

Dienste & Klassifizierung

Ein **Dienst** sendet/empfangt bestätigte und unbestätigte Daten. Klassifizierung:

	verbindungsorientiert	verbindungslos
	Verbindungs-Aufbau nötig (connect/close), Ziel muss bereit sein	jederzeit Nachrichten schicken, Ziel muss nicht «bereit» sein
	verbindungsorientiert	zuverlässig
Ethernet	O	O
IP	O	O
TCP	X	X
UDP	O	O

REZEPT Dienst aus Interface klassifizieren

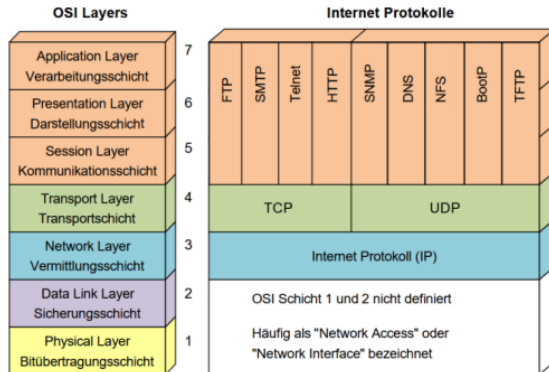
- Gibt es **connect_X(...)** / **close_X(...)**? → **verbindungsorientiert** (expliziter Auf-/Abbau).
- Nur **send/receive** mit Zieladresse pro Aufruf? → **verbindungslos** (jedes Telegramm eigenständig).

TCP vs. UDP

TCP (zuverlässig)	UDP (unzuverlässig)
kein Datenverlust; Sicherung durch Fehler-Erkennung/-Korrektur; Text-Nachrichten	möglicher Datenverlust; keine Sicherung; Streaming

OSI-Modell ↔ Internet-Protokolle

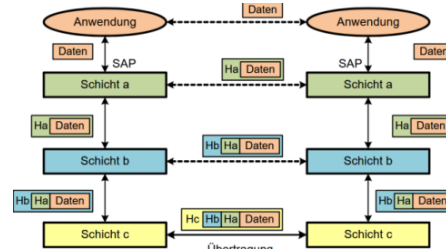
Beim Sender nach unten, beim Empfänger nach oben.



Aufgaben der Schichten 1-7

- Physical:** Rohdaten als elektrische Signale, Licht, Funk; Medium z. B. Kabel.
- Data Link:** gesicherte Strecke zwischen direkt verbundenen Knoten: Framing, Fehlererkennung/-korrektur, Flusskontrolle; MAC- & LLC-Teil; Adressierung (MAC), Media Access Control (CSMA/CD).
- Network:** Routing, IP-Adressierung, Weiterleitung. **Leitungsvermittelt:** feste Verbindung während ganzer Kommunikation (klass. Telefonie). **Paketvermittelt:** einzelne Pakete auf verschiedenen Wegen, beim Empfänger zusammengesetzt (Internet/IP).
- Transport:** Bereitstellung Übertragungsdienste.
- Session:** Verwaltung von Sitzungen, Steuerung von Verbindungen.
- Presentation:** Datenformatierung, Verschlüsselung, Komprimierung.
- Application:** benutzernahe Protokolle: HTTP, SMTP, E-Mail.

Kapselung (SAP, Header pro Schicht)



Signalgeschwindigkeit im Medium

$$c_{\text{Medium}} = 200'000 \text{ km/s} \approx \frac{2}{3} \cdot c_0 \quad (= 2 \cdot 10^8 \text{ m/s})$$

REZEPT Dämpfung in dB ↔ Faktor

$$\text{dB} = 10 \cdot \log_{10}(P_1/P_2) \quad \text{dB} = 10 \cdot \log_{10}(U_1/U_2)^2 \quad \log_a P = \log P / \log a$$

- Gesamtdämpfung: $Q = \text{Strecke} \cdot \text{Dämpfungsbelag}$ (z. B. 2.5 km · (-0.4 dB/km) = -1.0 dB).
- Faktor: $I_2 = I_1 \cdot 10^{(Q/10)} \rightarrow 10^{(-0.1)} \approx 0.8 \rightarrow \text{80 \% bleiben übrig.}$
- Umgekehrt (Faktor → dB): $Q = 10 \cdot \log_{10}(\text{Faktor})$.
Check: -3 dB ≈ halbe Leistung; -10 dB = Faktor 10.

SNR (Signal-Rausch-Verhältnis)

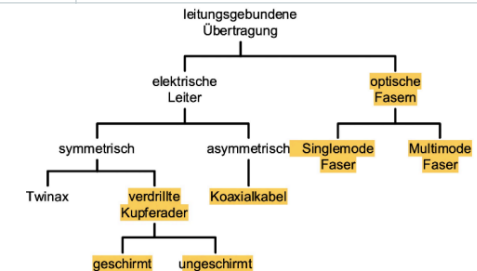
$$\text{SNR} = \frac{P_{\text{Signal}}}{P_{\text{Noise}}} = \frac{A^2_{\text{Signal}}}{A^2_{\text{Noise}}} \quad \text{SNR}_{\text{dB}} = 10 \cdot \log_{10}(\text{SNR}) \quad \text{SNR} = 10^{(\text{SNR}_{\text{dB}}/10)}$$

Störungen in Übertragungsmedien

- Kapazitive Störungen:** Überlagerung durch elektrische Felder benachbarter Leiter → **Shielding** als Lösung.
 - Induktive Störungen:** magnetische Wechselfelder verursachen Ströme in Nachbarleitungen → **Twisted Pair** als Lösung.
 - Übersprechen (Crosstalk):** Störungen durch Signale im selben Kabel (NEXT = Sender-, FEXT = Empfängerseite) → Kabel-/Steckerqualität (CAT 1-7).
 - Glasfaser:** Multimode günstig, aber begrenzte Reichweite & Dispersion.
- NEXT (dB) = 10 · log₁₀(Signalstärke/Störsignalstärke)** — je höher, desto geringer das Übersprechen

Kabel-Typen & Medien

Koaxialkabel	geeignet für hochfrequente Signale
Twinaxial	hoher Schutz
Twisted Pair	häufig im Einsatz (Shielded/Unshielded)
Glasfaser	hohe Bandbreite, geringe Dämpfung, resistent



Schirmung: xx/yTP

- xx** = Gesamtschirmung, **y** = Aderpaarschirmung.
- U** = ungeschirmt, **F** = Folienschirm, **S** = Geflechschirm, **SF** = Schirm aus Geflecht und Folie.
- Drahtgeflecht → niederfrequente Einstreuungen; metallisch beschichtete Folien → hochfrequente Störungen.

Glasfaser

Bestehen aus Glas, Prinzip **Totalreflexion**: Kernglas mit **hoher optischer Dichte**, Mantelglas mit **geringerer**.

- Multimode: Stufenfaser** = höherer Delay-Skew; **Gradientenfaser** = kürzerer. Klassen OM1-OM4 (OM1/2 LED-basiert, OM3/4 VCSEL).
- Monomode:** wesentlich grössere Bandbreite & Distanzen als Multimode.
- Vorteile:** unempfindlich gegen EM-Störungen, kleine Dämpfung → grosse Distanzen, grosse Bandbreiten/Übertragungsraten.

Kabel-Kategorien

CAT	Einsatz
1-4	Telefon- und Modem-Leitungen
5	weltweit akzeptierte Norm (100 MHz)
6	Gigabit-Ethernet am besten (250 MHz)
7	10-Gigabit-Ethernet geeignet (600 MHz)
8	für Datacenter

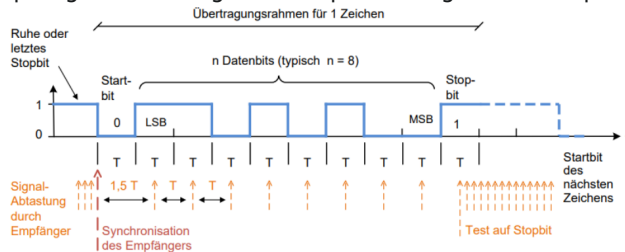
CAT5 nur 100 MHz vs. CAT7 600 MHz: nicht die Dämpfung limitiert, sondern das Übersprechen — bessere Kabel erlauben höhere Dämpfung.

Arten der Kommunikation

- **Simplex:** nur ein Kanal in eine Richtung.
- **Halbduplex:** ein Kanal, abwechselungsweise beide Richtungen.
- **Vollduplex:** je ein Kanal pro Richtung.
- **Punkt-Punkt:** direkte Verbindung zweier Partner (PC-Drucker). **Shared Medium:** mehrere Partner über dasselbe Medium (LAN).
- **Parallel:** mehrere Bits gleichzeitig über parallele Leitungen. **Seriell:** Bits zeitlich nacheinander über eine Leitung.

Serielle asynchrone Übertragung

→ asynchron, weil **kein Takt übertragen** wird; Sender/Empfänger haben eigene Taktquellen ≈ gleicher Frequenz.



- Empfänger justiert seinen Rahmen bei jedem Zeichen neu; Bitrate, Anzahl Datenbits + Stopbits müssen abgemacht sein.
- **Sender:** pro Zeichen 1 Startbit, vereinbarte Datenbits, dann Stopbit(s). **LSB zuerst!**
- **Empfänger:** tastet mit Vielfachem der Bitrate ab bis 0-Pegel des Startbits; Takte dürfen max. $\frac{1}{2}$ Bitzeit T abweichen; prüft Stopbit.
- **Taktrückgewinnung** möglich, solange regelmässig Zustandsänderungen auftreten.

REZEPT Serielle Schnittstelle dekodieren

Prüfungstyp

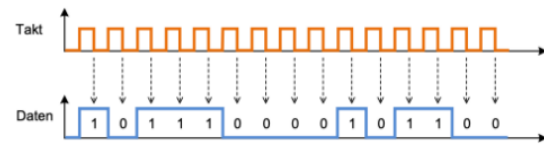
1. Ruhepegel = **1**; erstes **0** = Startbit.
2. Danach n Datenbits ablesen — sie kommen **LSB zuerst**: Reihenfolge umdrehen für den Wert.
3. Wert als Hex/Dezimal (Bit-Wertigkeiten 1,2,4,8,...).
4. Dann Paritybit, dann Stopbit (= 1).
5. **Parität bestimmen:** Einsen **inkl. Paritybit** zählen → gerade = Even, ungerade = Odd.

Bsp.: auf der Leitung 0 1 1 0 0 1 0 (zuerst gesendet = bit0, Wertigkeit 1·2·4·8·16·32·64) → 2+4+32 = 38 = **0x26**; MSB-Schreibweise: 0100110.

Serielle synchrone Übertragung

Empfänger arbeitet mit **gleichem Takt** wie Sender:

- keine Start-/Stopbits nötig → effizient, hohe Datenrate;
- Takt muss zusätzlich übertragen werden (Codierungsverfahren oder eigene Leitung).
 - Bitsynchrone Arbeitsweise erlaubt hohe Datenrate + ist effizient, da keine Start- und Stopbits benötigt werden
 - Nebst Datensignal muss auch dessen Takt übertragen werden



Begriffe Datenübertragungsrate

Baudrate	Symbole pro Sekunde
Zeichenrate	Zeichen pro Sekunde
Bandbreite B	Eigenschaft des Kanals, durch Medium begrenzt; abhängig von Strecke & Störstärke; Einheit Hz

REZEPT Nyquist: max. Symbolrate / Kompatibilität

$$f_s = 2 \cdot B \text{ [Baud]}$$

1. Gegebene Symboldauer? → Baudrate = $1/T_{\text{Sym}}$.
2. Vergleiche: Baudrate $\leq 2 \cdot B$ → kompatibel.
Bsp.: $T=66.67 \mu\text{s} \rightarrow 15 \text{ kBaud} < 2 \cdot 8 \text{ kHz} = 16 \text{ kBaud} \checkmark$

REZEPT Shannon: max. Kanalkapazität

$$C_S = B \cdot \log_2(1 + S/N) \cdot \log_2(x) = \log_2(x)/\log_2(2) \cdot S/N = 10^{\text{SNR}_{\text{dB}}/10}$$

1. SNR von dB in Faktor umrechnen (40 dB → 10'000).
2. Einsetzen: $C = 8000 \cdot \log_2(1+10'000) \approx 8000 \cdot 13.3 \approx \mathbf{106.3 \text{ kBit/s}}$.

REZEPT Bits pro Symbol & Anzahl Symbole

$$\text{Bits/Symbol} = \text{Bitrate} / \text{Baudrate} \quad \text{Anzahl Symbole} = 2^{\text{Bits/Symbol}}$$

1. **Maximal nutzbar** (Kapazität gegeben): **abrunden** → $106.3\text{k}/15\text{k} = 7.08 \rightarrow 7 \text{ Bit} \rightarrow 2^7 = 128 \text{ Symbole}$.
2. **Benötigt** (Bitrate gefordert, z. B. 320 Mb/s bei 80 MBaud): $\log_2(N) \geq 4 \rightarrow \mathbf{\text{aufrunden}}$ auf 2er-Potenz → $N \geq 2^4 = 16 \text{ Zustände}$.

Merke: max. erreichbar → abrunden; mindestens nötig → aufrunden.

Max. Bitrate & Signalzustände

$$R \leq 2 \cdot B \cdot \log_2(M) \quad M = 1 + A/\Delta V$$

M unterscheidbare Zustände bei Amplitude A = (M-1)·ΔV und Ungenauigkeit ΔV.



Übertragungsarten im LAN

Unicast	an einzelne Station
Broadcast	an alle Stationen
Multicast	an eine Gruppe von Stationen

Manchester-Code (Leitungscode)

- **1** = positive Flanke, **0** = negative Flanke.
- Erlaubt einfache Taktrückgewinnung.
- Benötigt **10 MHz Bandbreite** (doppeltes theoretisches Minimum).

Im LAN-Bereich gibt es drei Übertragungsarten

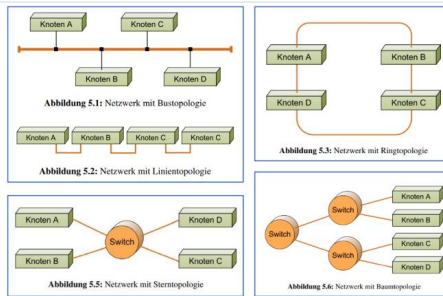
- Unicast an einzelne Stationen
- Broadcast an alle Stationen
- Multicast an eine Gruppe von Stationen

Als **Leitungscode** wird ein **Manchester-Code** eingesetzt.

- 1 positive Flanke, 0 negative Flanke
- Erlaubt die Taktrückgewinnung auf einfache Weise
- Bandbreite von 10 MHz benötigt (also das doppelte des theoretischen Minimums)



Topologien



Bus · Linie · Ring · Stern · Baum

REZEPT Kollisionserkennung & d_{max} Prüfungstyp

ohne Repeater: $t_{frame} > 2 \cdot t_{transfer}$ · mit: $t_{frame} > 2 \cdot (\sum t_{transfer} + \sum t_{forwarding})$

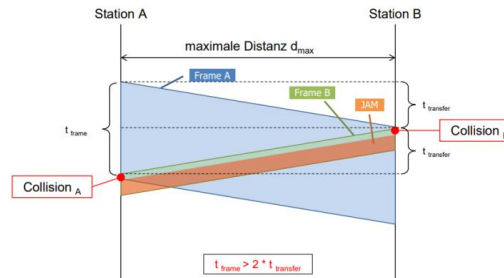
$$t_{frame} = \frac{\text{Framesize}_{min}}{\text{Bitrate}} \cdot t_{transfer} = \frac{d_{max}}{C_{Medium}} \cdot d_{max} < \frac{1}{2} \cdot \left(\frac{\text{Framesize}_{min}}{\text{Bitrate}} \right) \cdot C_{Medium}$$

1. Minimaler Frame in Bit (Ethernet: 576 bit inkl. Präambel).
2. t_{frame} rechnen, halbieren, mit $C_{Medium} = 2 \cdot 10^8$ m/s multiplizieren → d_{max}

Bsp.: $\frac{1}{2} \cdot 576 / (10 \cdot 10^6) \cdot 2 \cdot 10^8 = 5760$ m. Knoten erkennt Kollision nur, solange er selbst noch sendet!

Kollision im Weg/Zeit-Diagramm

Bedingung für Kollisionserkennung



Ethernet Frame-Format

Ethernet Format

Length/Type (2 Bytes)

- Fall 1: Länge von DATA ohne PAD (≤ 1500)
- Fall 2: Typ von Data = Protokoll der nächsten Schicht (≥ 1536)

Data / Padding (46 – 1500 Bytes)

- Enthält die eigentlichen Datenbytes
- Bei weniger als 46 Bytes wird mit PAD Bytes aufgefüllt

Frame Check Sequence, FCS (4 Bytes)

- IEEE CRC-32 Algorithmus

Interframe Gap, IFG (12 Bytes)

- «Zwangspause» zwischen aufeinanderfolgenden Frames
- Ist NICHT Teil des Ethernet Frames



- **Length/Type** (2 B): ≤ 1500 → Länge ohne Pad; ≥ 1536 (0x600) → Typ des nächsthöheren Protokolls.
- **Data/Pad** 46–1500 B: unter 46 B wird mit PAD aufgefüllt.
- **FCS** (4 B): IEEE CRC-32.
- **IFG** (12 B): Zwangspause zwischen Frames — **nicht** Teil des Frames.
- Präambel 7 B + SFD 1 B davor.

IEEE MAC-Adressen

IEEE MAC Adressen

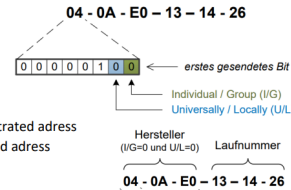
Die ersten beiden Bits des ersten Adress-Bytes haben eine spezielle Bedeutung:

Individual / Group Bit

- 0 = individual address
- 1 = group address

Universally / Locally Bit

- 0 = universally administrated address
- 1 = locally administrated address



- **I/G-Bit** (erstes gesendetes Bit): 0 = individual, 1 = group address.
- **U/L-Bit**: 0 = universally, 1 = locally administrated.
- Aufbau: 3 B Hersteller (OUI) + 3 B Laufnummer.

Repeater / Hub & Collision Domain

- Hub: ankommendes Signal wird an **alle** anderen Ports regeneriert ausgesendet.
- **Collision Domain** = Teilbereich, in dem Frames kollidieren können (mit Repeatern zusammengeschlossene Segmente).
- Kollisionserkennung: **Halbduplex** → Collision-Detection-Unit; **Vollduplex** → keine Kollisionen.

Normen für CSMA/CD

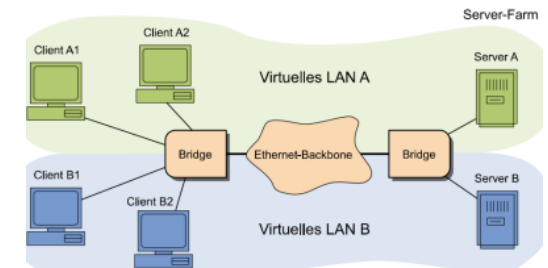
- Verbilligung (Thick → Thin Ethernet), Vereinfachung (Koaxial → Twisted Pair), Leistung 10 → 100 → 100'000 Mbit/s.

1000 BASE-T = Bitrate [Mbit/s] · BASE(Baseband)/BROAD(Broadband) · Art/Codierung (früher max. Segmentlänge)

VLANs

- Grosses Netz → unabhängige logische Netze; jeder Switch-Port einem beliebigen VLAN zuordnen.
- **VLAN-Tag**: VLAN-ID (Zuordnung), **Priority Code Point** (Priorisierung), **Discard Eligibility Indicator** (0 → wird bei Engpass zuerst verworfen).
- **Trunk = tagged, Access = untagged**. Ethernet-Backbone = Trunk.
- Ein virtuelles LAN bildet eine **Broadcast Domain** (Grenze für Broadcast-Frames).

Trunk = Tagged, Access = Untagged

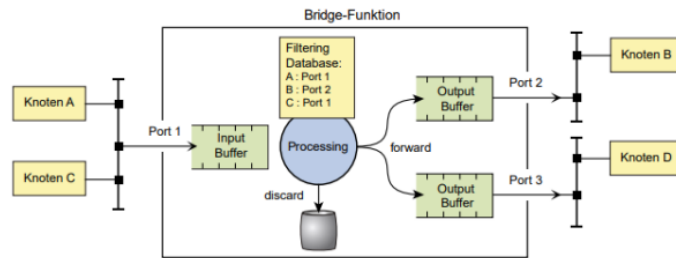


REZEPT Trunk-Ports markieren Prüfungstyp

1. Port mit **mehr als einem VLAN** in der Tabelle? → Trunk.
2. Auf Trunks **müssen** Frames getaggt übertragen werden (sonst VLAN nicht unterscheidbar).

Bridge / Switch

- Bridge hört Verkehr aller Ports ab, merkt sich Sender-Adressen in der **Filtering Database (FDB)**: pro MAC der Port, über den der Knoten erreichbar ist; unbenutzte Einträge altern heraus.
- Paket wird nur an die richtige Collision Domain geschickt; **Multi-Port-Bridge = Ethernet-Switch**, Standard-Kopplung von Segmenten.
- **Collision Domain**: mit Repeatern zusammengeschlossene Segmente — max. halb so lang wie der kürzeste Frame.
- **Broadcast Domain** endet am Router/VLAN.



REZEPT Address Learning / FDB-Tabelle Prüfungstyp

1. Frame trifft ein → Switch lernt **Quell-MAC → Eingangsport** (bereits bekannt: Eintrag wird aufdatiert).
2. Ziel-MAC **bekannt** → nur an diesen Port. **Unbekannt oder Broadcast FF-FF** → an alle Ports ausser Eingang fluten.
3. Empfangstabelle: Sender erhält sein Frame **nie** selbst; wer nie sendet, wird nie gelernt → Frames an ihn werden immer geflutet.

Store-and-Forward vs. Cut-Through

Store-and-Forward	(Adaptive) Cut-Through
Frame wird komplett empfangen, dann weitergeleitet → fehlerhafte Frames werden ausgefiltert	leitet schon nach Decodierung der Zieladresse weiter → leitet auch korrupte Frames; schaltet bei hoher Fehlerrate automatisch auf Store-and-Forward um

Weitere Switch-Merkmale: Anzahl Ports, Adresstabelle, Filter-/Transferrate, Backplane-Kapazität, (un)managed (Konsole/Web), VLAN-Fähigkeit, Energieverbrauch.

REZEPT Switch-Latenz (Weg/Zeit) Prüfungstyp

$$t_{\text{frame}} = \text{Bruttobits/Bitrate} \cdot t_{\text{transfer}} = d/C_{\text{Medium}}$$

$$t_{\text{total}} = 2 \cdot t_{\text{frame}} + 2 \cdot t_{\text{transfer}} + t_{\text{proc}}$$

1. Bruttoframe: Nutzdaten + 8 (Prä/SFD) + 12 (DA+SA) + 2 (Type) + 4 (FCS), alles · 8 in Bit.
2. t_{frame} zählt **zweimal** (Store-and-Forward: A→Switch und Switch→B), Laufzeit zweimal, plus Verarbeitungszeit.

Bsp.: $1000 \text{ B} \rightarrow 800 \mu\text{s}$; $50 \text{ m} \rightarrow 250 \text{ ns}$; total $2 \cdot 800 \mu\text{s} + 2 \cdot 250 \text{ ns} + 10 \mu\text{s} = 1610.5 \mu\text{s}$.

(Rapid) Spanning Tree Protocol

1. STP sperrt redundante Pfade bis auf einen; Root wird bestimmt, Baum aufgespannt, der alle Knoten genau einmal verbindet.
2. Austausch von **BPDUs** (spezielle Multicasts) — werden von Switches **nicht** weitergeleitet.
3. Jeder Knoten teilt Nachbarn die beste bekannte **Root-Bridge** + Kosten mit.
4. **Root-Bridge** = tiefster **Bridge Identifier** (wählbare Priorität + MAC).
5. Pfadkosten werden ab Root aufaddiert (pro Verbindung einstellbar).
6. BPDU alle **2 s**; bleiben sie aus → Netzwerk wird neu konfiguriert.

RSTP-Ablauf beim Einschalten: 1. jeder sieht sich selbst als Root (Kosten 0); 2. bessere Bridge-ID des Nachbarn → übernimmt dessen Root-ID; 3. bei redundanten Pfaden: Port mit tiefsten Pfadkosten zur Root = **Root-Port**; 4. **Designated Port** = Port der nächsten Bridge Richtung Root.

REZEPT Spanning Tree einzeichnen Prüfungstyp

1. Gleiche Pfadkosten → Kosten = **Hops** → **Breitensuche (BFS) ab Root**.
2. Ebene 1 = direkte Nachbarn der Root, Ebene 2 = deren neue Nachbarn, ...
3. Pro Knoten **genau eine Kante** Richtung Root (kürzester Weg).
4. Gleichstand → Kante über Knoten mit **tiefster Bridge-ID**.
5. Rest blockieren (v. a. Kanten innerhalb gleicher Ebene).

Check: Baum hat genau $N-1$ Kanten, keine Schleife, jeder Knoten 1 Weg zur Root.

Ethernet-Systeme

	10BASE-T	100BASE-TX	1000BASE-T	10GBASE-T
Kabel	CAT3-16 MHz / CAT5-100	CAT5-100 / CAT6-250	CAT5-100 / CAT6-250	CAT6A-500 / CAT7-600 / CAT7A-1000
Line Coding	Manchester, 2 Paare simplex	MLT-3, 4B5B, 2 Paare simplex	PAM-5, 8B/10B, 4 Paare duplex	PAM-16, 64B/65B, FEC, 4 Paare duplex
Baudrate	10 MBaud	125 MBaud	4 × 125 MBaud	4 × 800 MBaud
Link Pulses	NLP	FLP	FLP	FLP

Autonegotiation

- Ermittlung der besten Betriebsart durch Austausch der Leistungsmerkmale; danach höchstmögliches gemeinsames Profil (Speed/Duplex) aktiv; «Next Page» für optionale Features (EEE ...).
- **Link Pulses**: NLP = Link Presence Detection; FLP = Autonegotiation/Autopolarity. Gerät, das nur NLP sendet → als 10Base-T half-duplex identifiziert (Abwärtskompatibilität).

REZEPT Clock Drift (Eigentaktabweichung)

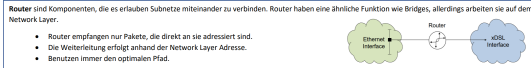
Jeder Port hat eigene Clock (Temperatur, Alterung, Toleranz). Sample-Zeitpunkt darf um max. $\frac{1}{2}$ Bitzeit verschoben sein:

$$N \cdot \delta_{\text{rel}} \leq \frac{1}{2} \Rightarrow \delta_{\text{rel,max}} = 1/(2 \cdot N)$$

1. N = Bits pro Rahmen (Datenbits + Stoppbit(s) + ...).
2. Bsp.: 8 Datenbits + 1 Stoppbit → $N = 9 \rightarrow \delta = 1/18 \approx 5.56 \%$.

Router

- Verbinden Subnetze; arbeiten auf dem **Network Layer** (ähnlich Bridges, aber Schicht 3).
- Empfangen nur Pakete, die direkt an sie adressiert sind; Weiterleitung anhand Network-Layer-Adresse; immer optimaler Pfad.



IP-Header (min. 20 Byte)

Version	IHL	Type of Service	Total Length
Identification Number		Flags	Fragment Offset
Time to Live	Protocol	IP Header Checksum	
IP Source Address			
IP Destination Address			
Options / Padding			

- **IHL:** Headerlänge in 4-Byte-Schritten (20 B → IHL = 5).
- **ToS:** Priorisierung.
- **Total Length:** Header + Nutzdaten. **ID:** identifiziert Paket/Fragmente.
- **Flags:** DF = Don't Fragment (Pos. 1), MF = More Fragments (Pos. 2), Bit 0 reserviert = 0. **FO:** Position des Fragments (-8 Byte).
- **TTL:** Hop-Counter, 0 → Paket verworfen. **Protocol:** übergeordnetes Protokoll (6 = TCP, 17 = UDP).

Das unterliegende Netz limitiert die Paketgröße (**MTU**, Maximum Transfer Unit) — der Sender kennt die MTU fremder Netze nicht.

REZEPT Fragmentierung Prüfungstyp

1. **Wer fragmentiert?** Der Router vor dem Netz mit kleinerer MTU.
2. Max. Nutzdaten/Fragment = $MTU - 20$; auf **Vielfaches von 8** abrunden (alle Fragmente ausser dem letzten gleich & grösstmöglich).
3. **FO** = bisherige Daten-Bytes / 8 (kumulativ); **TL** = Daten + 20; **MF** = 1, beim letzten 0.

Bsp. MTU 580, 1480 B Daten: 3 Fragmente à 560/560/360 B → FO 0/70/140, TL 580/580/380, MF 1/1/0. Check: Σ Daten = Original.

REZEPT Reassembly / Fragment erkennen

Originalgrösse = $FO \cdot 8 + (TL - 20) + 20$

1. MF = 0 → letztes Fragment; $FO \cdot 8$ = Bytes davor; $TL - 20$ = Daten dieses Fragments.
2. Bsp.: FO=500, TL=317 → $4000 + 297 + 20 = 4317$ Byte.

3. **Fragment erkennen:** MF = 1 oder FO > 0 (MF=1 bei FO=0 → erstes Stück eines grösseren Pakets).

Zusammensetzen erfolgt **beim Zielhost**.

IPv4-Adressierung

Netzadresse	tiefste Adresse im Subnetz = Interface-Adresse AND Subnetzmaske
Broadcast	höchste Adresse = Interface-Adresse OR invertierte Subnetzmaske
Subnetzmaske	Bit 1 → Network-, Bit 0 → Interface-Address; /nn = Länge der 1er

```

Maske /20 255.255.240.0 =
11111111.11111111.11110000.00000000
Subnetz 160.85.16.0 = IP AND Maske
Broadcast 160.85.31.255 = IP OR ~Maske

```

Binärwerte der Maske

Wert	binär	Wert	binär
0	00000000	248	11111000
128	10000000	252	11111100
192	11000000	254	11111110
224	11100000	255	11111111
240	11110000		

Adressklassen & private Bereiche

Klasse	Bereich	Netze / Interfaces
A	1.0.0.0 - 127.255.255.255	127 / 16'777'214
B	128.0.0.0 - 191.255.255.255	16'384 / 65'534
C	192.0.0.0 - 223.255.255.255	2'097'152 / 254
D	224.0.0.0 - 239.255.255.255	Multicast
E	240.0.0.0 - 255.255.255.255	reserviert

privat (nicht geroutet)	Anzahl Netze	Maske
10.0.0.0	1	255.0.0.0
172.16.0.0 - 172.31.0.0	16	255.255.0.0
192.168.0.0 - 192.168.255.0	256	255.255.255.0

Classless Routing: flexible Zuteilung von Adressräumen in Zweierpotenz-Schritten; Subnetmask legt Grenze Network/Host frei fest.

REZEPT Subnetting Prüfungstyp

Eckdaten von /n:

1. Hostbits $h = 32 - n$; Blockgrösse = 2^h Adressen.
2. Netzadresse = Blockanfang (Vielfaches der Blockgrösse); Broadcast = Netz + Block - 1.
3. Nutzbar: dazwischen; **Anzahl Hosts** = $2^h - 2$.

Netz aufteilen:

1. Pro Subnetz kleinstes h mit $2^h - 2 \geq$ geforderte Hosts → Maske = $/(32 - h)$.
2. **Grösstes Subnetz zuerst**, Blöcke lückenlos aneinanderreihen.

Bsp. 172.30.10.0/25 → 50 Hosts: /26 (.0-.63, 62), 2×25 Hosts: /27 (.64-.95, .96-.127, je 30). Σ Blöcke = 128 ✓

Routing & Forwarding

- **Routing:** optimalen Weg im Netz bestimmen (Aufbau der Tabellen) — statisch konfiguriert oder dynamisch via **Routing-Protokolle** (Router kennen Topologie, tauschen sich aus).
- **Forwarding:** Weiterleiten der Pakete entlang der Routen (anhand der Tabellen).

Flaches Routing

Router kennt explizite Wege zu jedem Zielnetz; Redundanz möglich; grosse Tabellen; eher in zentralen Teilen

Hierarchisches Routing (Default)

Router kennt nur direkt angeschlossene Netze + Default-Eintrag; kleine Tabellen; am «Rand» von Netzen; Host schickt Fremdes an Default-Router (Gateway)

Routing-Tabelle: Regeln

- Sortiert nach **Länge der Netzmaske** (längste zuerst), von oben nach unten durchsucht, verglichen werden die Netzadressen → **Longest Prefix Match**.

Netzadresse	Netzmaske	Port	Gateway
160.85.18.0	255.255.255.240	eth1	(direkt)
160.85.19.0	255.255.255.0	eth2	(direkt)
160.85.16.0	255.255.254.0	eth0	(direkt)
default	0.0.0.0	eth0	160.85.16.1

REZEPT Routing-Tabellen aufstellen Prüfungstyp

Router:

1. Pro direkt angeschlossenes Netz: Netzadresse, /Maske, Port, Gateway = «direkt».
2. Reihenfolge: **längste Maske zuerst** (spezifisch vor allgemein).
3. Letzte Zeile: **default** → Port Richtung Internet + Gateway-IP.

Host (minimal):

1. Eigenes Subnetz direkt via eth0; 2. default → Gateway = Router-Port im eigenen Subnetz.
- «Ports belegen die tiefste Adresse» → Router-Port = Netzadresse + 1 (z. B. 192.168.1.129 bei /26).

ARP — Address Resolution Protocol

Ermittelt die **HW-Adresse (MAC)** zu einer IP-Adresse. Ablauf: ARP-Request «who-has IP» als **Broadcast**; Besitzer antwortet «is-at MAC» als **Unicast**; Resultate landen im ARP-Cache.

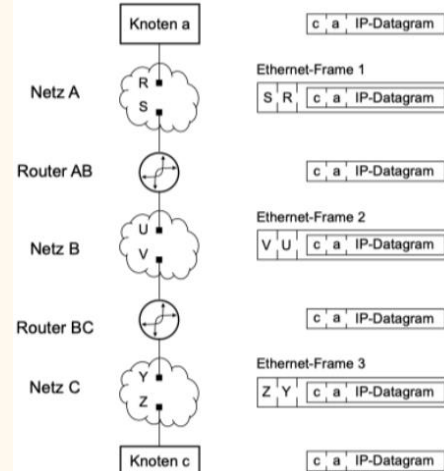
REZEPT ARP-Nachrichtentabelle Prüfungstyp

1. Vorab: Ziel im selben Subnetz (IP AND Maske)? Sonst fragt ARP nach der **Gateway-IP**!
2. **ARP-Request:** MAC-src = Sender, MAC-dst = **FF-FF**, IP-Felder leer «-», Inhalt «Who has [Ziel-IP]?»
3. **ARP-Reply:** MAC-src = Ziel, MAC-dst = Sender (Unicast), «[Ziel-IP] is at [Ziel-MAC]».
4. **Ping-Request:** ICMP Echo (Typ 8); MAC + IP normal Sender → Ziel.

REZEPT IP über Router (Encapsulation) Prüfungstyp

1. **IP-Adressen bleiben Ende-zu-Ende gleich** (Original-Quelle → End-Ziel) — nie ändern!
2. **MAC pro Netz neu:** src = sendendes Interface in diesem Netz; dst = nächster Hop (Router-Port dieses Netzes bzw. Ziel-Host).

Check: beide MACs einer Zeile liegen im selben Netz; Router-Interface des richtigen Netzes wählen (eth0 vs. eth1!).



ICMP — Internet Control Message Protocol

- Übertragung von Fehlermeldungen / Informationsaustausch; **in IP-Datagramm gekapselt** (Protocol = 1).
- Melden normalerweise Fehler bei Verarbeitung/ Weiterleitung von IP-Datagrammen.
- Keine ICMP-Meldungen über ICMP-Meldungen (kein Ping-Pong); bei Fragmenten nur Meldung über das **erste** Fragment.

Meldung	Bedeutung
Destination Unreachable	Datagram kann nicht ans Endziel befördert werden
Source Quench	Router-Buffer voll, musste Datagramme verwerfen (an Quell-Hosts)
Redirect	Host nutzt falschen Router → Route ändern
Time Exceeded	TTL = 0 oder Timer läuft ab, bevor alle Fragmente da sind
Parameter Problem	ungültiger Wert im Header gefunden
Echo Request/ Reply	an jeden Host möglich; Reply muss zurück (Ping)
Timestamp Req/ Reply	wie Echo, mit Zeitangabe
Information Req/ Reply	Informationsaustausch

Grundsätze des Internets

- Jedes Netzwerk soll für sich selbst funktionsfähig sein.
- Kommunikation basiert auf «**best effort**».
- Verbindung der Netze durch Black Boxes; keine zentrale Funktionssteuerung.

Kapselung & Demultiplexen

Applikationsdaten + UDP/TCP-Header = **User Datagramm (UDP)** / **Segment (TCP)**; danach IP-Header (Protocol-Feld: **17 = UDP, 6 = TCP**).

- Ethernet-Frame kommt an; Type-Feld 0x800 → IP-Modul.
- IP-Modul → Protocol-Feld → TCP-/UDP-Modul.
- Destination Port im TCP/UDP-Header → richtige Applikation.

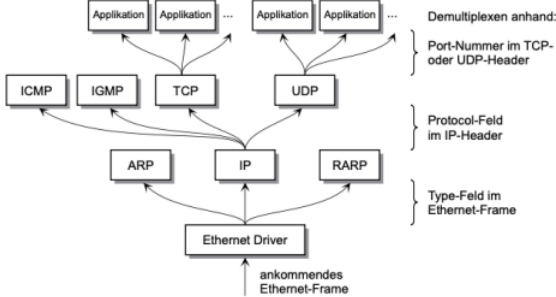


Figure 33: Demultiplexen: Bsp.: Ethernet-Schnittstelle

UDP & Ports

UDP dient dem Multi-/Demultiplexen der Datagramme zu den Applikationen — **verbindungslos, unzuverlässig**.

UDP Source Port	UDP Destination Port	
UDP Message Length	Checksum	
Data		
System (Well-Known)	User (Registered)	Dynamic / Private
0 – 1023, standardisiert reserviert	1024 – 49'151, für Hersteller	49'152 – 65'535, beliebig

Destination Port wird gesetzt; **Source Port meist zufällig durch OS**.

REZEPT Client-Server-Pakete (4-Tupel) Prüfungstyp

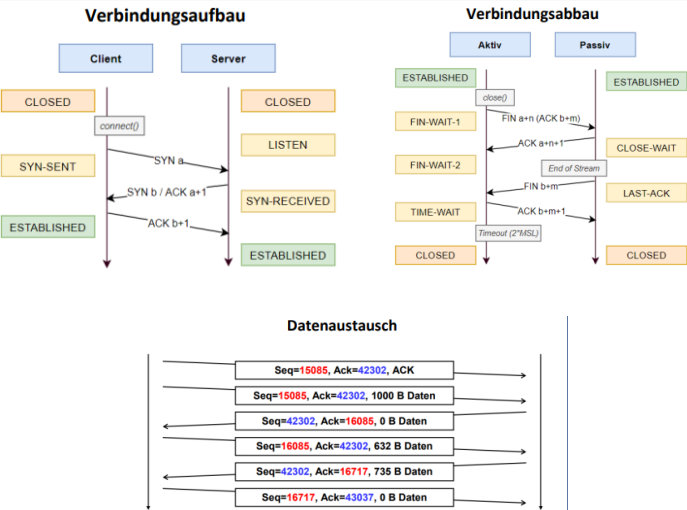
- Verbindung eindeutig durch (Src-IP, Src-Port, Dst-IP, Dst-Port).
- Antwortpaket = gespiegelt**: Server→Client hat Src-Port = Dienstport (443), Dst-Port = Client-Port.
- Zweite Verbindung des Clients → **anderer** lokaler Port (z. B. 40138).

TCP-Header

TCP Source Port			TCP Destination Port		
Sequence Number					
Acknowledgement Number					
Hdr-Len	unused	ECN	Control Bits		Window
Checksum			Urgent Pointer		
Options / Padding					

- Sequence-Nr.:** Ordnung der Segmente. **Ack-Nr.:** n+1 → Daten korrekt & vollständig.
- Control Bits:** URG, ACK, PSH, RST, **SYN** (Aufbau), **FIN** (Abbau).
- Window:** verfügbare Puffergrösse. **Options:** häufigste = MSS.

TCP-Verbindungsphasen



LISTEN	auf Anforderung warten	FIN-WAIT - 1	Abbauanforderung geschickt
SYN-SENT	Anforderung geschickt	FIN-WAIT - 2	Abbauanforderung bestätigt
SYN-RECEIVED	Anforderung erhalten	CLOSE-WAIT	auf lokale Verbindung warten
ESTABLISHED	Verbindung besteht	LAST-ACK / TIME-WAIT	Abbau bestätigt / letzte Bestätigung gesendet

REZEPT TCP-Phase aus Wireshark erkennen Prüfungstyp

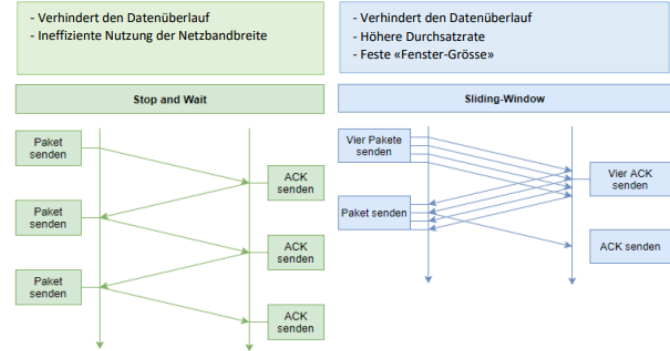
- SYN**-Flag → Verbindungsaufbau; **FIN** → Abbau; nur ACK/Daten → Datentransfer.
- TIME-WAIT:** Nach «Active Close» wartet der schliessende Host (Timeout 2·MSL, < 4 min), um bei verlorenem letztem ACK das wiederholte FIN regulär beantworten zu können. Beobachtung TIME-WAIT = kurz davor wurde hier eine Verbindung aktiv geschlossen.

REZEPT Seq-/Ack-Nummern fortschreiben Prüfungstyp

- Neue Seq** (gleicher Sender) = letzte eigene Seq + damals gesendete Datenbytes (0 Daten → Seq bleibt).
- Ack** = Seq der Gegenseite + deren Datenlänge = **nächstes erwartetes Byte**.
- Zeile für Zeile; gegebene Zellen als Kontrolle.

Fluss-Steuerung (Überlast des Empfängers)

TCP verwendet den **Sliding-Window**-Mechanismus; beide Seiten einen Buffer (Window).



Stop-and-Wait: verhindert Datenüberlauf, nutzt Bandbreite ineffizient · Sliding-Window: höhere Durchsatzrate, feste «Fenster-Grösse».

Congestion Control (Überlast des Netzes)

- TCP nutzt **Paketverlust** als Mass für Überlastung → Absenken der Rate (**Slow Start**).
- Jeder Sender pflegt **zwei Fenster** (vom Empfänger gewährtes + Überlastungsfenster); das **Minimum** ist sendbar.

REZEPT RTT / Retransmission Timeout

$$SRTT_n = (1-\alpha) \cdot SRTT_{n-1} + \alpha \cdot RTT_n \quad (\alpha=0.125)$$

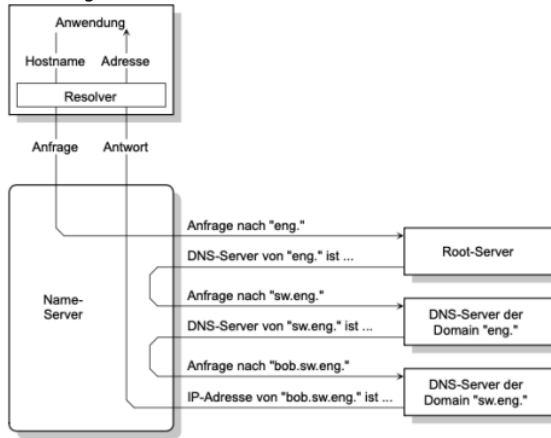
$$RTTVAR_n = (1-\beta) \cdot RTTVAR_{n-1} + \beta \cdot |SRTT_n - RTT_n| \quad (\beta=0.25)$$

$$RTO_n = SRTT_n + 4 \cdot RTTVAR_n$$

- RTT messen → SRTT glätten → Streuung RTTVAR → RTO.
- Keine Bestätigung bis RTO → Paket erneut übertragen.

DNS — Domain Name System

- Übersetzt Hostnamen ↔ IP-Adressen; hierarchischer **DNS Name Space** auf vielen Name-Servern verteilt.
- **Root-Domain** = Wurzel; **TLD** (oberste Ebene, direkt unter Wurzel), **Second/Third Level Domain**; Knoten-Namen bis 63 Zeichen; jeder Host eindeutig (**FQDN**); **Alias** = Zeiger von Domäne auf andere.
- **Verwaltung**: Name Space in **Zonen**; je Zone ein **Master**-Name-Server; **Slave**-Server erhalten Zonendaten via Transfer → **Redundanz**. Delegation an Sub-Domänen-Server.
- **Resolver** (Client-seitig) erzeugt DNS-Anfrage, sendet an Name-Server, verarbeitet Antwort; Resolver-Anfragen werden von DNS-Servern ausgeführt (Rekursion).



REZEPT Wer liefert die DNS-Antwort? Prüfungstyp

1. Rechner frisch gestartet → **lokaler Cache leer**.
2. Erste Anfrage, **populäre** Domain → **konfigurierter Name Server** (hat sie im Cache).
3. **Wiederholte** Anfrage (Browser egal!) → **Cache des lokalen Rechners/Resolvers**.
4. **Unbekannte** Domain → volle Rekursion Root → TLD → SLD; Antwort vom **Name Server der Second Level Domain**.

REZEPT dig-Ausgabe deuten Prüfungstyp

1. Record-Typ: **A** = IPv4, **AAAA** = IPv6, **CNAME** = Alias (kanonischer Name), Zahl = TTL.
2. CNAME-Zeile: angefragter Name ist Alias → rechts der echte Name.
3. A/AAAA folgt → gesuchte Adresse; **fehlt** der Record → keine Adresse dieses Typs bekannt (z. B. kein IPv6).

DHCP — Dynamic Host Configuration Protocol

- Verwendet BootP-Protokoll (Paketformat identisch); dynamische Zuweisung von IP-Adressen; reserviert nur IPs aktiver Geräte; Admin reserviert Adressbereich im Subnetz auf dem Server; Vergabe mit **Lease-Time** (danach Erneuerung durch Client).

Ablauf: 1. Client sucht Server per **Broadcast** → 2. Server antwortet (**DHCP offer**) → 3. Client wählt & fordert Parameter (**DHCP request**) → 4. Server bestätigt mit endgültigen Parametern → 5. vor Lease-Ablauf erneuert der Client.

NAT — Network Address Translation

- Versteckt privates IP-Netz hinter **einer** globalen Adresse (Bypass für ausgeschöpften IPv4-Raum).
- **Port Mapping**: Kommunikation muss vom lokalen Netz aus initiiert werden → von aussen kein Zugriff auf lokale Knoten.
- Dienste nach aussen: Portnummer **statisch** auf (lokale) IP des Servers konfigurieren; einkommende Pakete mit diesem Port → an den Server. **Pro Dienst/Port nur ein lokaler Server!**
- **Problem**: verletzt OSI-Layer (Network-Funktion verändert Transport-Header).

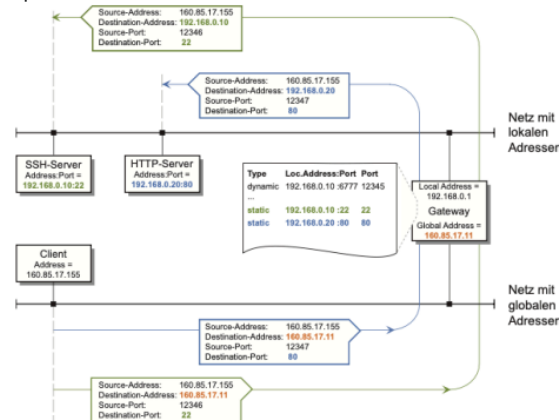


Figure 45: Port Mapping

9.3.2 Probleme mit NAT

- Verletzt eigentlich das OSI-Layer (Network-Funktion greift auf den Transport-Header zu und verändert diesen)

REZEPT NAT-Paket umschreiben Prüfungstyp

1. **Source-IP** → öffentliche Gateway-IP.
2. **Source-Port** → globaler Port aus NAT-Tabelle (Zeile mit lokaler IP:Port suchen).
3. **Destination IP + Port: unverändert**.
4. **Statischer Eintrag** (Server erreichbar machen): Static | lokale IP : Dienstport | globaler Port (z. B. TFTP: 10.1.1.44:69 → 69).

TFTP · SMTP · HTTP

TFTP	Trivial File Transfer: einfaches, zuverlässiges Protokoll neben FTP; z. B. für diskless Systeme (Betriebssystem-Image vom Server); UDP Port 69
SMTP	Versand von E-Mails (POP/IMAP = Empfang)
HTTP	Zugriff auf verteilte Dokumente, adressiert via URL

Physical / Medien	
Medium	$C = 200'000 \text{ km/s} \approx \frac{2}{3} \cdot c_0 = 2 \cdot 10^8 \text{ m/s}$
Dämpfung	$\text{dB} = 10 \cdot \log_{10}(P_1/P_2) = 10 \cdot \log_{10}(U_1/U_2)^2$; Faktor = $10^{(\text{dB}/10)}$
SNR	$\text{SNR} = P_S/P_N = A^2_S/A^2_N$; $\text{SNR}_{\text{dB}} = 10 \cdot \log \text{SNR}$; $\text{SNR} = 10^{(\text{dB}/10)}$
log-Wechsel	$\log_a P = \log P / \log a$; $\text{ld}(x) = \log x / \log 2$
Nyquist	$f_s = 2 \cdot B [\text{Baud}]$; Baudrate = $1/T_{\text{Sym}}$
Shannon	$C_S = B \cdot \text{ld}(1 + S/N)$
Bitrate	$R \leq 2 \cdot B \cdot \log_2(M)$; $M = 1 + A/\Delta V$
Bits/Symbol	= Bitrate/Baudrate → max: abrunden · nötig; aufrunden; Symbole = 2^{Bits}
NEXT	$10 \cdot \log_{10}(\text{Signal}/\text{Störsignal})$ — höher = besser
Clock Drift	$\delta_{\text{max}} = 1/(2 \cdot N)$, $N = \text{Bits pro Rahmen}$

Data Link / Ethernet	
Framerate	$F_R = \text{Bitrate} / (8 \cdot (F_L + \text{IFG}))$; Bytes +8 Prä/SFD +12 IFG
Nutzbitrate	$N = F_R \cdot P \cdot 8$; Netto = Brutto·Nutz/(Nutz+Header)
opt. Framelänge	$\approx \sqrt{(H/p_e)}$
Frame-Fehler	$P_{\text{Erfolg}} = (1-p_e)^N$; $P_{\text{Fehler}} \approx N \cdot p_e$ (FER)
Kollision	$t_{\text{frame}} > 2 \cdot t_{\text{transfer}} (+ \Sigma t_{\text{forwarding mit Repeater}})$
d_max	$< \frac{1}{2} \cdot (\text{Framesize}_{\text{min}}/\text{Bitrate}) \cdot C_{\text{Medium}}$; Ethernet min = 576 bit
Switch-Latenz	$t_{\text{total}} = 2 \cdot t_{\text{frame}} + 2 \cdot t_{\text{transfer}} + t_{\text{proc}}$ (Store-and-Forward)
Frame-Overhead	Prä+SFD 8 · DA+SA 12 · Type 2 · FCS 4 · IFG 12 · Pad bis 46 · Data ≤ 1500

Network / Transport	
Subnetz /n	$h = 32-n$; Block = 2^h ; Hosts = $2^h - 2$; Broadcast = Netz+Block-1
Netz/Broadcast	Netz = IP AND Maske; Broadcast = IP OR ~Maske
Fragment	Daten/Frag = (MTU-20) auf 8 gerundet; FO = Bytes/8; TL = Daten+20; MF=1 ausser letztes
Reassembly	Grösse = FO·8 + (TL-20) + 20
Seq/Ack	Seq_neu = Seq + Datenbytes; Ack = Seq_Gegenseite + deren Daten
RTO	$\text{SRTT} = (1-\alpha)\text{SRTT} + \alpha\text{RTT}$ ($\alpha=.125$); $\text{RTTVAR} = (1-\beta)\text{RTTVAR} + \beta \text{SRTT}-\text{RTT} $ ($\beta=.25$); $\text{RTO} = \text{SRTT} + 4 \cdot \text{RTTVAR}$
Ports	System 0-1023 · User 1024-49151 · Dynamic 49152-65535
privat	10/8 · 172.16-31/12 · 192.168/16
Protocol-Feld	1 = ICMP · 6 = TCP · 17 = UDP; Ethernet-Type 0x800 = IP

Methoden-Spickzettel (1 Zeile pro Prüfungstyp)	
Parität	Einsen je Zeile/Spalte (inkl. Paritybit); Mehrheit = Typ; abweichende Zeile x Spalte = Fehlerbit.
Seriell dekodieren	Idle=1, Start=0, Datenbits LSB zuerst → umdrehen, Hex; Einsen inkl. Parity zählen → Even/Odd.
Framerate	$(\text{Bytes}+8+12) \cdot 8 \rightarrow \text{Bitrate/Bruttobit}$.
Nyquist/Shannon	$f_s=2B$ prüfen; $C=B \cdot \text{ld}(1+10^{(\text{SNR}/10)})$; Bits=[C/Baud], Symbole= 2^{Bits} .
FDB/Learning	Quell-MAC→Port lernen; Ziel bekannt→Port, sonst/Broadcast→fluten; Sender kriegt nichts.
Spanning Tree	BFS ab Root; je Knoten 1 Kante (kürzester Weg); Tiebreak tiefste Bridge-ID; N-1 Kanten.
VLAN	Port mit >1 VLAN = Trunk = tagged.
Switch-Zeit	$t = 2 \cdot t_{\text{frame}} + 2 \cdot t_{\text{transfer}} + t_{\text{proc}}$.
ARP	anderes Subnetz→Gateway fragen; Request Broadcast «who has», Reply Unicast «is at», dann Ping.
Encapsulation	IP Ende-zu-Ende konstant; MAC pro Netz: sendendes Interface → nächster Hop.
Subnetting	kleinstes h mit $2^h-2 \geq \text{Hosts} \rightarrow /(32-h)$; grösstes Subnetz zuerst, lückenlos.
Routing	direkte Netze (längste Maske zuerst, «direkt»), default → Gateway zuletzt; Host: eigenes Netz + default.
Fragmente	Router fragmentiert; (MTU-20)→Vielfaches v. 8; FO kumulativ/8; MF=1 ausser letztes.
TCP-Phase	SYN=Aufbau, FIN=Abbau; TIME-WAIT = Wartezeit nach Active Close (2·MSL).
Seq/Ack	Seq=eigene Daten; Ack = nächstes erwartetes Byte der Gegenseite.
4-Tupel	Antwort = gespiegelt; Client-Ports ephemere, je Verbindung neu.
NAT	Src-IP→öffentlich, Src-Port→global (Tabelle); Ziel bleibt; Server: Static lokal:Port→global.
DNS-Instanz	Neustart=Cache leer; populär→konfig. NS; Wiederholung→lokaler Cache; unbekannt→SLD-NS.
dig	A=IPv4, AAAA=IPv6, CNAME=Alias; Record fehlt = Adresse unbekannt.
Dienst-Typ	connect/close vorhanden = verbindungsorientiert, sonst verbindungslos.