Kommunikationsarten

Leitungs- vs. paketbasierte Verbindungen

Leitungsbasierende Kommunikation

Leitungsvermittelte Verbindungen öffnen einen exklusiven Kommunikationskanal zwischen zwei Kommunikationspartnern - ein Beispiel ist das frühere Telefonnetz.[1)]

Durch Änderung der Steckverbindung, später durch Relais und elektronische Schaltungen wurden die Teilnehmer direkt verbunden. Der gesamte Kommunikationsvorgang nutzte diesen Kanal. Bei leitungsbasierten Verbindungen ist der logistische Aufwand recht hoch, die Vermittlungszeit ebenfalls, dafür kann man sehr genau vorhersagen, mit welcher Bandbreite die Kommunikation durchgeführt werden kann, da die Verbindung für diesen einen Kommunikationsvorgang "reserviert" ist.

Paketbasierte Kommunikation

Bei der Paketbasierten Kommunikation (z.B. per TCP oder UDP), werden die Kommunikationsinhalte in "Datenpakete aufgeteilt und über ein vermaschtes Netz an den Empfänger vermittelt - jedes Paket einzeln. Dabei können verschiedene Teile der Kommunikation durchaus unterschiedliche Wege durch das Netzwerk nehmen und in anderer Reihenfolge am Ziel ankommen, als sie abgesendet wurden.

Grafik: Oddbodz - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=29033823

TCP genauer angeschaut

Mit dem TCP Protokoll kann man Informationen in beide Richtungen übertragen. Kommunikation über TCP ermöglicht es, zur gleichen Zeit Daten zu senden und zu empfangen. Die Übertragungseinheiten, auf die TCP bei der Datenübertragung zurückgreift, sind Segmente (Pakete), die zusätzlich zu den Nutzdaten auch Metainformationen enthalten können. Die maximale Paketgröße ist auf 1.500 Byte beschränkt (MTU - "Maximum Transfer Unit"). Verbindungsauf- und -abbau und auch die Datenübertragung werden von TCP-Software im Netzwerk-Protokollstapel des Betriebssystems übernommen.

Der TCP-Stack des Betriebssystems wird von den Netzwerkanwendungen wie Webbrowsern oder Serverprogrammen über spezifische Schnittstellen bedient, jede Verbindung ist dabei durch zwei Endpunkte – **Client** und **Server** – definiert. Welcher Kommunikationspartner die Client- und welche die Serverrolle übernimmt, spielt dabei zunächst keine Rolle – wichtig ist, dass der TCP-Stack an jedem Endpunkt ein eindeutiges, Paar aus **IP-Adresse** und **Port** verwenden kann.

Beispiel:

TCP-Verbindungsaufbau: Three-Way-Handshake

Damit Client und Server eine TCP Verbindung aufbauen können, müssen beide über eine IP-Adresse verfügen und den Port für die Datenübertragung definiert und freigegeben haben[2)].

Der Ablauf beim Verbindungsaufbau mit TCP sieht folgendermaßen aus:

- Im ersten Schritt sendet der Client dem Server ein **SYN-Paket** (**syn**chronize) mit einer individuellen, zufälligen Sequenznummer. Diese Nummer stellt die Übertragung in der korrekten Reihenfolge und ohne Duplikate sicher.
- Hat der Server das Paket erhalten, stimmt er dem Verbindungsaufbau zu, indem er ein **SYN-ACK-Paket** (**ack**nowledgement) zusammen mit der um 1 erhöhten Sequenznummer des Clients zurückschickt. Außerdem übermittelt er dem Client eine eigene (zufällige) Sequenznummer.
- Zum Abschluss des Verbindungsaufbaus bestätigt der Client den Erhalt des SYN-ACK-Segments, indem er ein eigenes ACK-Paket versendet, das die um 1 erhöhte Sequenznummer des Servers enthält.

(A1)

Baue mit Filius das Netz im Abschnitt Simulation des WWW auf, zunächst **ohne DNS und die Erweiterungen** - es genügt wenn du auf dem Clientcomputer mit dem Webbrowser eine Webseite auf dem Server öffnen kannst.

Vollziehe den TCP Verbindungsaufbau nach, wenn du mit einem Webbrowser eine Webseite öffnest und den Datenverkehr betrachtest.

Kannst in der Analyse des Datenverkehrs herausfinden, ob eine TCP Verbindung auch beendet wird? Recherchiere und erstelle einen Heftaufschrieb.

1)
Bildquelle: https://commons.wikimedia.org/wiki/File:Telefonistinde.jpg, Lizenz: Creative-Commons-Lizenz "Namensnennung 3.0 nicht portiert", Beschreibung: *Dansk: Manuel betjening på telefoncentral* Datum: *2. April 2017* Quelle: *MyNewsDesk.com* Urheber: *Arkiv*

2)
Hier greifen z.B. Firewalls ein, indem Sie den Zugriff auf bestimmte Ports blockieren