

# Kommunikationstechnik - S4

Raphael Nambiar

Version: 12. Juni 2023

## OSI-Modell

### Dienst

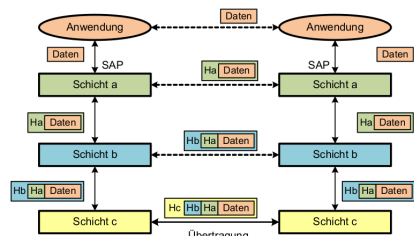
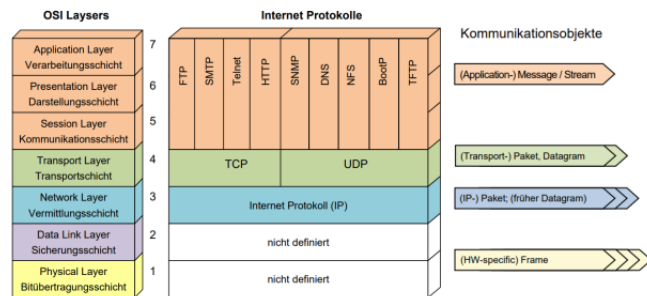
Klassifizierung von Diensten:

|   |   |
|---|---|
| Verbindungsorientiert                             | verbindungslos  |
| Verbindungs-Aufbau nötig<br>Ziel muss bereit sein | Jederzeit Nachrichten schicken<br>Ziel muss nicht «bereit» sein |

| Zuverlässig  | Unzuverlässig                             |
|--|---|
| Kein Datenverlust<br>Sicherung durch<br>Fehler-Erkennung<br>-/ Korrektur | Möglicher Datenverlust<br>Keine Sicherung |
| Text-Nachrichten, Backup<br>Dateidienste                                 | Streaming<br>Voip                         |

## Schicht

Eine Schicht hat die Aufgabe der darüberliegenden Schicht bestimmte Dienste zur Verfügung zu stellen. Die Schichten benötigen kein Wissen über die Realisierung der darunterliegenden Schicht.



## Protokoll

Ein Protokoll ist eine Sammlung von Nachrichten, Nachrichtenformaten und Regeln zu deren Austausch.

## Übertragungsmedien

### Ausbreitungsgeschwindigkeit

Lichtgeschwindigkeit im Vakuum:

$$c_0 = 299'792'458 m/s$$

Ausbreitungsgeschwindigkeit in Medien:

$$c_{Medium} = 200'000 km/s = \frac{2}{3} c_0$$

### Beispiel:

Licht im Glas, Brechungsindex  $n = 1.5$

$$c_{Glas} = \frac{c_0}{n} = 200'000 km/s$$

## Signaldämpfung

Signaldämpfung bezeichnet die Leistungsabnahme eines Signals.

- Je grösser die Bandbreite (Hz), desto höhere Datenraten (bit/s) übertragen
- Je kleiner die Dämpfung ist, desto grössere Distanzen können erreicht werden
- Senkt man die Bitrate (bei gleicher Dämpfung), können grössere Distanzen erreicht werden

$$dB = 10 \cdot \log\left(\frac{P_1}{P_2}\right)$$

$$dB = 10 \cdot \log\left(\frac{U_1}{U_2}\right)^2$$

## Signal-Rausch-Verhältnis (SNR)

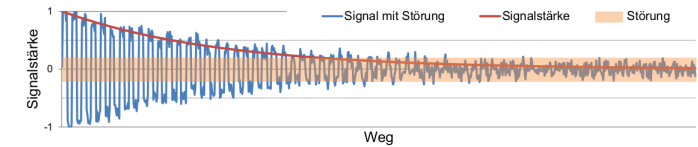
Das SNR ist ein Mass für die Qualität eines Signals. Es gibt an, wie stark das Signal im Vergleich zum Rauschen ist.

$$SNR = 10 \cdot \log\left(\frac{P_{Signal}}{P_{Noise}}\right)$$

In dB angegeben.

$P_{Signal}$  die Sendeleistung (Watt),  $P_{Noise}$  die Empfangsleistung (Watt)

## Signale und Störungen



Mögliche Ursachen der Störungen:

- Übersprechen zwischen den Leitungen
- Rauschen des Empfängers
- Einstreuungen durch andere Geräte / Anlagen (Motoren etc.)

## Kabeltypen

- Koaxialkabel → Geeignet für hochfrequente Signale
- Twinaxial-Kabel → Hoher Schutz
- Twisted Pair (TP) → Häufig im Einsatz (Shielded / Unshielded)
- Glasfaser → Hohe Bandbreite, Geringe Dämpfung, Resistent

## Schirmeigenschaften

- Drahtgeflecht → niederfrequente Einstreuungen
- Metallisch beschichtete Folien → hochfrequente Störungen

xx/lyTP worin TP für Twisted Pair steht:

xx steht für die Gesamtschirmung:

U = ungeschirmt

F = Folienschirm

S = Geflechschirm

SF = Schirm aus Geflecht und Folie

y steht für die Aderpaarschirmung:

U = ungeschirmt

F = Folienschirm

S = Geflechschirm

## TP Kabel und Störungen

- TP Kabel sind anfälliger auf Störungen als Koaxialkabel oder Glasfasern
- Störungen werden kapazitiv oder induktiv eingekoppelt z.B. von parallel geführten Leitungen oder Motoren etc.
- Bei Störungen von benachbarten Leitungen spricht man von Übersprechen oder Nebensprechen (crosstalk)

## Fausregel:

- Kapazitive Störung → Abschirmung
- Induktive Störung → twisted

## Lichtwellenleiter

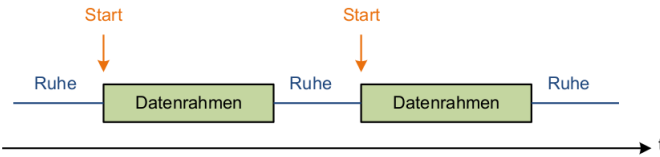
- Zentrum aus Kernglas mit hoher optischer Dichte (Brechungsindex 1.5)
- Vom Mantelglas umschlossen, geringere optische Dichte (Brechungsindex 1.48)
- Lichtstrahlen breiten sich im Kernglas aus und werden am Mantelglas totalreflektiert
- Die Eigenwellen (Ausbreitungswege der Lichtstrahlen) werden als Moden bezeichnet.

in km

## Data Link Layer (Sicherungsschicht)

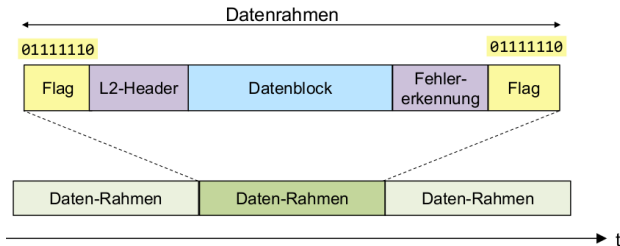
### Framing (Asynchron)

- Keine Daten → Nichts wird gesendet
- Zu Beginn eines Frames wird ein Start-Bit gesendet



### Framing (Synchron)

- Frames werden ohne Unterbruch gesendet
- Stehen keine Daten an, werden Flags gesendet
- Frames werden durch ein Start- und ein End-Flag begrenzt



### Bitstopfen

Wird verwendet, um ein Bitmuster zu garantieren.

- Sender fügt im Datenstrom nach 5 Einsen immer eine 0 ein.
- Empfänger wirft nach 5 Einsen immer ein Bit weg.

### Fehlererkennung / Fehlerkorrektur

- FER (Frame Error Ratio)
- RER (Residual Error Ratio)
- BER (Bit Error Ratio) Anzahl fehlerhafte Bits im Verhältnis zu Gesamtzahl der Bits

### Wahl der Framelänge

- Lange Frames → Höhere Nutzdatenrate, Fehleranfällig
- Kurze Frames → Tiefere Nutzdatenrate, Zuverlässig

### Datenraten

$$F_R = \text{FrameRate}, B = \text{BitRate}, F_L = \text{FrameLength}$$

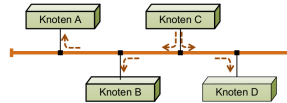
$$N = \text{NutzBits}, P = \text{Payload}$$

$$F_R = \frac{B}{8 \cdot (F_L + IFG)}$$
$$N = F_R \cdot P \cdot 8$$

## Ethernet 1 (LAN, Grundlagen)

### Topologien

#### Bus



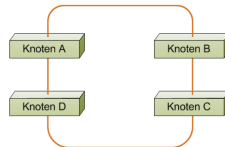
- Alle Stationen: sind passiv angeschlossen, horchen Leitung permanent ab, werden aktiv, wenn sie etwas senden wollen
- Taktrückgewinnung erlauben, um eine separate Taktleitung einzusparen
- Empfänger erkennt anhand einer Adresse, ob die Daten für ihn relevant sind

#### Linie



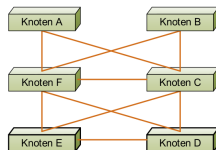
- Punkt-zu-Punkt Verbindungen zwischen benachbarten Knoten
- Alle Stationen müssen: Daten empfangen, Daten regenerieren, falls nötig weiterleiten
- Der Ausfall einer Station führt zur Segmentierung des LAN in zwei Teile

#### Ring



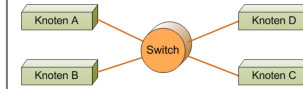
- Benötigt Verfahren zur Verhinderung von endlosem Kreisverkehr
- Gewisse Redundanz: beim Ausfall einer Station kann immer noch jede Station erreicht werden

#### Vermascht



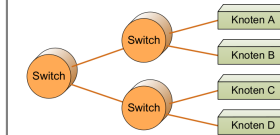
- Weitere Erhöhung der Redundanz:
- Ausfall einer oder eventuell auch mehrerer Stationen oder Verbindungen kann toleriert werden
- Zusätzliche Kosten und Aufwand, um mehrfache Lieferung von Daten zu verhindern

#### Stern



- Jede Station an zentralen Verteiler (Switch/Bridge) angeschlossen
- Verteiler entkoppelt Knoten elektrisch und macht LAN weniger störungsanfällig
- Verteiler sendet Daten, die er von einer Station erhält, an die anderen Knoten weiter

#### Baum



- Hierarchische Erweiterung der Sterntopologie
- Intelligenten Switches ermöglichen einen Grossteil der Kommunikation „lokal“
- Dadurch Verringerung der Last für die einzelnen Switches

## Übertragungsarten

#### Unicast



- Genau ein, klar spezifizierter Empfänger
- Frame trägt die Adresse dieses Empfängers
- Analogie: Briefpost

#### Broadcast



- An alle Knoten im LAN gerichtet
- Frame trägt die Broadcast-Adresse des LAN
- Analogie: Radio-Sendestation

#### Multicast



- Gruppe von Empfängern
- Frame trägt die Multicast-Adresse der Gruppe
- Analogie: Mailing-Liste

## MAC Adressen

Adressierung in LANs, bestehen aus 6 Bytes

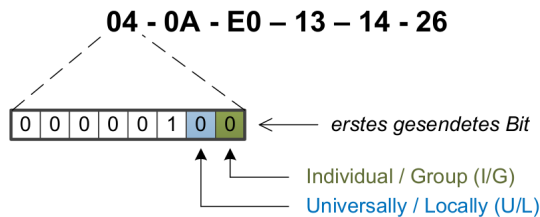
Registrierung bei IEEE

- 3-Byte "OUI" identifiziert Hersteller
- 3-Byte Laufnummer durch Hersteller verwaltet



|                 |               |
|-----------------|---------------|
| 00-00-0C        | Cisco         |
| 00-00-0E        | Fujitsu       |
| 00-00-AA        | Xerox         |
| 00-01-02        | 3Com          |
| 00-AA-00        | Intel         |
| <b>00-15-12</b> | <b>ZHAW ©</b> |
| 08-00-11        | Tektronix     |
| 08-00-20        | Sun           |
| 08-00-46        | Sony          |
| 08-00-5A        | IBM           |

Zwei Bits klassifizieren die MAC Adresse:



Individual/Group Bit (I/G):

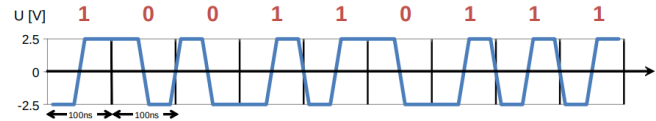
- 0 = individual address (Normalfall),
- 1 = group address z.B. Broadcast FF-FF-FF-FF-FF-FF

Universally/Locally Bit (U/L):

- 0 = universally administrated address (Normalfall)
- 1 = locally administrated address

## 10 Mbit/s (10BASE-T) Manchester Codierung

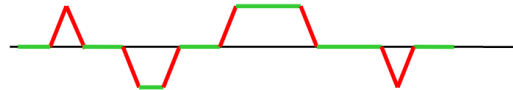
- Erlaubt die Taktrückgewinnung auf einfache Weise: weil Bei jedem Bit gibt es einen Signalwechsel
- Bandbreite von 10 MHz benötigt (also das doppelte des theoretischen Minimums)
- 1 positive Flanke, 0 negative Flanke



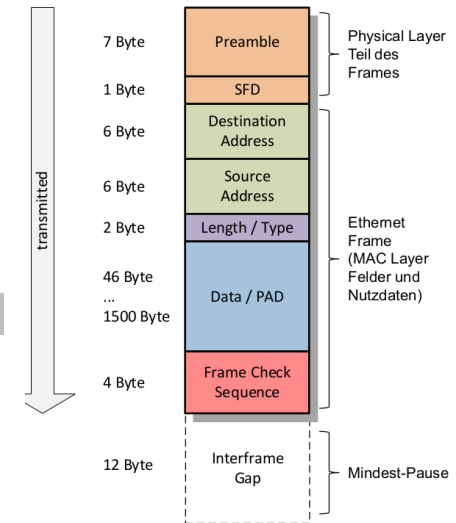
## 100 Mbit/s (100BASE-TX) NRZI-Codierung

- NRZI-Codierung (Non Return to Zero Inverted), kombiniert mit MLT-3 (MLT-3 = Multi-Level Transmit) 125 MBaud → 1 Symbol entspricht 8 ns
- 4B/5B Code Leitungscodierung
- 4 Bits des MII (Zeichen) werden mit einem 5 Bit-Zeichen (Code Group) auf der Leitung codiert

**011001010010001000110**



## Ethernet - Frame Format



- **Length/Type (2 Bytes):**
  - Fall 1: Länge von DATA ohne PAD ( $\leq 1500$ )
  - Fall 2: Typ von DATA = Protokoll der nächsten Schicht ( $\geq 1536$ )
    - Beispiel: 0x0800 für IP
- **Data / Padding (46 – 1500 Bytes):**
  - Enthält die eigentlichen Datenbytes (Nutzinformation)
  - Bei weniger als 46 Bytes Nutzdaten wird mit Padding (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

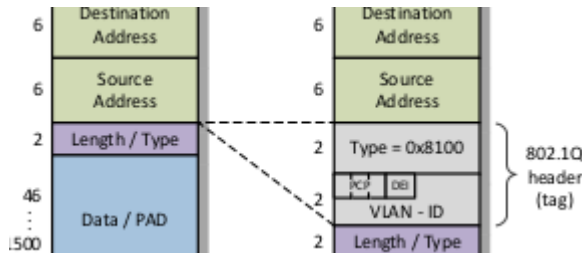
## Ethernet 2 (Ethernet Systeme)

### Virtuelle LANs

**Trunk-Links** Trunk Links sind Teil von mehreren VLANs. Auf den Trunk Links müssen Frames der verschiedenen VLANs eindeutig gekennzeichnet werden!

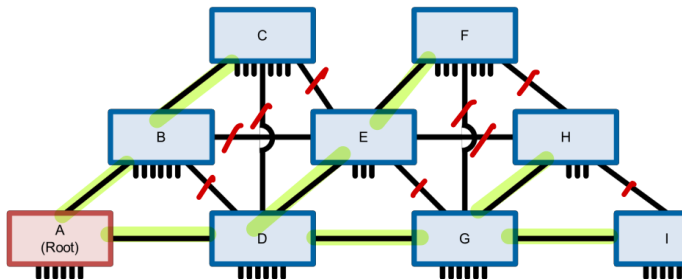
**VLAN-Tag** Erweiterung des Ethernet Headers durch einen VLAN-Tag. Die maximalen Nutzdatenlänge bleibt erhalten, der Ethernet Frame wird 4 Bytes länger

- Type 0x8100 → getaggttes Frame
- Priority Code Point ermöglicht die Priorisierung gewisser Applikationen
- Discard Eligibility Indicator 0 → Frame wird bei Überlastsituationen zuerst verworfen
- VLAN Tagging erfolgt oft beim Eintritt / Austritt ins Netz
- Für Endgeräte unsichtbar



### Spanning Tree

**Ziel:** Alle Segmente in einer loop-freien Topologie verbinden. Beim Spanning-Tree werden von redundanten Pfaden alle ausser einer gesperrt. Im Fehlerfall wird falls möglich ein ausgefallener Port ersetzt. Der Algorithmus bestimmt eine Root-Bridge, von welcher aus dem Baum aufgespannt wird.



### Autonegotiation

**Ziel:** Ermittlung der besten Betriebsart durch Austausch der Leistungsmerkmale zweier Netzwerkkomponenten.

## Bridges

Bridges verfügen über einen Mechanismus zum Erlernen von Adressen. Eine Bridge hört den Verkehr von allen Ports ab und merkt sich die Sender-Adressen aus den empfangenen Frames in der sogenannten «Filtering Database».

**Filtering Database** beinhaltet für jede bekannte Mac-Adresse das Bridge-Port, über welches der zugehörige Knoten erreichbar ist. Unbenutzte Einträge in der Filtering Database werden nach einer gewissen Zeit automatisch gelöscht

## Router

**Router** sind Komponenten, die es erlauben Subnetze miteinander zu verbinden. Router haben eine ähnliche Funktion wie Bridges, allerdings arbeiten sie auf dem Network Layer.

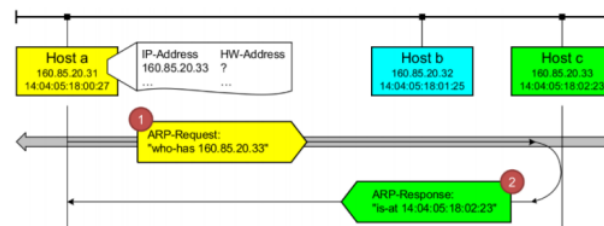
- Router empfangen nur Pakete, die direkt an sie adressiert sind.
- Die Weiterleitung erfolgt anhand der Network Layer Adresse.
- DBenutzen immer den optimalen Pfad.
- Für Endgeräte unsichtbar

## Routing-Tabelle

- Sortiert nach der Länge der Netzmaske
- Von oben nach unten durchsucht
- Verglichen werden die Netzadressen

## ARP (Address Resolution Protocol)

**Ziel:** Ermittlung der MAC-Adresse zu einer IP-Adresse.



## ARP (Internet Protokoll Format (IP-Header))

Ein IP-Packet besteht aus einem Header (min. 20 Byte) und Nutzdaten.

- **Version** IPv4 / IPv6
- **IHL** Header Length in 4-Byte (20 Byte → IHL = 5)
- **Type of Service** Erlaubt Priorisierung
- **Total Length** Länge des IP-Packets (Header + Nutzdaten)
- **ID Number** Identifikation des IP-Pakets / Fragmente
- **Flags** Kontroll-Flags für Fragmentierung
- **Fragment Offset** Gibt an, wo ein Fragment hingehört
- **Time to Live** Hop-Counter, 0 → Paket wird verworfen
- **Protocol** Übergeordnetes Protokoll

| 1. Byte                |   |   |   |     |   |   |   | 2. Byte         |   |    |    |    |    |    |    | 3. Byte            |    |                 |    |    |    |    |    | 4. Byte |    |    |    |    |    |    |    |
|------------------------|---|---|---|-----|---|---|---|-----------------|---|----|----|----|----|----|----|--------------------|----|-----------------|----|----|----|----|----|---------|----|----|----|----|----|----|----|
| 0                      | 1 | 2 | 3 | 4   | 5 | 6 | 7 | 8               | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16                 | 17 | 18              | 19 | 20 | 21 | 22 | 23 | 24      | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| 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      |   |   |   |     |   |   |   |                 |   |    |    |    |    |    |    |                    |    |                 |    |    |    |    |    |         |    |    |    |    |    |    |    |

Das unterliegende Netz limitiert die Grösse eines Pakets (Maximum Transfer Unit). Der Sender kennt die MTU der Netze nicht.

## Fragmentierung

Um über Netze mit verschiedenen Maximum Transfer Units (MTU) arbeiten zu können, unterstützt IP Fragmentierung und Reassembly.

- Länge der Nutzdaten = Vielfaches von 8 Bytes
- Die Pakete haben die gleiche und grösstmögliche Länge

Jedes IP Fragment beinhaltet alle notwendigen Daten um den Endknoten zu erreichen (IP Header) und ein Vielfaches von 8 Bytes an Transportlayer-Daten (Ausnahme: letztes Fragment)

## Reassembly

1. Zusammensetzen beim Zielhost
2. Letztes Fragment: MF = 0

| Feld | Position | Werte | Funktion               |
|------|----------|-------|------------------------|
|      | 0        | 0     | Reserved, must be Zero |
| DF   | 1        | 0 / 1 | May / Don't Fragment   |
| MF   | 2        | 0 / 1 | Last / More Fragments  |

## Internet-Adressierung (IPv4)

- Netzadresse → Tiefste Adresse im Subnetz
- Broadcast → Höchste Adresse im Subnetz



Beispiel im Anhang.

| Klasse | Adressbereich               | Anzahl Netze                      | Interfaces pro Netz |
|--------|-----------------------------|-----------------------------------|---------------------|
| 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 – 239.255.255.555 | Multicast Adressen                |                     |
| E      | 240.0.0.0 – 255.255.255.255 | Reserviert für zukünftige Nutzung |                     |

Private Adressbereiche (werden im Internet nicht weitergeleitet):

| Klasse | Netzadresse(n)              | Anzahl Netze | Subnetzmaske  |
|--------|-----------------------------|--------------|---------------|
| A      | 10.0.0.0                    | 1            | 255.0.0.0     |
| B      | 172.16.0.0 – 172.31.0.0     | 16           | 255.255.0.0   |
| C      | 192.168.0.0 – 192.168.255.0 | 256          | 255.255.255.0 |

## Transport Layer

### User Datagram Protocol (UDP)

- unzuverlässiges, verbindungsloses Protokoll
- UDP-Sitzungen werden durch eine 2-Tupel-Adresse (Ziel-IP, Zielport) identifiziert.
- keine Mechanismen zur Fehlererkennung oder Fehlerkorrektur, Zuverlässigkeit und Flusskontrolle
- UDP-Pakete → beliebiger Reihenfolge ankommen oder verloren gehen, ohne Empfänger benachrichtigt
- für Anwendungen: geringe Latenz → Echtzeitkommunikation oder Streaming.
- Multicasting und Broadcasting,

### Transmission Control Protocol (TCP)

- zuverlässiges, verbindungsorientiertes Protokoll
- implementiert Fehlererkennung, Flusskontrolle und Sequenzierung.
- TCP-Sitzungen werden durch eine 4-Tupel-Adresse (Quell-IP, Quellport, Ziel-IP, Zielport) identifiziert.
- 3-Way-Handshake, um eine Verbindung aufzubauen: SYN, SYN-ACK, ACK.
- Sequenznummern, um Reihenfolge empfangener Pakete zu überprüfen und fehlende oder beschädigte Pakete zu erkennen.
- Bestätigungen (ACKs), um Empfang von Datenpaketen zu bestätigen und Sender informieren, welche Daten erfolgreich übertragen.
- Flusskontrolle, um sicherzustellen, dass der Sender Daten nicht schneller sendet, als der Empfänger verarbeiten kann.
- Überlastkontrolle, um die Netzwerküberlastung zu vermeiden
- Segmentierung von Daten in kleinere Einheiten (Segmenten) für Übertragung → stellt sicher, dn Reihenfolge zusammengefügt.
- TCP bietet Mechanismen zur Zuverlässigkeit, Flusskontrolle und Fehlerbehebung, ist jedoch im Vergleich zu UDP (User Datagram Protocol) langsamer und erzeugt einen höheren Overhead.

## Ports

- System Ports (Well-Known)
- User Ports (Registered)
- Dynamic / Private Ports

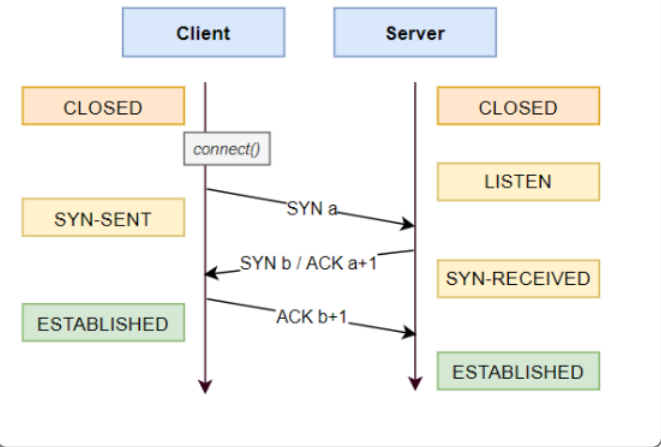
| System Ports | User Ports    | Dynamic Ports   |
|--------------|---------------|-----------------|
| 0 - 1023     | 1024 - 49'151 | 49'152 - 65'535 |

## Verbindungsauf und -abbau und Datenaustausch

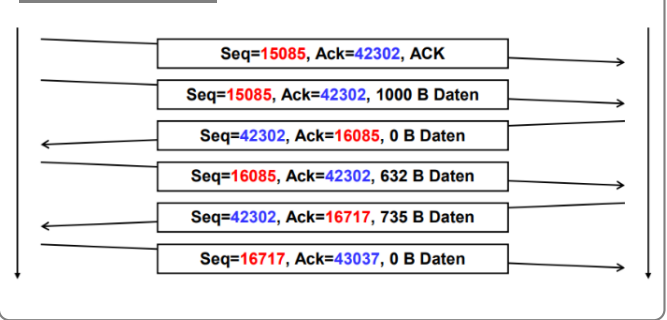
### Abkürzungen:

|              |                              |
|--------------|------------------------------|
| LISTEN       | Auf Anforderung warten       |
| SYN-SENT     | Auf Anforderung warten       |
| SYN-RECEIVED | Anforderung erhalten         |
| ESTABLISHED  | Verbindung besteht           |
| SYN          | Verbindungsaufbau            |
| ACK          | Paket bestätigen             |
| FIN          | Verbindungsabbau             |
| FIN-WAIT-1   | Verbindungsaufbau            |
| FIN-WAIT-2   | Abbauanforderung bestätigt   |
| CLOSE-WAIT   | Auf Lokale Verbindung warten |
| LAST-ACK     | Verbindungsabbau bestätigt   |
| TIME-WAIT    | Letzte Bestätigung gesendet  |

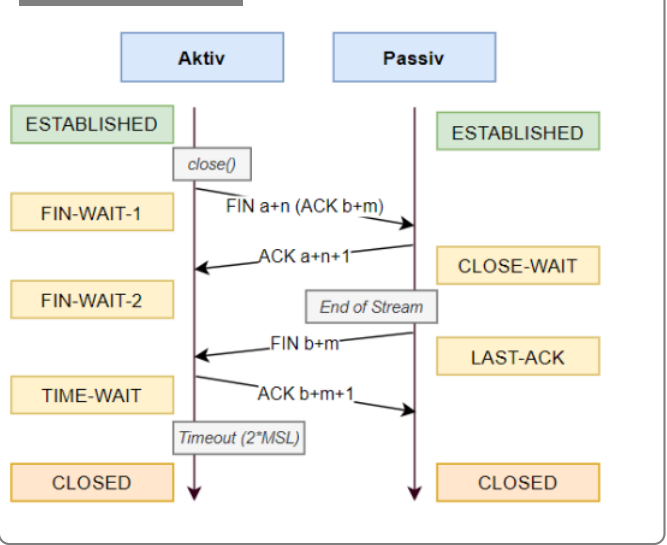
### Verbindungsaufbau:



### Datenaustausch:

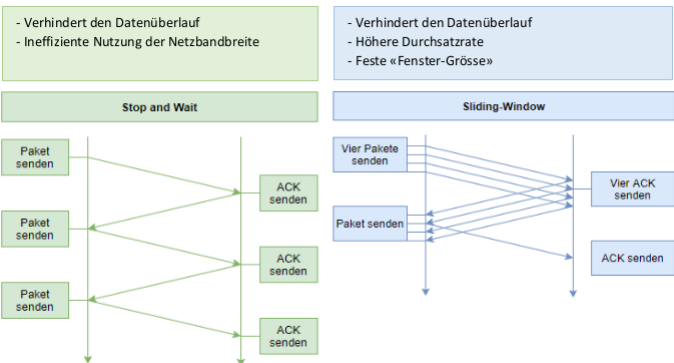


### Verbindungsabbau:

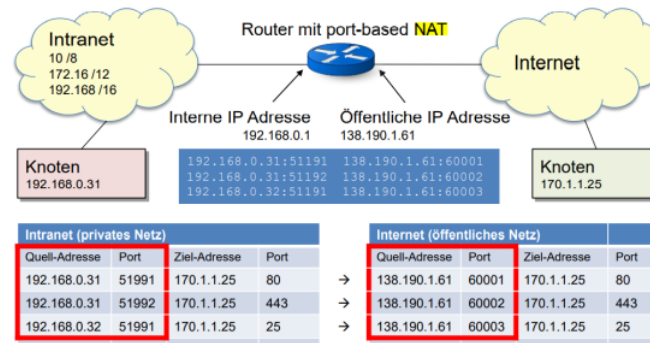
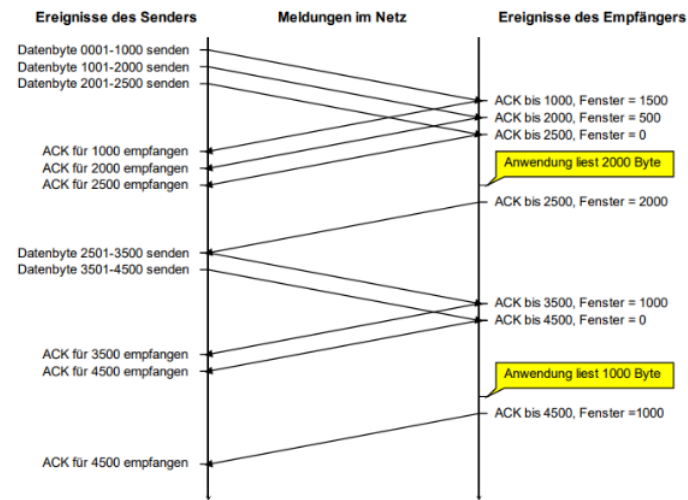


### Überlast des Empfängers: Fluss-Steuerung

TCP verwendet den Sliding-Window Mechanismus. Beide Seiten einen Buffer (Window).



## Fluss-Steuerung bei TCP

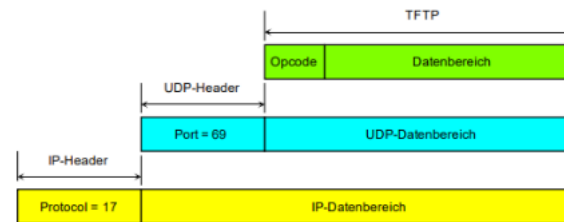


## BOOTP

- Manuelle Verwaltung
- Heimanwender sind überfordert
- Statische Adresszuordnung

## Trivial File Transfer Protocol (TFTP)

Basiert auf UDP



## Erkennung von verlorene Telegramme (Round Trip Time)

Um Fehler Paketverluste und andere Fehler zu verhindern, werden Pakete nach einer bestimmten Zeit erneut übertragen, wenn keine Bestätigung gesendet wurde. Um diese Zeit zu optimieren, misst TCP bei jeder aktiven Verbindung die Round-Trip Time (RTT).

## Retransmission Time-Out RTO

### Application Layer

### Domain Name Space (DNS)

## Dynamic Host Configuration Protocol (DHCP)

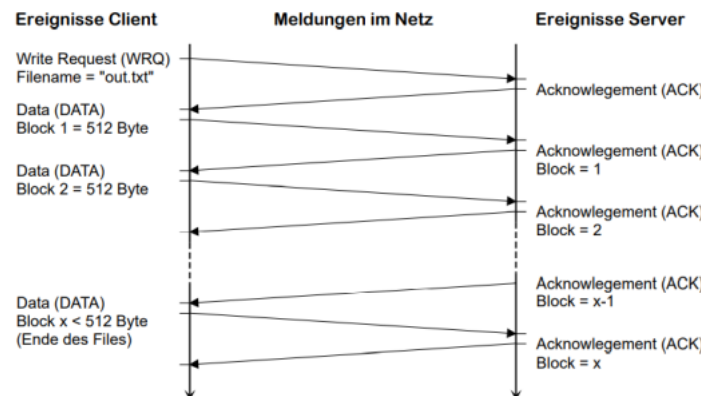
- Paketformat identisch zu BOOTP
- Dynamische Zuweisung von IP-Adressen
- PReserviert nur IP's von aktiven Geräte

Ablauf (DHCP):

1. Client sucht DHCP Server mittels Broadcast
2. DHCP Server antwortet (DHCP offer)
3. Der Client wählt einen Server und fordert eine
4. Der Server bestätigt mit einer Message, welche die endgültigen Parameter enthält
5. Vor Ablauf der Lease-Time erneuert der Client die Adresse

## Network Address Translation (NAT)

NAT verletzt das Konzept der OSI-Layer, da eine Network-Funktion auf den Transport-Header zugreift. IP-Adresse und Portnummer werden dabei verändert.



## Simple Mail Transfer Protocol (SMTP)

## Filtering-Database

Aging-Time: 50 Sekunden

| Zeit              | Step | S → E | Port 1 | Port 2 | Port 3 | Port 4 |
|-------------------|------|-------|--------|--------|--------|--------|
| Gelernte Adressen |      |       |        |        |        |        |
| 0 s               | 1    | AtoB  |        |        | A      |        |
| 20 s              | 2    | AtoC  |        |        | A      |        |
| 40 s              | 3    | BtoC  | B      |        | A      |        |
| 60 s              | 4    | CtoA  | B      |        | A      | C      |
| 80 s              | 5    | CtoA  | B      |        |        | C      |
| 100 s             | 6    | BtoA  | B      |        |        | C      |
| 120 s             | 7    | BtoA  | B      |        |        | C      |
| 140 s             | 8    | AtoB  | B      |        | A      |        |
| 160 s             | 9    | AtoB  | B      |        | A      |        |
| 180 s             | 10   | AtoB  |        |        | A      |        |

## Addressierung IPv4 Beispiel:

- Interface 000...000 32 – Länge vom Subnetz
- Subnetzmaske 255.255.240.0 1111'1111.1111'1111'0000.0000'0000
- Subnetz 160.85.16.0/20 20 = Länge

|              |     |     |     | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|--------------|-----|-----|-----|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Subnetzmaske | 255 | 255 | 240 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |    |
| Subnetz      | 160 | 85  | 16  | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0  | 1  | 0  | 1  | 0  | 1  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| Netzadresse  | 160 | 85  | 16  | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0  | 1  | 0  | 1  | 0  | 1  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

AND

|                           |     |     |     | 0   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---------------------------|-----|-----|-----|-----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Subnetzmaske (invertiert) | 255 | 255 | 240 | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |
| Subnetz                   | 160 | 85  | 16  | 0   | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0  | 1  | 0  | 1  | 0  | 1  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| Broadcast                 | 160 | 85  | 31  | 255 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0  | 1  | 0  | 1  | 0  | 1  | 0  | 0  | 0  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |

OR

## Umrechnung Dezimal zu Binär

177<sub>10</sub>

|     |   |
|-----|---|
| 177 | 1 |
| 88  | 0 |
| 44  | 0 |
| 22  | 0 |
| 11  | 1 |
| 5   | 1 |
| 2   | 0 |
| 1   | 1 |

10110001

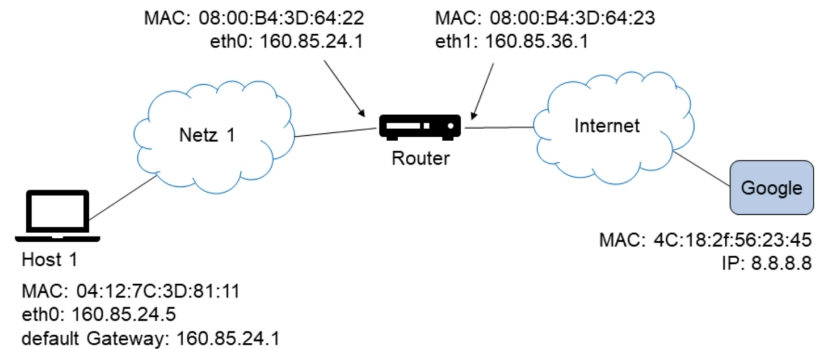
## Umrechnungstabelle

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

## IP Subnetzmasken

| Netzmaske (DEC) | Netzmaske (BIN)                     | Netz | Anzahl IPs |
|-----------------|-------------------------------------|------|------------|
| 255.255.0.0     | 11111111'11111111'00000000'00000000 | /16  | 65'534     |
| 255.255.128.0   | 11111111'11111111'10000000'00000000 | /17  | 32'766     |
| 255.255.192.0   | 11111111'11111111'11000000'00000000 | /18  | 16'382     |
| 255.255.224.0   | 11111111'11111111'11100000'00000000 | /19  | 8'190      |
| 255.255.240.0   | 11111111'11111111'11110000'00000000 | /20  | 4'094      |
| 255.255.248.0   | 11111111'11111111'11111000'00000000 | /21  | 2'046      |
| 255.255.252.0   | 11111111'11111111'11111100'00000000 | /22  | 1'022      |
| 255.255.254.0   | 11111111'11111111'11111110'00000000 | /23  | 510        |
| 255.255.255.0   | 11111111'11111111'11111111'00000000 | /24  | 254        |
| 255.255.255.128 | 11111111'11111111'11111111'10000000 | /25  | 126        |
| 255.255.255.192 | 11111111'11111111'11111111'11000000 | /26  | 62         |
| 255.255.255.224 | 11111111'11111111'11111111'11100000 | /27  | 30         |
| 255.255.255.240 | 11111111'11111111'11111111'11110000 | /28  | 14         |
| 255.255.255.248 | 11111111'11111111'11111111'11111000 | /29  | 6          |
| 255.255.255.252 | 11111111'11111111'11111111'11111100 | /30  | 2          |

## ARP Table Beispiel



| Request Typ (ARP-Request, ARP-Reply, IP Paket) | MAC Source | MAC Destination | IP source | IP Destination | What            |
|--|------------|-----------------|-----------|----------------|-----------------|
| ARP Request                                    | 11         | FF              | -         | -              | Who has IP 24.1 |
| ARP Reply                                      | 22         | 11              | -         | -              | 24.1 has 22     |
| IP Paket                                       | 11         | 22              | 24.5      | 8.8.8.8        | -               |



### Subnet Beispiel 1

Gegeben ist das Netz 172.30.10.0/25. Dieses Netz soll in drei Subnetze aufgeteilt werden: ein größeres Subnetz 1 für 50 IP-Hosts und zwei kleinere Subnetze 2 und 3 für je 25 IP-Hosts.

#### 1. Subnetz 1 für 50 Hosts:

Wir benötigen 6 Bits für die Host-IDs, um mindestens 50 Hosts zu unterstützen ( $2^6 - 2 = 62$ ). Das führt zu einer Subnetzmaske von /26 ( $32 - 6 = 26$ ).

- Netzadresse: 172.30.10.0/26
- Broadcastadresse: 172.30.10.63/26
- Anzahl adressierbarer Hosts: 62

#### 2. Subnetz 2 und 3 für jeweils 25 Hosts:

Wir benötigen 5 Bits für die Host-IDs, um mindestens 25 Hosts zu unterstützen ( $2^5 - 2 = 30$ ). Das führt zu einer Subnetzmaske von /27 ( $32 - 5 = 27$ ).

- **Subnetz 2:**
  - Netzadresse: 172.30.10.64/27
  - Broadcastadresse: 172.30.10.95/27
  - Anzahl adressierbarer Hosts: 30
- **Subnetz 3:**
  - Netzadresse: 172.30.10.96/27
  - Broadcastadresse: 172.30.10.127/27
  - Anzahl adressierbarer Hosts: 30

### Subnet Beispiel 2

Sie bekommen von Ihrem Internet Service Provider (ISP) ein privates Klasse-C Netz zugeteilt. In Ihrem Haus befinden sich 4 Parteien, welche sich den Internet-Anschluss teilen. Sie geben jeder Partei ein gleich grosses Subnetz, indem sie das Klasse-C Netz 192.168.1.0/24 in 4 Subnetze aufteilen.

Wir teilen das gegebene Klasse-C Netz 192.168.1.0/24 in vier gleich große Subnetze auf, indem wir zwei zusätzliche Bits für die Subnetz-ID verwenden. Dies führt zu einer neuen Subnetzmaske von /26 und 62 adressierbaren Hosts pro Subnetz ( $2^6 - 2 = 62$ ).

#### 1. Subnetz 1:

- Netzadresse: 192.168.1.0/26
- Netzmaske: 255.255.255.192
- Broadcastadresse: 192.168.1.63/26
- Default Gateway: 192.168.1.1
- Anzahl adressierbarer Hosts: 62

#### 2. Subnetz 2:

- Netzadresse: 192.168.1.64/26
- Netzmaske: 255.255.255.192
- Broadcastadresse: 192.168.1.127/26
- Default Gateway: 192.168.1.65
- Anzahl adressierbarer Hosts: 62

#### 3. Subnetz 3:

- Netzadresse: 192.168.1.128/26
- Netzmaske: 255.255.255.192
- Broadcastadresse: 192.168.1.191/26
- Default Gateway: 192.168.1.129
- Anzahl adressierbarer Hosts: 62

#### 4. Subnetz 4:

- Netzadresse: 192.168.1.192/26
- Netzmaske: 255.255.255.192
- Broadcastadresse: 192.168.1.255/26
- Default Gateway: 192.168.1.193
- Anzahl adressierbarer Hosts: 62

## Wireshark Hex-Dump

```

0000:  08 00 2B C3 AC A5 00 00 F8 1A 84 1A 08 00 45 00
0010:  00 2C 1B 31 40 00 80 06 99 5E A0 55 82 2A A0 55
0020:  83 67 04 1A 12 67 00 00 C0 C5 00 00 00 00 60 02
0030:  20 00 5A A3 00 00 02 04 05 B4 00 00 A3 7C 51 FB
  
```

- Oktett 0-5: Destination MAC-Address
- Oktett 6-11: Source MAC-Address
- Oktett 12/13: Length / Type → hier Type = 0x0800
- Oktett 14-59: Data / padding
- Oktett 60-63: Frame Check Sequence

3. Oktett =  $0x2B = 00101011$

## Dezimal, Hexadezimal, Binär

| Dezimal | Hexadezimal | Binär |
|---------|-------------|-------|
| 0       | 0           | 0000  |
| 1       | 1           | 0001  |
| 2       | 2           | 0010  |
| 3       | 3           | 0011  |
| 4       | 4           | 0100  |
| 5       | 5           | 0101  |
| 6       | 6           | 0110  |
| 7       | 7           | 0111  |
| 8       | 8           | 1000  |
| 9       | 9           | 1001  |
| 10      | A           | 1010  |
| 11      | B           | 1011  |
| 12      | C           | 1100  |
| 13      | D           | 1101  |
| 14      | E           | 1110  |
| 15      | F           | 1111  |

## Fragmentierung

| Frame# | 1   | 1   | 1   | 1   |
|--------|-----|-----|-----|-----|
| TL     | 420 | 420 | 380 | 340 |
| MF     | 1   | 1   | 1   | 0   |
| FO     | 0   | 50  | 100 | 145 |

Header := 4 \* IHL  
 Daten = TL - Header

