

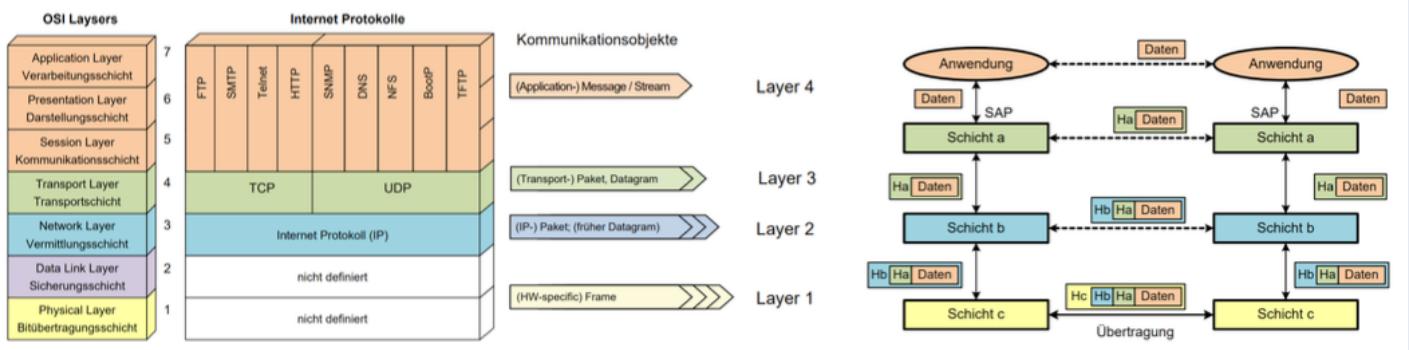
# 1 OSI-Modell

Ein Dienst sendet und empfängt bestätigte und unbestätigte Daten.  Klassifizierung von Diensten <ul style="list-style-type: none"> <li>• Verbindungsorientiert oder verbindungslos</li> <li>• Zuverlässig oder unzuverlässig</li> </ul>	<b>Verbindungsorientiert</b> Verbindungs-Aufbau nötig Ziel muss bereit sein	<b>Verbindungslos</b> Jederzeit Nachrichten schicken Ziel muss nicht «bereit» sein
	<b>Zuverlässig</b> Kein Datenverlust Sicherung durch Fehler-Erkennung -/ Korrektur Text-Nachrichten	<b>Unzuverlässig</b> Möglicher Datenverlust Keine Sicherung Streaming

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.

Ein **Protokoll** ist eine Sammlung von Nachrichten, Nachrichtenformaten und Regeln zu deren Austausch. Im zwischenmenschlichen Bereich könnte man die Kniege als Protokoll bezeichnen. Sie legt einen gewissen «Verhaltens-Standard» nach welchem wir uns richten.

In der Technik ist ein **Kommunikationsprotokoll** eine Vereinbarung, die festlegt wie eine Datenübertragung zwischen Kommunikationspartnern abläuft.



## (I) Physical Layer

Übertragung eines Bit-Stromes zwischen zwei Knoten über physikalisches Medium.

- Elektrische & optische Eigenschaften (Pegel, Timing)
- Codierung (Manchester etc.)
- Mechanische Eigenschaften (Pinbelegung)

## (II) Data Link Layer

Stellt der höheren Schicht gesicherte Übertragungsstrecke zur Verfügung.

Für zwei Teilnehmer (Peer to Peer):

- Massnahmen zur Fehlererkennung & Korrektur
- Verpacken der Datenblöcke vom Network Layer in Datenrahmen und Auspacken der Datenblöcke aus empfangenen Datenrahmen
- Flow Control: Massnahmen, dass Sender nicht zu schnell sendet

Mehrere Teilnehmer (gilt zusätzlich):

- Adressierung der Teilnehmer durch eindeutige Adresse
- Medium Access Control: Steuerung des Zugriffs

## (III) Network Layer

Daten austauschen zwischen Knoten vereinheitlichen. Dazu netzweite Layer 3 Adressierung erforderlich sowie Routing-Verfahren.

**Verbindungsorientierter Dienst:**

- Datenpakete werden für Weiterleitung entscheid mit lokalem Identifier vers.
- Vor der Übertragung wird Pfad durch das Netzwerk festgelegt
- Nach Übertragung werden Ressourcen wieder freigegeben
- Wichtiges Verfahren: Multi Protocol Label Switching (MPLS)

**Verbindungsloser Dienst:**

- Paket erhalt vollständige Adresse des Empfängers
- Jeder einzelne Knoten entscheid selbst, welcher Weg genutzt wird  
→ Dazu werden Informationen über Verbaubarkeit in Routing-Tabellen abgeleitet (Beispiel: IP-Protokoll)

### Verbindungsorientiert vs Verbindungslos

- |                                                                          |                                               |
|--------------------------------------------------------------------------|-----------------------------------------------|
| ◦ Charakteristika (Delay, Verlust) durch vorgegebene Pfad sichergestellt | ◦ Keine Pfadberechnung = kein Ablauf          |
| ◦ Ganzheitliche Lenkung des Verkehrsstroms                               | ◦ Beim Unterbruch wird andere Route verwendet |
| ◦ Reihenfolge der Daten bleibt                                           |                                               |

## (IV) Transport Layer

Stellt End to End Übertragungsqualität sicher. Wird abhängig von Layer 1-3 entsprechend Zuverlässigkeit geleiht.

- User Data Protocol (UDP): Verbindungslos, unsicher
- Transmission Control Protocol (TCP): Verbindungsorientiert, sicher

## (V) Session Layer

Legt Ablauf der Kommunikation fest. (Aufbau und Abbau von Sessions)

Beim Abbau einer Session für Wiederaufbau verantwortlich.

## (VI) Presentation Layer

Für Darstellung der zu übertragenden Daten zur Verfügung.

Typisches Beispiel: Kodierung der Daten auf Standardisierte Art:  
◦ ASCII, ISO, Unicode

## (VII) Application Layer

Bindetglied zur eigentlichen Anwendung.

Typische Protokolle:

- File Transfer Protocol (FTP)
- Simple Mail Transfer Protocol (SMTP)
- Hypertext Transfer Protocol (HTTP)
- Domain Name System (DNS)

## 2 Übertragungsmedien

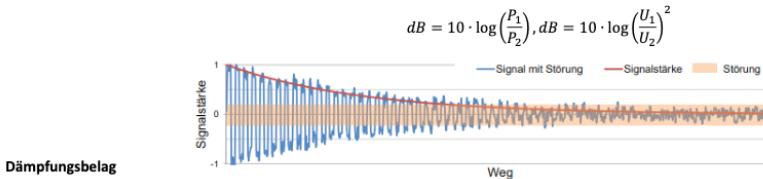
### Ausbreitungsgeschwindigkeit

Funk- oder Licht-Signale sind elektromagnetische Wellen, die sich im Vakuum mit Lichtgeschwindigkeit  $c_0 = 299'792'458 \frac{m}{s}$  ausbreiten. Die Vakuumgeschwindigkeit kann nicht überschritten werden.

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

### Signaldämpfung

Die Signaldämpfung bezeichnet die Leistungsabnahme eines Signals auf einer Übertragungsstrecke. Sie ist ein wesentlicher Faktor, der die erreichbare Distanz beschränkt. Die Angabe der Signaldämpfung erfolgt in dB als logarithmische Verhältniszahl von Eingangsleistung  $P_1$  zur Aufgangsleistung  $P_2$ .



### Dämpfung von 6dB

Leistungsabnahme: Faktor 4  
Spannungsabnahme: Faktor 2

Für Übertragungsmedien ist die Dämpfung pro Distanz massgebend. Typischerweise in dB pro 100 m angegeben.

### Kabel-Typen

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

### Schirmegenschaften

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

$xx/y$  worin  $TP$  für Twisted Pair steht:

$xx$  steht für die Gesamtschirmung:

U = ungeschirmt

F = Folenschirm

S = Geflechtschirm

SF = Schirm aus Geflecht und Folie

$y$  steht für die Aderpaarschirmung:

U = ungeschirmt

F = Folenschirm

S = Geflechtschirm

### (2.2) Koaxialkabel

Gut geeignet für Übertragung von hochfrequenten Signalen.

- Kleiner Dämpfungsbelag
- Unempfindlich gegenüber elektromagnetischen Störungen
- Für grosse Distanzen geeignet
- Dürfen nicht geknickt / verquält werden

⇒ Heute: Für Peer to Peer in Hochgeschwindigkeitsnetzen

### Funktionsprinzip

Störungen können kapazitiv, induktiv oder galvanisch auftreten.

Durch das Senden eines komplementären Signals und das Addieren beider Signale beim Empfänger durch einen Differenzverstärker, hat sich das Signal vollständig auf und das Störsignal bleibt abgäng. Bei den direkten Signalausführungen ändern sich lediglich die Art / Umwandlung der Aufteilung des Signals. Das Prinzip bleibt dasselbe.

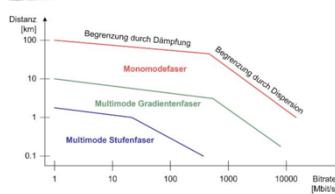
### (2.4) Lichtwellenleiter

Bestehen aus Glas und haben folgende Vorteile:

- Vollständige Unempfindlichkeit gegen elektromagnetische Störungen
- Kleine Dämpfung → Große Distanzen
- Große Bandbreite und große Übertragungsgeschwindigkeiten

Dispersion: Verzerrung des Signals → Nicht mehr erkennbar

Arten von Glasfaserkabel:



Das Glasfaserkabel wird durch die Dämpfung und die Dispersion beeinträchtigt. Die Dispersion stellt eine grössere Problematik dar, da die Dämpfung im Vergleich zu elektrischen Leitern minim ist. Die Moden-Dispersion verursacht bei grösseren Bitraten einen grösseren Effekt der Verbreitung der Signale. Somit "verschmieren" die Pulse schneller, so dass diese nicht mehr als solche erkennbar sind. Die Distanz und die Lichtimpulsdichte beschränken sich dabei gegenseitig.

- Folgende Medien können über folgende Eigenschaften beeinträchtigt werden:

- Koaxialkabel
  - Dämpfung
- Twisted Pair Kabel
  - Dämpfung
  - Crosstalk
  - Kapazitiv eingekoppelte Störungen
  - Induktiv eingekoppelte Störungen
- Glasfaserkabel
  - Dämpfung
  - Dispersion

## 2.1 Umrechnungen

DEC	HEX	BIN	DEC	HEX	BIN	DEC	HEX	BIN
0	00	00000000	43	2B	00101011	86	56	01010110
1	01	00000001	44	2C	00101100	87	57	01010111
2	02	00000010	45	2D	00101101	88	58	01010100
3	03	00000011	46	2E	00101110	89	59	01010101
4	04	00000100	47	2F	00101111	90	5A	01010110
5	05	00000101	48	30	00110000	91	5B	01010111
6	06	00000110	49	31	00110001	92	5C	01011100
7	07	00000111	50	32	00110010	93	5D	01011101
8	08	00001000	51	33	00110011	94	5E	01011110
9	09	00001001	52	34	00110100	95	5F	01011111
10	0A	00001010	53	35	00110101	96	60	01100000
11	0B	00001011	54	36	00110110	97	61	01100001
12	0C	00001100	55	37	00110111	98	62	01100010
13	0D	00001101	56	38	00111000	99	63	01100011
14	0E	00001110	57	39	00111001	100	64	01100100
15	0F	00001111	58	3A	00111010	101	65	01100101
16	10	00010000	59	3B	00111011	102	66	01100110
17	11	00010001	60	3C	00111100	103	67	01100111
18	12	00010010	61	3D	00111101	104	68	01101000
19	13	00010011	62	3E	00111110	105	69	01101001
20	14	00010100	63	3F	00111111	106	6A	01101010
21	15	00010101	64	40	01000000	107	6B	01101011
22	16	00010110	65	41	01000001	108	6C	01101100
23	17	00010111	66	42	01000010	109	6D	01101101
24	18	00011000	67	43	01000011	110	6E	01101110
25	19	00011001	68	44	01000100	111	6F	01101111
26	1A	00011010	69	45	01000101	112	70	01110000
27	1B	00011011	70	46	01000110	113	71	01110001
28	1C	00011100	71	47	01000111	114	72	01110010
29	1D	00011101	72	48	01001000	115	73	01110011
30	1E	00011110	73	49	01001001	116	74	01110100
31	1F	00011111	74	4A	01001010	117	75	01110101
32	20	00100000	75	4B	01001011	118	76	01110110
33	21	00100001	76	4C	01001100	119	77	01110111
34	22	00100010	77	4D	01001101	120	78	01111000
35	23	00100011	78	4E	01001110	121	79	01111001
36	24	00100100	79	4F	01001111	122	7A	01111010
37	25	00100101	80	50	01010000	123	7B	01111011
38	26	00100110	81	51	01010001	124	7C	01111100
39	27	00100111	82	52	01010010	125	7D	01111101
40	28	00101000	83	53	01010011	126	7E	01111110
41	29	00101001	84	54	01010100	127	7F	01111111
42	2A	00101010	85	55	01010101			

45834

Vorsatzsilbe	Zeichen	Zehnerpotenz
<b>Exa</b>	E	$10^{18}$
<b>Peta</b>	P	$10^{15}$
<b>Tera</b>	T	$10^{12}$
<b>Giga</b>	G	$10^9$
<b>Mega</b>	M	$10^6$
<b>Kilo</b>	k	$10^3$
<b>Hekto</b>	h	$10^2$
<b>Deka</b>	da	$10^1$
<b>Dezi</b>	d	$10^{-1}$
<b>Zenti</b>	c	$10^{-2}$
<b>Milli</b>	m	$10^{-3}$
<b>Mikro</b>	μ	$10^{-6}$
<b>Nano</b>	n	$10^{-9}$
<b>Piko</b>	p	$10^{-12}$
<b>Femto</b>	f	$10^{-15}$
<b>Atto</b>	a	$10^{-18}$

$$\begin{aligned} 2^1 &= 2, \\ 2^2 &= 4, \\ 2^3 &= 8, \\ 2^4 &= 16, \\ 2^5 &= 32, \\ 2^6 &= 64, \\ 2^7 &= 128, \\ 2^8 &= 256, \\ 2^9 &= 512, \\ 2^{10} &= 1024, \\ 2^{11} &= 2048, \\ 2^{12} &= 4096, \dots \end{aligned}$$

### 3 Physical Layer

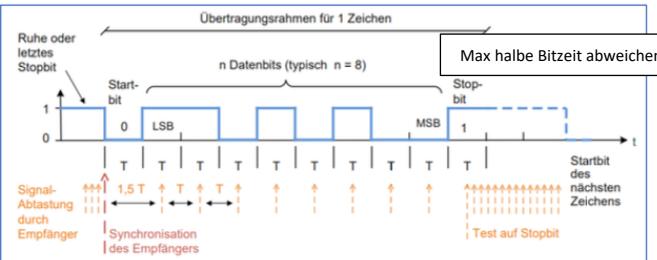
⇒ Ungesicherte Übertragung eines Bitstroms zwischen 2 Teilnehmern über das physikalische Medium

#### Serielle asynchron Übertragung (ohne Synchronisations-Takt)

Zwischen Sender und Empfänger werden folgende Abmachungen benötigt

- Bitrate
- Anzahl Datenbits Typischerweise 1 Byte
- Anzahl Stopbits Typischerweise 1 Bit

Nachteil: bei einem Bitfehler => Datenblock wertlos

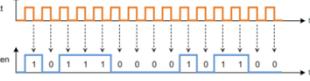


#### Serielle synchron Übertragung

Bei der synchronen Übertragung arbeitet der Empfänger mit dem gleichen Takt wie der Sender.

- Es werden keine Start- und Stopbits benötigt
- Der Takt muss zusätzlich übertragen werden

Die Übertragung des Takts erfolgt über ein Codierungsverfahren oder eine zusätzliche Leitung.



Effizientere Bandbreitennutzung -> keine Start + Stopbits

#### Arten der Kommunikation (Verkehrsbeziehung)

- Simplex Ein Kanal, in eine Richtung
- Halbduplex Ein Kanal, abwechselndweise in zwei Richtungen
- Voll duplex Ein Kanal pro Richtung

#### Arten der Verbindungen (Kopplung)

- Punkt-Punkt Direkte Verbindung zweier Kommunikationspartner
- Shared Medium Mehrere Partner verwenden das gleiche Medium

#### Datenübertragungsrate

- Baudrate Symbole pro Sekunde
- Zeichenrate Zeichen pro Sekunde

Die maximale Symbolrate  $f_s$  (Baud) ist gleich der doppelten Bandbreite B (Hz) des Übertragungskanals.  $f_s = 2B$

#### Bandbreite

Die Bandbreite hängt von der Übertragungsstrecke und der Stärke des Signals im Vergleich zu den vorhandenen Störungen, ab.

- Eigenschaft des Übertragungskanals und durch das Medium begrenzt
- Maßeinheit Hertz (Hz)

#### Maximal erreichbare Bitrate

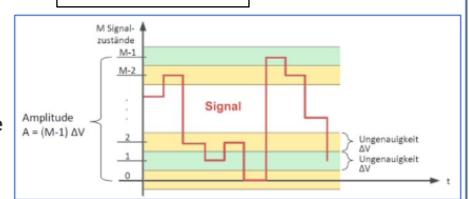
Maximal Bitrate R [bit/s]

- $R \leq 2B \cdot \log_2(M)$

#### Unterscheidbare Signalzustände

- $M = 1 + \frac{A}{\Delta V}$

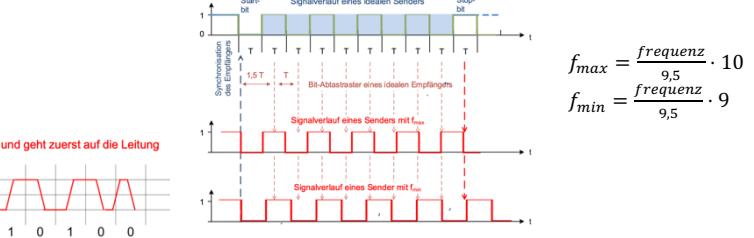
$$\log_2(M) = \frac{\log(M)}{\log(2)}$$



#### Kanalkapazität nach Shannon (Cs)

$$C_s = B \cdot \log_2(1 + \frac{S}{N}) \quad S/N: \text{Signal to Noise Ratio}$$

$$S/N \text{ ist } \mu = \sqrt{1+S/N}$$



#### Leistungscodes

- Separate Leitung zur Übertragung des Takts

Anforderungen: Taktrückgewinnung & Gleichspannungsfreiheit

#### Gleichspannungsfreiheit (AMI, HDB3)

- Mittlerer Spannungswert eines Signals = 0
  - o 0 und 1 Bits wechseln sich regelmäßig ab

#### Taktrückgewinnung (HDB3, Manchester 10Base-2)

- Lokaler Takt darf nie verloren gehen
- Zu viele 0 sorgen für Taktverlust

### 4 Data Link Layer

$$\text{FrameLength} = \text{Preamble} + \text{SA} + \text{DA} + \text{Type} + \text{FCS} + \text{IFG} = 38$$

$$\text{Framerate} = \text{Bitrate} / ((\text{Framegröße in Bytes} + \text{IFG}) * 8) = 100'000'000 \text{ bit/s} / (26 + 1500 + 12) * 8$$

#### Framing (Asynchron)

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



#### Fehlererkennung / Fehlerkorrektur

- FER (Frame Error Ratio)
- RER (Residual Error Ratio)
- BER (Bit Error Ratio)

#### Wahl der Framelänge

- Lange Frames Höhere Nutzdatenrate, Fehleranfälligkeit
- Kurze Frames Tiefer Nutzdatenrate, Zuverlässigkeit

$$\text{Nettobitrate} = \text{Bruttobitrate} * \frac{\text{Nutzdaten}}{\text{Nutzdaten} + \text{Header}}$$

Einfluss der Frame Länge auf den Durchsatz

Frame Erfolgswahrscheinlichkeit:  $(1 - p_e)^N$

Frame Fehlerwahrscheinlichkeit:  $1 - (1 - p_e)^N$

#### Optimale Frame-Länge:

$$N = \sqrt{\frac{H}{p_e}} \quad (H = \text{Header-Länge}; p = \text{BER})$$

#### Fehlerkorrektur

Erkennbare Fehler = Hamming-Distanz - 1

#### Framing (Synchron)

- Frames werden ohne Unterbruch gesendet



#### Datenraten

- $F_R = \text{FrameRate}, B = \text{BitRate}, F_L = \text{FrameLength}$
- $N = \text{NutzBitRate}, P = \text{Payload}$

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

#### (4.6) Medium Access Control

##### Häufige Slave Verzweigen

Häufig tritt zyklisches Sintern ab und konkurriert zueinander.

Vorteil: Keine Korrasie.

Nachteil: Häufig Single Point of Failure (SPOF).

#### Framing: 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.



**Deterministisch:** Master/Slave, Token-, Zeitgesteuert  
**Undeterministisch:** Kollisionserkennung und -auflösung (CSMA/CD und CSMA/CR)

#### Token Passing

Sendeunterdrückung in einer Reihenfolge wahrgenommen

Master: Dezentralisiert

Master: Anforderte (Startup, Token Verlust)

Master: Fehlerfreiheit, jeder kann nur den seinen holen

Master: Sehr einfache Realisierung (CSMA)

Dezentral: kein Kontroll der Bus (es kann nicht direkt ablesen)

Vorteil: kein Master, Stationen gleichberechtigt

Nachteile: Polling notwendig

CSMA/CD = Collision Detection, Überlappungsbereich, starker Konflikt

CSMA/CR = Collision Resolution, Konflikt-Vermeidung

#### (4.5) Fehlerkorrektur

Backward Error Correction:

Nach jedem Paket muss Bestätigung folgen. Bei hoher BER wird das sehr ineffizient.

Forward Error Correction (FEC):

Nach eiemem Fehler wird die am wahrscheinlichsten gesendete Wiedergabe geschätzt.

Korrigierbare Fehler =  $\frac{\text{Hamming Distanz} - 1}{2}$

#### Parity

- eine der einfachsten Methoden für die Fehlererkennung
- 1-Bit Fehler erkennbar

Jedem Datenelement wird ein zusätzliches Parity-Bit hinzugefügt, so dass die Anzahl der Einsen inklusive Parity-Bit entweder gerade (Even Parity) oder ungerade (Odd Parity) ist.

Längs- und Querparity: Hamming-Distanz 4

# 5 Local Area Network

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)



Kollisionen können durch Überlagerung von Signalen entstehen. Kollisionen müssen erkannt werden.

Bedingung für Kollisionserkennung

- Ohne Repeater  $t_{frame} > 2 \cdot t_{transfer}$
- Mit Repeater  $t_{frame} > 2 \cdot (\sum t_{transfer} + \sum t_{forwarding})$

Maximale Ausdehnung eines Segments  $t_{frame} = \text{Framesize} / \text{Bitrate} = ((B + 18 + 1'500) * 8) / \text{Bitrate}$

$$t_{frame} = \frac{\text{Framesize}_{min}}{\text{Bitrate}}, \quad t_{transfer} = \frac{d_{max}}{C_{Medium}}$$

Ein Knoten kann Kollisionen lokal nur erkennen, solange er selbst am Senden ist.

$$d_{max} < \frac{1}{2} \cdot \frac{\text{Framesize}_{min}}{\text{Bitrate}} \cdot C_{Medium}, \quad d_{max} < \frac{1}{2} \cdot \frac{576 \text{ Bit}}{10 \cdot 10^6 \cdot \text{Bit/s}}$$

Bustopologie:

- Stationen passiv angeschlossen
- Werden nur beim Senden aktiv
- Empfänger erkennt, falls Paket für ihn relevant



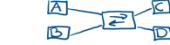
Ringtopologie:

- Besitzt Verfahren zur Vermeidung von Loops.



Sterntopologie:

- Jeder Knoten an Switch



Linientopologie:

- Peer to Peer von Knoten
- Jeder Knoten muss Daten empfangen und weiterleiten
- Ausfall Knoten = Spaltung LAN



Dopperringtopologie:

- Erhöhung Redundanz: Ausfall eines Rings => Anderer übernimmt

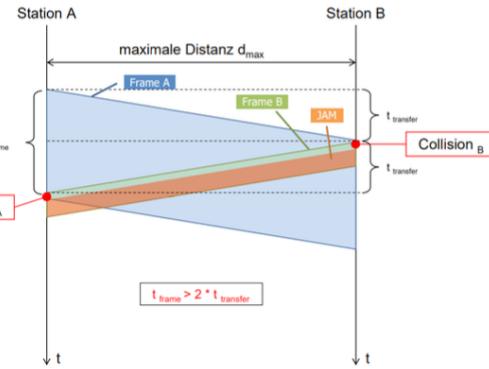


Baumtopologie:

- Hierarchische Erweiterung Stern-topologie



Bedingung für Kollisionserkennung



## Ethernet Format

Length/Type (2 Bytes)

- Fall 1: Länge von DATA ohne PAD ( $\leq 1500$ )
- Fall 2: Typ von Data = Protokoll der nächsten Schicht ( $\geq 1536$ )

Data / Padding (46 – 1500 Bytes)

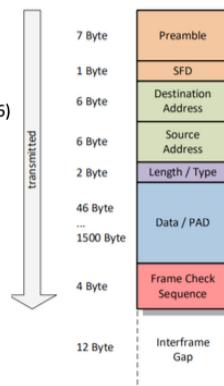
- Enthält die eigentlichen Datenbytes
- Bei weniger als 46 Bytes wird mit PAD Bytes abgefüllt

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



## Repeater and Collision Domain

Eine Collision Domain ist ein Teilbereich eines LANs, in dem die Frames der Stationen miteinander kollidieren können.

### Erkennen von Kollisionen

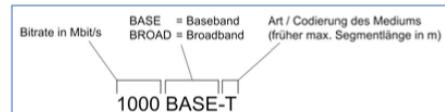
- Halbduplex Collision Detection Unit
- Voll duplex Keine Kollisionen

### Shared Medium Ethernet

- Carrier Sense Multiple Access with Collision Detection (CSMA/CD)

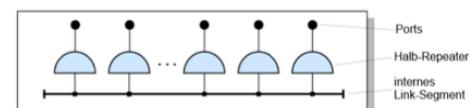
### Normen für CSMA/CD

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



### Repeater / HUB

Ankommendes Signal wird an alle anderen Ports weitergeleitet, regeneriert und ausgesendet.



## CSMA/CD = Carrier Sense Multiple Access with collision detection

### (I) Senden ohne Kollision:

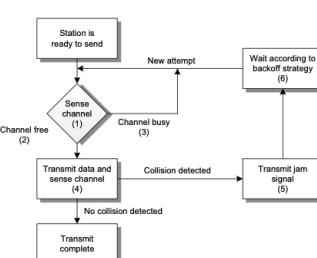
- Warten bis Kanal frei
- Physical Layer sendet Präambel und SFD, damit Empfänger sich synchronisieren kann
- Physical Layer überwacht während Sendezzeit Signalepegel  
⇒ Bei Kollision (Signaleinstieg) wird Collision Detected Signal an Data Link Layer gesendet

### (II) Empfangen ohne Kollision:

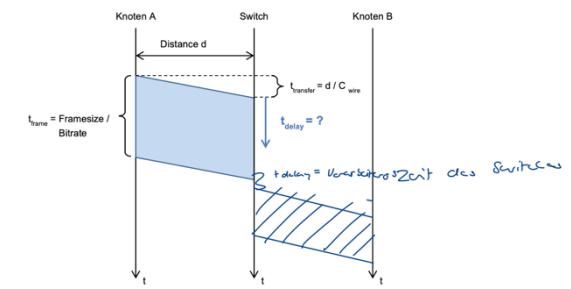
- Alle Knoten erkennen Senderseiten und medien ihrem Data Link Layer, keine Jendversuche zu unternehmen
- Alle Knoten im LAN werden zu Empfangsknoten

### (III) Senden mit Kollision:

- Falls mehrere Knoten gleichzeitig Kanal als frei sehen
- JAH - Signal wird an alle versendet
- Data Link Layer des Empfängers erkennt durch Prüfsumme Fehler



## Switch Delays (Store and Forward)



# 6 Switched LAN und Ethernet-Technologien

Wird die Adresse nicht gefunden, wird das Frame an alle gesendet!

**Bridges** = L2-Switch mit mehreren Ports

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*». Diese 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.

Diese Verarbeitung benötigt etwas Zeit, ist aber dennoch vorteilhaft, da das Paket nur an die richtige *Collision Domain* geschickt wird.

**Multi-Port-Bridges** verbinden mehr als zwei Segmente.

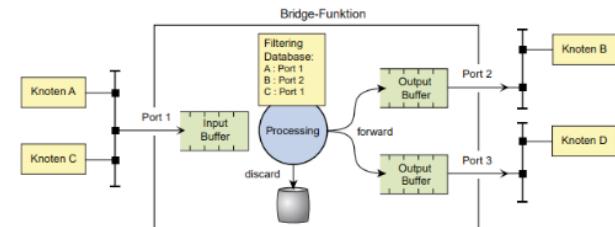
- Daten werden ausschliesslich an den richtigen Port weitergeleitet.
- Standard-Komponente zur Kopplung von Segmenten
- Werden als Ethernet-Switch bezeichnet

## Broadcast and Collision Domain

Eine **Collision Domain (CD)** besteht aus mit Repeatern zusammengeschlossenen Segmenten.

- Max. halb so lange wie die Ausdehnung des kürzesten Frames

Ein virtuelle LAN bildet eine **Broadcast Domain**. Das heisst die Grenzen für die Verteilung der Broadcast-Frames.



BD = alle Stationen, die auf Layer 2 (durch Repeater, Bridges, etc.) zusammengeschlossen sind

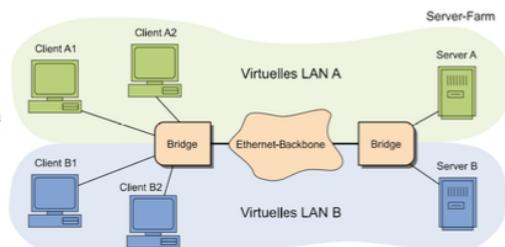
## VLANs

Mithilfe von virtuellen LAN kann ein grosses Netz in unabhängige logische Netze aufgeteilt werden. Jedes Switch-Port kann einem beliebigen VLAN zugeordnet werden.

### VLAN-Tag

- VLAN-ID im VLAN-Tag wird zur Zuordnung verwendet
- Priority Code Point ermöglicht die Priorisierung gewisser Applikationen
- Discard Eligibility Indicator 0 → Frame wird bei Engpässen zuerst verworfen

Trunk = Tagged, Access = Untagged



## Spanning Tree (Redundanz-Protokoll)

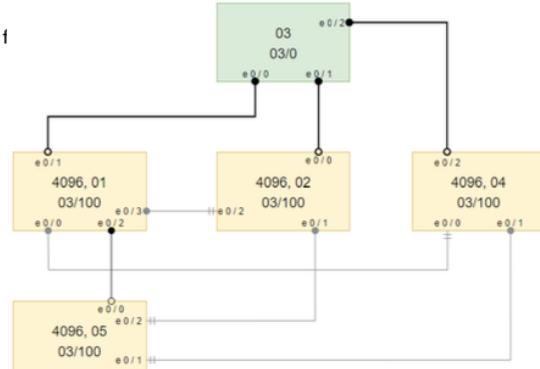
Beim Spanning-Tree werden von redundanten Pfaden alle außer einer gesperrt. Im Fehlerfall wird f

Der Algorithmus bestimmt eine Root-Bridge, von welcher aus dem Baum aufgespannt wird.

- Alle Knoten werden genau einmal verbunden
- Verbindungen, die zu Schleifen führen werden gesperrt
- Die Auswahl der Root-Bridge ist vom *Bridge-Identifier* abhängig
- Der *Bridge-Identifier* besteht aus einer wählbaren Priorität und der MAC-Adresse

### Vorgehen

1. Root bestimmen mittels *Bridge-Identifer* (Priorität, MAC-Adresse)
2. Direkt angeschlossene Bridges bestätigen (verbinden)
3. Weitere Verbindungen abhängig von Kosten und *Bridge-Identifer* eintragen



## Ethernet-Systeme

- *Autonegotiation* Ermittlung der besten Betriebsart durch Austausch der Leistungsmerkmale zweier Netzwerkkomponenten.
- *Link Pulses* NLP = Link Presence Detection  
FLP = Autonegotiation, Autopolarity

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

## Merkmale von Bridges

Anzahl Ports	Steckergroesse ist im Extremfall die Limitierung
Adressabelle	Wie viele Stationen können im LAN existieren
Filtrate	Maximale Frames / s / Port (Empfangsrichtung)
Transferrate	Maximale Frames / s / Port (Senderichtung)
Backplane / Fabric Kapazität	Maximaler Gesamtdurchsatz zwischen allen Ports
Architektur	<b>Store-and-Forward:</b> Frame wird komplett empfangen und dann weitergeleitet <b>Cut-Through:</b> Frame wird schon nach Decodierung der Zieladresse weitergeleitet Leitet auch korrupte Frames weiter, in der Regel aber kein Problem <b>Adaptive Cut-Through:</b> Schaltet bei hohen Fehlerraten automatisch auf Store-and-Forward um
Konfigurierbarkeit	Unmanaged (keine Möglichkeit z.B. VLANs einzurichten) oder Managed (via Konsole oder Web Interface)
Energieverbrauch	Wird zunehmend wichtiger in Data Center Anwendungen

Herausforderungen bei Vervielfachen der Datenrate:

- (1) CSMA/CD: Maximale Segmentgrösse umgekehrt proportional zur Datenrate.
- (2) Baudrate: Höhere Datenrate = Höhere Baudrate / Bandbreite  
Dämpfung nimmt zu
- (3) Heterogene Datenraten: Möglichst keine Konfiguration systemrelevanter Parameter beibehalten

Lösung : 100BASE-TX

### 100BASE-TX:

#### CSMA/CD

- CSMA/CD ist unterstützt
- VollDuplex
  - Optisch (100BASE-FX)

Segmentlänge 200 m (einschließlich 2 Hubs)  
100 m Kabellänge  
MM Faser 2000 m, SM bis 40 km

#### Baudrate / Bandbreite

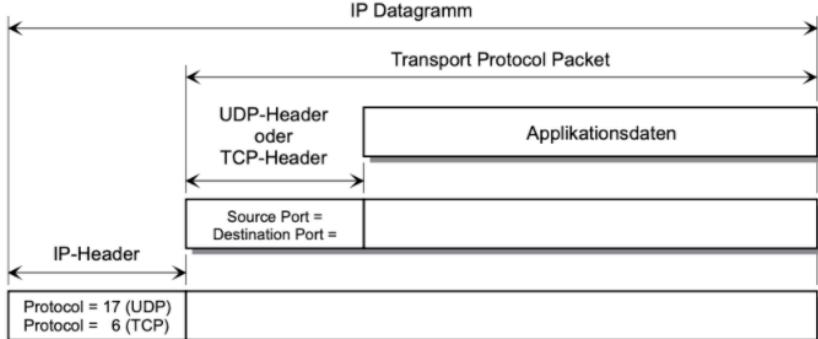
- CAT3 Kabel nicht mehr unterstützt (16 MHz Bandbreite ungünstig für 125 MBaud)
- MLT-3 und 4B5B Codierung reduziert höherfrequente Anteile und erlaubt Verwendung von CAT5

#### Kompatibilität und Unterstützung von Systemen mit heterogenen Datenraten

- Einführung Autonegotiation mithilfe von Fast Link Pulsen
- Definition geschwindigkeitsunabhängiger Schnittstellen (MII - Media Independent Interface)
- Sonst alles gleich



## 8 Transport Layer



<b>UDP</b> dient dem <i>Multi- und Demultiplexen</i> der Datagramme zu den Applikationen.		<b>TCP-Header</b>																																				
<ul style="list-style-type: none"> <li>• Verbindungslos</li> <li>• Unzuverlässig</li> </ul>		<table border="1"> <tr> <td>Min length = 20 Byte / Max Length = 60 Byte</td> </tr> </table>	Min length = 20 Byte / Max Length = 60 Byte																																			
Min length = 20 Byte / Max Length = 60 Byte																																						
<table border="1"> <tr> <td>Message-Length (in Byte)</td> <td>Header + Daten = min. 8 B</td> </tr> </table>		Message-Length (in Byte)	Header + Daten = min. 8 B	<ul style="list-style-type: none"> <li>• Sequence-Nr. Nummer zur Ordnung der Segmente</li> </ul>																																		
Message-Length (in Byte)	Header + Daten = min. 8 B																																					
<table border="1"> <tr> <td>1. Byte</td> <td>2. Byte</td> <td>3. Byte</td> <td>4. Byte</td> </tr> <tr> <td>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</td> <td>UDP Source Port</td> <td>UDP Destination Port</td> <td></td> </tr> <tr> <td>UDP Message Length</td> <td>Checksum</td> <td></td> <td></td> </tr> <tr> <td></td> <td>Data</td> <td></td> <td></td> </tr> </table>		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	UDP Source Port	UDP Destination Port		UDP Message Length	Checksum				Data			<ul style="list-style-type: none"> <li>• Acknowledgement-Nr. <math>n + 1 \rightarrow</math> Daten korrekt und vollständig</li> </ul>																				
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	UDP Source Port	UDP Destination Port																																				
UDP Message Length	Checksum																																					
	Data																																					
<b>System Ports (Well-Known)</b> <b>User Ports (Registered)</b> <b>Dynamic / Private Ports</b>		<ul style="list-style-type: none"> <li>• Data Offset Gibt an wo Daten beginnen / enden</li> </ul>																																				
<b>Feste Port-Nummern</b> , für bekannte Appl. reserviert Reservierter Bereich für herstellerspezifische Appl. <b>Frei verfügbare Ports</b>		<ul style="list-style-type: none"> <li>• ECN-Flags Explicit Congestion Notification</li> </ul>																																				
<table border="1"> <tr> <td>System Ports</td> <td>User Ports</td> <td>Dynamic Ports</td> </tr> <tr> <td>0 - 1023</td> <td>1024 - 49'151</td> <td>49'152 - 65'535</td> </tr> </table>		System Ports	User Ports	Dynamic Ports	0 - 1023	1024 - 49'151	49'152 - 65'535	<ul style="list-style-type: none"> <li>• Control Bits URG, ACK, PSH, RST, SYN, FIN</li> </ul>																														
System Ports	User Ports	Dynamic Ports																																				
0 - 1023	1024 - 49'151	49'152 - 65'535																																				
<ul style="list-style-type: none"> <li>• LISTEN Auf Anforderung warten</li> <li>• SYN-SENT Anforderung geschickt</li> <li>• SYN-RECEIVED Anforderung erhalten</li> <li>• ESTABLISHED Verbindung besteht</li> </ul>		<ul style="list-style-type: none"> <li>• Window Verfügbare Puffergrösse</li> </ul>																																				
<ul style="list-style-type: none"> <li>• SYN Verbindungsaufbau</li> <li>• ACK Paket bestätigen</li> <li>• FIN Verbindungsabbau</li> </ul>		<ul style="list-style-type: none"> <li>• Urgent Pointer URG = 1 <math>\rightarrow</math> Position der wichtigen Daten</li> </ul>																																				
<ul style="list-style-type: none"> <li>• FIN-WAIT-1 Abbauforderung gesickt</li> <li>• FIN-WAIT-2 Abbauforderung bestätigt</li> <li>• CLOSE-WAIT Auf Lokale Verbindung warten</li> <li>• LAST-ACK Verbindungsabbau bestätigt</li> <li>• TIME-WAIT Letzte Bestätigung gesendet</li> </ul>		<ul style="list-style-type: none"> <li>• Options Häufigste Verwendung: MSS</li> </ul>																																				
<b>Verbindungsaufbau</b>		<table border="1"> <tr> <td>1. Byte</td> <td>2. Byte</td> <td>3. Byte</td> <td>4. Byte</td> </tr> <tr> <td>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</td> <td>TCP Source Port</td> <td>TCP Destination Port</td> <td></td> </tr> <tr> <td></td> <td>Sequence Number</td> <td></td> <td></td> </tr> <tr> <td></td> <td>Acknowledgement Number</td> <td></td> <td></td> </tr> <tr> <td>Header Length</td> <td>unused</td> <td>ECN Control Bits</td> <td>Window</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td>Checksum</td> <td></td> <td>Urgent Pointer</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td>Options / Padding</td> </tr> </table>	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	TCP Source Port	TCP Destination Port			Sequence Number				Acknowledgement Number			Header Length	unused	ECN Control Bits	Window						Checksum		Urgent Pointer								Options / Padding
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	TCP Source Port	TCP Destination Port																																				
	Sequence Number																																					
	Acknowledgement Number																																					
Header Length	unused	ECN Control Bits	Window																																			
	Checksum		Urgent Pointer																																			
			Options / Padding																																			
<b>Datenaustausch</b>		<b>Daten werden zur Sequenznummer addiert</b>																																				
<b>Verbindungsabbau</b>																																						

### 8.1 TCP

Zuverlässig & verbindungsorientiert  $\rightarrow$  sichert Reihenfolge & Schutz vor Datenverlust

#### Wichtige Merkmale:

- o Verbindungsorientierte Übertragung: Verbindung muss aufgebaut werden
- o Hohe Zuverlässigkeit: Schutz vor Datenverlust, Reihenfolge bleibt
- o Vollduplex Übertragung: Gleichzeitige Übertragung in beide Richtungen
- o Stream Schnittstelle: Anwendung sender/empfängt unstrukturierte Bytefolge
- o Eleganter Verbindungsabbau: Gewährt Zustellung aller Daten auch beim Verbindungsabbau.

#### Sicherstellung der Zuverlässigkeit

(1) Retransmission: Empfänger schickt Acknowledgement an Sender, falls erfolg. Falls Sender nach Ablauf des Timers kein Ack hat, sendet er nochmals.

(2) Adaptive Wartezeit: Wartezeit vor dem Neusenden wird mit Roundtrip Time berechnet (RTT)  
Wartezeit bis Neuübertragung  $\rightarrow$  Retransmission Time out (RTO)

Überlast des Empfängers: Fluss-Steuerung		Fluss-Steuerung bei TCP	
TCP verwendet den <i>Sliding-Window Mechanismus</i> . Beide Seiten einen Buffer (Window).			
<ul style="list-style-type: none"> <li>- Verhindert den Datenüberlauf</li> <li>- Ineffiziente Nutzung der Netzbandbreite</li> </ul>		<ul style="list-style-type: none"> <li>- Verhindert den Datenüberlauf</li> <li>- Höhere Durchsatzrate</li> <li>- Feste «Fenster-Größe»</li> </ul>	
<b>Überlast des Netzes: Congestion Control</b>		<p>bei Ankunft eines Segments gibt der Empfänger in der Bestätigung das <u>restliche Window</u> an → Window Advertisement</p>	
<p>TCP benutzt den Paketverlust als Masseneinheit für Überlastung und reagiert durch Absenken der Übertragungsrate (<i>Slow Start</i>). Dadurch kann die Überlastung überwacht und verhindert werden.</p> <p>Hierfür pflegt jeder Sender zwei Fenster (vom Sender gewährtes Fenster, Überlastungsfenster). Das Minimum der Fenster stellt die Anzahl Bytes dar, die gesendet werden können.</p>			
<b>Erkennung von verlorengegangenen Telegrammen (Round Trip Time)</b>		<p>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 <i>Round-Trip Time (RTT)</i>.</p> <p>Gewichteter Mittelwert <i>SRTT (Smoothed Round-Trip Time)</i>  <math>\alpha = 0.125: SRTT_n = (1 - \alpha) \cdot SRTT_{n-1} + \alpha \cdot RTT_n</math>  <i>Streuung RTTVar des SRTT</i> der Abweichungen  <math>\beta = 0.25: RTTVar_n = (1 - \beta) \cdot RTTVar_{n-1} + \beta \cdot  SRTT_n - RTT_n </math>  <i>Retransmission Time-Out RTO</i>  <math>RTO_n = SRTT_n + 4 \cdot RTTVar_n</math></p>	

## 8.2 Autoconfig bei IPv4 (APIPA)

Ohne DHCP-Server muss sich ein Client selbst seine IPv4-Adresse geben. Diese wird im 169.254.0.0/16-Netz zufällig gewählt und dann via Gratuitous ARP-Request überprüft, ob sie schon vergeben ist.

→ nach dem booten kennt ein Client typischerweise nur seine MAC-Adresse → DHCP

→ falls **kein DHCP-Server** gefunden wird:

1. Client wählt eine IP-Adresse im Bereich von **169.254.1.0 bis 169.254.244.255** → Klasse B
2. Alle 5 Minuten wird ein DHCP-Request abgesetzt

Der Router in so einem Netzwerk muss **Proxy-ARP** aktiviert haben. Dadurch gaukelt er dem Client vor, dass alle Geräte auf der ganzen Welt im lokalen Netz sind, in dem er alle *ARP-Requests* an externe Geräte selbst beantwortet.

### Gratuitous ARP

Ein Gratuitous ARP-Request wird verwendet, um **IP-Adresskonflikte** zu erkennen. Dazu versendet der Knoten nach jeder Adresszuweisung beim Booten oder bei Änderungen der IP-Adresse einen ARP-Request für seine eigene IP-Adresse. Falls diese IP-Adresse schon vergeben ist, also ein Konflikt besteht, wird der anfragende Knoten einen ARP-Reply empfangen. Keine Antwort bedeutet also, dass (zurzeit) kein Konflikt besteht.

Ein Gratuitous ARP-Reply wird von einem Knoten ebenfalls, nachdem **Setzen oder Ändern der IP- Adresse** verschickt, aber mit dem Zweck, die **ARP-Cache** der anderen Knoten zu berichtigen. Dazu wird - ohne dass ein Request vorliegt - ein ARP-Reply mit der eigenen IP-/MAC-Adress-Paarung verschickt. Beim Gratuitous ARP-Reply wird ebenfalls ein **Broadcast** verwendet, damit alle Knoten adressiert werden. Es gibt jedoch **keine Gewähr**, dass die Knoten einen Gratuitous ARP auch verwerten.

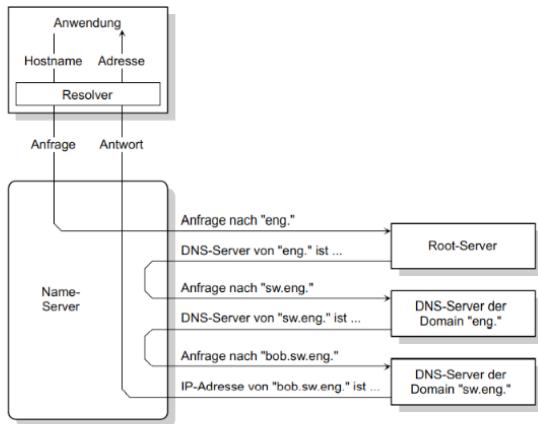
# 9 Application Layer

## Domain Name Space (DNS)

- Leserliche Darstellung von IP-Adressen
- Hauptdomäne = Root

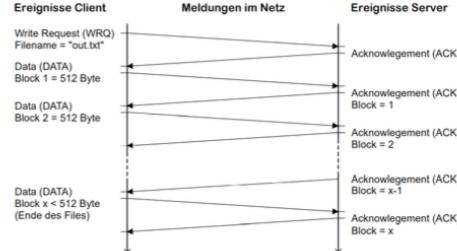
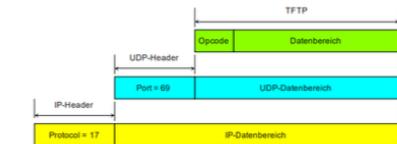
### Beispiel

- *bob.sw.eng.* Fully Qualified Domain Name
- Root
- *eng* Top Level Domain
- *sw* Second Level Domain



## Trivial File Transfer Protocol (TFTP)

- Basiert auf UDP



## Hypertext Transfer Protocol (HTTP)

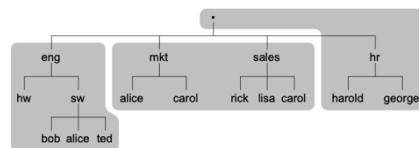
- WWW basiert auf HTTP

### Funktionsweise von HTTP

- Basiert auf TCP, Port 80
- ASCII-Basiert, MIME-Typen, Codierungen
- Transaktionsbasiert: HTTP Request → HTTP Response

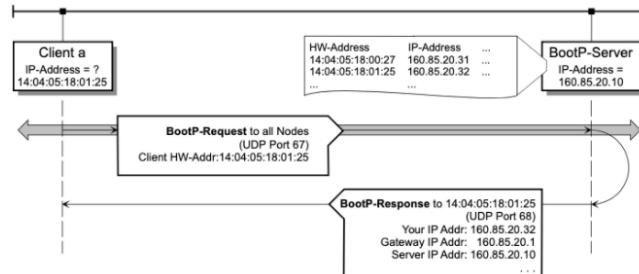
## 9.1 Name-Server

- verantwortlich für meistens 1 Zone
- Mehrere Name-Server pro Zone = Redundanz
- ➔ Jede Zone muss von einem Master-Name-Server bedient werden



## 9.2 BootP

- ➔ Wie erhält ein Knoten seine IP-Adresse in einem Netzwerk?



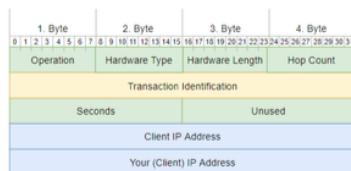
baut auf UDP auf:

- > UDP Port 67: Kommunikation mit BootP-Server
- > UDP Port 68: Kommunikation mit dem Client

-> Server kann auch in anderem IP-Subnetz sein (UDP)

## BOOTP

- Manuelle Verwaltung
- Heimanwender sind überfordert
- Statische Adresszuordnung



## Dynamic Host Configuration Protocol (DHCP)

- Paketformat identisch zu BOOTP
- Dynamische Zuweisung von IP-Adressen
- Reserviert nur IP's von aktiven Geräten

## Ablauf (DHCP)

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

## Simple Mail Transfer Protocol (SMTP)

Standard-Protokoll zum Versenden oder Weiterleiten von E-Mails. Es können nur ASCII-Zeichen versendet werden. Für weitere Zeichen wird MIME verwendet.

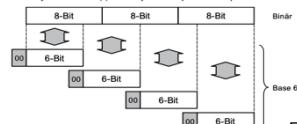
### MIME-Standard (Multipurpose Internet Mail Extension)

Ermöglicht eine Codierung zu wählen, um auch nicht-ASCII-Zeichen zu versenden.

- Maximale Zeilenlänge = 76 Zeichen
- «B»-Encoding (Base64)
- Beispiel: Züri → WvxyaQ==
- echo «Text» / base64

Base 64 Encoding  
resultierende Werte liegen zwischen 0..63

Gruppen von 3 Bytes → 4 Gruppen mit je 6 Bits (RFC 2045)



Binär

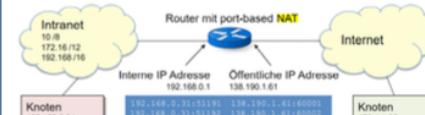
Base 64

## Network Address Translation (NAT)

- NAT (Historisch)
- NAPT (Port Trans.)

Sicherheit durch „Verstecken“ von lokalen Adressen  
Lokale IP-Adresse → Öffentliche IP-Adresse

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



Intranet (privates Netz)	Internet (öffentliche Netze)
Quell-Adresse: 192.168.0.31	Ziel-Adresse: 138.190.1.61
Port: 51991	Port: 60001
170.1.1.25	170.1.1.25
80	80
Quell-Adresse: 192.168.0.31	Ziel-Adresse: 138.190.1.61
Port: 51992	Port: 60002
170.1.1.25	170.1.1.25
443	443
Quell-Adresse: 192.168.0.32	Ziel-Adresse: 138.190.1.61
Port: 51991	Port: 60003
170.1.1.25	170.1.1.25
25	25

Anzahl öffentlicher Adressen kann reduziert werden  
Port Mapping hilft, um Pakete von außerhalb des Netzes auf einen Host innerhalb weiterzuleiten.  
Dabei speichert der Gateway den Port und die interne Adresse eines Hosts um diese später aufzulösen.  
Pro Portnummer kann es nur ein lokaler Host geben.

### Problem

1. Datenverschlüsselung unterhalb des Transport-Layers ergibt Probleme
2. Ist NAT nicht Port basierend so muss die Kommunikation vom

## 9.3 NAT

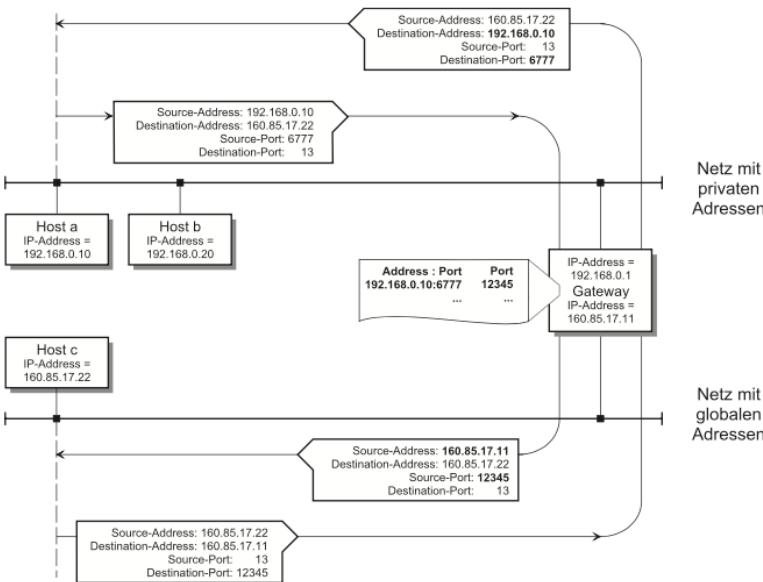
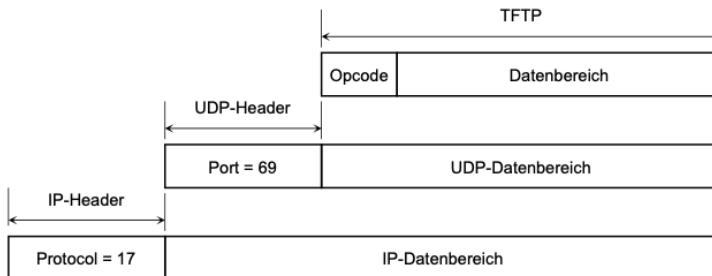


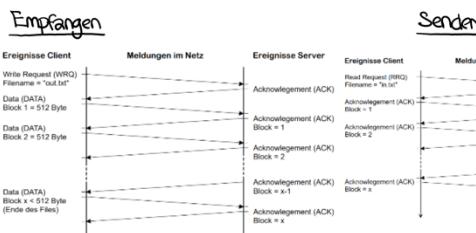
Abbildung 9.12: Beispiel einer Network Address and Port Translation

## 9.4 TFTP

- Protokoll zur Datenübertragung
- Verwendet UDP



- **Read Request (RRQ):** Anfordern einer Datei vom anderen Host, es wird eine TFTP-„Verbindung“ aufgebaut
- **Write Request (WRQ):** Senden einer Datei zum anderen Host, es wird eine TFTP-„Verbindung“ aufgebaut
- **Acknowledgement (ACK):** Bestätigung eines korrekt empfangenen WRQ oder DATA Pakets
- **Data (DATA):** Übermittlung der eigentlichen Daten, immer in Blöcken von 512 Bytes, ein kürzerer Datenblock beendet die Übertragung, Blöcke sind fortlaufend nummeriert
- **Error (ERROR):** Ein Fehler ist aufgetreten (siehe Error Code), TFTP-Verbindung wird beendet



## 9.5 SMTP (TCP)

### Aufbau der Verbindung

- gesicherte Verbindung über TCP Port 25

- Der Sender öffnet eine TCP-Verbindung zum Empfänger.
- Der Empfänger identifiziert sich gegenüber dem Sender.
- Der Sender identifiziert sich gegenüber dem Empfänger.
- Der Empfänger akzeptiert die Identifikation des Senders.

### Abbau der Verbindung

Der Verbindungsabbau durch den SMTP-Sender erfolgt in zwei Schritten: Nach Übertragung des QUIT-Kommandos wird auf eine Antwort (Reply) des Empfängers gewartet. Anschließend wird ein TCP-CLOSE für die Verbindung initiiert. Unmittelbar nach Empfang des QUIT-Kommandos baut der Empfänger die TCP-Verbindung ab.

### Beispiel:

```
Sender: Quit
Empfänger: Bye-Bye
Empfänger: <TCP-Close>
Sender: <TCP-Close>
```

## 9.6 Mime-Standard

### Quoted Printable Encoding

Jeder 8 Bit Wert wird durch 3 ASCII Zeichen ersetzt. (G → =F0)

Da nur 7 Bit ASCII erlaubt, werden Zeichen im Bereich 128-255 umcodiert

### BASE64 Encoding

Für rein binäre Daten besser geeignet. 3 Byte werden 4 Blöcke à 6 Bit aufgelegt. Somit werden binäre Werte in ASCII umgedeutet

Welche Datenmenge muss übertragen werden für ein 793815 Byte grosses JPEG-File?

Es wird Base64 Encoding verwendet. Aus drei Bytes werden vier Zeichen gebildet. Also rechnet sich die Datenmenge zu  $793815 * 4/3 = 1058420$  Byte.

Wegen dem zeilenorientierten Aufbau: pro Zeile à 76 Zeichen und auch für die angebrochene Zeile braucht es zusätzlich je 2 Zeichen (CR, LF), in unserem Fall also  $2 * 13927$  Zeichen. So kommt man auf total  $1058420 + 2 * 13927 = 1086274$  Zeichen

### ○ Content-Transfer-Encoding:

- Definiert, wie codiert wird
- Quoted-Printable:
  - Nur Nicht-ASCII-Zeichen werden durch = + zwei Hex-Ziffern codiert
  - Effizient bei Text, der fast nur aus ASCII besteht
- Base64:
  - 3 Bytes werden in vier Gruppen à sechs Bit aufgeteilt.
  - Am Schluss mit = aufgefüllt, falls es nicht aufgeht.
  - Effizient bei Binär-Daten