TCP einfach erklärt

So kommen Daten zuverlässig bei der richtigen App an.

TCP steht für **Transmission Control Protocol**. Es ist verbindungsorientiert: Die Geräte bauen zuerst eine Verbindung auf und tauschen dann Daten aus.

Übertragung kontrollieren

Empfang bestätigen, fehlende Daten erneut senden und die Reihenfolge sichern.

Die richtige App erreichen

Portnummern helfen, Daten der passenden Anwendung am Endgerät zuzuordnen.



Max öffnet **firma-xyz.at**. DNS ermittelt dazu **80.30.20.10**. Im Video folgt eine HTTPS-Verbindung über TCP.

Vereinfachtes Beispiel: öffentliche IPv4-Adressen, ohne NAT. Netzwerkgeräte dazwischen sind mitgedacht.

1 Verbindung aufbauen: Three-Way-Handshake

Laptop → Server	**SYN**	Laptop sendet seine Startnummer.
Server → Laptop	**SYN + ACK**	Server bestätigt und sendet seine Startnummer.
Laptop → Server	**ACK**	Laptop bestätigt das SYN des Servers.

Beide Seiten sind synchronisiert. Die TCP-Verbindung steht.

2 Ports und Socket-Adressen

TCP-Ports sind **virtuelle Nummern für Anwendungen**, keine LAN-Buchsen. Sie werden bereits beim Verbindungsaufbau genutzt.

Laptop: **200.30.20.10:65000** → Server: **80.30.20.10:443**

IP-Adresse + Port = **Socket-Adresse**. Beide Seiten kennzeichnen die TCP-Verbindung.

Dienst	Standard-Port	Aufgabe
HTTPS / HTTP	443 / 80	Website verschlüsselt / unverschlüsselt
FTP	21 / 20	Steuerung / Daten im aktiven Modus
SMTP / IMAP	465 / 993	E-Mails senden / abrufen, TLS von Beginn an

Der Laptop wählt einen freien Absender-Port, häufig zwischen 49152 und 65535; hier 65000.

Übertragen und bestätigen

Vom verschlüsselten HTTPS-Request bis zum kontrollierten Verbindungsabbau.

3 Anfrage in TCP-Segmenten übertragen

Die HTTPS-Anfrage wird verschlüsselt und in einem oder mehreren **TCP-Segmenten** übertragen. Bei HTTPS übernimmt **TLS** die Verschlüsselung. Die **Sequenznummer** eines Datensegments nennt die Position seines ersten Datenbytes im Datenstrom.

ACK bestätigt den Empfang.
Eine Bestätigung kann mehrere Segmente abdecken.

Über Ziel-Port **443** ordnet der Server die Daten seiner Webserver-App zu.

Fehlende Daten erneut senden

Geht ein Segment verloren oder wird es wegen eines Fehlers verworfen, sendet TCP die fehlenden Daten bei Bedarf erneut, etwa wenn Bestätigungen ausbleiben.

FCS im Frame: erkennt Übertragungsfehler, repariert sie aber nicht. **TCP** kann daraus entstandene Datenverluste durch erneutes Senden ausgleichen.

Die Datenmenge passend steuern

Flusssteuerung: Der Empfänger teilt mit, wie viele Daten er aufnehmen kann. Der Sender berücksichtigt diesen verfügbaren Platz.

Im Video startet die Antwort mit wenigen Segmenten. Mit Bestätigungen kann die Menge noch unbestätigter Daten wachsen, wenn die Verbindung es zulässt. Diese Anpassung an das Netzwerk gehört zur **Staukontrolle**.

4 Die Website kommt zurück

Der Server antwortet mit **Absender-Port 443** an **Ziel-Port 65000**. TCP bringt empfangene Daten anhand der Sequenznummern in die richtige Reihenfolge. Nach dem Entschlüsseln kann der Browser die Website anzeigen.

5 Verbindung beenden: Four-Way-Handshake

Laptop → Server	**FIN + ACK**	Laptop wird keine weiteren Daten senden.
Server → Laptop	**ACK**	Server bestätigt das FIN des Laptops.
Server → Laptop	**FIN + ACK**	Auch der Server wird nichts mehr senden.
Laptop → Server	**ACK**	Laptop bestätigt das FIN des Servers.

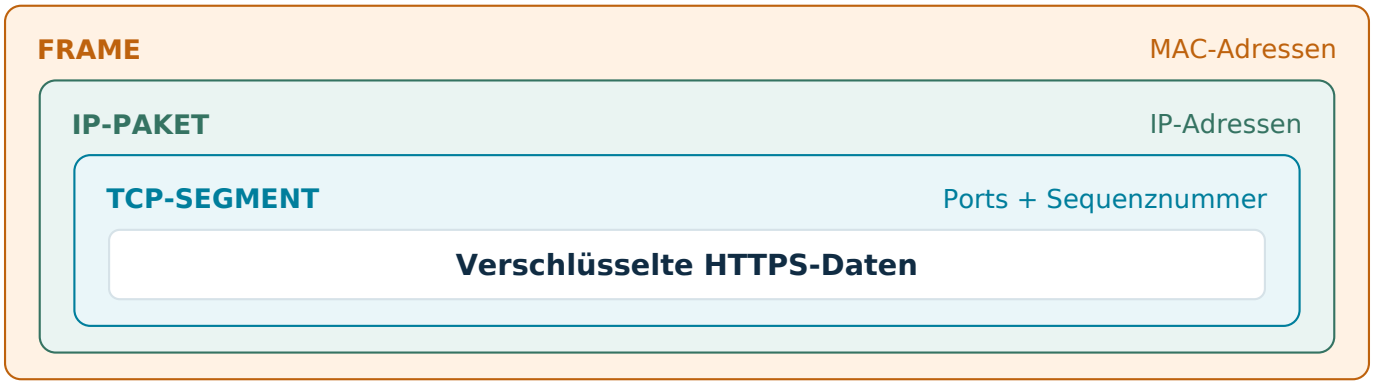
Häufiger Ablauf in vier Schritten. FIN beendet eine Senderichtung; ACK bestätigt den Empfang. Bei FIN + ACK werden mit ACK die bisher empfangenen Daten bestätigt.

Wie TCP durchs Netzwerk reist

Segmente, IP-Pakete und Frames arbeiten bei jeder Übertragung zusammen.

6 Daten verpacken: die Kapselung

Die Daten werden für die Übertragung schrittweise verpackt.



Beim Sender: einpacken

Daten in TCP-Segmente, diese in IP-Pakete und die Pakete in Frames.

Beim Empfänger: auspacken

Frame, IP-Paket und TCP-Segment auswerten. Daten der passenden App zuordnen.

Wer verarbeitet welche Informationen?

Gerät	Aufgabe im Beispiel
Switch / Access Point	Verarbeitet Frames und leitet die enthaltenen Daten weiter.
Router	Verarbeitet Frames und IP-Pakete, um Pakete in das nächste Netzwerk weiterzuleiten.
Laptop / Server	Verarbeitet für die eigene TCP-Verbindung auch Segmente, bestätigt Daten und ordnet sie den Apps zu.

Bei der normalen Weiterleitung transportieren die Netzwerkgeräte TCP-Segmente **mit**. Sie bauen diese TCP-Verbindung nicht auf und bestätigen keine TCP-Daten.

An den Übergängen zwischen Netzwerken entstehen neue Frames. Die Absender- und Ziel-IP-Adresse bleiben in unserem Beispiel **ohne NAT** gleich.

TCP/IP: gemeinsam zum Ziel

IP bringt Pakete über Router und Netzwerke zum Ziel.
TCP sorgt für eine zuverlässige, geordnete Übertragung zwischen Apps.

Weiterlesen (Links): Portübersicht bei Wikipedia | IANA-Portregister
Technische Grundlagen: TCP (RFC 9293) | Staukontrolle (RFC 5681)