7.1 OSI-Schicht 4 – Transportschicht

Die Transportschicht ist Schicht 4 des OSI-Modells.

Sie liegt zwischen:

```
Schicht 3: Vermittlungsschicht / Netzwerkschicht
und
Schicht 5: Sitzungsschicht
```

Die Hauptaufgabe von Schicht 4 ist:

```
Daten zwischen Anwendungen auf Endgeräten transportieren.
```

Dabei geht es besonders um:

```
- TCP
- UDP
- Ports
- Verbindungen
- Zuverlässigkeit
- Reihenfolge
- Flusskontrolle
- Fehlerbehandlung
- Segmentierung
```

Merksatz:

```
Schicht 4 = Transport zwischen Anwendungen mit TCP, UDP und Ports.
```

Grundidee von Schicht 4

Schicht 3 bringt IP-Pakete von einem Gerät zu einem anderen Gerät.

Schicht 4 sorgt dafür, dass die Daten zur richtigen Anwendung auf diesem Gerät gelangen.

Beispiel:

Ein Server hat eine IP-Adresse.

Auf dem Server laufen mehrere Dienste.

Zum Beispiel:

Webserver

SSH

Mailserver

Datenbank

Damit klar ist, welcher Dienst gemeint ist, nutzt Schicht 4 Ports.

Merksatz:

IP bringt zum Gerät.

Port bringt zur Anwendung.

Einordnung im OSI-Modell

OSI-Schicht	Name	Aufgabe
7	Anwendungsschicht	Netzwerkdienste für Anwendungen
6	Darstellungsschicht	Darstellung, Codierung, Verschlüsselung
5	Sitzungsschicht	Sitzungen verwalten
4	Transportschicht	Transport zwischen Anwendungen
3	Vermittlungsschicht / Netzwerkschicht	IP-Adressierung und Routing
2	Sicherungsschicht	Frames, MAC-Adressen, lokale Übertragung
1	Bitübertragungsschicht	Bits als Signale übertragen

Merksatz:

Schicht 4 sitzt über IP

und unter den Anwendungsschichten.

Warum braucht man Schicht 4?

Schicht 3 kennt IP-Adressen.

Eine IP-Adresse sagt:

```
welches Gerät?
```

Aber ein Gerät kann viele Dienste gleichzeitig anbieten.

Beispiel Server:

Dienst	typischer Port
HTTP	80
HTTPS	443
SSH	22
DNS	53
SMTP	25
IMAP	143
RDP	3389

Schicht 4 sorgt dafür, dass Daten beim richtigen Dienst landen.

Merksatz:

```
Ohne Ports wüsste ein Gerät nicht,
für welchen Dienst die Daten bestimmt sind.
```

Ports

Ein Port ist eine logische Nummer auf Schicht 4.

Ports werden verwendet, um Anwendungen oder Dienste auf einem Gerät zu unterscheiden.

Beispiel:

```
IP-Adresse:
192.168.10.50

Port:
443
```

Zusammen:

```
192.168.10.50:443
```

Das bedeutet:

```
Dienst auf Port 443
auf dem Gerät 192.168.10.50
```

Merksatz:

```
Port = logische Dienstnummer auf Schicht 4.
```

Port ist nicht gleich Switch-Port

Der Begriff Port kann verwirrend sein.

Begriff	Bedeutung	Schicht
Switch-Port	physischer Anschluss am Switch	1 / 2
TCP-Port	logische Dienstnummer	4
UDP-Port	logische Dienstnummer	4

Beispiel:

```
Switch-Port 5:
Kabel steckt am Switch.

TCP-Port 443:
HTTPS-Dienst auf einem Server.
```

Merksatz:

```
Switch-Port = physisch.
TCP-/UDP-Port = logisch.
```

TCP

TCP steht für:

```
Transmission Control Protocol
```

TCP ist ein verbindungsorientiertes Transportprotokoll.

Das bedeutet:

```
Vor der eigentlichen Datenübertragung wird eine Verbindung aufgebaut.
```

TCP bietet unter anderem:

```
- Verbindungsaufbau
- zuverlässige Übertragung
- Reihenfolge der Daten
- Bestätigungen
- erneute Übertragung bei Verlust
- Flusskontrolle
- Staukontrolle
```

Merksatz:

```
TCP = zuverlässig und verbindungsorientiert.
```

UDP

UDP steht für:

```
User Datagram Protocol
```

UDP ist ein verbindungsloses Transportprotokoll.

Das bedeutet:

```
Es wird keine Verbindung wie bei TCP aufgebaut.
```

UDP sendet Datagramme ohne eingebaute Garantie für:

- Zustellung
- Reihenfolge
- erneute Übertragung
- Bestätigung

UDP ist dadurch einfacher und schneller, aber weniger zuverlässig.

Merksatz:

UDP = schnell und verbindungslos.

TCP und UDP im Vergleich

Merkmal	TCP	UDP
Verbindungsaufbau	ja	nein
Zuverlässigkeit	hoch	keine eingebaute Garantie
Reihenfolge	wird sichergestellt	keine eingebaute Garantie
Bestätigung	ja	nein
erneute Übertragung	ja	nein
Overhead	höher	geringer
typische Nutzung	Web, Mail, SSH	DNS, VoIP, Streaming, Gaming

Merksatz:

TCP = sicherer Transport.
UDP = schlanker Transport.

Verbindungsorientiert

Verbindungsorientiert bedeutet:

Vor der Datenübertragung wird eine Verbindung aufgebaut.

TCP arbeitet verbindungsorientiert.

Dabei wird geprüft:

```
Ist die Gegenstelle erreichbar?
Können beide Seiten kommunizieren?
Welche Anfangswerte werden verwendet?
```

Erst danach werden Nutzdaten übertragen.

Merksatz:

```
Verbindungsorientiert = Verbindung vor Datenübertragung.
```

Verbindungslos

Verbindungslos bedeutet:

```
Es wird keine feste Verbindung aufgebaut.
```

UDP arbeitet verbindungslos.

Ein UDP-Datagramm wird einfach gesendet.

Der Sender weiß durch UDP allein nicht sicher:

```
ob es angekommen ist
ob es in richtiger Reihenfolge angekommen ist
ob es mehrfach angekommen ist
```

Wenn eine Anwendung Zuverlässigkeit braucht, muss sie das selbst lösen.

Merksatz:

```
Verbindungslos = senden ohne vorherigen Verbindungsaufbau.
```

TCP-Handshake

TCP baut eine Verbindung mit dem sogenannten Drei-Wege-Handshake auf.

Die drei Schritte sind:

```
1. SYN
2. SYN-ACK
3. ACK
```

Vereinfacht:

```
Client:
Ich möchte eine Verbindung aufbauen.

Server:
Einverstanden, ich bestätige.

Client:
Bestätigung erhalten.
```

Danach beginnt die eigentliche Datenübertragung.

Merksatz:

```
TCP-Verbindung startet mit SYN, SYN-ACK, ACK.
```

Drei-Wege-Handshake

Schritt	Richtung	Bedeutung
1	Client → Server	SYN
2	Server → Client	SYN-ACK
3	Client → Server	ACK

SYN bedeutet:

```
Synchronize
```

ACK bedeutet:

```
Acknowledgement
```

Nach dem dritten Schritt gilt die TCP-Verbindung als aufgebaut.

Merksatz:

```
Drei-Wege-Handshake = SYN, SYN-ACK, ACK.
```

TCP-Bestätigungen

TCP arbeitet mit Bestätigungen.

Wenn Daten ankommen, bestätigt der Empfänger den Empfang.

Wenn Daten verloren gehen, können sie erneut übertragen werden.

Dadurch erreicht TCP eine zuverlässige Datenübertragung.

Beispiel:

```
Sender sendet Daten.
Empfänger bestätigt Empfang.
Sender weiß:
Daten sind angekommen.
```

Merksatz:

```
TCP nutzt ACKs zur Bestätigung.
```

Reihenfolge bei TCP

TCP sorgt dafür, dass Daten in der richtigen Reihenfolge an die Anwendung übergeben werden.

Wenn Datenpakete unterwegs vertauscht ankommen, kann TCP sie wieder sortieren.

Beispiel:

```
gesendet:
Teil 1, Teil 2, Teil 3

angekommen:
Teil 1, Teil 3, Teil 2
```

TCP kann die Reihenfolge für die Anwendung wieder herstellen.

Merksatz:

```
TCP stellt die Reihenfolge der Daten sicher.
```

Sequenznummern

TCP nutzt Sequenznummern.

Sie helfen dabei:

```
- Daten in Reihenfolge zu bringen
- fehlende Daten zu erkennen
- Daten korrekt zu bestätigen
```

Wichtig:

```
Sequenznummern sind keine Ports.
```

Ports zeigen den Dienst.

Sequenznummern zeigen die Position von Daten im TCP-Datenstrom.

Merksatz:

```
Port = Dienst.
Sequenznummer = Position im TCP-Datenstrom.
```

TCP-Segment

Die Datenform bei TCP auf Schicht 4 heißt:

```
Segment
```

Ein TCP-Segment enthält unter anderem:

```
- Quell-Port
- Ziel-Port
- Sequenznummer
```

- Bestätigungsnummer
- Steuerbits
- Nutzdaten

Merksatz:

TCP arbeitet mit Segmenten.

UDP-Datagramm

Die Datenform bei UDP auf Schicht 4 heißt:

Datagramm

Ein UDP-Datagramm enthält unter anderem:

- Quell-Port
- Ziel-Port
- Länge
- Prüfsumme
- Nutzdaten

UDP ist einfacher aufgebaut als TCP.

Merksatz:

UDP arbeitet mit Datagrammen.

Segment und Datagramm unterscheiden

Protokoll	Datenform auf Schicht 4
TCP	Segment
UDP	Datagramm

Wichtig:

In der Praxis wird oft allgemein von Paketen gesprochen.
Fachlich genauer sollte man unterscheiden.

Merksatz:

TCP = Segment.
UDP = Datagramm.

Portnummern

Portnummern liegen im Bereich:

0 bis 65535

Warum?

Ports sind 16 Bit groß.

16 Bit ermöglichen:

65536 verschiedene Werte

Von:

0

bis:

65535

Merksatz:

Portnummern gehen von 0 bis 65535.

Well-Known Ports

Well-Known Ports sind bekannte, standardisierte Ports.

Sie liegen im Bereich:

```
0 bis 1023
```

Beispiele:

Dienst	Protokoll	Port
FTP	TCP	21
SSH	TCP	22
SMTP	TCP	25
DNS	TCP / UDP	53
HTTP	TCP	80
HTTPS	TCP	443

Merksatz:

```
Well-Known Ports = 0 bis 1023.
```

Registered Ports

Registered Ports liegen im Bereich:

```
1024 bis 49151
```

Sie werden häufig von Anwendungen, Herstellern oder Diensten genutzt.

Beispiele können je nach Anwendung unterschiedlich sein.

Für die Prüfung ist meistens wichtiger:

```
Well-Known Ports kennen
und
Portbereiche grob einordnen.
```

Merksatz:

```
Registered Ports = 1024 bis 49151.
```

Dynamic oder Ephemeral Ports

Dynamic Ports werden auch genannt:

```
Ephemeral Ports
```

Sie liegen im Bereich:

```
49152 bis 65535
```

Clients nutzen solche Ports oft als temporäre Quell-Ports.

Beispiel:

```
Client öffnet Webseite.

Quell-Port:
52344

Ziel-Port:
443
```

Der Quell-Port wird temporär für diese Verbindung genutzt.

Merksatz:

```
Ephemeral Port = temporärer Client-Port.
```

Quell-Port und Ziel-Port

Ein TCP- oder UDP-Paket enthält:

```
- Quell-Port
- Ziel-Port
```

Beispiel HTTPS-Aufruf:

```
Client:
192.168.10.20:52344

Server:
93.184.216.34:443
```

Der Ziel-Port 443 zeigt:

```
Der Client möchte HTTPS erreichen.
```

Der Quell-Port 52344 hilft dem Client, die Antwort der richtigen Verbindung zuzuordnen.

Merksatz:

```
Ziel-Port = gewünschter Dienst.
Quell-Port = Rückzuordnung beim Absender.
```

Socket

Ein Socket beschreibt eine Kombination aus:

```
IP-Adresse
und
Port
```

Beispiel:

```
192.168.10.20:52344
```

Oder:

```
93.184.216.34:443
```

Bei TCP-Verbindungen betrachtet man oft die Kombination aus:

Quell-IP
Quell-Port
Ziel-IP
Ziel-Port
Protokoll

Diese Kombination macht eine Verbindung eindeutig.

Merksatz:

Socket = IP-Adresse plus Port.

Typische Dienste und Ports

Dienst	Protokoll	Port
FTP Steuerverbindung	TCP	21
SSH	TCP	22
Telnet	TCP	23
SMTP	TCP	25
DNS	TCP / UDP	53
DHCP Server	UDP	67
DHCP Client	UDP	68
HTTP	TCP	80
POP3	TCP	110
IMAP	TCP	143
HTTPS	TCP	443
RDP	TCP	3389

Merksatz:

Die wichtigsten Standardports sollte man sicher kennen.

DNS und TCP/UDP

DNS nutzt häufig UDP Port 53.

Warum?

> Normale DNS-Anfragen sind meist klein und schnell.

DNS kann aber auch TCP Port 53 verwenden.

Zum Beispiel:

> - Zonentransfers
> - große Antworten
> - bestimmte DNS-Sicherheits- oder Sonderfälle

Prüfungsfalle:

> DNS ist nicht nur UDP,
> sondern kann TCP und UDP 53 nutzen.

Merksatz:

> DNS = UDP 53 häufig,
> TCP 53 ebenfalls möglich.

DHCP und UDP

DHCP nutzt UDP.

Wichtige Ports:

Richtung	Port
DHCP Server	UDP 67
DHCP Client	UDP 68

DHCP wird genutzt, um IP-Konfiguration automatisch zu vergeben.

Merksatz:

> DHCP nutzt UDP 67 und 68.

HTTP und HTTPS

HTTP nutzt typischerweise:

```
TCP 80
```

HTTPS nutzt typischerweise:

```
TCP 443
```

HTTPS ist HTTP über TLS-Verschlüsselung.

Wichtig:

```
HTTPS ist nicht einfach nur ein anderer Port,
sondern nutzt Verschlüsselung über TLS.
```

Merksatz:

```
HTTP = TCP 80.
HTTPS = TCP 443.
```

SSH und Telnet

SSH nutzt typischerweise:

```
TCP 22
```

Telnet nutzt typischerweise:

```
TCP 23
```

Wichtiger Unterschied:

```
SSH ist verschlüsselt.
Telnet ist unverschlüsselt.
```

In modernen Netzen sollte SSH verwendet werden, nicht Telnet.

Merksatz:

```
SSH sicherer als Telnet.
```

Transport und Anwendung

Schicht 4 transportiert Daten zwischen Anwendungen.

Die Anwendung selbst liegt höher.

Beispiel:

```
HTTPS nutzt TCP 443.
```

Dabei ist:

```
TCP = Schicht 4
HTTPS = Anwendungsschicht mit TLS-Bezug
```

Merksatz:

```
Port gehört zu Schicht 4.
Dienst gehört zu höheren Schichten.
```

TCP und Firewall

Firewalls nutzen häufig Schicht-4-Informationen.

Beispiel-Regel:

```
Erlaube TCP 443 von Client-Netz zum Internet.
```

Das bedeutet:

```
HTTPS-Verbindungen nach außen sind erlaubt.
```

Andere Ports können blockiert werden.

Merksatz:

```
Firewall-Regeln nutzen oft IP-Adressen und Ports.
```

UDP und Firewall

Auch UDP kann durch Firewalls erlaubt oder blockiert werden.

Beispiele:

```
DNS über UDP 53
DHCP über UDP 67/68
VoIP über UDP
VPN über UDP
NTP über UDP 123
```

Da UDP verbindungslos ist, muss eine Firewall den Verkehr anders verfolgen als bei TCP.

Merksatz:

```
UDP braucht passende Firewall-Regeln,
auch ohne TCP-Verbindung.
```

TCP-Verbindungszustände

TCP kennt verschiedene Verbindungszustände.

Beispiele:

```
LISTEN
SYN-SENT
ESTABLISHED
FIN-WAIT
TIME-WAIT
CLOSED
```

Für AP1/AP2 ist besonders wichtig:

LISTEN bedeutet:
Dienst wartet auf eingehende Verbindungen.

ESTABLISHED bedeutet:
Verbindung besteht.

Merksatz:

LISTEN = Dienst wartet.
ESTABLISHED = Verbindung steht.

Port offen oder geschlossen

Ein Port kann aus Sicht eines Clients unterschiedlich wirken.

Zustand	Bedeutung
offen	Dienst nimmt Verbindungen an
geschlossen	kein Dienst hört auf diesem Port
gefiltert	Firewall blockiert oder verwirft

Wichtig:

Ein offener Port bedeutet,
dass ein Dienst erreichbar ist.
Ein geschlossener Port bedeutet nicht zwingend,
dass das Gerät offline ist.

Merksatz:

Portstatus sagt etwas über den Dienst,
nicht nur über das Gerät.

Typische Schicht-4-Fehler

Typische Fehler auf Schicht 4 sind:

- falscher Port
- Dienst hört nicht auf erwartetem Port
- Firewall blockiert TCP oder UDP
- TCP-Verbindung wird nicht aufgebaut
- UDP-Antwort kommt nicht zurück
- Portweiterleitung auf falschen Port
- NAT/PAT-Portzuordnung falsch
- Anwendung nutzt anderes Protokoll als erwartet
- TCP-Handshake scheitert

Merksatz:

Schicht-4-Fehler betreffen TCP, UDP und Ports.

Schicht 3 oder Schicht 4 unterscheiden

Frage	eher Schicht
Ist die IP-Adresse erreichbar?	3
Funktioniert Ping auf IP?	3
Gibt es eine Route?	3
Ist TCP-Port 443 erreichbar?	4
Ist UDP-Port 53 erreichbar?	4
Baut TCP eine Verbindung auf?	4
Antwortet der Webdienst korrekt?	7

Merksatz:

IP erreichbar = Schicht 3.
Port erreichbar = Schicht 4.

Beispiel: Ping geht, Webseite geht nicht

Fehlerbild:

ping server funktioniert.
Webseite funktioniert nicht.

Dann ist Schicht 3 wahrscheinlich grundsätzlich erreichbar.

Mögliche Ursachen:

```
- TCP-Port 80 oder 443 blockiert
- Webserver läuft nicht
- Dienst hört auf anderer IP
- Dienst hört auf anderem Port
- Firewall blockiert
- Reverse Proxy falsch
- TLS-Problem
```

Merksatz:

```
Ping geht,
Dienst nicht:
Schicht 4 und 7 prüfen.
```

Beispiel: Port 443 blockiert

Ein Client möchte HTTPS nutzen.

Ziel:

```
server.example.local:443
```

Mögliche Fehler:

```
- Server ist erreichbar,
aber TCP 443 ist geschlossen
- Firewall blockiert TCP 443
- Dienst läuft nur auf TCP 8443
- Portweiterleitung zeigt auf falschen Port
```

Merksatz:

```
HTTPS braucht erreichbaren TCP-Port 443.
```

Beispiel: DNS funktioniert nicht

DNS nutzt häufig UDP 53 und teilweise TCP 53.

Wenn DNS nicht funktioniert, prüft man:

```
- DNS-Server-IP korrekt?
- DNS-Server erreichbar?
- UDP 53 erlaubt?
- TCP 53 erlaubt?
- Firewall blockiert?
- DNS-Dienst läuft?
- Antwort kommt zurück?
```

Merksatz:

> DNS-Probleme können Schicht 3, 4 oder 7 betreffen.

Beispiel: UDP-Dienst antwortet nicht

Bei UDP gibt es keinen TCP-Handshake.

Wenn ein UDP-Dienst nicht antwortet, kann das viele Ursachen haben:

```
- Dienst läuft nicht
- Firewall blockiert
- Antwortweg fehlt
- NAT/PAT-Problem
- Anwendung antwortet nur auf gültige Anfragen
- Paketverlust
- falscher Port
```

Merksatz:

> UDP-Fehler sind oft schwerer eindeutig zu erkennen als TCP-Fehler.

Einordnung in das OSI-Modell

Thema	Schicht

IP-Adresse	3
Routing	3
ICMP	3
TCP	4
UDP	4
Port	4
Socket	4 mit IP-Bezug
TCP-Handshake	4
DNS-Protokoll	7, nutzt TCP/UDP 53
HTTP	7, nutzt TCP 80
HTTPS	7, nutzt TCP 443 und TLS
DHCP	7, nutzt UDP 67/68

Merksatz:

> TCP, UDP und Ports = Schicht 4.

Was Schicht 4 nicht macht

Schicht 4 macht nicht:

- IP-Routing zwischen Netzen
- MAC-Adressen im LAN auflösen
- DNS-Namen in IP-Adressen auflösen
- Webseiten darstellen
- Benutzer authentifizieren
- Zertifikate prüfen
- Inhalte interpretieren
- Dateien speichern

Diese Aufgaben liegen auf anderen Schichten.

Merksatz:

> Schicht 4 transportiert,
> interpretiert aber keine Anwendung.

Typische IHK-Fragen

In AP1 und AP2 kann zum Beispiel gefragt werden:

- Welche Aufgabe hat die Transportschicht?
- Was ist ein Port?
- Was ist der Unterschied zwischen Switch-Port und TCP-Port?
- Was ist TCP?
- Was ist UDP?
- Was ist der Unterschied zwischen TCP und UDP?
- Was bedeutet verbindungsorientiert?
- Was bedeutet verbindungslos?
- Wie funktioniert der TCP-Handshake?
- Welche Ports haben HTTP, HTTPS, SSH, DNS und DHCP?
- Warum nutzt DNS TCP und UDP?
- Was bedeutet Socket?
- Was ist ein Ephemeral Port?
- Warum kann Ping funktionieren, aber HTTPS nicht?
- Warum nutzt ICMP keine Ports?

Typische Prüfungsfallen

TCP gehört zu Schicht 4.

UDP gehört zu Schicht 4.

Ports gehören zu Schicht 4.

ICMP nutzt keine TCP- oder UDP-Ports.

Switch-Port ist nicht TCP-Port.

TCP ist verbindungsorientiert.

UDP ist verbindungslos.

TCP ist zuverlässiger als UDP.

UDP ist schlanker als TCP.

TCP-Handshake = SYN, SYN-ACK, ACK.

HTTP nutzt TCP 80.

HTTPS nutzt TCP 443.

SSH nutzt TCP 22.

DNS nutzt UDP 53 und auch TCP 53.

DHCP nutzt UDP 67 und 68.

Ping funktioniert heißt nicht,
dass ein TCP-Port offen ist.

Port offen heißt nicht automatisch,
dass die Anwendung korrekt funktioniert.

Wichtige Begriffe kurz erklärt

Begriff	Kurze Erklärung
Transportschicht	OSI-Schicht 4
TCP	verbindungsorientiertes Transportprotokoll
UDP	verbindungsloses Transportprotokoll
Port	logische Dienstnummer
TCP-Port	Port für TCP-Dienst
UDP-Port	Port für UDP-Dienst
Socket	IP-Adresse plus Port
Segment	TCP-Datenform auf Schicht 4
Datagramm	UDP-Datenform auf Schicht 4
SYN	TCP-Verbindungsaufbau-Anfrage
ACK	Bestätigung bei TCP
Drei-Wege-Handshake	SYN, SYN-ACK, ACK
Sequenznummer	Position im TCP-Datenstrom
Well-Known Port	Port 0 bis 1023

Begriff	Kurze Erklärung
Registered Port	Port 1024 bis 49151
Ephemeral Port	temporärer Client-Port
LISTEN	Dienst wartet auf Verbindung
ESTABLISHED	TCP-Verbindung besteht

IHK-sichere Kurzformulierung

Die Transportschicht ist Schicht 4 des OSI-Modells. Sie ist für den Transport von Daten zwischen Anwendungen auf Endgeräten zuständig. Dafür werden vor allem TCP, UDP und Ports verwendet. TCP ist verbindungsorientiert und bietet zuverlässige Übertragung mit Verbindungsaufbau, Bestätigungen, Reihenfolge und erneuter Übertragung. UDP ist verbindungslos, schlanker und bietet keine eingebaute Garantie für Zustellung oder Reihenfolge. Ports dienen dazu, Dienste auf einem Gerät zu unterscheiden. Ein Socket besteht aus IP-Adresse und Port. Der TCP-Verbindungsaufbau erfolgt über den Drei-Wege-Handshake mit SYN, SYN-ACK und ACK.

Merksätze

Schicht 4 = Transportschicht.

Schicht 4 = TCP, UDP und Ports.

IP bringt zum Gerät.

Port bringt zur Anwendung.

Port = logische Dienstnummer.

Switch-Port ist physisch.

TCP-Port ist logisch.

TCP = verbindungsorientiert.

UDP = verbindungslos.

TCP = zuverlässig.

UDP = schlank und schnell.

TCP-Handshake = SYN, SYN-ACK, ACK.

TCP nutzt Segmente.

UDP nutzt Datagramme.

Portnummern gehen von 0 bis 65535.

Well-Known Ports = 0 bis 1023.

Ephemeral Ports = temporäre Client-Ports.

HTTP = TCP 80.

HTTPS = TCP 443.

SSH = TCP 22.

DNS = UDP 53 und TCP 53.

DHCP = UDP 67 und 68.

ICMP nutzt keine Ports.

Ping prüft nicht,
ob ein TCP-Port offen ist.

IP erreichbar = Schicht 3.

Port erreichbar = Schicht 4.

Anwendung korrekt = höhere Schicht.