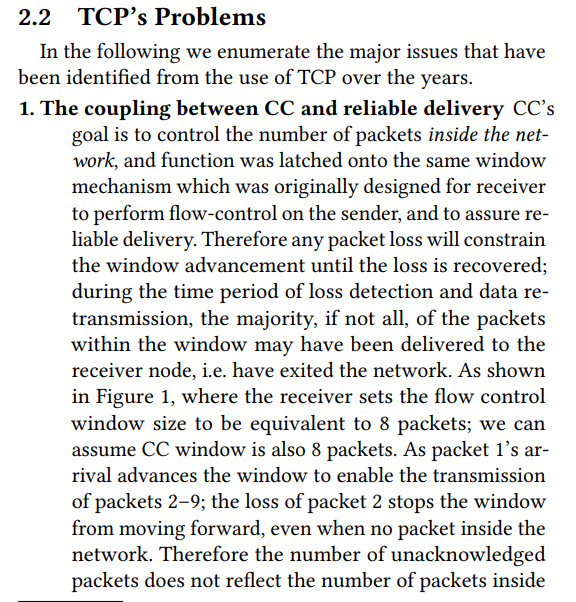
1. מגבלות של כותרת (Header) קבוע:

ה-Header של TCP מורכבת משדות בגודל קבוע, כולל שדה אפשרויות המוגבל ל-40 בתים. השדות של ה-Sequence Number ושל ה-ACK בגודל 4 בתים, ושל חלון בקרת הזרימה בגודל 2 בתים, מגבילים את הביצועים של TCP במהירויות רשת גבוהות. ה-Sequence Number וה-ACK מתמלאים מהר מדי, וגודל החלון הקטן מגביל את קצב השידור המרבי של החיבור.

1. מזהה ייחודי (ID) לחיבור וכתובת IP ייחודי:

ה- TCP מזהה את החיבור לפי שילוב של כתובת IP ומספר ה- PORT, אבל כתובת ה-IP או המספר PORT של אחת נקודות הקצה יכול להשתנות במהלך החיבור TCP הקיים ממספר סיבות שונות. לכן כל שינוי של אחד מהנתונים האלו ישבור את החיבור הקיים, מה שיוביל לאיבוד של מידע.

בנוסף, על מנת לחדש קשר עם IP חדש או PORT חדש נצטרך לחיצת יד משולשת.



A black text on a white background

Description automatically generatedA screenshot of a document

Description automatically generated

A page of a document

Description automatically generatedA black text on a white background

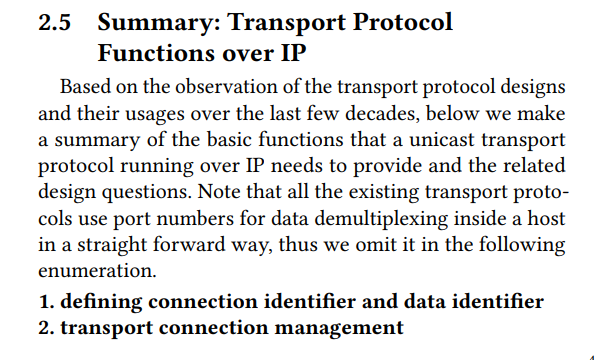
Description automatically generatedA diagram of a network

Description automatically generated

**שאלה 2:**

נתאר 5 תפקידים שפרוטוקול תעבורה צריך למלא:

1. הגדרת המזהה הייחודי של הקשר
2. ניהול החיבור של התעבורה:
3. מזהה החיבור ייחודי גלובלי שמקשר בין שני קצוות החיבור, חייב למפות את הכתובות IP בצורה אמינה.
4. הקמת מצב חיבור (Connection State) ופירוקו בסוף הקשר
5. בקרת החילוף מידע בין שני הקצוות. נרצה שיהיה גמיש.
6. תמיכה בשינוי כתובת IP.
7. אמינות של העברת החבילות:
8. מזהה נתונים ייחודי שמועבר בצורה אמינה לקצה השני.
9. בקרת זרימה בחלון. נרצה להימנע מחסימת ראש התור (HLB).
10. בקרת עומס על הרשת:
11. נרצה לשלוט על כמות החבילות שנמצאים ברשת בכל רגע נתון.
12. אבטחת המידע:
13. הצפנת ואבטחת המידע שעובר ברשת. (TLL מספקים רק סודיות נתונים)

 A close up of a text

Description automatically generated

A close up of a text

Description automatically generated

**שאלה 3:**

בשאלה זו נתאר את אופן פתיחת הקשר ב QUIC והיכן הוא משפר את חלק מהחסרונות של TCP:

**אופן פתיח הקשר בפרוטוקול QUIC:**

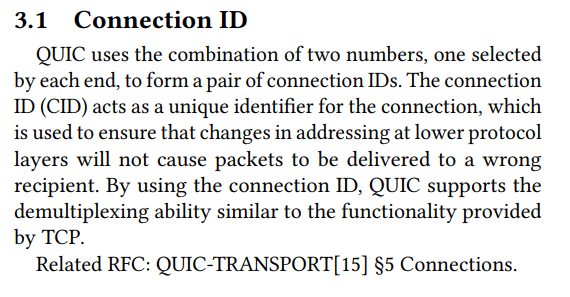
פרוטוקול QUIC משלב את תהליך היצירת קשר הראשוני עם אימות של מפתחות הצפנה כדי להשיג את המידע הנדרש לשני התהליכים בRTT אחד.

QUIC משתמש בשליחת החבילה התחלתית כדי לתאם את המזהה של הקשר (CID) כל קצה תורם את החלק שלו בשדות של המזהה. בנוסף, ה TLS 1.3 גם כן נמצא בחבילה ההתחלתית, וכך אפשר לקבוע את הפרמטרים של ההצפנה .

כך הוא משפר את הלחיצת יד של TCP אשר שם נדרשים שני סבבי RTT, הראשון עבור יצירת הקשר הראשוני והעברת פרמטרים תחבורתיים, והשני לתהליך האבטחה.

בנוסף, המזהה קשר הייחודי (CID) מורכב משני מספרים שכל צד בוחר אחד מהם. מספר זה לא תלוי בPORT ולכן שינוי בפרמטרים בשכבה יותר נמוכה לא משפיע על הCID, ואפשר לשמור על החיבור במקרה כזה, בשונה מ TCP.

שיפור נוסף שך QUIC, הוא שאפשר לשלוח מידע מוצפן ב RTT-0 ואם לקצה השני כבר קיימים הנתונים של ההצפנה, הוא יוכל להמשיך את הקשר הקודם ללא צורך ביצירת קשר ראשוני.

 A white text on a black background

Description automatically generated

A screenshot of a document

Description automatically generated

**שאלה 4:**

בשאלה זו נתאר את מבנה של חבילהת QUIC, וכיצד הוא משפר חסרונות של TCP.

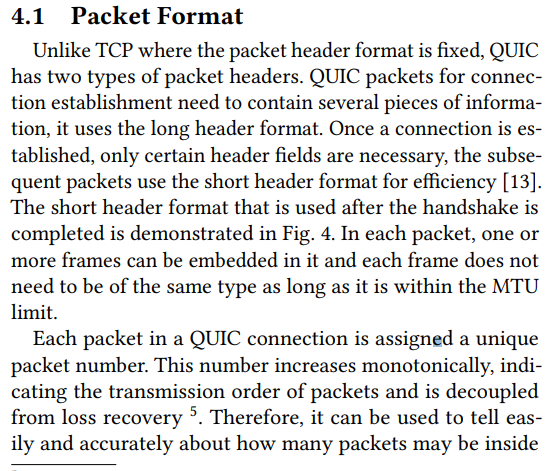
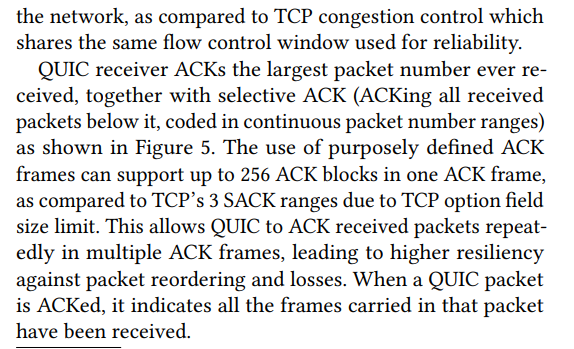
בניגוד ל TCP שבו פורמט החבילה קבוע (ובפרט ה HEADER קבוע), ל- QUIC יש שני סוגים שונים של כותרות Headers: חבילות להקמת החיבור מכילים HEADER בפורמט הארוך, אך לאחר שהחיבור הוקם נשתמש בפורמט הקצר.

ה HEADER מורכב מ:

1. מזהה החיבור (מזהה החיבור המקור והיעד נבחרים בכל נקודת קצה)
2. מספר חבילה
3. דגלים ( מידע על סוג החבילה ומצבה)
4. תוכן מוצפן (Protected Payload)

איך הוא משפר חסרונות ב TCP:

* חסימת ראש תור (HLB): מבנה החבילה מאפשר לשלוח חביוות בסדר לא רציף, בשונה מ TCP.
* שינוי כתובת IP : הקישור בין נקודות הקצה הוא בעזרת Connection ID (CID) וכמו שפירטנו בסעיף הקודם, מאפשר שינוי של ה IP וה PORT בשונה מה TCP



A diagram of a computer program

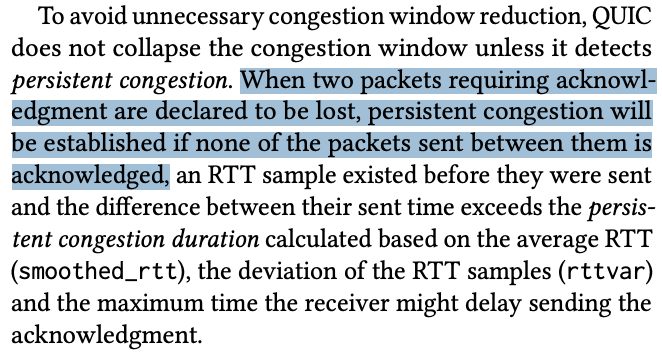
Description automatically generated A screenshot of a computer

Description automatically generated

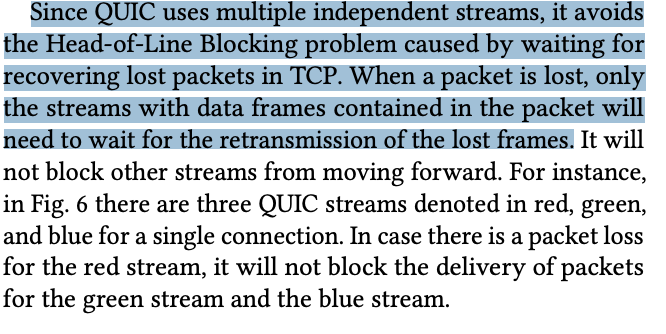
**שאלה 5:**

מה QUIC עושה כאשר חבילות מגיעות באיחור או לא מגיעות כלל?

QUIC משתמש במספר מנגנונים כאשר חבילות מגיעות באיחור או לא מגיעות בכלל.  
המנגנון הראשון הוא בקרת עומס. הוא עוזר לפרוטוקול להתאים את קצב השליחה בהתאם לתנאים של הרשת. רק אם יש חבילות שמגיעות באיחור באופן קבוע, אז זה מעיד על עומס קיים ויגרום לQUIC להוריד את הקצב כדי למנוע איבוד של חבילות נוספות.

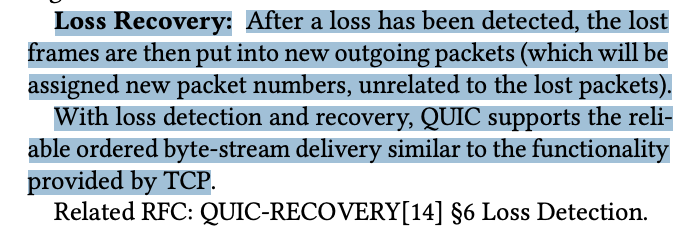


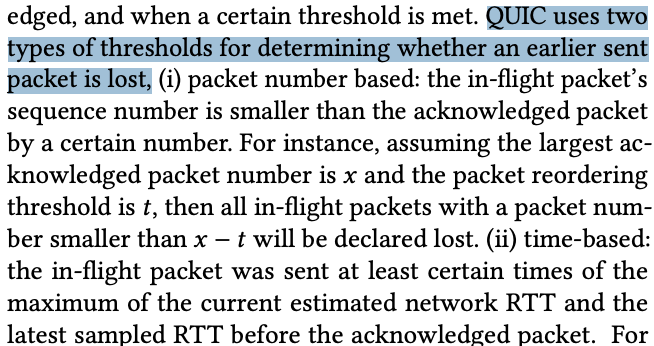
מנגנון נוסף הוא שימוש בשידורים עצמיים, וזה מאפשר לו להימנע מחסימות שנגרמות כתוצאה מהמתנה לשיחזור חבילות שנאבדו. רק השידורים של החבילות שנאבדו יצטרכו לחכות לשידור חוזר וזה לא יחסום את שאר החבילות מלהתקדם.



איתור ושיקום אובדן של חבילות. לכל חבילה QUIC משתמש בשתי שיטות על מנת לקבוע אם החבילה נאבדה. הראשונה היא מבוססת על מספר חבילה: מספר הרצף קטן מהמספר של החבילה המוכרת. השיטה השנייה מבוססת על זמן: החבילה נשלחה לפחות מספר פעמים בזמנים מסויימים של המקסימום לרשת המשוערת הנוכחית RTT וה-RTT שנדגם אחרון לפני החבילה שנשלחה.

לאחר שהQUIC איתר את החבילה שנאבדה, הוא ינסה לשחזר אותה. המסגרות שנאבדו מוכנסות לחבילות חדשות, שיהיו להם מספרי חבילות חדשים - שונים מהמספרים של החבילות שנאבדו.





**שאלה 6:**

נתאר מהו בקרת העומס (CC) של QUIC:

QUIC משתמש במספרי החבילות לבקרת העומס וב OFFSET לבקרה על אמינות.

בדומה ל TCP, QUIC משתמש במנגנון בקרת עומס מבוסס על גודל חלון המגביל את מספר הבייטים שהשולח יכול להעביר. חשוב לציין, QUIC לא מפתח אלגוריתם העברה או משתמש באלגוריתם ספציפי, אלא שולח אותות לבקרה על העומס והשולח יכול ליישם את מנגנון בקרת העומס שלו.

על מנת להימנע מהקטנת חלון מיותרת, QUIC לא מצמצם את החלון עבור חבילה אחת שנאבדה, אלא מחכה לקלוט עומס מתמשך ורק אז מצמצם את החלון. עומס מתמשך ייקבע אם לפחות שתי חבילות שדורשות ACK נאבדו ולא קיבלנו ACK על החבילות ביניהם.

בנוסף, QUIC שולח חבילות באופן מדורג על מנת להפחית את הסיכוי לעומס על ידי שליחת חבילות במרווח זמן לפי החישוב שלו.

A close-up of a text

Description automatically generated A screenshot of a document

Description automatically generated