TCP Grundlagen

TCP steht für das Transmission Control Protocol, was zu Deutsch so viel bedeutet wie Übertragungssteuerungsprotokoll. Es handelt sich dabei um eine Vereinbarung zur Datenübertragung zwischen verschiedenen Computern. Das Protokoll wird heutzutage von allen Betriebssystemen genutzt und ist zudem ein Bestandteil der Internetprotokollfamilie (IP).

In einer ausführlichen Artikelreihe, die aus sechs Teilen besteht, werden die unterschiedlichen Verbindungsarten zur Datenübertragung sowie der Aufbau des Protokolls vorgestellt. Der erste Beitrag gibt einen allgemeinen Überblick über TCP und beschäftigt sich mit der Funktionsweise des Verbindungsaufbaus.

Entwicklung von TCP

Das TCP dient zur Herstellung einer zuverlässigen bidirektionalen Datenverbindung zwischen Computern. In dem Protokoll werden Vereinbarungen dazu getroffen, auf welche Art und Weise verschiedene Rechner miteinander kommunizieren. TCP nutzt zur Übertragung von Daten meist das Internetprotokoll (IP), weshalb auch von einem TCP/IP-Protokoll gesprochen wird. Als Teil des Internetprotokolls ist TCP in die Transportschicht implementiert.

Die Geschichte von TCP geht in das Jahr 1973 zurück, als Robert E. Kahn und Vinton G. Cerf mit der Forschung an dem Protokoll begannen. Bis zum ersten standardisierten TCP im Jahr 1981 vergingen jedoch einige Jahre. Zu diesem Zeitpunkt wurde die Entwicklung in dem RFC 793 (Request für Comments) festgelegt. Diese „Bitte um Kommentare“ wurde über die Zeit ständig weiterentwickelt und auch heute werden noch technische und organisatorische Dokumente zum Internet in neuen RFCs spezifiziert.

In der heutigen Entwicklungsstufe ist das TCP in der Lage, eine Verbindung zwischen zwei Endpunkten (Sockets) einer Netzverbindung herzustellen, auf der eine beidseitige Datenübertragung möglich ist. Das Protokoll wird aufgrund seiner vorteilhaften Eigenschaften, wie der automatischen Erkennung von Datenverlusten und dem bidirektionalen Datenaustauch, weit verbreitet eingesetzt. Als Beispiel für die Anwendung von TCP gehören neben E-Mails und dem Internet auch zahlreiche weitere Netzdienste.

Allgemeine Funktionsweise von TCP

Vergleichbar mit einem Telefongespräch, lässt TCP eine Übertragung von Informationen in beide Richtungen zu. Dies wird mit dem Prinzip des Vollduplex oder durch zwei Halbduplexverbindungen, bei denen die Informationen nicht gleichzeitig übertragen werden können, realisiert. Neben den Datenpaketen werden bei der Übertragung auch Steuerungsinformationen übersendet.

Um die Datenverbindung zu verwalten, wird eine TCP-Software eingesetzt, welche sich im Netz-Protokollstack des Betriebssystems befindet. Die speziellen Anwendungsprogramme, wie Windows oder

Linux, benutzen als Schnittstelle sogenannte Sockets, die sich je nach Betriebssystem unterscheiden. Die letztendlichen Anwender machen von TCP häufig Gebrauch, indem Webbrowser oder –server genutzt werden.

Bei jeder Verbindung über das TCP-Protokoll werden zwei eindeutige Endpunkte identifiziert, die ein geordnetes Paar darstellen. Jedes Paar besteht aus einer IP-Adresse und einem Port, das die Software-Schnittstelle bildet und dann auch als Socket bezeichnet wird. Dabei sind nicht nur direkte Verbindungen möglich, sondern durch den Einbezug von Clients auch Weiterleitungen auf mehrere Server erlaubt.

Ein Webserver ist damit in der Lage, zur gleichen Zeit mehrere Verbindungen zu unterschiedlichen Rechnern aufzubauen, indem verschiedene Ports eingesetzt werden. Jeder Client erhält einen Port, auf dem eine einzigartige Verbindung zum Server hergestellt wird. Diese Verbindung zwischen dem Server und dem Endpunkt (Socket) wird durch die Zuordnung einer Portnummer und einer IP-Adresse gewährleistet.

Verbindungsaufbau des TCP

Zum Aufbau einer Verbindung wird ein Server und ein Client (Socket) benötigt, die durch die Zuweisung einer eindeutigen IP-Adresse und des Ports miteinander kommunizieren. Die Rollenverteilung von Client und Server sind dabei aus Sicht des TCP-Protokolls völlig symmetrisch, so dass beide Rechner eine Verbindung aufbauen können. Die benötigten Bestandteile zum Verbindungsaufbau über TCP, sind hier noch einmal zusammengefasst:

- Server: Der Server wartet auf einem Port auf eingehende Verbindungen, was auch als „listen" bezeichnet wird. Ein Server kann durch die Nutzung unterschiedlicher Ports mit mehreren Rechnern gleichzeitig kommunizieren.

- Client: Der Client ist ein Rechner, dem eine eindeutige IP-Adresse zugewiesen ist, über welche er mit dem Server kommuniziert.

- IP-Adresse: Am weitesten verbreitet sind IPv4-Adressen, die 4 Zahlen mit Werten zwischen 0 und 255 einnehmen. Damit wird einem Datenpaket bzw. einem Computer eine eindeutige Adresse zugewiesen.

- Port: Portnummern sind 16-Bit-Zahlen zwischen 0 und 65535, wobei die Ports bis 1023 bereits durch die IANA vergeben sind.

Im ersten Schritt des Verbindungsaufbaus sendet der Client dem Server ein SYN-Paket mit einer bestimmten Sequenznummer, welche die ordnungsgemäße Übertragung kontrolliert. Diese Sequenznummer befindet sich im SYN-Bit, dem Header des Pakets, und wird für gewöhnlich zufällig ausgewählt, um Sicherheitsrisiken zu vermeiden. Insofern der Port zum Server geöffnet ist, antwortet dieser indem ein SYN/ACK-Paket an den Client zurückschickt. Damit wird der Erhalt des SYN-Paketes bestätigt und einem Verbindungsaufbau zugestimmt.

Nachdem der Erhalt des SYN/ACK-Paketes durch den Client, durch das Senden eines eigenen ACK-Paketes mit nächsthöherer Sequenznummer, bestätigt wurde, ist die Verbindung aufgebaut. Von der TCP-Perspektive aus, sind nun beide Kommunikationspartner gleichberechtigt und es spielt keine Rolle mehr, wer der Server und wer der Client ist.

Der TCP Header

Neben dem Header beinhaltet das TCP-Segment noch das Payload (Nutzlast). Darin befinden sich die Daten, die übertragen werden sollen. Auch Protokollinformationen wie HTTP oder FTP können Bestandteil des Payloads sein.

Die wichtigen Daten, die zum Aufbau der Verbindung und zur Kommunikation zwischen zwei Rechnern benötigt wird, befinden sich allerdings im TCP-Header. Dieser Beitrag geht daher ausführlich auf die Funktionsweise und den Aufbau des Kernstückes im TCP-Protokoll ein.

Aufbau des TCP-Headers

Der Header enthält verschiedene Felder, die für den Aufbau der Kommunikation verantwortlich sind. Jedem Feld kommt eine bestimmte Funktion zu, die im TCP-Protokoll spezifiziert ist. In der Regel hat ein TCP-Header eine Größe von 20 Bytes, dessen Werte folgende sind:

- Source Port: Der Quellport besteht aus zwei Byte und gibt die Portnummer auf der Senderseite an.
- Destination Port: Der Zielport besteht ebenfalls aus zwei Byte und gibt die Portnummer auf der Empfängerseite an.
- Sequenz Number: Die Sequenznummer hat eine Größe von vier Byte und vergibt eine zufällig generierte Nummer bei der Initialisierung der Verbindung (wenn das SYN-Flag gesetzt ist). Während der Datenübertragung sind die Sequenznummern für die Sortierung der TCP-Segmente verantwortlich.
- Acknowledgement Number: Die Quittierungsnummer hat ebenfalls eine Größe von vier Byte und gibt die Sequenznummer an, die der Empfänger als nächstes erwartet. Allerdings ist diese nur bei gesetztem ACK-Flag gültig.
- Data Offset: Gibt die Länge des TCP-Headers in 32-Bit-Blöcken an und zeigt die Startadresse der Nutzdaten (Payload).
- Reserved: Dieses Feld hat eine Größe von sechs Bit. Es muss Null sein und wird nicht verwendet.
- Control-Flags: Die sechs Bit großen Variablen können verschiedene Funktionen auszuführen. Abhängig von den Zuständen, gesetzt oder nicht gesetzt, kennzeichnen diese einen Zustand, der zur Weiterverarbeitung der Daten benötigt wird. Auf die einzelnen Typen wird im nächsten Absatz näher eingegangen.
- Receive Window: Das Empfangsfenster gibt die Anzahl der Bytes wieder, die der Sender des Paketes empfangen möchte. Dabei wird mit dem indizierten Daten-Oktett begonnen. Die Größe des Feldes beträgt zwei Bytes.
- Checksum: Mit Hilfe der Prüfsumme werden Übertragungsfehler erkannt und berechnet. Die Prüfsumme ist zwei Byte groß und beinhaltet die Ziel-IP, Quell-IP, die TCP-Protokollkennung und die Länge des Headers und der Nutzdaten.
- Urgent Pointer: Wenn das URG-Flag gesetzt ist, gibt dieser Wert zusammen mit der Sequenznummer die Position des ersten Bytes nach den Urgent-Daten, welche sofort nach dem Header beginnen, an. Die Größe beträgt zwei Byte.
- Options: In der Regel wird das Options-Feld nicht genutzt. Es kann jedoch Zusatzinformationen von einer Größe bis zu 40 Byte enthalten. Es wird nur angewendet, wenn zusätzliche Verbindungsdaten ausgehandelt werden müssen, die nicht bereits im TCP-Header enthalten sind.

Verschiedene Typen von Control-Flags

Die bereits kurz angesprochenen Control-Flags sind zweiwertige Variablen, welche bestimmte Zustände kennzeichnen, die zur weiteren Datenübertragung benötigt werden. Alle Flags können entweder „gesetzt“ oder „nicht gesetzt“ sein und indizieren damit ihren Zustand. Die unterschiedlichen Typen von Control-

Flags, die im TCP-Header eingesetzt werden können, sind im Folgenden dargestellt:

- Urgent-Flag (URG): Wird die Urgent-Flag gesetzt, werden die Daten, die dem Header folgen, sofort durch die Anwendung bearbeitet. Die Verarbeitung der aktuellen Daten wird dabei unterbrochen und die Anwendung liest alle Bytes bis zu dem Punkt, an dem sie durch das Urgent-Pointer-Feld begrenzt werden. Nützlich ist dieses Flag für den Empfänger, um eine Anwendung abzubrechen.
- Acknowledgement-Flag (ACK): Dieses Flag ist für die Bestätigung des Empfangs von TCP-Segmenten verantwortlich. Dies geschieht in Verbindung mit der Acknowledgement-Nummer, die allerdings nur bei einem gesetzten Flag gültig ist.
- Push-Flag (PSH): Bei der Datenübertraungg mit TCP werden zwei Puffer verwendet, um mehrere kleine Daten effizienter an den Empfänger zu versenden. Bei einem gesetzten PSH-Flag ist sowohl der eingehende als auch der ausgehende Puffer aktiv. Dieser hilft nun dabei, den kontinuierlichen Datenstrom zu bündeln und dann als ein großes Paket an die Anwendung zu schicken.
- Reset-Flag (RST): Bei technischen Problemen oder der Abweisung von unerwünschten Verbindungen wird der Reset-Flag gesetzt, um die Kommunikation sofort abzubrechen.
- Synchronization-Flag (SYN): Eine gesetzte SYN-Flag initiiert eine Verbindung und synchronisiert Sequenznummern beim Verbindungsaufbau. Nach der Versendung einer SYN-Flag antwortet der Server normalerweise mit einer Bestätigung (SYN/ACK), sobald er bereit ist die Verbindung aufzunehmen.
- Finish-Flag (FIN): Das Schlussflag zeigt an, dass vom Sender keine weiteren Daten mehr kommen. Damit wird die Verbindung aufgenommen. Um die SYN- und FIN-Flags in der korrekten Reihenfolge abzuarbeiten, werden diese mit Sequenznummern versehen.

Mit Hilfe der enthalten TCP-Segmente im Header, ist es möglich die Nutzdaten (Payload) zuverlässig zu übertragen. Durch den Zustand der unterschiedlichen Flags wird die Kommunikation außerdem noch zusätzlich gesteuert, so dass die empfangen Daten von der Anwendung richtig gelesen und behandelt werden können.

Datenübertragung durch TCP/IP

Bevor mit dem Datenaustausch begonnen wird, einigen sich Sender und Empfänger über den TCP-Header auf die maximale Größe des TCP-Segments (MSS). Im Normalfall ist ein Segment maximal 1.500 Bytes groß, wobei es in die darunter liegende Übertragungsschicht passen muss. Bei dieser Schicht handelt es sich um das Internetprotokoll (IP).

Gemeinsam legen TCP und IP einen Header mit einer Größe von 1.500 Bytes fest, so dass insgesamt Platz für bis zu 1.460 Bytes für Nutzdaten zur Verfügung steht. Wenn der Sender einen Datenblock mit einer Größe von mehreren Kilobyte versenden möchte, wird dieser durch die TCP-Software in mehrere kleine Pakete geteilt und versendet. Vor der Versendung wird jedes Datenpaket mit einem eigenen Header ausgestattet. Die TCP-Segmente landen zunächst in einem Puffer, wo dann die Weiterleitung reguliert wird.

Die einzelnen Datenpakete, die alle mit einer Sequenznummer versehen sind, werden nun nacheinander aus dem Puffer an den Empfänger verschickt und dort wieder zugeordnet. Der Empfänger bestätigt den Erhalt der TCP-Segmente durch den Abgleich mit der Prüfsumme bzw. fordert diese erneut an.

Wiederholte Sendung über den Retransmission Timeout

Insofern bei der Datenübertragung eines oder mehrere TCP-Segmente verlorengegangen sind, verschickt der Empfänger dem Sender dafür keine Bestätigung. Wenn nach einer festgelegten Zeit beim Sender keine

Empfangsbestätigung eingegangen ist, läuft ein Timer ab, der zum Zeitpunkt der Versendung des Segments gestartet wurde. Nach Ablauf des Timers wird das Datenpaket erneut versendet.

Die Schwierigkeit ist es hier, den Timer für die richtige Zeit einzustellen. Eine zu kurze Zeit kann bewirken, dass korrekt gelieferte Pakete erneut gesendet werden, wobei ein zu hohes Timeout die wiederholte Versendung unnötig verspätet. Da die Laufzeiten eines Datenpaketes sehr unterschiedlich sein können, wird der Timer dynamisch an die vorliegende Verbindung angepasst.

Der Retransmission Timeout (RTO), also die Zeit bis zur erneuten Sendung eines TCP-Segments, ergibt sich aus zwei Variablen, die der Sender mitführt. Das ist zum einen die geschätzte Round Trip Time (SRTT) und zum anderen die Varianz (RTTVAR). Die Messung der Werte basiert im Grunde auf den ständig aktualisierten Durchlaufzeiten der erfolgreich übertragenen Pakete.

Flusssteuerung und Staukontrolle

Neben dem Timeout, der den Verlust von Datenpaketen verhindert, gibt es mit der Flusssteuerung und der Staukontrolle zwei weitere TCP-Konzepte. Diese dienen zum einen zur Regulierung eines effizienten Datenflusses und zum anderen dazu, Überlastungen in der Verbindung zu vermeiden.

Da die Anwendung Daten aus dem Puffer liest, ist der Füllstand ständig wechselhaft. Diese Änderungen im Füllstand müssen gesteuert werden, damit eine flüssige Datenübertragung möglich ist. Die Flusssteuerung basiert auf dem Sliding Window, welches den Empfang von Datenpaketen, die Gruppierung und die Weitersendung an den Empfänger reguliert.

Das Konzept der Staukontrolle bzw. Überlaststeuerung soll dabei helfen, Datenstaus in Netzen vorzubeugen. Wenn Verbindungen stark belastet werden, kommt es häufiger vor, dass Pakete ihren Empfänger nicht erreichen und erneut gesendet werden müssen, was letztendlich in einem Datenstau enden kann.

Eine Überbelastung des Netzwerks wird durch eine ständige Beobachtung der Verlustrate von Datenpaketen erreicht. Die TCP/IP-Verbindung wird grundsätzlich langsam gestartet und die Senderate dann schrittweise erhöht, bis es zum Verlust von Daten kommt. Die Senderate wird durch den Datenverlust vermindert und ohne Verlust erhöht. Durch einen Algorithmus kann die Senderate dem Maximum nahe gebracht werden und sich dort einpegeln.

Mit dem genauen Einsatz des Algorithmus sowie der Gestaltung der Überlaststeuerung, beschäftigen sich auch heute noch viele Wissenschaftler. Besonders im Bereich der drahtlosen Übertragungstechnik, bei der es oft zu hohen Verzögerungen in der Laufzeit kommt, wird kontinuierlich nach Verbesserungen und Anpassungen an die äußeren Gegebenheiten gesucht.

Sicherung durch die TCP-Prüfsumme

Ein weiterer Mechanismus zur Garantie einer zuverlässigen Datenübertragung ist die TCP-Prüfsumme, welche sich im Header des Datenpakets befindet. Besonders wenn IP mit TCP eingesetzt wird, sollte eine zusätzliche Sicherung in dem TCP-Header integriert werden. Dazu wird ein Pseudo-Header gebildet, der aus den IP-Adressen von Sender und Empfänger, einem Null-Byte, einem Byte mit dem Wert sechs und der Länge des TCP-Segments besteht.

Der Pseudo-Header wird vor den eigentlichen TCP-Header gelegt. Nach der Berechnung der Prüfsumme wird der Wert in dem Feld „checksum“ im TCP-Header abgelegt und mit dem Datenpaket versendet. Der Pseudo-Header an sich wird jedoch niemals versendet. Vom Empfänger wird auf der anderen Seite ebenfalls ein Pseudo-Header erstellt und eine Berechnung der Prüfsumme durchgeführt.

Die TCP/IP-Protokoll-Familie

Die Begriffe der Internetprotokollfamilie und TCP/IP-Protokoll-Familie werden oft synonym verwendet, jedoch beinhaltet diese rund 500 verschiedenen Netzwerkprotokolle, die weit über die beiden eigentlichen Protokolle TCP und IP hinausgehen. Aus der Anwendersicht und besonders bei der Verwendung des World Wide Web nehmen TCP und IP jedoch die wichtigste Rolle ein.

Das TCP/IP-Referenzmodell

Die Grundlage für die Internetprotokollfamilie bildet das DoD-Schichtenmodell, welches vom Verteidigungsministerium der Vereinigten Staaten in den 70er-Jahren, zu Zwecken der Datenkommunikation entwickelt wurde. Die Kommunikation zwischen Rechnern basiert auf Netzwerkprotokollen, welche in funktionale Schichten unterteilt werden. Für die Gliederung der Internetprotokollfamilie wird dabei das aus vier Schichten bestehende TCP/IP-Referenzmodell genutzt.

Das Referenzmodell ist für den Datenaustausch, über die Grenzen lokaler Netzwerk hinweg, konzipiert. In dem Modell werden allerdings keine Technik für die Übertragung und Zugriffe auf das Übertragungsmedium festgelegt, sondern eine Weitervermittlung von Datenpaketen über sogenannte Hops (Punkt-zu-Punkt-Verbindungen) als Basis zum Verbindungsaufbau definiert.

Im Allgemeinen wird die Netzwerkkommunikation meist durch das ISO/OSI-Referenzmodell beschrieben, welches sehr viel detaillierter ist, als das TCP/IP-Modell. Im Folgenden werden die einzelnen Schichten der Internetprotokollfamilie, in Anlehnung an die OSI-Schicht, beschrieben.

- Anwendungsschicht: Die oberste Schicht leitet die anwendungsspezifischen Anforderungen an die Transportschicht weiter.
- Transportschicht: Hier wird der Header hinzugefügt und die Daten, in Form von Paketen, an die Internetschicht weitergeleitet.
- Internetschicht: Die Internetschicht fügt dem Header die IP-Ursprungs- und Zieladresse hinzu und leitet die Datenpakete weiter.
- Netzzugangsschicht: Die letzte Schicht kann unterschiedliche Techniken zum Host-Netz-Anschluss enthalten.

1. Anwendungsschicht

Die TCP/IP-Schicht der Anwendungen kann mit den Schichten 5 – 7 im ISO/OSI-Referenzmodell (Anwendungsschicht, Darstellung, Sitzung) gleichgesetzt werden. Diese Schicht umfasst alle Protokolle, welche die Netzwerkinfrastruktur zum Austausch anwendungsspezifischer Daten nutzen und mit den Anwendungsprogrammen zusammenarbeiten.

Einige bekannte Beispiele für Protokolle, die in der Anwendungsschicht zu finden sind, lauten:

- DNS: Das Domain Name System dient zur Übersetzung von numerischen IP-Adressen in Domainnamen.
- HTTP: Hypertext Transfer Protocol dient zum Laden von Webseiten im Webbrowser.
- FTP: Das File Transfer Protocol gewährleistet den Datentransfer zwischen Client und Server.
- POP: Das Post Office Protocol wird zum Abrufen von E-Mails genutzt.

2. Transportschicht

Die zweite Schicht im TCP/IP-Modell ist ähnlich der vierten OSI-Schicht, die sich ebenfalls Transportschicht nennt. In dieser Schicht wird eine Ende-zu-Ende-Verbindung zwischen den beiden Teilnehmern des Netzwerks hergestellt. Das wichtigste Protokoll dieser Schicht stellt das TCP dar, das zuverlässige Verbindungen für das Versenden von Datenströmen herstellt.

Neben dem TCP gehören in diese Schicht auch Datagramm-Protokolle, die für eine zuverlässige Zustellung an den richtigen Dienst sorgen, jedoch keine Verbindung aufbauen. Beispiele dafür sind:

- UDP: Das User Datagram Protocol überträgt Datenpakete verbindungslos und mit einem geringen Overhead.
- SCTP: Das Stream Control Transmission Protocol ist wie das TCP ein Transportprotokoll.
- TLC: Bei dem Transport Layer Security handelt es sich um eine Erweiterung von TCP um die Verschlüsselung (ehemals SSL).

3. Internetschicht

In dem OSI/ISO-Referenzmodell wird diese Schicht als Vermittlungsschicht bezeichnet. Bei TCP/IP ist die Internetschicht für das Routing, also der Wahl des richtigen Weges von Datenpakten, zuständig. In dieser und der nächsten Schicht werden Direktverbindungen betrachtet, bei denen es darum geht, für ein empfangenes Paket das nächste Zwischenziel zu ermitteln und dieses weiterzuleiten.

Den Kern dieser Schicht bildet das Internet Protocol (IP) in der Version 4 (IPv4) oder 6 (IPv6). Dieser stellt einen sogenannten Paketauslieferungsdienst zur Verfügung, bei dem Dual-Stacks erkennen können, ob ein Teilnehmer im Netzwerk über IPv4 oder IPv6 erreicht werden kann. Über das Internet Protocol hinaus, können sich die folgenden Protokolle in der Internetschicht befinden:

- ICMP: Das Internet Message Protocol ist Teil einer jeden IP-Implementierung und sendet Kontrollnachrichten.
- BGP: Durch das Border Gateway Protocol werden Informationen zwischen autonomen Systemen über TCP ausgetauscht.
- RIP: Das Routing Information Protocol erlaubt den Informationsaustausch zwischen Routern.

4. Netzzugangsschicht

Die letzte Schicht im TCP/IP-Model ist mit den ersten beiden Schichten im OSI/ISO-Referenzmodell gleichzusetzen, welche die Aufgaben der Sicherung und Bitübertragung übernehmen. Im TCP/IP-Model enthält diese Schicht keine Protokolle der Internetprotokollfamilie, sondern wird als Platzhalter für unterschiedliche Techniken zur Übertragung von Daten verstanden.

Das Ziel der Netzzugangsschicht besteht in der Zusammenführung verschiedener Subsysteme, um diese durch andere Protokolle wie Ethernet, FDDI oder PPP an die Host-zu-Netz-Schicht anzuschließen. Einige dieser Techniken zur Datenübertragung sind:

- WLAN: Das Wireless Local Area Network ermöglicht die drahtlose Datenübertragung über den Netzwerkstandard IEEE 802.11.
- PPP: Mit dem Point-to-Point Protokoll, das in der RFC 1661 definiert ist, werden Verbindungen zwischen zwei Punkten aufgebaut.

- ARP: Mit Hilfe des Adress Resolution Protocol werden IP-Adressen in Geräteadressen umgesetzt.

TCP im OSI-Modell

Dieses OSI-Modell stellt einen weit verbreiteten Standard für die Datenkommunikation dar. Das Ziel ist es dabei, eine Kommunikation zu ermöglichen, die unabhängig von einzelnen technischen Systemen und Protokollen funktioniert.

Eine zentrale Funktion in diesem Schichtenmodell wird von TCP eingenommen, das für den Transport der Datenpakete verantwortlich ist. Aus welchen weiteren Schichten die Architektur des OSI-Modells besteht und wie sich TCP in die Struktur eingliedert, beschreibt der folgende Beitrag.

Entwicklung des OSI-Modells

Mit der Entwicklung des Open Systems Interconnections Modells (OSI) wurde bereits im Jahr 1977 begonnen. Seit 1983 bzw. 1984 wird es von der International Telecommunication Union und von der International Organization for Standardization als offizieller Standard geführt.

Das OSI-Modell wurde mit dem Ziel entwickelt, eine systemunabhängige Kommunikation zu ermöglichen und die ständige Weiterentwicklung zu begünstigen. Aus diesem Grund wurden insgesamt sieben Schichten definiert, denen jeweils eng begrenzte Aufgaben zugeordnet sind. Innerhalb einer Schicht ist der Austausch einzelner Netzwerkprotokolle, durch die Vorgabe klarer Schnittstellen, möglich.

Notwendigkeit eines Referenzmodells

Anders als es auf den ersten Blick erscheinen mag, ist die Kommunikation in einem Netzwerk äußerst komplex. Es werden zahlreiche abstrakte Aufgaben erfüllt und Anforderungen an die Zuverlässigkeit, die Sicherheit und die Effizienz der Kommunikation gestellt. Dabei müssen Probleme berücksichtigt werden, die bei der allgemeinen Übertragung elektronischer Signale oder der spezifischen Kommunikation zwischen den einzelnen Anwendungen auftreten können.

Um der Masse an Aufgaben und Problemen gerecht zu werden, wurde das Modell in verschiedene Schichten mit festgelegten Anforderungen aufgeteilt. In jeder Schicht werden einzelne Instanzen eingesetzt, die bestimmte Funktionen ausführen. Diese Instanzen sorgen für ein gemeinsames Regelwerk zwischen Sender und Empfänger, welches in Protokollen festgeschrieben ist und eine horizontale Verbindung der Teilnehmer erlaubt.

Die Dienste, die von einer Instanz in der jeweiligen Schicht zur Verfügung gestellt werden, können von der direkt darüberliegenden Schicht genutzt werden. Somit ist auch der vertikale Datenfluss gewährleistet. Ein Austausch der Instanzen ist dann möglich, wenn diese beim Sender und Empfänger gleichermaßen Akzeptanz finden.

Sieben Schichten des OSI-Modells

Wie bereits angedeutet, übernimmt jede einzelne Schicht spezielle Aufgaben bei der Kommunikation in einem Netzwerk. Die festgelegten Aufgaben und die Funktionalitäten werden mit höherer Schicht auch immer abstrakter. Eine Übersicht der sieben OSI-Schichten wird im Folgenden gegeben.

1. Schicht – Bitübertragung (Physical)

Die unterste Schicht im OSI-Modell stellt mechanische, elektrische und weitere funktionale Hilfsmittel, zur Aktivierung bzw. Deaktivierung physischer Verbindungen, zur Verfügung. Bei den angewendeten

übertragungstechnischen Verfahren handelt es sich um elektrische Signale, optische Signale wie Lichtleiter oder Laser, elektromagnetische Wellen oder Schall. Der Bitübertragungsschicht zugeordnete Geräte und Netzkomponenten, können Antennen, Verstärker, Repeater oder Transceiver sein.

2. Schicht - Sicherung (Data Link)

Die Sicherungsschicht hat zur Aufgabe, eine zuverlässige Übertragung zu gewährleisten. Außerdem wird in dieser Schicht der Zugriff auf das Übertragungsmedium geregelt, indem der Bitdatenstrom in einzelne Blöcke aufgeteilt wird. Den Blöcken, die auch als Frames bekannt sind, werden im Rahmen der Kanalkodierung Prüfsummen hinzugefügt, wodurch fehlerhafte Blöcke vom Empfänger erkannt werden können. Als Hardware existieren in dieser Schicht Bridges und Switches, wobei es sich bei den Protokollen hauptsächlich um Ethernet-Protokolle handelt.

3. Schicht - Vermittlung (Network)

Die dritte Ebene wird auch als Paketebene oder Netzwerkschicht bezeichnet. Diese sorgt für eine Schaltung von Verbindungen und die Weitervermittlung von Datenpaketen. In die Schicht eingeschlossen ist auch das Routing, welches für die Wegsuche im gesamten Kommunikationsnetz verantwortlich ist. Konsequenterweise werden in der Vermittlungsschicht auch Router als Hardware eingesetzt. Als bekannteste Protokolle sind IP, IPsec und ICMP zu nennen.

4. Schicht – Transport (Transport)

In der Transportschicht wird eine Ende-zu-Ende-Verbindung hergestellt und der Datenstrom segmentiert. Eine weitere wichtige Aufgabe dieser Schicht besteht in der Stauvermeidung und damit der Herstellung eines effizienten Datenstroms. In der Transportebene ist auch TCP zu finden, das als eines der zentralen Protokolle im OSI-Modell, die Erstellung und Kontrolle der Datenübertragung übernimmt.

5. Schicht – Sitzung (Session)

Die fünfte Schicht sorgt für die Prozesskommunikation zwischen zwei Teilnehmern. Die Sitzungsschicht gewährleistet einen organisierten und synchronisierten Datenaustausch, um Zusammenbrüche der Verbindung zu vermeiden. Beispielsweise befindet sich das Remote Procedure Call Protokoll (RPC) in dieser Schicht, welches Wiederaufsetzpunkte erstellt, die nach einem möglichen Ausfall der Sitzung wieder synchronisiert werden können.

6. Schicht – Darstellung (Presentation)

In der Darstellungsschicht werden die Daten von einer systemabhängigen Form (z.B. ASCII) in eine unabhängige Form umgewandelt. Dadurch wird ein syntaktisch korrekter Datenaustausch zwischen zwei unterschiedlichen Systemen ermöglicht. Außerdem gehört die Kompression und Verschlüsselung zu den Aufgaben dieser Schicht.

7. Schicht – Anwendungen (Application)

Die oberste Schicht des OSI-Modells beinhaltet Dienste, Anwendungen und das Netzwerkmanagement. Die Anwendungsschicht stellt verschiedene Funktionalitäten für die auszuführenden Dienste und Anwendungen bereit. Außerdem stellt diese Schicht die Verbindung zu den darunterliegenden Schichten her und bietet Möglichkeiten zur Dateneingabe und –ausgabe.

TCP-Definition im RFC 793

TCP wurde bereits 1981 im RFC 793 festgeschrieben. Im Laufe der Jahre gab es eine Vielzahl an Erweiterungen, die in das Dokument eingeflossen sind. Dieser Beitrag gibt einen Überblick zu relevanten

RFC's und geht speziell auf das RFC 793 ein.

Hintergrund zu RFCs

Bei den Requests for Comments (RFC) handelt es sich in der wortwörtlichen Übersetzung, um eine „Bitte um Kommentare". Es sind technische und organisatorische Dokumente des RFC-Editors, die seit 1969 zum Internet geführt werden. In der ursprünglich eingereichten Version werden die Dokumente zur Diskussion gestellt. Sie behalten aber auch dann ihren Namen, wenn aus ihnen durch allgemeine Akzeptanz ein Standard geworden ist.

Allerdings werden die RFCs mit fortlaufenden Nummern versehen, die bei jeder abschließenden Aktualisierung neu vergeben werden. Jeder kann im Grunde ein RFC verfassen und bei der Internet Engineering Steering Group (IESG) einreichen, die mit der Verwaltung der Dokumente beauftragt ist. Sobald ein Entwurf zur Diskussion freigegeben wurde, kann sich jeder Interessierte daran beteiligen.

Alle RFCs besitzen einen Status, der auf die Gültigkeit und Reichweite des Dokuments hinweist. Dies können ältere RFCs sein, die nicht mehr benutzt werden (Historic), oder es kann sich um einen Entwurf (Draft Standard) bzw. einen Vorschlag für einen Standard (Proposed Standard) handeln. Wenn ein RFC über den Entwurfsstatus hinaus ist, wird dieses als offizieller Standard (STDn), als potenzieller Standard zum Ausprobieren (Experimental) oder als informative Mitteilung an die Netzgemeinde (Informational), deklariert.

Wichtige RFC's für TCP/IP

Für die Definition des TCP/IP-Protokoll sind besonders die folgenden Standards, die in den RFCs definiert wurden, von großer Bedeutung:

- RFC 768 – User Datagram Protocol (UDP): Das UDP ist ein minimales, verbindungsloses Netzwerkprotokoll, dessen Aufgabe es ist, Daten über das Internet zu übertragen. UDP stellt, anders als TCP, eine nicht-zuverlässige und ungesicherte Datenübertragung bereit, die den Empfang des Datenpaketes nicht garantieren kann.
- RFC 791 – Internet Protocol (IP): Das Internet Protocol stellt als Netzwerkprotokoll die Grundlage des Internets dar. Es handelt sich um ein verbindungsloses Protokoll, das für die Adressierung von Rechnern in einem Netzwerk sorgt und damit das Routing ermöglicht.
- RFC 792 – Internet Control Message Protocol (ICMP): Das ICMP dient zum Austausch von Informations- und Fehlermeldungen. Das Protokoll ist ein Bestandteil von IPv4, wird jedoch als eigenständiges Protokoll behandelt.
- RFC 793 – Transmission Control Protocol (TCP): Neben IP stellt TCP das Herzstück der Internetprotokollfamilie dar. TCP sorgt für eine zuverlässige, verbindungsorientierte und paketvermittelte Datenübertragung zwischen den Rechnern in einem Netzwerk.

Die inhaltliche Strukturierung der RFCs ist sehr formal und ist in eigens dafür vorgesehenen RFCs gesondert festgelegt (RFC 2119 und RFC 2223). Alle Änderungen und Vorschläge für ein RFC werden vor der formellen Veröffentlichung exakt dokumentiert. Danach kann dieses nicht mehr korrigiert, sondern lediglich durch ein neues RFC abgelöst werden. Schreibweise, Begrifflichkeiten und Zusammensetzung von Zeichenketten sind klar definiert, um Verwirrung bei der Interpretation zu vermeiden.

RFC 793: TCP

Die erste Standardisierung von TCP erfolgte im Jahr 1981 mit dem Eintrag in das RFC 973. Das Dokument hält sich, wie alle anderen RFCs, an eine vorgeschriebene Struktur und Sprache. In den folgenden

Absätzen sollen die Inhalte des RFC 973 zum TCP, in kurzer Form wiedergegeben werden.

1. Einleitung

In der RCF 793 ist die Rede von dem DoD Standard Transmission Control Protocol, das heutzutage unter der Abkürzung TCP geläufig ist. Es wird auch kurz auf den Vorgänger, das Arpanet, eingegangen und die Weiterentwicklung beschrieben. Zur Einführung in den TCP-Standard werden die Motivation und der Umfang der TCP-Beschreibung festgelegt. Außerdem werden in dem Dokument Aussagen zur Oberfläche und der Ausführung des TCP-Standards getätigt

2. Philosophie

Unter dem zweiten Abschnitt finden sich die Grundideen des TCP-Protokolls wieder. Es wird von den verschiedenen Elementen des Internetwork Systems gesprochen und ein modellhafter Ablauf der Datenkommunikation beschrieben. Weitere Kapitel gehen auf die Host-Umgebung, die Interrelation mit anderen Protokollen, die zuverlässige Kommunikation und die Robustheit des TCP-Protokolls ein. Im Grunde findet sich hier das gesamte Gedankengut wieder, das zur Entwicklung des Standards in den 1970er Jahren geführt hat.

3. Funktionale Spezifikationen

In dem letzten Abschnitt werden die Spezifikationen des Protokolls genau definiert. Dabei wird der Aufbau des TCP-Headers beschrieben und Terminologien festgelegt. Auch die Vergabe von Sequenznummern wurde spezifiziert. Relativ ausführlich beschäftigt sich das Dokument dann mit dem Aufbau und der Beendigung einer Verbindung, bevor es auf die Fehlererkennung und –vermeidung eingeht. Abschließend wird der Ablauf einer allgemeinen Datenkommunikation mit den dazugehörigen Schnittstellen und der Datenverarbeitung spezifiziert.