**פרויקט גמר ברשתות – QUIC**

חלק יבש:

1. נפרט את חמשת החסרונות של TCP.

**חיסרון ראשון** טמון בחיבור בין אמינות העברת החבילות בTCP לבין הדרך שבה TCP מטפל בעומסים.

ב – TCP, על מנת לטפל בעומסים, מוגדר חלון בגודל מסויים שמטרתו היא לתרום לשליטת הזרימה ברשת, החלון עובר חבילה חבילה וכאשר חבילה נשלחת ומתקבלת כראוי, הוא ממשיך לחבילה הבאה. אך כאשר יש עומס, הוא מקטין או מגדיל אותו בדרכים שונות בהתאם לסיטואציה והאלגוריתם שהוגדר.

מצד שני, אמינות העברת חבילות ב – TCP דואגת לשלוח בשנית חבילות שנאבדו כדי לוודא את שלמות הdata ברשת.

כיוצא מזאת, כל סוג של איבוד חבילה, עלול לגרום להגבלה בתנועת חלון הזרימה קדימה, אפילו אם הסיבה לא נובעת מעומס בכלל, בנוסף לכך, החלון יפסיק להתקדם עד אשר האובדן ישוחזר, דבר זה יוצר עיכובים בצד השרת. לכן השילוב בין שני ההיבטים האלה גורם לעיכוב ברשת.

מעבר לכך שכל הסיבות האלה גורמות להגדלת הdelay ברשת, עיכובים, ואי ניצול מירבי של רוחב הפס הנתון, כמות החבילות שלא קיבלו ack, לא משקפת את כמות החבילות ברשת.

**חיסרון שני** קשור בעיקרו ליצירות של צוואר בקבוק בהעברת המידע הנובע מתוך מנגנון ההעברה האמינה של TCP. כאשר נאבדת חבילה, השרת שולח אותה שוב ללקוח, אך מכיוון שב-TCP החבילות ממוספרות, הצד המקבל לא יהיה מוכן לקבל את החבילות לא בסדר הנכון, כלומר אם ברצף מסויים נאבדה חבילה הלקוח לא ימשיך לקבל חבילות עד אשר תגיע אליו החבילה הבאה בסדר הנכון. כלומר חבילה אחת שנאבדת יוצרת צוואר בקבוק ועוצרת את זרימת המידע ללקוח של כל החבילות שמגיעות אחריה, דבר זה יוצר עיכובים ברשת.

**חיסרון שלישי** הוא לחיצת היד של TCP שנדרשת בכל יצירת חיבור בין שני צדדים, לחיצת היד משולשת היא הכרחית ולא יכול להישלח מידע לפני שהיא קוראת, כלומר גם אם הלקוח והשרת תיקשרו כבר פעם אחת, פנייה נוספת חדשה תדרוש שוב לחיצת יד משולשת, יתר על כן, אם אחד הצדדים ירצה גם לבטח את הקשר באמצעות TLS, תידרש עוד שליחה וקבלה של חבילות על מנת להחליף פרטי אבטחה בין הצדדים. כלומר, שלבים רבים חוזרים על עצמם ביצירת קשר בTCP כחלק ממנגנון האבטחה ואמינות הקשר שלו, דבר זה חשוב, אך בכל זאת יוצר trade off ב- delay ברשת.

**חיסרון רביעי** קשור בheader של TCP, ה-header מכיל את כל פונקציות השליטה שדרושות לפרוטוקול בתוך אותם 20 בתים, הבעיה בכך היא שיש לו גודל קבוע לפונקציות האלה ולכן הוא מושפע מאוד מעלייה במהירות הרשת, כלומר הגודל הקבוע של זיכרון עבור הפונקציות מגביל מאוד את הביצועים של TCP במהירות גבוהה, בTCP אכן יש עוד 40 בתים אופציונליים לשימוש אך גם כמות זאת מוגבלת ובמהירות מסוימת תגביל גם את הביצועים של TCP במהירויות גבוהות.

**חיסרון חמישי** קשור בדרך שבה TCP מבדיל בין קשרים.

TCP מתאים לכל צד בקשר זוג סדור של כתובת IP ו – PORT, הבעיה בכך היא שבמידה ו – IP פתאום משתנה מסיבה כלשהי (ולא חסר) כל המידע שהוחלף עד לשינוי ה – IP נעלם, מכיוון שאחד הצדדים כבר לא עונה על הidentifier ש – TCP הצמיד לו כדי לזהות אותו. לכן שני הצדדים יצטרכו להקים קשר מחדש ולחיצת יד נוספת תידרש מההתחלה.

2. חמשת התפקידים שפרוטוקול תעבורה צריך למלא הם:

1. להגדיר מזהי חיבור ומזהי מידע, מזהי חיבור הם לדוגמה כתובות IP, PORTS כמו שהפרוטוקולים שלמדנו עליהם עד כה משתמשים.

מזהי מידע הם דרך לסמן את סדר המידע הנכון, לדוגמה כמו שTCP ממספר חבילות בצורה סדרתית ובכך יודע את סדרן.

2. ניהול תעבורת החיבור

בעצם איזו שהיא רשימת חוקים שמתארת מה הפרוטוקול עושה, איך הוא מחבר בין שני צדדים, איך הוא מקים חיבור או סוגר אותו, איך תעבורת המידע תתנהל פרקטית בין שני צדדים לפי הפרוטוקול, דברים שהפרוטוקול תומך בהם בחיבור, למשל ריבויי כתובות IP.

3. העברת מידע אמינה

איך המידע מסומן ברשת ( כדי לדעת את סדר שליחתו או קבלתו), דרך העברת המידע.

4. טיפול בעומסים

בסופו של דבר הפרוטוקול צריך להגיע לידי שליטה בכמות החבילות שיש ברשת.

5. אבטחה

הכוונה באבטחה היא יצירת חיבורים מוצפנים, בדרך כלל הקמת החיבור מתרחשת ועליה מוסיפים אבטחה והצפנה של הקשרים.

3.

פרוטוקול QUIC, בשונה מ- TCP משתמש בדרך שונה ללחיצת יד, ראשית הוא לא משתמש בIP או PORT בתור מזהי חיבור, אלא בID פשוט.

הQUIC שולח את הID שלו למקבל והמקבל מחזיר לו ID, וכך בשתי שליחות מהירות הצלחנו כבר להקים חיבור.

לעומת זאת, בבעיה שתיארנו קודם בTCP, כל פעם שרוצים להקים קשר, צריך לחיצת ידיים משולשת של חבילת SYN, חבילת ACK-SYN וחבילת ACK. בנוסף לכך, QUIC תומך בלחיצת יד אחת, ולאחר מכן מאפשר חיבורים שכבר קרו ללא לחיצת יד, בשונה לTCP שכן מבקש לחיצת יד בכל הקמת קשר שהיא.

לכן אנו מבינים שQUIC אכן משפר חלק מהחסרונות של TCP בהקשר של לחיצת היד שלו, שכן היא יותר קצרה וקומפקטית, וכמובן דורשת פחות RTT מלחיצת היד של TCP.

4.

מבנה חבילת QUIC מחולק לשני סוגים של חבילות header, סוג אחד נועד עבור חבילות של הקמת חיבור, שהוא הפורמט הארוך יותר, סוג זה משומש רק בהקמת החיבור, לאחר מכן החבילות שעוקבות להם בחיבור משתמשות בסוג השני, בסוג הקצר יותר מטעמי יעילות.

פורמט הheader הקצר מכיל את הדגלים הבאים: ID של היעד, מספר פאקטה, מטען של פריימים של מידע שכל פריים מכיל Stream ID, אורך המידע, offset ואת המידע עצמו, סך משקל הפריים לא חוצה את גבול הMTU הנוכחי.

ניתן לראות שQUIC אכן משפר את TCP במקרה זה. בעוד שלTCP היה גודל קבוע בheader שלו, QUIC הרבה יותר גמיש ומאפשר שני גדלים עבור תפקידים שונים ואכן במהירויות גבוהות יותר ישיג כנראה ביצועים טובים יותר בשל ניצול טוב יותר של מקום לתפקיד מתאים.

בנוסף לכך, QUIC מאפשר עד ל256 בלוקים שלACK בתוך פריים יחיד, בעוד שTCP מאפשר 3 SACK עקב הגודל הקבוע והמגביל שלו. זה מאפשר לQUIC להחזיר ACK עבור חבילות שהתקבלו בקלות רבה. עובדה זאת מהווה שיפור גדול של QUIC על TCP במקרים של אובדן חבילות או בעיות בסדר קבלת המידע.

5.