TCP

Das Transmission Control Protocol (TCP) ist ein Standard, der definiert, wie eine Netzwerkkonversation aufgebaut und aufrechterhalten wird, über die Anwendungen Daten austauschen können.

TCP arbeitet mit dem Internet-Protokoll (IP) zusammen, das festlegt, wie Computer Datenpakete aneinander senden. Zusammen bilden TCP und IP die Grundregeln, die das Internet definieren. Die Internet Engineering Task Force (IETF) definiert TCP in dem Standarddokument Request for Comment (RFC) mit der Nummer 793.

Wie das Transmission Control Protocol funktioniert

TCP ist ein verbindungsorientiertes Protokoll, das heißt, eine Verbindung wird aufgebaut und aufrechterhalten, bis die Anwendungen auf beiden Seiten den Nachrichtenaustausch beendet haben.

TCP führt die folgenden Aktionen durch:

- Es legt fest, wie die Anwendungsdaten in Pakete zerlegt werden, die das Netzwerk übertragen kann.
- TCP sendet Pakete an die Netzwerkschicht und nimmt Pakete von ihr entgegen.
- TCP verwaltet die Flusskontrolle.
- Es behandelt die erneute Übertragung von verworfenen oder verstümmelten Paketen, da es eine fehlerfreie Datenübertragung gewährleisten soll.
- TCP quittiert alle ankommenden Pakete.

Im OSI-Kommunikationsmodell (Open Systems Interconnection) deckt TCP Teile von Layer 4 (Transportschicht) und Teile von Layer 5 (Sitzungsschicht) ab.

Wenn ein Webserver eine HTML-Datei an einen Client sendet, verwendet er dazu das Hypertext Transfer Protocol (HTTP). Die HTTP-Programmschicht fordert die TCP-Schicht auf, die Verbindung aufzubauen und die Datei zu senden. Der TCP-Stack teilt die Datei in Datenpakete auf, nummeriert sie und leitet sie dann einzeln an die IP-Schicht zur Zustellung weiter.

Obwohl jedes Paket in der Übertragung die gleiche Quell- und Ziel-IP-Adresse hat, können die Pakete über mehrere Routen gesendet werden. Die TCP-Programmschicht auf dem Client-Computer wartet, bis alle Pakete eingetroffen sind. Dann bestätigt sie die empfangenen Pakete und bittet um die erneute Übertragung der Pakete, die sie nicht erhalten hat, da die Paketnummern fehlen. Die TCP-Schicht setzt dann die Pakete zu einer Datei zusammen und übergibt die Datei an die empfangende Anwendung.

TCP vs. UDP

Dieser Prozess der Fehlererkennung, bei dem TCP Pakete nach ihrem Eintreffen erneut überträgt und neu anordnet, kann zu Latenzen in einem TCP-Stream führen. Zeitkritische Anwendungen wie Voice over IP (VoIP), Videostreaming und Spiele sind in der Regel auf ein Transportverfahren wie das User Datagram Protocol (UDP) angewiesen, da es Latenz und Jitter reduziert, indem es Pakete nicht neu anordnet oder fehlende Daten erneut überträgt.

UDP wird als Datagramm-Protokoll oder verbindungsloses Protokoll eingestuft, da es keine Möglichkeit hat, zu erkennen, ob beide Anwendungen ihre Hin- und Her-Kommunikation beendet haben. Anstatt ungültige Datenpakete zu korrigieren, wie dies bei TCP der Fall ist, verwirft UDP diese Pakete und überlässt der Anwendungsschicht eine genauere Fehlererkennung.

Der Header eines UDP-Datagramms enthält weit weniger Informationen als ein TCP-Segmentheader. Der UDP-Header wird auch auf der Transportschicht viel weniger verarbeitet, um die Latenzzeit zu verringern.

Wofür wird TCP verwendet?

TCP wird verwendet, um Daten so zu organisieren, dass die sichere Übertragung zwischen Server und Client gewährleistet ist. Es garantiert die Integrität der über das Netz gesendeten Daten, unabhängig von ihrer Menge. Deshalb wird es zur Übertragung von Daten aus anderen übergeordneten Protokollen eingesetzt, bei denen alle übermittelten Daten ankommen müssen.

Beispiele für solche Protokolle sind:

- Secure Shell (SSH), File Transfer Protocol (FTP), Telnet: Für den Peer-to-Peer-Dateiaustausch und, im Falle von Telnet, für das Einloggen in den Computer eines anderen Benutzers, um auf eine Datei zuzugreifen.
- Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP), Internet Message Access Protocol (IMAP): Für das Senden und Empfangen von E-Mails.
- HTTP: Für den Web-Zugang.

Diese Beispiele befinden sich alle auf der Anwendungsschicht des TCP/IP-Stacks und senden Daten an TCP auf der Transportschicht.

Warum TCP wichtig ist

TCP ist wichtig, weil es die Regeln und Standardverfahren für die Art und Weise der Informationsübermittlung im Internet festlegt. Es bildet die Grundlage für das Internet in seiner heutigen Form und sorgt dafür, dass die Datenübertragung einheitlich erfolgt, unabhängig von Standort, Hardware oder Software.

TCP ist flexibel und hochgradig skalierbar, das heißt, es können neue Protokolle eingeführt werden, und es wird sie aufnehmen. Außerdem ist TCP nicht proprietär, es ist also nicht im Besitz einer Person oder eines Unternehmens.

Position im TCP/IP-Stack

Der TCP/IP-Stack ist ein Modell, das darstellt, wie Daten über Netzwerke mit dem TCP/IP-Protokoll organisiert und ausgetauscht werden. Es zeigt eine Reihe von Schichten, die die Art und Weise darstellen, wie Daten auf ihrem Weg vom Client zum Server und umgekehrt von einer Reihe von Protokollen behandelt und verpackt werden.

TCP existiert in der Transportschicht zusammen mit anderen Protokollen wie UDP. Die Protokolle dieser Schicht gewährleisten die fehlerfreie Übertragung der Daten zur Quelle, mit Ausnahme von UDP, da es nur über eine begrenzte Fähigkeit zur Fehlerprüfung verfügt.

Wie das OSI-Modell ist auch der TCP/IP-Stack ein konzeptionelles Modell für Datenaustauschstandards. Die Daten werden auf jeder Schicht auf der Grundlage ihrer Funktionalität und der Transportprotokolle neu verpackt.

Die Anfragen gelangen über den Stack zum Server, wobei sie auf der Anwendungsschicht als Daten beginnen. Von dort aus werden die Informationen auf jeder Schicht in Pakete verschiedener Typen aufgeteilt. Die Daten bewegen sich auf den folgenden Wegen:

- von der Anwendung zur Transportschicht, wo sie in TCP-Segmente sortiert werden;
- zur Internet-Schicht, wo sie zu einem Datagramm werden;
- zur Netzwerkschnittstellenschicht, wo sie wieder in Bits und Frames zerlegt werden; und
- wenn der Server antwortet, wandern sie durch den Stapel, um als Daten in der Anwendungsschicht anzukommen.

TCP/IP vs. OSI-Modell

Der Hauptunterschied zwischen dem TCP/IP-Modell und dem OSI-Modell ist der Grad der Spezifität.

Das OSI-Modell ist eine abstraktere Darstellung der Art und Weise, wie Daten ausgetauscht werden, und nicht spezifisch für ein bestimmtes Protokoll. Es ist ein Framework für allgemeine Netzwerksysteme. Der TCP/IP-Stack ist spezifischer und umfasst die vorherrschenden Protokolle, die zum Datenaustausch verwendet werden.

Das OSI-Modell ist abstrakt und basiert eher auf Funktionalität, während der TCP/IP-Stack konkret und protokollbasiert ist. Außerdem hat das OSI-Modell sieben Schichten, während das TCP/IP-Modell nur vier hat.

UDP

UDP steht für User Datagram Protocol und ist ein Kommunikationsprotokoll im Internet, das vorrangig dazu dient, Verbindungen mit geringer Latenz und Verlusttoleranz zwischen Anwendungen herzustellen.

Anders gesagt, UDP ist eine Grundfunktion im Internet, die eine verbindungslose Datenübertragung ermöglicht. Zum Beispiel läuft bei Livestreams oder Video-Calls UDP im Hintergrund automatisch mit und versendet währenddessen sogenannte Datenpakete, ohne dass der Empfang bestätigt werden muss. Wenn bei einem Videogespräch beispielsweise die Verbindung kurz unterbrochen wird, verwirft UDP zwar diese Datenpakete, das Gespräch läuft aber trotzdem weiter. Was das genau bedeutet, wie diese Übertragungsstrecke funktioniert und was das verwandte TCP damit zu tun hat? In diesem Artikel erfahren Sie alles rund um das Thema UDP.

Was ist UDP?

UDP – auf Deutsch Benutzer-Datagramm-Protokoll – regelt die Datenübertragung vom Sender zum Empfänger im Internet. Das verbindungslose Kommunikationsprotokoll ermöglicht es, einen Datentransfer zum Empfänger herzustellen, ohne dessen vorheriger Zustimmung.

Was erstmal kompliziert klingt, ist in der Praxis so einfach wie hilfreich: Beispielsweise kann es bei Liveübertragungen vorkommen, dass das Bild ruckeln, der Ton aber reibungslos weiterläuft. Hierfür sorgt UDP. Im Hintergrund verwirft das User Datagramm Protocol einige Datenpakete, die Verbindung bleibt aber trotzdem bestehen. Dadurch, dass UDP äußerst kleine Datagramme versendet, können diese Übertragungen schnell geschehen, was bei Echtzeit-Übertragungen und Video-Calls wichtig ist. Und was schnell und einfach funktioniert, verbraucht auch weniger Ressourcen: UDP bietet beispielsweise keine Garantien dafür, dass die gesendeten Daten auch tatsächlich zugestellt werden. Bei dieser Prozess-zu-Prozess-Kommunikation muss der Empfänger den Erhalt der Daten beziehungsweise den Verbindungsaufbau nämlich nicht bestätigen. Wie oben gelernt: Bei Verlust eines Pakets läuft die

Verbindung trotzdem automatisch weiter und muss nicht (erneut) bestätigt werden. Daher wird UDP in Fällen angewendet, wo Datenverluste keine kritischen Auswirkungen auf die Qualität der Verbindung beziehungsweise Übertagung haben.

UDP-Anwendungsfälle.

Das User Datagram Protocol kommt zum Einsatz, wo Datenverluste toleriert werden können, wo sich die Anwendungen sowie Dienste selbst um das Verbindungsmanagement kümmern oder es auf eine geringe Latenz, also eine niedrige Reaktionszeit ankommt. Was das genau bedeutet? Bei Echtzeitanwendungen ist es nicht tragisch, wenn gesendete Pakete verworfen werfen oder anders gereiht beim Empfänger ankommen. Auch bei einer größeren Anzahl an Clients, die bei diesem verbindungslosen Kommunikationsprotokoll auf Empfängerseite liegen, und wo keine Fehlerkorrekturen in Echtzeit nötig sind, kommt UDP zum Einsatz. Hier einige UDP-Beispiele aus der Praxis:

Datenübertragungen.

UDP kommt bei Datenübertragungen zur Anwendung, wo das Protokoll verloren gegangene Pakete selbst verwaltet und eine neue Übertragung der Pakete selbstständig anordnet. Auch bei schnellen Vermittlungen wird auf UDP gesetzt, wo die Geschwindigkeit der Übertragung wichtiger ist als die Zuverlässigkeit.

Multicasting.

Da das User Datagram Protocol der Paketvermittlung hilft, kommt auch beim Multicasting UDP zum Einsatz. Hintergrund: Multicasting ist eine Kommunikationstechnik, bei der Daten von einem Absender an mehrere Empfänger gleichzeitig in einem Netzwerk übertragen werden.

Spiele.

Bei Videogames beziehungsweise Onlinespielen ist die wahrgenommene Qualität wichtiger als die eigentliche, da hier die Reaktionszeit entscheidend für die Qualität, also das Spielerlebnis ist.

Sprachen, Audio und Video.

Wie bei Spielen bereits erklärt, zählt auch bei der digitalen Sprach- und Videokommunikation die wahrgenommene Latenz, also die Qualität der Übertragung. Hier kann ein kleiner Datenverlust auftreten, ohne dass die Qualität bei der Übertragung beziehungsweise Nutzung allzu großen Schaden nimmt. Sprich: Bei Livestreams beispielsweise läuft UDP automatisch mit und sorgt dafür, dass der Ton weiterläuft und die Verbindung zum Stream erhalten bleibt, sollte das Video mal ins Stocken geraten. Diese kleinen Datenverluste tun der Qualität der Verbindung demnach keinen erheblichen Abbruch. Um dieser Verlustquelle jedoch vorzubeugen, kommt manchmal FEC (Forward Error Correction) zum Einsatz. Bei dieser Vorwärtsfehlerkorrektur wird ein Fehlerkorrekturverfahren angewendet, um die Fehlerrate bei einer Datenübertragung via Audio oder Video zu senken. Beim Audio- sowie Videostreaming hat Datenverlust aber keine kritische Auswirkung. Da sich die Anwendung selbst um die Verbindung kümmert, können Datenpakete kontinuierlich fließen, ohne dass eine Bestätigung zurückkommt oder zurückkommen muss.

Aufbau eines UDP-Headers.

Wie oben erwähnt, benötigt es bei der Datenübertragung keine Bestätigung des Empfängers, was dazu führt, dass der UDP-Header, auch Paket-Header genannt, relativ klein ist. Der UDP-Header ist Teil des IP-Pakets und sorgt für eine geordnete Datenübertragung. Das IP-Paket besteht aus den Kopfdaten, also dem Header mit Informationen über den Quell- und Ziel-UDP-Port sowie den Status. Außerdem sind im

IP-Paket noch die Nutzdaten der Kommunikation enthalten. Im UDP-Header befindet sich zusätzlich die Checksumme, die die Fehlerfreiheit des Datagramms testet.

- Die Quell-Port-Nummer gibt die Nummer des Senders an.
- Mit Ziel-Port ist der UDP-Port gemeint, an den das Datagramm übermittelt wird.
- Die Länge beschreibt, wie lang das UDP-Paket ist. Dieser Wert zeigt die Daten in Byte an, woran die Vollständigkeit des Pakets ermittelt wird.
- Die Prüf- beziehungsweise Check-Summe kommt bei der Fehlerkontrolle zum Einsatz und prüft, ob das UDP-Paket korrekt übermittelt wurde.

Unterschied zu Transmission Control Protocol (TCP).

UDP und TCP sind beides Datenprogramme zwischen zwei Stationen und basieren auf dem Internet Protocol (IP). Das Transmission Control Protocol ist dabei eine Alternative zum User Datagram Protocol und kommt beispielsweise beim E-Mail-Verkehr zum Einsatz. Beim Versand von E-Mails sorgt TCP dafür, dass die Datenpakete in derselben Reihenfolge ankommen, wie sie versendet werden. Sprich: Eine geschriebene und versendete E-Mail landet, ohne fehlender Zeichen, genauso im Posteingang des Empfängers, wie sie versendet wurde.

Die beiden Programme UDP und TCP unterscheiden sich wesentlich in folgenden Punkten:

- Im Gegensatz zum verbindungslosen UDP ist TCP ein verbindungsorientiertes Protokoll und garantiert, dass keine Pakete während der Datenübertragung verloren gehen. Das hat die Ursache, dass der Empfänger den Verbindungsaufbau beziehungsweise Datenempfang bestätigen muss.
- Dieser Schritt verzögert beim Versenden der Datenpakete die Latenz und sorgt dafür, dass UDP weniger Ressourcen und TCP mehr Ressourcen benötigt.
- Ein weiterer Unterschied: Während beim UDP die Datenpakete teilweise verloren oder nicht in der richtigen Reihenfolge übertragen werden, sendet TCP die Pakete der Reihe nach.
- Während sich das User Datagram Protocol für Anwendungen wie Spiele, Audio- und Videoübertragungen dank der schnellen Übertragung eignet, ist die Übertragungszeit beim Transmission Control Protocol eher unwichtig, da hier die Zuverlässigkeit im Vordergrund steht.