

Transmission Control Protocol (TCP) 14

כאשר מישו מוכן דחפור לכל סיבים אופטיים, חבילות עלולות להיאבד. (מדינות שלמות הוצאו מקו על ידי סירת עוגן שגררה כבל רשת תת-ימי!) שגיאות בתוכנה, קריסות מחשבים ותקלות בנתבים יכולים לגרום לבעיות.

אבל אנחנו לא רוצים לדאוג לזה. אנחנו רוצים שלישות מטפל בכל זה ואז יגיד לנו כשהנתונים שלמים, תקינים ובסדר.

TCP הוא אותו יישות. הוא דואג לחבילות שאבדו כדי שאנחנו לא נצטרך. וכאשר הוא בטוח שיש לו את כל הנתונים הנכונים, הוא מעביר אותם אלינו.

נבחן את:

- המטרות הכלליות של TCP
- היכן הוא משתלב במערכת הרשת
- רעיון על פורטים של TCP
- כיצד TCP יוצר, משתמש וסוגר חיבורים
- מנגנוני שלמות נתונים
- שמירה על סדר החבילות
- זיהוי שגיאות
- בקרת זרימה – כיצד מקבל מידע נמנע מהצפה
- בקרת עומס – כיצד שליחים נמנעים מלהעמיס על האינטרנט

TCP הוא נושא מאוד מורכב ואנחנו רק נוגעים בנקודות המרכזיות כאן. אם ברצונכם ללמוד יותר, הספר המומלץ הוא TCP/IP Illustrated Volume 1 של ו. ריצ'רד סטיבנס.

14.1 מטרות TCP

- לספק תקשורת אמינה
- לדמות חיבור סיבתי על רשת מנות
- לספק בקרת זרימה
- לספק בקרת עומס
- לתמוך בנתונים מחוץ לערוץ

14.2 מיקום במערכת הרשת

זכרו את השכבות של מערכת הרשת:

פרוטוקולים לדוגמה	שכבה	אחריות
HTTP, FTP, TFTP, Telnet, SSH, SMTP, POP, IMAP	יישום	נתונים מאורגנים ביישום
TCP, UDP	תחבורה	שלמות נתונים, פיצול וארגון חבילות

IP, IPv6, ICMP

אינטרנט
ניתוב

Ethernet, PPP, token ring

קישור פיזי, אותות על חוטים

אתם יכולים לראות את TCP בשכבת התחבורה. IP מתחתיו אחראי על ניתוב. ושכבת היישום שמעליו עושה שימוש בכל היתרונות ש-TCP מציע.

זו הסיבה שכאשר כתבנו את לקוח ושרת ה-HTTP שלנו, לא היינו צריכים לדאוג לשלמות הנתונים בכלל. השתמשנו ב-TCP וזה טיפל בזה בשבילנו!

14.3 פורטים של TCP

זכרו שכאשר השתמשנו ב-TCP היינו צריכים לציין מספרי פורטים כדי להתחבר. אפילו הלקוחות שלנו הוקצו אוטומטית פורטים מקומיים על ידי מערכת ההפעלה (אם לא קיבלנו אותם בעצמנו).

IP משתמש בכתובת IP כדי לזהות מארחים.

אך ברגע שנמצאים על אותו מארח, מספר הפורט הוא מה שמשתמשת מערכת ההפעלה כדי להעביר את הנתונים לתהליך המתאים.

בהשוואה, כתובת ה-IP היא כמו כתובת רחוב, ומספר הפורט הוא כמו מספר דירה באותה כתובת.

14.4 סקירה כללית של TCP

ישנם שלושה דברים עיקריים ש-TCP עושה במהלך חיבור:

1. יצירת החיבור
2. שליחת נתונים
3. סגירת החיבור

כל אחד מהם כרוך בשליחת חבילות מיוחדות (שאינן נתוני משתמש) הלוך ושוב בין הלקוח לשרת. נבחן את סוגי החבילות המיוחדות SYN, SYN-ACK, ACK ו-FIN.

14.4.1 יצירת החיבור

החיבור מתבצע באמצעות "לחיצת יד של שלושה צעדים". מכיוון שיכולות להיאבד חבילות במהלך השידור, TCP מקפיד לוודא ששני הצדדים של החיבור מוכנים לנתונים לפני שמתקדמים.

1. הלקוח שולח חבילת SYN (סנכרון) לשרת.
2. השרת עונה בחבילת SYN-ACK (סנכרון אישור) חזרה ללקוח.
3. הלקוח עונה בחבילת ACK (אישור) חזרה לשרת. אם אין תגובה בזמן סביר לכל אחד מהשלבים הללו, החבילה נשלחת מחדש.

14.4.2 שליחת נתונים

TCP לוקח זרם של נתונים ומחלק אותו לחתיכות. כל חתיכה מקבלת כותרת TCP ומועברת לשכבת ה-IP למשלוח. הכותרת והחתיכה יחד נקראות "סגמנט TCP".

(נשתמש במונחים "חבילת TCP" ו-"סגמנט TCP" לסירוגין, אך סגמנט הוא המונח הנכון יותר.)

כאשר TCP שולח סגמנט, הוא מצפה לקבל אישור מהנמען של הנתונים, הנקרא ACK. אם TCP לא מקבל את ה-ACK, הוא יניח שמהו השתבש וימשיך לשלוח את הסגמנט מחדש.

הסגמנטים ממוספרים כך שגם אם הם מגיעים בסדר לא נכון, TCP יכול למיין אותם לסדר הנכון.

14.4.3 סגירת החיבור

כאשר צד רוצה לסגור את החיבור, הוא שולח חבילת FIN (סיום). הצד המרוחק בדרך כלל מגיב ב-ACK ו-FIN משלו. הצד המקומי ישלים את הסיום עם ACK נוסף.

במערכות הפעלה מסוימות, אם מארח סוגר חיבור עם נתונים שלא נקראו, הוא ישלח RST (איפוס) כדי להצביע על כך. תוכניות סוקט עשויות להדפיס את ההודעה "החיבור אופס על ידי עמית".

14.5 שלמות נתונים

יש הרבה דברים שיכולים להשתבש. נתונים יכולים להגיע לא בסדר. הם יכולים להתקלקל. הם יכולים להיות כפולים. או אולי בכלל לא יגיעו.

ל-TCP יש מנגנונים לטפל בכל אחד מהמקרים האלה.

14.5.1 סדר החבילות

השולח מציב מספר סיקוונס עולה על כל סגמנט. "הנה סגמנט 3490". הנה סגמנט 3491.

המקבל מגיב לשולח עם חבילת ACK שמכילה את מספר הסיקוונס הזה. "קיבלתי סגמנט 3490. קיבלתי סגמנט 3491".

אם שני סגמנטים מגיעים לא בסדר, TCP יכול להחזיר אותם לסדר על ידי מיון לפי מספר הסיקוונס.

14.5.2 זיהוי שגיאות

לפני שהשולח שולח סגמנט, מחושב סכום בדיקה (Checksum) עבור הסגמנט.

כאשר המקבל מקבל את הסגמנט, הוא מחשב את סכום הבדיקה שלו.

אם שני סכומי הבדיקה תואמים, הנתונים נחשבים תקינים. אם הם שונים, הנתונים נזרקים והשולח צריך לשלוח את הסגמנט מחדש.

סכום הבדיקה הוא מספר בגודל 16 ביט שמתקבל על ידי פיענוח כל הנתונים של כותרת ה-TCP וטעינת המידע, יחד עם כתובות ה-IP המעורבות, לפונקציה ש'מעכלת' את הנתונים.

14.6 בקרת זרימה

בקרת זרימה היא המנגנון שבו שני מכשירים בתקשורת מודיעים זה לזה שצריך לשלוח את הנתונים לאט יותר.

14.7 בקרת עומס

בקרת זרימה פועלת בין שני מחשבים, אך ישנה בעיה גדולה יותר של האינטרנט כולו. אם נתב מוצף, הוא יכול להתחיל לזרוק חבילות, דבר שגורם לשליח להתחיל לשלוח מחדש, מבלי לפתור את הבעיה.

כדי לתקן את זה, מספר מנגנונים הוספו ל-TCP כדי להעריך ולהפחית את הצפיפות ברשת.

14.7.1 התחלה איטית (Slow Start)

כאשר החיבור מתחיל, השולח לא יודע מהו מצב העומס ברשת. שלב זה הוא בעיקר על מנת להעריך את המצב.

14.7.2 המנעות מעומס (Congestion Avoidance)

האלגוריתם הזה דומה להתחלה איטית, אבל הוא עושה צעדים הרבה יותר מבוקרים.