TCP ו UDP

Transmission Control Protocol, הוא פרוטוקול הפועל בשכבת התעבורה ומשמש להעברת מידע בצורה אמינה ומסודרת בין שני התקנים. TCP דואג לכך שהמידע יגיע ליעד בצורה תקינה, לפי הסדר, ובמקרה של אובדן נתונים ניתן יהיה לשלוח אותם מחדש.

TCP משתמש במספרי **Port** כדי לזהות לאיזה יישום מיועד המידע. כך מספר יישומים יכולים להשתמש באותה תקשורת במקביל, כאשר לכל תקשורת יש את מספר הפורט המתאים.

כיצד TCP עובד

לפני שהתחלת העברת המידע, TCP צריך להקים חיבור בין שני הצדדים. תהליך זה נקרא **Three Way Handshake**, והוא מאפשר לשני הצדדים לתאם את תחילת התקשורת.בתחילה הלקוח שולח הודעת **SYN** לשרת כדי לבקש לפתוח חיבור ולסנכרן את מספרי ה Sequence.השרת מחזיר הודעת **SYN ACK**, אשר מאשרת שקיבל את הבקשה וגם מציינת שהוא מוכן להקים את החיבור.לאחר מכן הלקוח שולח הודעת **ACK**, ובכך החיבור נחשב כמוקם ושני הצדדים יכולים להתחיל להעביר מידע.

התהליך מתבצע בצורה הבאה:

SYN → SYN ACK → ACK

לאחר יצירת החיבור, TCP מתחיל להעביר את הנתונים. לכל חלק מהמידע מוקצה **Sequence Number**, אשר מאפשר ל TCP לזהות את סדר הנתונים ולוודא שהם מגיעים בצורה הנכונה.

כאשר הצד המקבל מקבל את המידע, הוא שולח **ACK** כדי לאשר שהנתונים התקבלו. אם חלק מהמידע לא הגיע או שהתקבלה בעיה במהלך ההעברה, TCP יכול לזהות זאת ולבצע **Retransmission**, כלומר לשלוח את המידע מחדש.

באופן זה TCP מספק תקשורת אמינה יותר, במיוחד במקרים שבהם חשוב לקבל את כל המידע ובסדר הנכון.

מבנה TCP Header

לכל מקטע TCP מצורף Header המכיל את המידע הדרוש לניהול התקשורת.

בתחילת ה Header נמצא **Source Port**, המציין את מספר הפורט של היישום ששולח את המידע.

אחריו נמצא **Destination Port**, המציין את הפורט של היישום שאליו המידע מיועד.

לאחר מכן נמצאים **Sequence Number** ו **Acknowledgment Number**, המשמשים לניהול סדר המידע ואישור קבלת הנתונים.

בהמשך קיימים שדות בקרה, כאשר בין החשובים נמצאים הדגלים **SYN, ACK** ו **FIN**, המשמשים בין היתר להקמת החיבור ולסגירתו.

בנוסף קיימים שדות כמו **Window Size**, המשמש לשליטה בכמות המידע שניתן להעביר לפני קבלת אישור, ו **Checksum**, המשמש לבדיקת תקינות המידע.

UDP

User Datagram Protocol, הוא פרוטוקול הפועל גם הוא בשכבת התעבורה ומשמש להעברת מידע בין יישומים. בניגוד ל TCP, UDP אינו מקים חיבור לפני תחילת העברת המידע ואינו מספק מנגנונים מובנים להבטחת הגעת הנתונים או לסידור שלהם.

UDP משתמש גם הוא במספרי **Port**, ולכן הוא מאפשר לזהות לאיזה יישום מיועד המידע.

כיצד UDP עובד

ב UDP ניתן להתחיל להעביר מידע ללא ביצוע של Three Way Handshake. היישום מוסר את המידע ל UDP, ה UDP מוסיף את ה Header שלו והמידע ממשיך לשכבות שמתחתיו.

UDP אינו ממתין לאישור קבלה מהצד השני ואינו מבצע שליחה מחדש של מידע שאבד. לכן אם Datagram מסוים לא הגיע ליעד, UDP עצמו אינו מבצע פעולה לתיקון הבעיה.

היתרון בכך הוא שמבנה הפרוטוקול פשוט יותר והתקורה של התקשורת נמוכה יותר. הדבר מאפשר זמן תגובה מהיר ומתאים ליישומים שבהם העברת המידע בזמן חשובה יותר מהבטחה שכל המידע יגיע.

מבנה UDP Header

UDP Header פשוט יחסית ומכיל מספר שדות עיקריים.

בתחילת ה Header נמצא **Source Port**, המציין את הפורט שממנו נשלח המידע, ולאחריו **Destination Port**, המציין את הפורט שאליו המידע מיועד.

בהמשך נמצא **Length**, המציין את אורך הודעת ה UDP, ולאחר מכן **Checksum**, המשמש לבדיקת תקינות המידע.

לאחר ה Header נמצא המידע עצמו שהיישום רוצה להעביר.

TCP לעומת UDP

TCP ו UDP משמשים שניהם בשכבת התעבורה, אך צורת העבודה שלהם שונה.

TCP מבצע יצירת חיבור, משתמש ב Sequence Numbers וב ACK, שומר על סדר המידע ויכול לבצע שליחה מחדש במקרה של אובדן.

UDP אינו מבצע יצירת חיבור ואינו מספק מנגנונים מובנים לשליחה מחדש או לשמירה על סדר המידע. לכן הוא פשוט ומהיר יותר מבחינת מנגנון התקשורת.

TCP מתאים כאשר חשוב שהמידע יגיע בצורה מלאה ומסודרת, לדוגמה בהעברת קבצים ובתקשורת שבה אובדן נתונים אינו רצוי.

UDP מתאים כאשר חשוב יותר לקבל מידע במהירות ובזמן תגובה נמוך, לדוגמה בתקשורת בזמן אמת, משחקי רשת ושירותים שבהם ניתן להתמודד עם אובדן של חלק מהמידע.

בסופו של דבר, הבחירה בין TCP ל UDP תלויה בדרישות של היישום ובחשיבות של אמינות המידע לעומת מהירות וזמן תגובה.