Layer 4 (UDP/TCP)

Andere Begriffe dafür:

- Transport
- Transmission
- Übertragung

Layer 3 ermöglichte also, dass Pakete jeden Ort im Netzwerk erreichen und von dort wieder zurückkommen, durch das *Routing*. Layer 4 kümmert sich nun darum, ob und wie die Nachrichten sicher das Ziel erreichen, von dem Ziel richtig empfangen werden, und der Sender auch eine "Quittung" für ein richtig empfangenes Paket bekommt.

Um die Übertragung abzusichern, gibt es einerseits:

- das ICMP Protokoll https://de.wikipedia.org/wiki/Internet_Control_Message_Protocol sowie
- Prüfsummen bei UDP (s.u.) und zusätzlich
- Handshake - Algorithmen (TCP)

Der Begriff "Socket"

Auf Deutsch etwa "Steckdose". Siehe https://de.wikipedia.org/wiki/Socket_(Software).

IP Sockets gibt es als UDP-Sockets und TCP-Sockets.

Ein Socket ist die Kombination (IP-Adresse:Port). Port ist ein 16bit-Integer und kann somit die Werte 0-65535 annehmen.

Eine UDP oder TCP Verbindung ist die Verbindung zweier Sockets. Der Socket des Servers ist im "listening" Modus, der Socket des Clients ist im "sending" Modus. Eine solche Verbindung wird auch als `socket-pair` bezeichnet.

Durch das Konzept von Ports können somit viele Netzwerkverbindungen zugleich auf nur einer IP Adresse realisiert werden.

Unter Linux sind Sockets mit z.B. mit `netstat -46an` anzeigbar.

Programmiertechnisch ist ein Socket ein Objekt, oder "handle", wie ein File-Handle, aus dem das Programm lesen und wohin es auch schreiben kann.

Eine TCP (oder UDP) *Connection* (also Netzwerkverbindung) besteht aus je einem Socket am Server und einem Socket am Client, welche miteinander verbunden sind .
Ein Server hat für jeden bereitgestellten Dienst einen eigenen "listen Port". Man kann sich das wie eine Gegensprechanlage vorstellen mit mehreren Türen (Ports) an einer Adresse.
Standard-Portnummern (zB 80:http 443:https 123:ntp 53:dns 443:https 25:smtp) sind der Datei `/etc/services` zu entnehmen und werden von der IANA verwaltet.

UDP (User Datagram Protocol)

- https://de.wikipedia.org/wiki/User_Datagram_Protocol
- Wichtig: Konzept von Ports (s.u. bei Sockets)
- *Keine gesicherte Datenübertragung*, aber dafür sehr simples Protokoll
- z.B. IP-Telefonie (Pakete können verloren gehen, dann schlechte Qualität)
- Video Streaming: Moderne Codecs können mit verlorenen
- DNS Dienst: hauptsächlich UDP
- wenn die Antwort zu lange dauert: Retransmits in größer werdenden Abständen

TCP (Transmission Control Protocol)

- https://de.wikipedia.org/wiki/Transmission_Control_Protocol
- "TCP" ist eine *Implementierung der Transportschicht*.
- TCP-Pakete *sind* IP Pakete.
- Das Betriebssystem gibt der Anwendung bei TCP eine **Garantie**:
  - Unversehrtheit der Übertragung: Die Daten kommen beim Empfänger unversehrt an (Prüfsummen) oder werden von diesem verworfen.
  - Bei Fehlern in der Datenübertragung bekommt die Anwendung den Fehler mit! (socket Errors)
- notwendig und geeignet für Anwendungen, die verläßliche Datenübertragung brauchen
  - Fileservice
  - Email
  - http(s) Cryptographie, wo fast immer ein fehlerhaftes Bit reicht, damit die Kommunikation fehlschlägt!
  - remote console, und viele viele andere

TCP Verbindungsphasen

- Verbindungs**aufbau**phase: `SYN` Paket
- **verbindung**sphase: "normale" Pakete, jeweils vom anderen mit `ACK` quittiert.
- Verbindungs**abbau**phase: `FIN`

Werkzeuge

- ping
- traceroute
- speedtest.net
- netstat
- wireshark