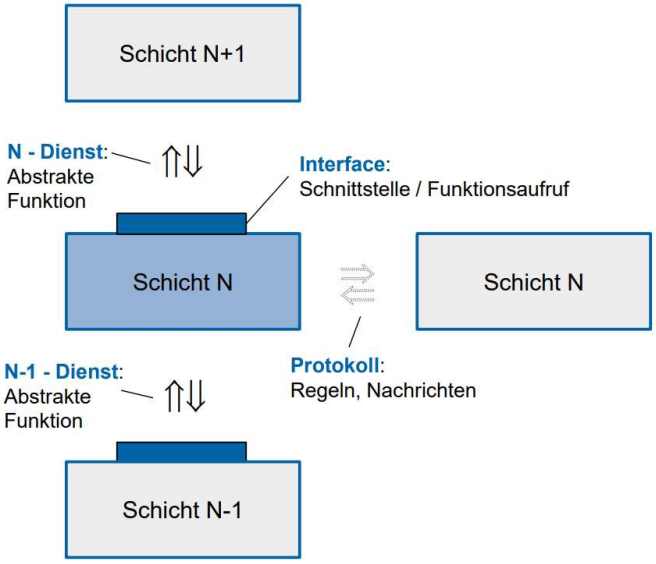
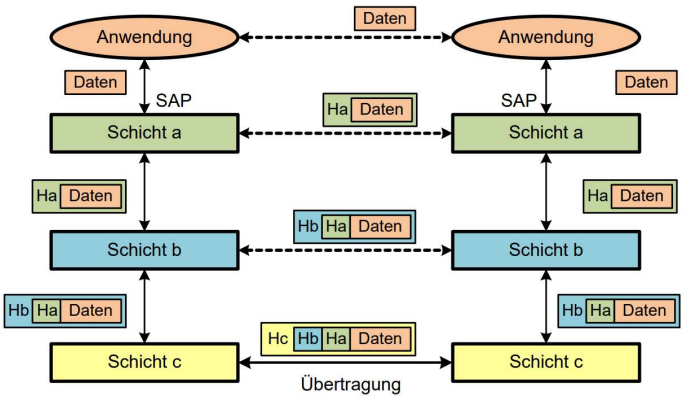
OSI Referenzmodell

Kommunikationsschicht N

- bietet der höheren Schicht N+1 einen Interface *Dienst* an. (nur der Schicht N+1)
- verwendet zu Erfüllung ihrer Aufgaben den *Dienst* der Schicht N-1. (nur der Schicht N-1)
- kommuniziert mit der korrespondierenden Schicht N über ein *Protokoll*.



Datenübertragung in einem Schichtenmodell



Zuverlässiger & unzuverlässiger Dienst

Zuverlässiger Dienst: Es gehen grundsätzlich keine Daten verloren.

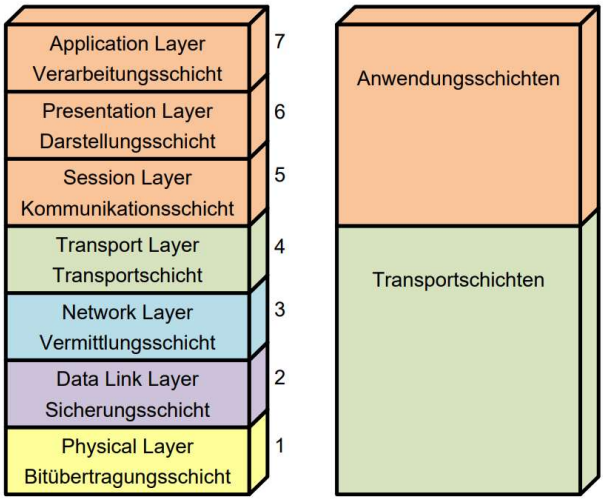
Unzuverlässiger Dienst: Es können Daten verloren gehen.

Verbindungsorientierter & Verbindungsloser Dienst

Test

test

Aufgabe der OSI-Schichten

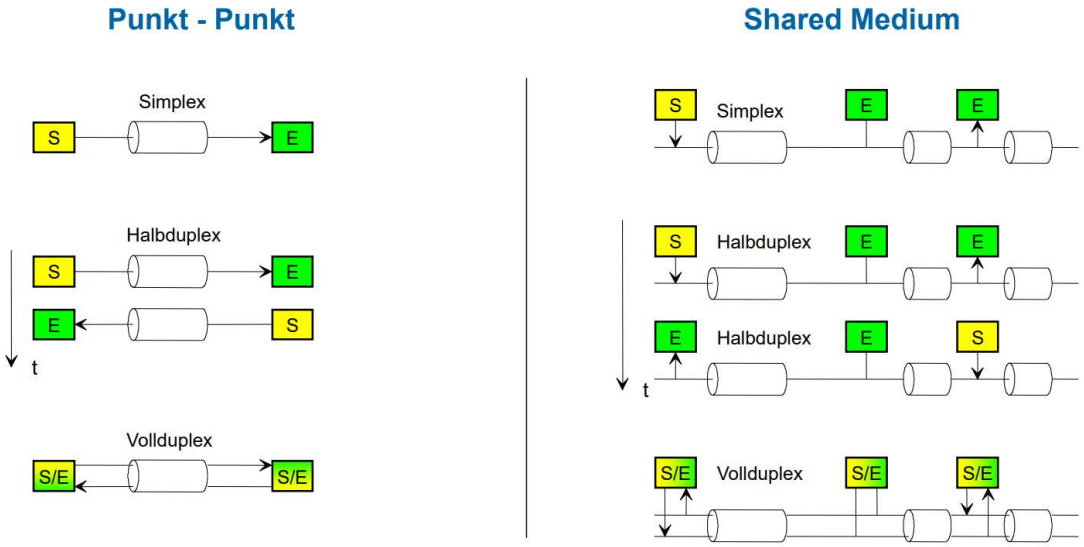


1. Physical Layer
   i. Verbunden mit dem Übertragungsmedium.
   ii. Codiert oder Encodiert die Daten / elektrische Signale.
2. Data Link Layer
   i. Framing: Verpacken / Auspacken von Datenblöcken
   ii. Fehlererkennung und -korrektur
   iii. Fluss-Steuerung
   iv. Adressierung
3. Network Layer (IP)
   i. Kontrolle und gezielte Lenkung von Verkehrsströmen
4. Transport Layer (TCP, UDP)
   i. Kommunikationsphasen
      a. Verbindungsaufbau
      b. Datenaustausch
      c. Verbindungsabbau
   ii. Reihenfolge der Daten

5. Session Layer
   i. Auf- und Abbau einer Session
6. Presentation Layer
   i. Umwandlung der Darstellung von Daten
   ii. Konvertierung von ASCII und Unicode
7. Application Layer
   i. Anwendung für den Nutzer

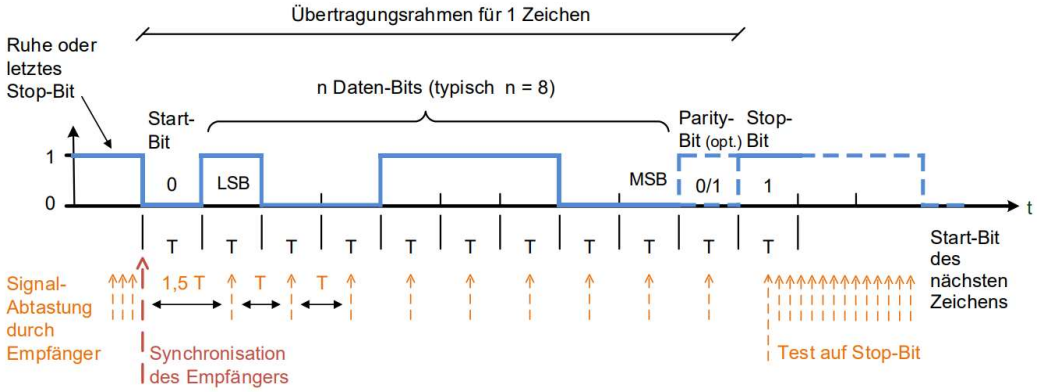
Physical Layer

Verkehrsbeziehung & Kopplung



Serielle asynchrone Übertragung

Die Daten werden einfach geschickt, der Empfänger ist zuständig für das richtige Abschätzen des Taktes.



Folgendes muss beachtet werden:

1. Start-Bit / Stopp-Bit gehören nicht zu den Daten-Bits
2. Parity-Bit ist optional

Wir empfangen: 1001 1100 → **lesen beginnend mit MSB**: 0011 1001b = 0x39; ASCII Code 57 = «9».

Clock Drift, Real-World Beispiel

Angaben:

- Max. Framegrösse Ethernet: 1500 Bytes
- Genauigkeit Oszillatoren: $\pm$50ppm → Fehler von 0.00005. Worst-Case Szenario wäre der Sender würde einen Fehler von -50ppm und der Empfänger +50ppm aufweisen.

Frage: Können in diesem Fall die Daten sicher abgetastet werden?

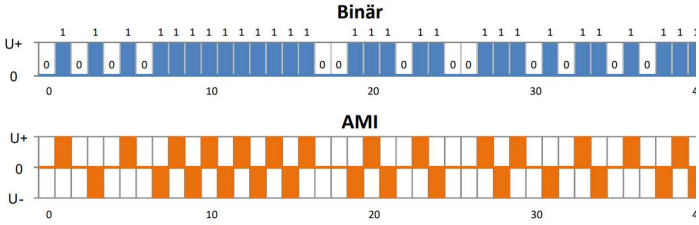
Antwort:

- $1500\ \text{Bytes} * 8\ \text{Bit/Byte} = 12'000\ \text{Bit}$; 100ppm Differenz zwischen Sender & Empfänger $= 10^{-4}$
- Pro Bit entsteht so ein Fehler von 10^{-4} Bit Zeiten T_{Bit}.
- Die Abweichung ist somit $1.2 * 10^4 * 10^{-4} \frac{T_{\text{Bit}}}{\text{Bit}} = 1.2\ T_{\text{Bit}}$
- Eine fehlerfreie Abtastung ist nicht mehr möglich (ohne weitere Massnahmen).

Serielle synchrone Übertragung

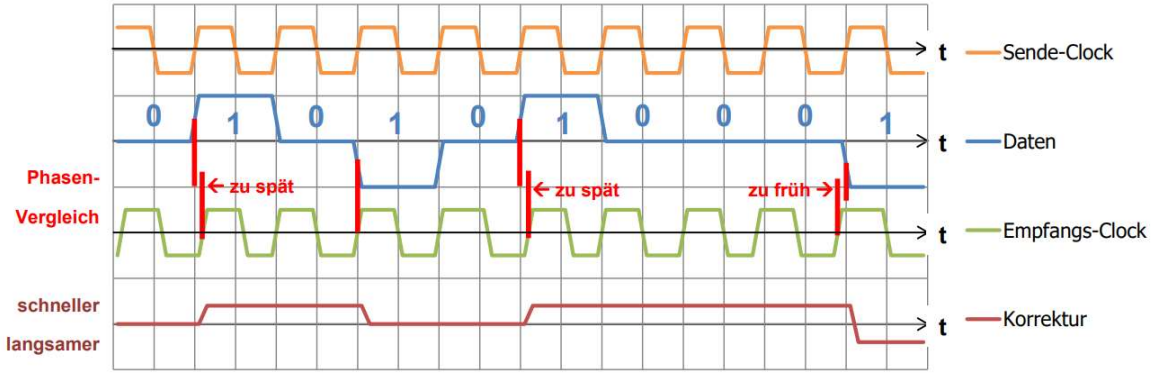
Entweder wird parallel die Daten mit dem Takt geschickt, besser ist der Takt mit den Daten zu codieren. Somit wird nur eine Leitung benötigt.

Beispiel Leitungscode AMI



Nachteil: Wenn viele Nullen geschickt werden, kann der Takt beim Empfänger nicht rekonstruiert werden.

Prinzip der Taktrückgewinnung



Beispiel Leitungscode PAM3

Hierbei wird das Problem bei AMI berücksichtig indem für alle 4 Bit unterschiedliche Codierungen existieren, je nach Situation wird eine andere Codierung verwendet, damit es nicht zu viele 0en an einem Stück hat.

MMS 43 coding table[1]

Input		Accumulated DC offset			
Hex	Binary	1	2	3	4
0	0000	+ 0 + (+2)	0–0 (–1)		
1	0001	0 – + (+0)			
2	0010	+ – 0 (+0)			
3	0011	0 0 + (+1)		– – 0 (–2)	
4	0100	– + 0 (+0)			
5	0101	0 + + (+2)	– 0 0 (–1)		
6	0110	– + + (+1)		– – + (–1)	
7	0111	– 0 + (+0)			
8	1000	+ 0 0 (+1)		0 – – (–2)	
9	1001	+ – + (+1)		– – – (–3)	
A	1010	+ + – (+1)		+ – – (–1)	
B	1011	+ 0 – (+0)			
C	1100	+ + + (+3)	– + – (–1)		
D	1101	0 + 0 (+1)		0 – – (–2)	
E	1110	0 + – (+0)			
F	1111	+ + 0 (+2)	0 0 – (–1)		

Datenrate, Bandbreite und Baudrate

Begriffe

(Leitungs-)Symbol:	physikalisches Signal, das mit einer bestimmten Rate seinen Wert (Amplitude) verändert.
Bit:	Informationsgehalt (des Symbols / der Nachricht), es gilt: $N_{Bit} = \log_2(Anzahl)$
Zeichen:	Einheit der übertragenen Daten, z.B. ASCII Zeichen
Bitrate:	= Datenübertragungsrate / = Durchsatz, Bit pro Sekunde
Baudrate:	= Schrittgeschwindigkeit, (Leitungs-)Symbole pro Sekunde
Zeichenrate:	Anzahl übertragene ASCII Zeichen pro Sekunde

Beispiel Bitrate / Baudrate

Beispiel ASK-4:
4-wertige Symbole, die sich nur in der Amplitude unterscheiden.

Baudrate:

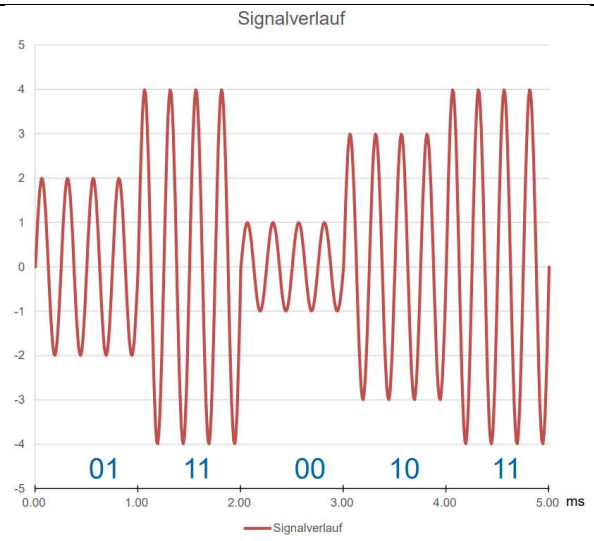
$$\frac{1\ \text{Symbol}}{1 * 10^{-3}\text{s}} = 1000\ \text{Baud} = 1\ \text{kBaud}$$

Bit pro Symbol;

$$2\ \text{Bit/Symbol}$$

Bitrate:

$$1\ \text{kBaud} * 2\ \text{Bit/Symbol} = 2\ \text{kBit/s}$$



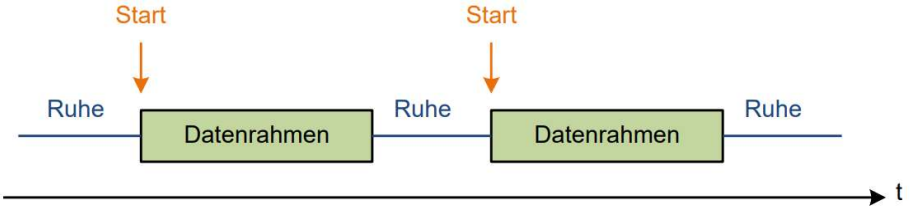
Data Link Layer

Die Aufgabe des Data Link Layer sind:

- Die Realisierung einer zuverlässigen Verbindung zwischen direkt miteinander verbundenen Systemen.
- Einpacken der zu senden Nutzerdaten in Frames.
- Entpacken der empfangenden Datenblöcke.
- Fluss Steuerung: «langsamer» Empfänger kann «schnellen» Sender bremsen.
- Adressierung der Teilnehmer
- Medium Zugriff: Welche Station darf wann senden.

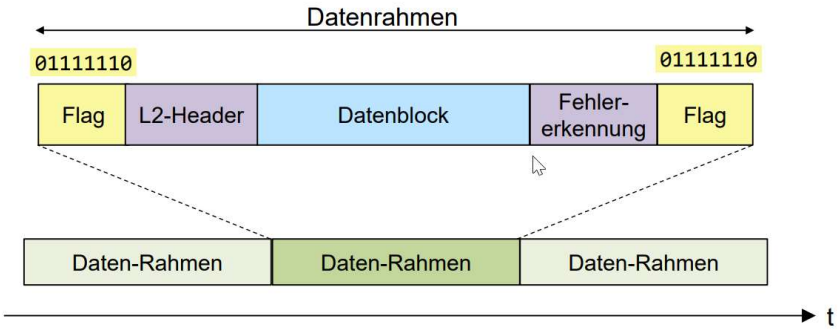
Framing: Asynchrone Übertragung

Der Beginn eines Frame wird mit einem Start-Bit markiert, im Header wird dann die Framelänge begrenzt. Stehen keine Daten zur Übertragung so wird nichts gesendet (**Ruhezustand**). Beispiel:



Framing: Synchrone Übertragung

Frames werden ohne Unterbrechung gesendet. Stehen keine Daten zur Übertragung an, so werden Flags gesendet. Jeder Frame ist mit einem Start-Flag und einem End-Flag versehen. Beispiel:



Wie wird nun verhindert dass, das vordefiniert Flag-Bitmuster in den Daten nicht vorkommt? :

Man *stopft* die gesendeten Daten mit zusätzlichen 0en. Z.B. Flag: 01111110 muss im Datenblock & Header nach jedem 5ten 1 eine 0 *gestopft* werden. Beim lesen werden diese weggeschmissen.

Fehlererkennung / Fehlerkorrektur

Definition BER / FER

BER = Bitfehlerwahrscheinlichkeit ε

$$\text{BER} = 1 \Rightarrow \text{Alle Bits falsch}$$

$$\text{BER} = 0.001 \Rightarrow \text{Jedes 1000. Bit ist falsch}$$

FER = Framefehlerwahrscheinlichkeit

$$\text{FER} \cong N * \varepsilon$$

Fehlererkennung: Hamming-Distanz

Die Hamming-Distanz gibt an wie viel Bits müssen geflippt werden zu einem nächsten gültigen Codewort. Die Grösse der Fehlererkennung ergibt sich aus:

$$\text{Fehlererkennung} = \text{Hamming-Distanz} - 1$$

Beispiele Fehlererkennung

Beispiele wurden nach der stärke ihrer Fehlererkennung eingestuft.

1. 32-bit CRC code
2. Längs- / Querparität
3. 16-bit Check-Summe
4. Odd Parity & Even Parity (RS-232)

Der Unterschied von Längs- / Querparität zu Odd / Even Parity, beim Längs- / Querparität werden ganze Datenblöcke mit Odd oder Even Parity versehen.

Fehlerkorrektur: Hamming-Distanz

Aus der Hamming-Distanz können wir entnehmen die Anzahl korrigierbaren Bitfehler k:

$$k \leq \frac{\text{Hamming-Distanz} - 1}{2}$$

Beispiele Fehlerkorrektur

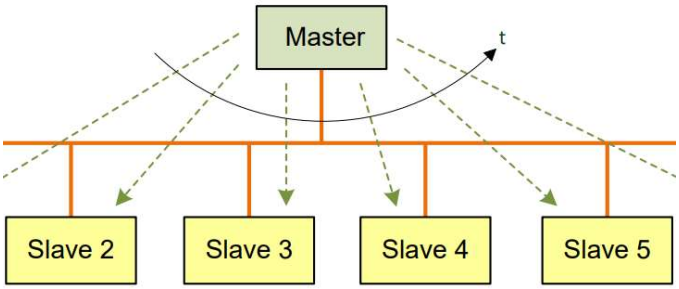
1. Faltungscode
2. Blockcodes
3. Längs- / Querparität

Mit CRC Codes lässt sich keine Fehler korrigieren.

Zugriffsmechanismen

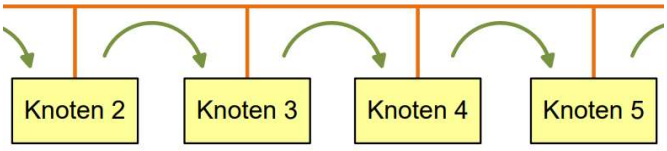
Master-Slave Verfahren

Ein Master sagt wann, welcher Slave reden kann. Wenn dieser aber ausfällt, kann keine Kommunikation stattfinden.



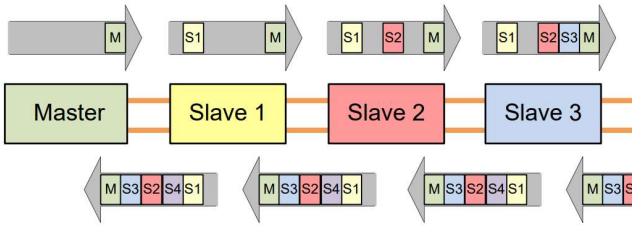
Token Verfahren

Durch einen Token wird definiert wer reden darf, nach einer gewissen Zeit wird dieser Token weitergegeben. Das ganze ist aber eher Aufwändig da jeder Knoten ein solcher Token unterstützen muss.



Alternative: Token Verfahren

Anstelle eines Token wird ein Frame von dem Master geschickt nun kann jeder Slave sein eigenes Pakete anhängen (mit einer korrekten Adresse), beim Weg zurück kann jeder Slave lesen was im Knoten steht.



Zeitsteuerung

Es wird definiert wann welcher Knoten reden darf. Ist aber für das Einrichten auch sehr Aufwändig.

Random Medium Zugriff

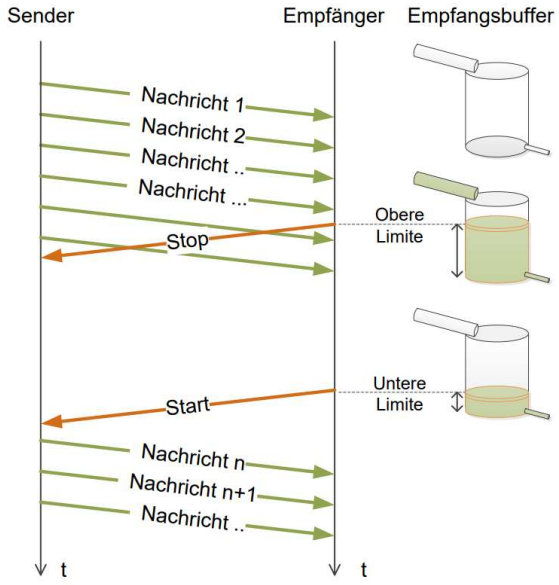
Hier sind alle Knoten gleichberechtigt und haben jederzeit Zugriff auf das Übertragungsmedium. Vor dem Senden wird abgehört ob das Übertragungsmedium frei ist, wenn ja wird gesendet.

Kollisionsbehandlung

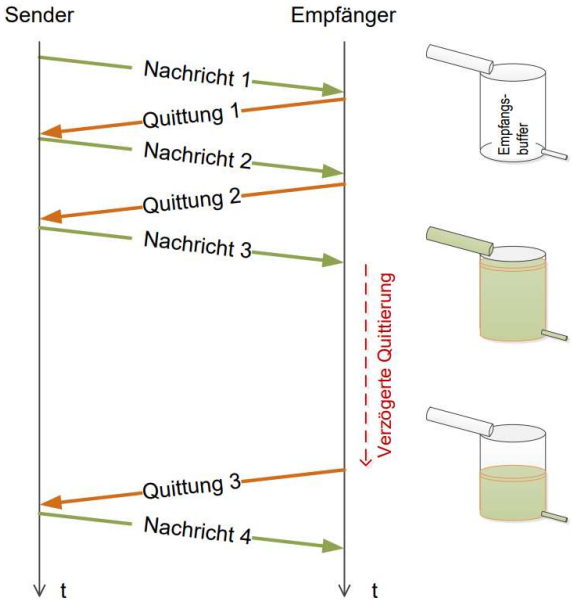
- CSMA/CD (Collision Detection): Kollision entdeckt → Abbrechen und später nochmals versuchen.
- CSMA/CR (Collision Resolution): Erkennt eine Kollision und bricht diese kontrolliert ab.
- CSMA/CA (Collision Avoidance): Prüft ob ein Medium frei ist, sendet erst wenn Medium frei ist.

Flow Control

Explizit Start-Stopp Signalisierung



Implizite Stop & Wait – Protokoll



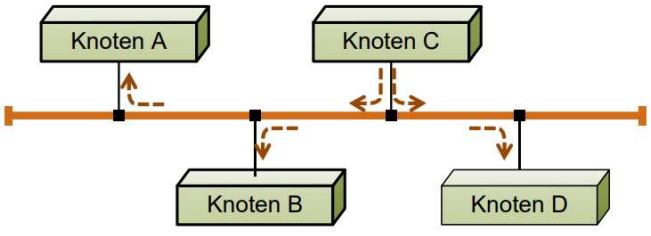
Ethernet

Eigenschaften LAN

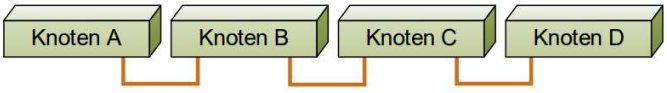
Reichweite:	10m bis wenige km
Datenrate:	100Mbit/s bis 100Gbit/s, typisch heute 1Gbit/s
Verbindet:	Server, Workstation, PCs, Drucker, NAS ...

Topologien

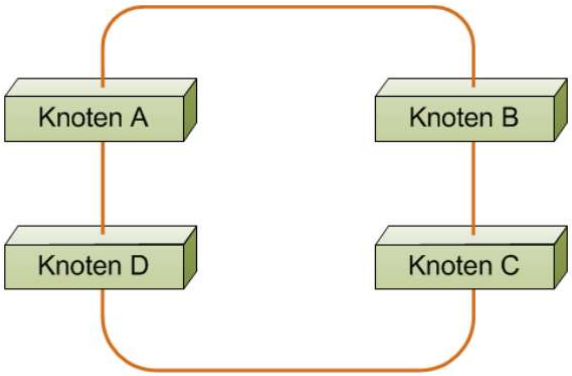
Bus



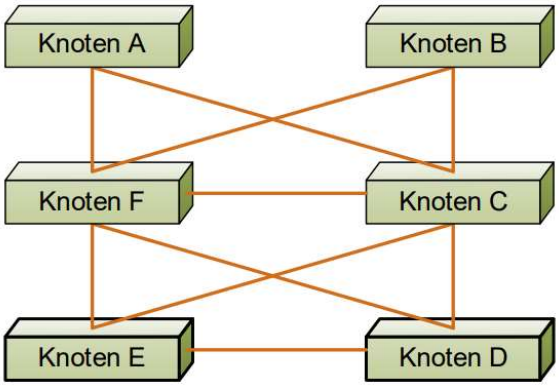
Linie



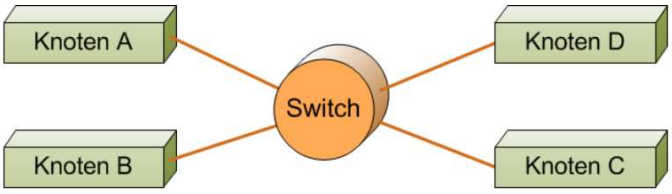
Ring



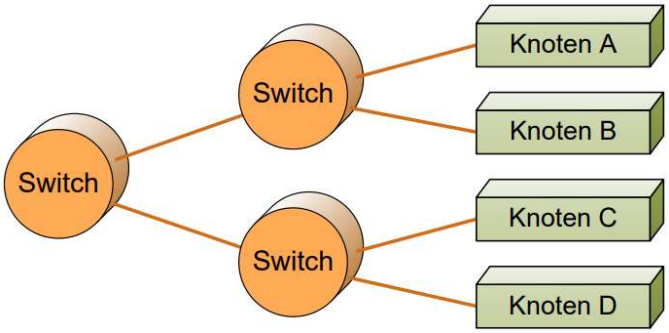
Vermascht



Stern



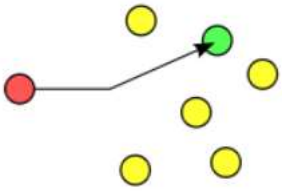
Baum



Übertragungsarten

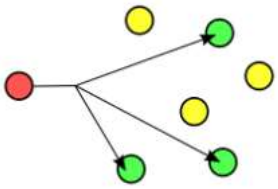
Unicast

Genau ein klar spezifizierter Empfänger. Frame trägt die Adresse dieses Empfängers.



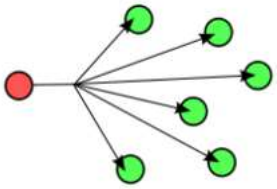
Multicast

Eine Gruppe von Empfängern. Frame trägt die Multicast-Adresse der Gruppe.



Broadcast

An alle Knoten im LAN gerichtet. Frame trägt die Broadcast-Adresse des LAN.



Adressierung in LANs

IEEE MAC Adressen

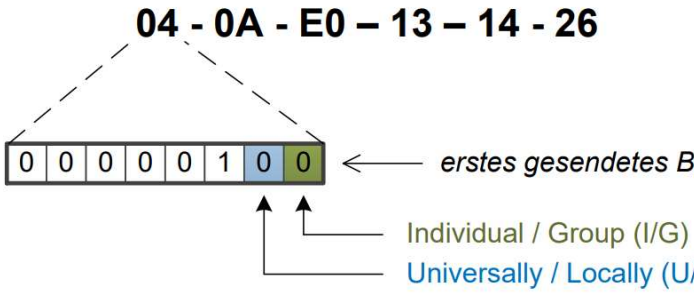
- Werden nicht konfiguriert
- Sind fix einem Interface des Gerätes zugeordnet
- Bestehen aus 6 Bytes
- Darstellung in hexadezimal: **1A-2B-3C-4E-5F-67**

Registrierung globale MAC Adressen (bei IEEE)

- Die ersten **3** Bytes identifizieren den Hersteller «OUI»
- Die letzten **3** Bytes ist die Laufnummer welche von dem Hersteller verwaltet werden.

Klassifizierung MAC Adresse

Bei den MAC Adressen sind auch die Bits vertauscht, aber nur pro Byte. Bedeutet das **erste** geschickte Byte = das **erste** gelesene Byte, aber das **erste** geschickte Bit = das letzte geschickte Bit des Bytes.

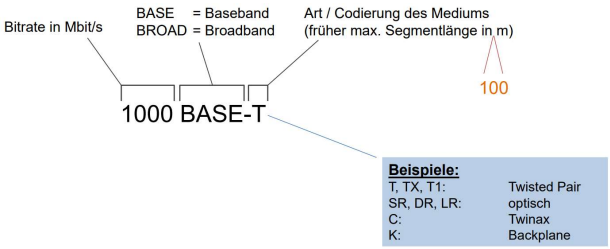


Individual/Group Bit:
- 0 = individual address (Normalfall)
- 1 = group address z.B. Broadcast FF-FF-FF-FF-FF-FF

Universally/Locally Bit:
- 0 = universally administrated address (Normalfall)
- 1 = locally administrated address

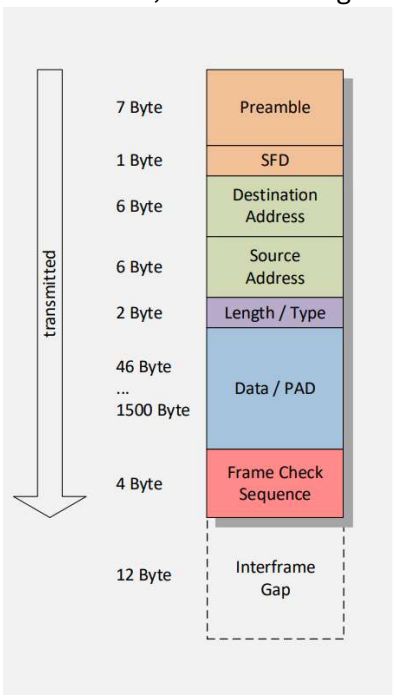
Ethernet Grundlagen / Frameformat

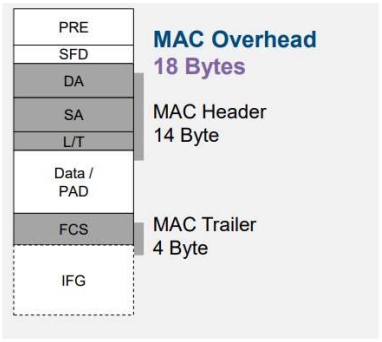
Datenraten:
- 10BASE-T: 10Mbit/s
- 100BASE-TX: 100Mbit/s
- 1000BASE-T: 1Gbit/s

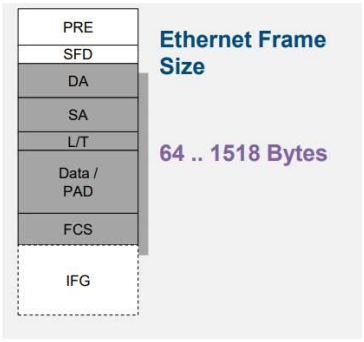


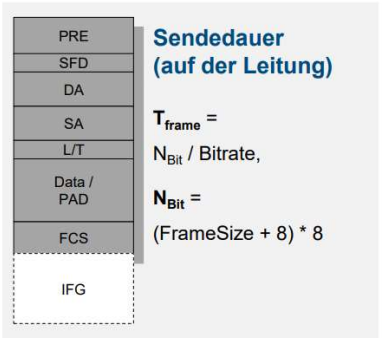
Frameformat

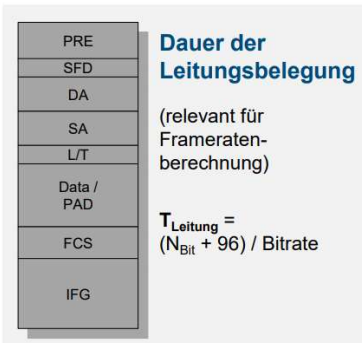
Pro Byte wird immer das niederwertigste Bit **zuerst** und das höchstwertigste Bit **zuletzt** übertragen. Ausnahme bei Zahlenwerte, z.B. beim Length/Type-Feld.











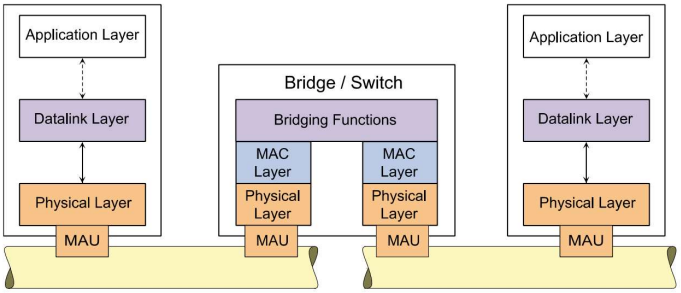
Ethernet-Geräte

Repeater / Hubs

Verstärkt ankommende Signale auf einem Port und leitet sie «in bester Qualität» weiter. (Veraltet)

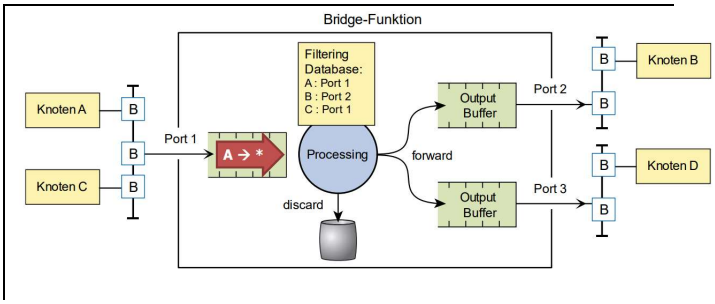
Switch /Bridges

Signale werden auch verstärkt und «weitergeleitet» wie beim Hub, aber prüft zusätzlich Checksummen und kann Layer 2 Adressen auswerten.



Filterung Datenbank

- Im Switch / Bridge verwaltet.
- Mapped MAC Adressen zu Ports.
- Speichert immer nur die **Sender** Adresse zu ihrem Port.
- Bei Unbekannten Empfänger werden alle Ports geflutet.
- Bei Bekannten Empfänger wird direkt weitergeleitet.
- Gespeicherte Adressen werden nach einer eingestellten Zeit wieder gelöscht (**Aging Time**)

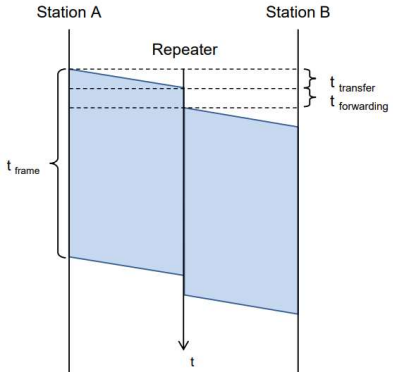


Weg/Zeit-Diagramm für das Senden eines Frames

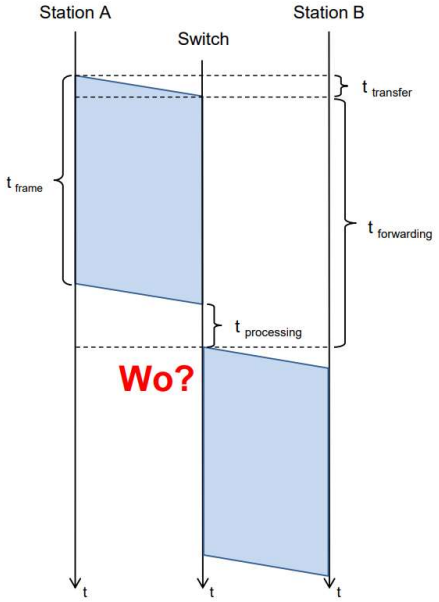
$$t_{\text{frame}} = \frac{\text{Framesize}}{\text{Bitrate}}$$

$$t_{\text{transfer}} = \frac{\text{Distanz}}{2/3\ \text{Lichtgeschw.}} = \frac{d}{2 * 10^8 \text{m/s}}$$

Beispiel Hub



Beispiel Switch



Redundanz (Spanning Tree)

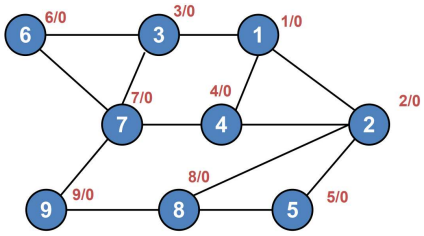
Wenn ein Netzwerk komplexer und es mehre Wege gibt von A nach B. Werden diese Wege durch einen Spanning Tree Algorithmus definiert.

Spanning Tree Algorithmus

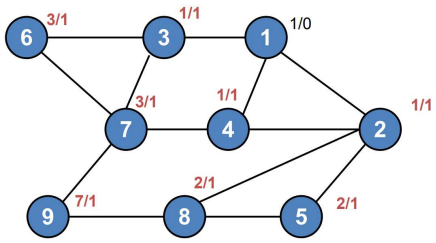
1. Initialisierung
   - Alle Ports für Nutzdaten blockiert
   - Annahme: «Ich bin Root»
   - Austausch BPDUs mit Nachbar. BPDU: Root-ID, Root-Cost, Bridge-ID, Port-ID
2. Aufbau des Spanning Tree (Iteration)
   - «Kleinster» Nachbar als Root gesetzt → Anzahl Hops +1. (**Beachten des Prioritätswert**)
   - So oft wiederholen bis alle dieselbe Root ID besitzen.
3. Setzen der Port Rollen
   - Weg zum «kleinsten» Nachbar wird bevorzugt. (ID & Anzahl Hops)
   - Alle anderen Verbindungen werden geschlossen.

Beispiel Rapid Spanning Tree

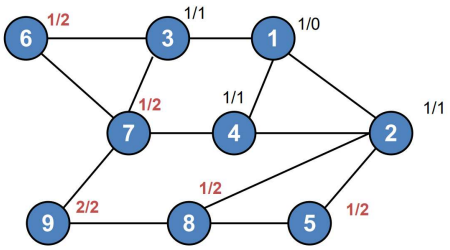
Initialisierung:



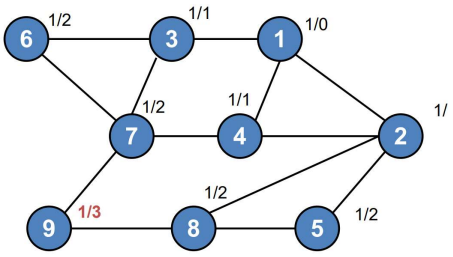
Iteration 1:



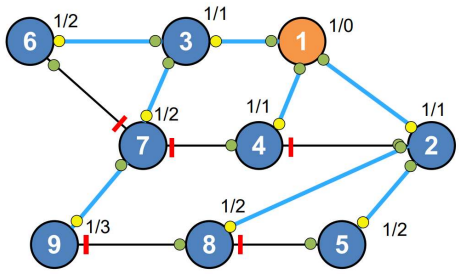
Iteration 2:



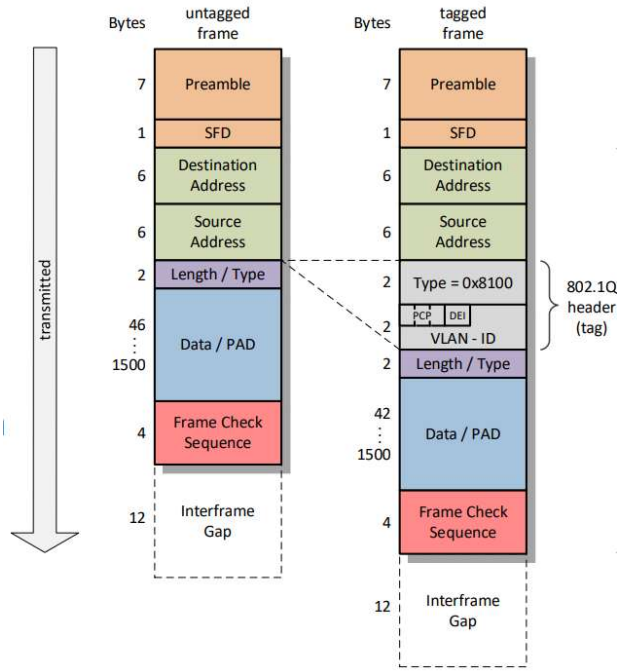
Iteration 3:



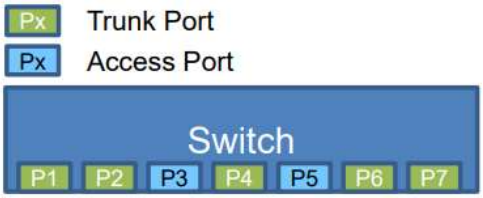
Final



Virtuelle LANs



Der Switch ist wie folgt konfiguriert:



VLAN IDs	All	2 4 5	7	2 3 7	1	2 3 4 8	2 5 7 8

Welche Frames werden an welchen Ports gesendet und sind diese getagged oder ungetagged?

Frame Nr	P2	P3	P4	P5	P6	P7
1	T		T		T	T
2		U	T			T
3	T				T	
4	U		U	U	U	U

Übungsbeispiel VLAN

Es werden folgende Frames gesendet:

Frame Nr	DA	tagged?	VLAN ID
1	ff:ff:ff:ff:ff:ff	ja	2
2	ff:ff:ff:ff:ff:ff	ja	7
3	ff:ff:ff:ff:ff:ff	ja	4
4	ff:ff:ff:ff:ff:ff	nein	N/A

Internet Protokolle des Network Layers

Das Internet verbindet mehrere LANs miteinander durch Router.

Router

Der Router selbst hat die Layer 1 bis 3 implementiert plus zusätzliche Router Funktionen. Ein Router verbindet immer zwei LANs miteinander oder LANs mit dem Internet.

IPv4

Eine IPv4 Adresse besteht aus 4 Bytes, je nach Subnetzmaske gehört ein Teil **Netz** und der Rest zum **Interface**. Die Adresse sieht z.B. so aus: 160.85.16.0 /20 → die 20 ist die Anzahl gesetzten Bits in der Subnetzmaske.

IPv4 Netzwerk Klassen

Die Netzwerkklasse wird anhand der erste 4 Bits bestimmt:

Klasse A: 0..., B: 10..., C: 110..., D: 1110..., E: 1111...

Mögliche Werte für die Subnetzmaske:

Binär Wert	Dezimal Wert
1111 1111	255
1111 1110	254
1111 1100	252
1111 1000	248
1111 0000	240
1110 0000	224
1100 0000	192
1000 0000	128
0	0

Binär Wert	Dezimal Wert	Binär Wert	Dezimal Wert	Binär Wert	Dezimal Wert	Binär Wert	Dezimal Wert	Binär Wert	Dezimal Wert
1	**1**	**11 0100**	**52**	**110 0111**	**103**	**1001 1010**	**154**	**1100 1101**	**205**
10	2	11 0101	53	110 1000	104	1001 1011	155	1100 1110	206
11	3	11 0110	54	110 1001	105	1001 1100	156	1100 1111	207
100	4	11 0111	55	110 1010	106	1001 1101	157	1101 0000	208
101	5	11 1000	56	110 1011	107	1001 1110	158	1101 0001	209
110	6	11 1001	57	110 1100	108	1001 1111	159	1101 0010	210
111	7	11 1010	58	110 1101	109	1010 0000	160	1101 0011	211
1000	8	11 1011	59	110 1110	110	1010 0001	161	1101 0100	212
1001	9	11 1100	60	110 1111	111	1010 0010	162	1101 0101	213
1010	10	11 1101	61	111 0000	112	1010 0011	163	1101 0110	214
1011	11	11 1110	62	111 0001	113	1010 0100	164	1101 0111	215
1100	12	11 1111	63	111 0010	114	1010 0101	165	1101 1000	216
1101	13	100 0000	64	111 0011	115	1010 0110	166	1101 1001	217
1110	14	100 0001	65	111 0100	116	1010 0111	167	1101 1010	218
1111	15	100 0010	66	111 0101	117	1010 1000	168	1101 1011	219
1 0000	16	100 0011	67	111 0110	118	1010 1001	169	1101 1100	220
1 0001	17	100 0100	68	111 0111	119	1010 1010	170	1101 1101	221
1 0010	18	100 0101	69	111 1000	120	1010 1011	171	1101 1110	222
1 0011	19	100 0110	70	111 1001	121	1010 1100	172	1101 1111	223
1 0100	20	100 0111	71	111 1010	122	1010 1101	173	1110 0000	224
1 0101	21	100 1000	72	111 1011	123	1010 1110	174	1110 0001	225
1 0110	22	100 1001	73	111 1100	124	1010 1111	175	1110 0010	226
1 0111	23	100 1010	74	111 1101	125	1011 0000	176	1110 0011	227
1 1000	24	100 1011	75	111 1110	126	1011 0001	177	1110 0100	228
1 1001	*25*	*100 1100*	*76*	*111 1111*	*127*	*1011 0010*	*178*	*1110 0101*	*229*
1 1010	26	100 1101	77	1000 0000	128	1011 0011	179	1110 0110	230
1 1011	27	100 1110	78	1000 0001	129	1011 0100	180	1110 0111	231
1 1100	28	100 1111	79	1000 0010	130	1011 0101	181	1110 1000	232
1 1101	29	101 0000	80	1000 0011	131	1011 0110	182	1110 1001	233
1 1110	30	101 0001	81	1000 0100	132	1011 0111	183	1110 1010	234
1 1111	31	101 0010	82	1000 0101	133	1011 1000	184	1110 1011	235
10 0000	32	101 0011	83	1000 0110	134	1011 1001	185	1110 1100	236
10 0001	33	101 0100	84	1000 0111	135	1011 1010	186	1110 1101	237
10 0010	34	101 0101	85	1000 1000	136	1011 1011	187	1110 1110	238
10 0011	35	101 0110	86	1000 1001	137	1011 1100	188	1110 1111	239
10 0100	36	101 0111	87	1000 1010	138	1011 1101	189	1111 0000	240
10 0101	37	101 1000	88	1000 1011	139	1011 1110	190	1111 0001	241
10 0110	38	101 1001	89	1000 1100	140	1011 1111	191	1111 0010	242
10 0111	39	101 1010	90	1000 1101	141	1100 0000	192	1111 0011	243
10 1000	40	101 1011	91	1000 1110	142	1100 0001	193	1111 0100	244
10 1001	41	101 1100	92	1000 1111	143	1100 0010	194	1111 0101	245
10 1010	42	101 1101	93	1001 0000	144	1100 0011	195	1111 0110	246
10 1011	43	101 1110	94	1001 0001	145	1100 0100	196	1111 0111	247
10 1100	44	101 1111	95	1001 0010	146	1100 0101	197	1111 1000	248
10 1101	45	110 0000	96	1001 0011	147	1100 0110	198	1111 1001	249
10 1110	46	110 0001	97	1001 0100	148	1100 0111	199	1111 1010	250
10 1111	47	110 0010	98	1001 0101	149	1100 1000	200	1111 1011	251
11 0000	48	110 0011	99	1001 0110	150	1100 1001	201	1111 1100	252
11 0001	49	110 0100	100	1001 0111	151	1100 1010	202	1111 1101	253
11 0010	50	110 0101	101	1001 1000	152	1100 1011	203	1111 1110	254
11 0011	**51**	**110 0110**	**102**	**1001 1001**	**153**	**1100 1100**	**204**	**1111 1111**	**255**

IPv4 – Header Format

Version:
4 oder 6 (IPv4 / IPv6)

IHL:
Gibt die Länge des Headers an (inkl. Options), max. 15 → 15 * 4 Bytes = 60 Bytes

DiffServ:
Erlaubt Priorisierung von IP-Datenpakete, 0-5 → DSCP & 6-7 → ECN

Total Length:
Länge des IP-Pakets in Bytes (inkl. Header), max. 65'535 Bytes / normal <1500 Bytes

Identification Number:
Eindeutige Erkennung des ursprünglichen IP-Pakets.

Flags:
Bestehend aus 3 Bits, 0, DF und MF. DF (=Don't Fragment), MF (=More Fragments)

Fragment Offset:
Gibt an wo in einem fragmentierten IP-Paket ein Fragment hingehört.

Time to Live:
Verbleibende Lebenszeit für ein Paket. Bei jedem Router wird dieser Wert dekrementiert.

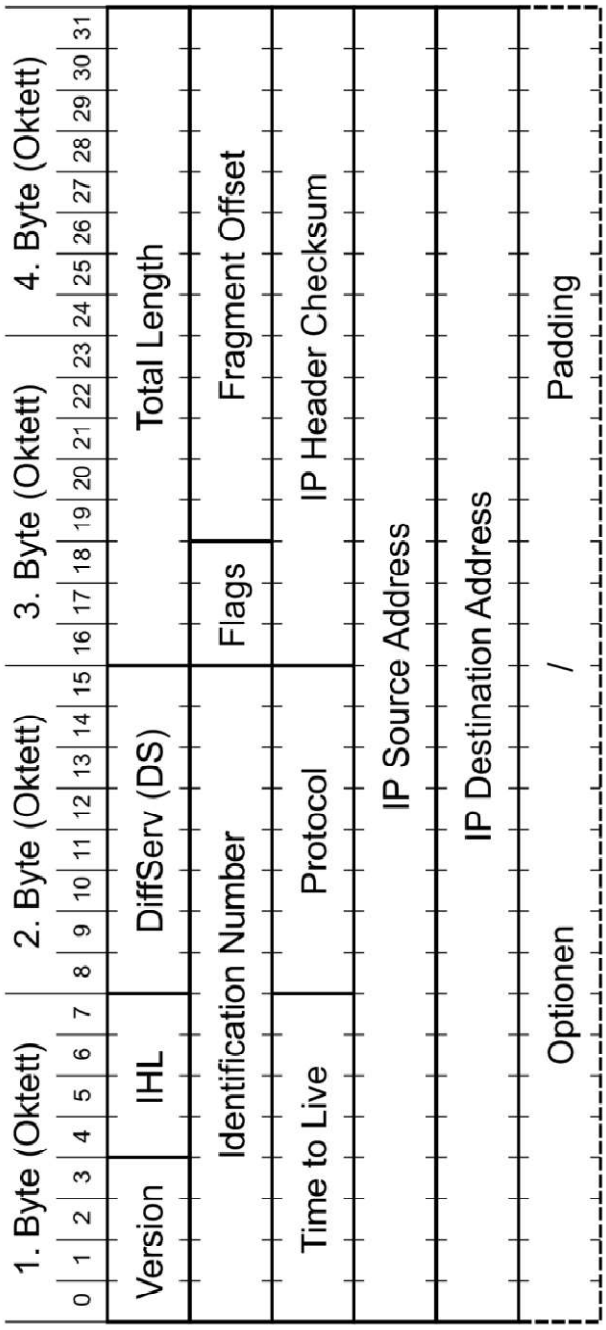
Protocol:
1, 6 oder 17 (ICMP, TCP oder UDP)

Checksum:
16-Bit Prüfsumme über den Header. Wird bei jedem Router neu berechnet.

Source:
IP-Adresse des Hosts

Destination:
IP-Adresse des Hosts

Options / Padding:
Options werden selten verwendet, Padding um ein Vielfaches von 32 Bits aufzufüllen.

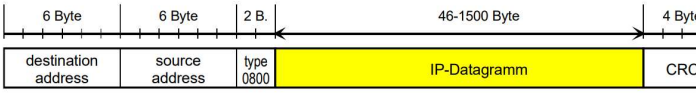


IPv4 Fragmentieren

Für die Fragmentierung sind die Felder Identification Number, Flags und Fragment Offset wichtig. Früher hat der Router übernommen heutzutage (Ipv6) werden Pakete vom Host in bereits schon passender Grösse geschickt.

Kapselung & Adressauflösung

Wenn ein IP Paket in ein Netz gerät muss dieses von dem Router in ein Ethernet Frame verpackt werden. Dabei wird das Typen Feld auf 0x0800 gesetzt.



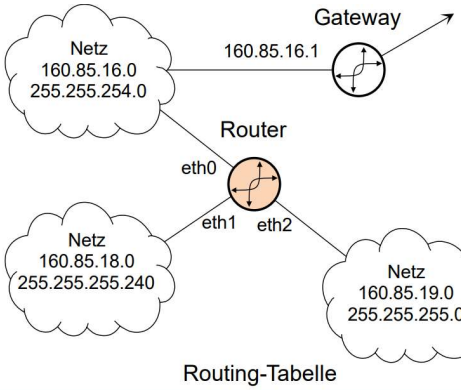
Spezielle IPv4 Adressen

Private Netzadressbereiche (werden im Internet nicht weitergeleitet):

Klasse	Netzadresse(n)	Anzahl Netze	Subnetzmaske
A	10.0.0.0	1	255.0.0.0 / 8
B	172.16.0.0 – 172.31.0.0	16	255.255.0.0 /16
C	192.168.0.0 – 192.168.255.0	256	255.255.255.0 /24

Routing im Internet Layer

Jede Router besitzt eine Routing-Tabelle diese gibt vor mit welcher Netzadresse & Netzmaske welchen Port & Gateway erreicht wird.

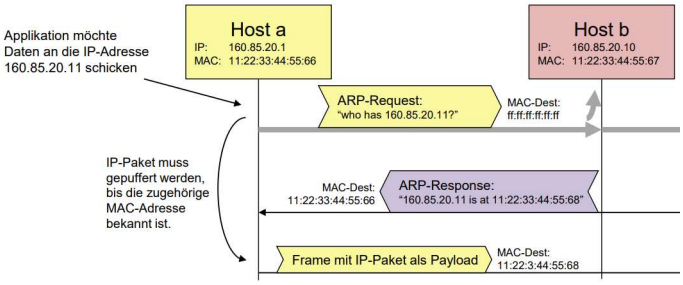


Routing-Tabelle

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

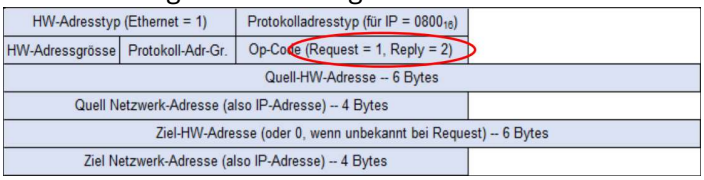
ARP (Adressauflösung)

Wenn ein IP Paket an einem Router ankommt, weiss der vielleicht nicht wie die destination-address für diese IP-Adresse heisst. Dafür macht er einen ARP Request um heraus zu finden wo das Ethernet Frame (mit IP-Paket drin) hinmuss.



ARP Frame

ARP Request & Response befinden sich jeweils in einem Ethernet Frame mit Typ 0x0806. In den Daten wird dann folgendes hinein gesetzt:



ICMP (Internet Control Message Protocol)

Dient zur Übertragung von Fehlermeldungen im Internet Layer. Z.B:

- Wenn Time to live den Wert 0 erreichte
- Ein Host möchte testen, ob ein anderer Host «up» ist.

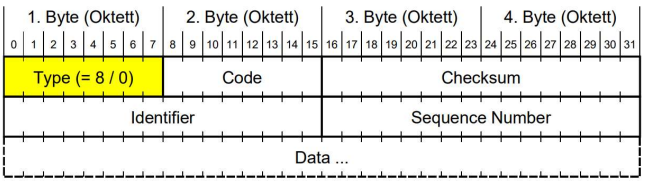
ICMP Meldungen **werden in IP Pakete gekapselt**.

Meldungstypen bei ICMP: Fehler / Information

- **3**: Destination Unreachable
- **11**: Time Exceeded
- **0**: Echo Reply
- **8**: Echo

Codes:

- 0 = net unreachable (Router)
- 1 = host unreachable (Router)
- 2 = protocol unreachable (Ziel Host)
- 3 = port unreachable (Ziel Host)
- 4 = fragmentation needed and DF set (Router)
- 13 = communication administratively prohibited (Firewall)

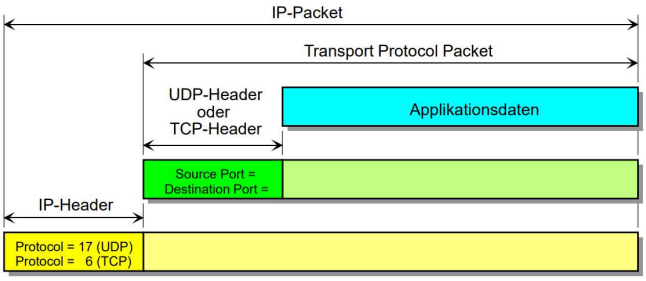


IPv6

Eine IPv6 Adressen haben eine Länge von 16 Byte bzw. 128 Bit. Beispiel:
2001:0620:0000:0004:0A00:20FF:FE9C:7E4A → 2001:620:0:4:A00:20FF:FE9C:7E4A (Verkürzte Schreibweise)

Transport Layer

Kapselung TCP & UDP: TCP & UDP Headers werden in einem IP Paket gekapselt.



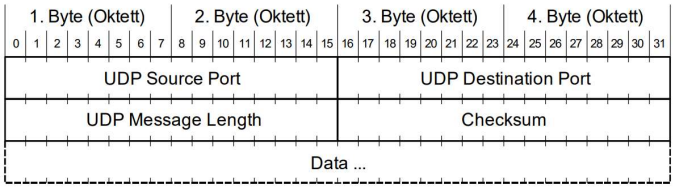
Well-known Ports

Port	Protocol
20 / TCP	FTP - Data
21 / TCP	FTP - Control
22 / TCP	SSH
23 / TCP	Telnet
25 / TCP	SMTP
43 / TCP	WHOIS
53 / UDP/TCP	DNS
80 / TCP	HTTP
67 / UDP	BOOTPs / DHCPs
68 / UDP	BOOTPc / DHCPc
69 / UDP	TFTP
110 / TCP	POP3
143 / TCP	IMAP4
443 / TCP	HTTPS
465 / TCP	SMTPS
993 / TCP	IMAP4S
995 / TCP	POP3S

UDP-User Datagram Protocol

UDP ist **verbindungslos** & **unzuverlässig.** UDP dient dem Multiplexen und Demultiplexen der Datagramme zu den Applikationen.

UDP Header



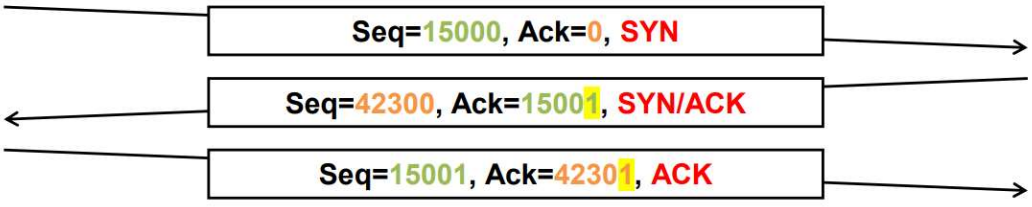
TCP-Transmission Control Protocol

TCP ist **verbindungsorientiert** & **zuverlässig**. TCP kann vollduplexübertragen.

Verkehrssteuerung

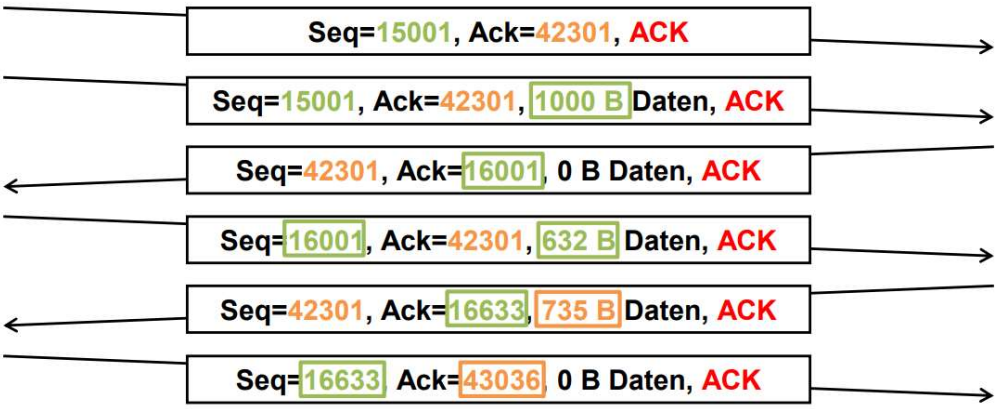
Verbindungsaufbau

- Beim Verbindungsaufbau «horcht» der Server auf einer bestimmten Port Nummer (z.B. 80).
- Nun kommt ein Client sendet ein Frame mit einem SYN Flag und einer zufälligen **Sequenznummer s1** (z.B. 15'000).
- Server bestätigt die Sequenznummer s1 mit einer **Acknowledgement Nummer s1+1** (15'001) und wählt eine zufällige **Sequenznummer s2** (z.B. 42'300), das Frame wird mit einem SYN/ACK Flag.
- Client bestätigt s2 mit **Acknowledgement s2+1** (42'301).



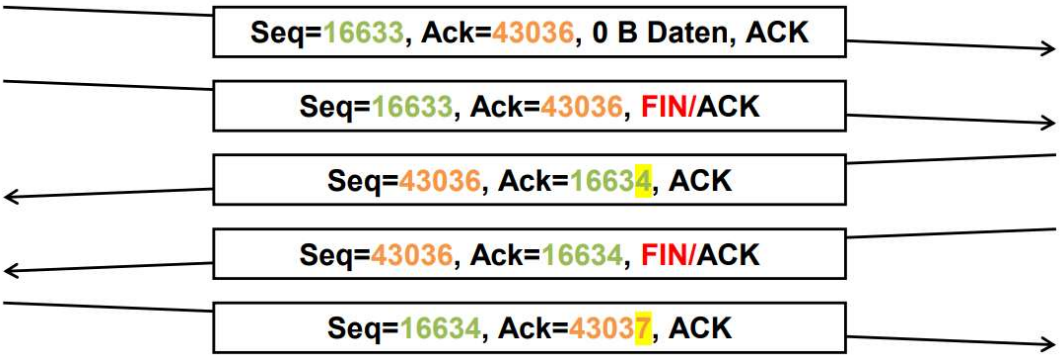
Datenaustausch

- Nach dem Verbindungsaufbau können Daten geschickt werden.
- Wenn der Server oder Client Daten schickt muss von dem anderen die **Acknowledgement Nummer** mit den Anzahl Bits der geschickten Daten aktualisieren.

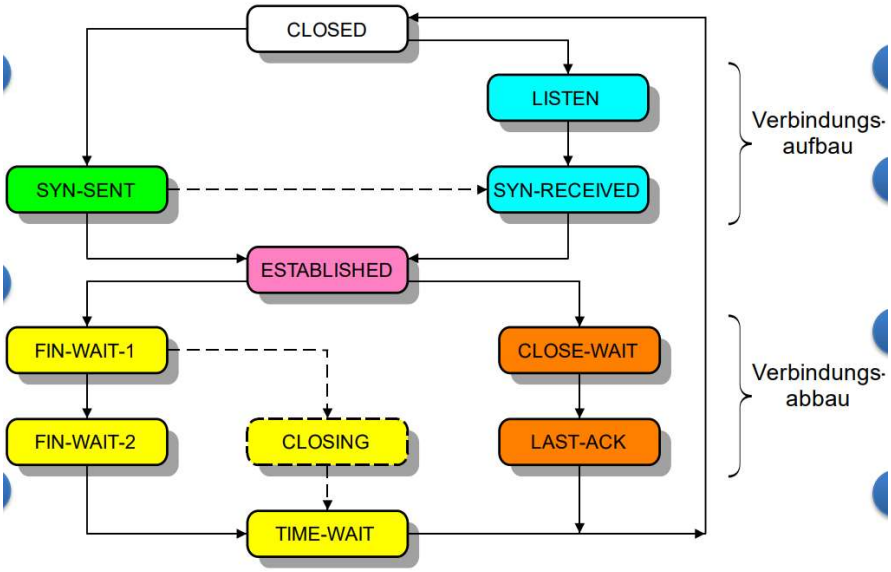


Verbindungsabbau

- **Beide Seiten** können den Verbindungsabbau einleiten. Die Verbindung ist erst geschlossen wenn beide Seiten den Verbindungsabbau eingeleitet haben.
- Eine Seite kann die Verbindung schliessen mit einem FIN/ACK Flag. Diese muss dann die andere Seite rückmelden, die Acknowledgement Nummer wird dadurch um 1 inkrementiert.



Zustandsdiagramm



Adaptive Elemente

Retransmission Time-Out (RTO)

Ist eine dynamische Anpassung der Wartezeit bis zum senden des nächsten Pakets (Überlastung des Netztes). TCP misst bei jeder aktiven Verbindung die Round-Trip Time (RTT), zur Berechnung:

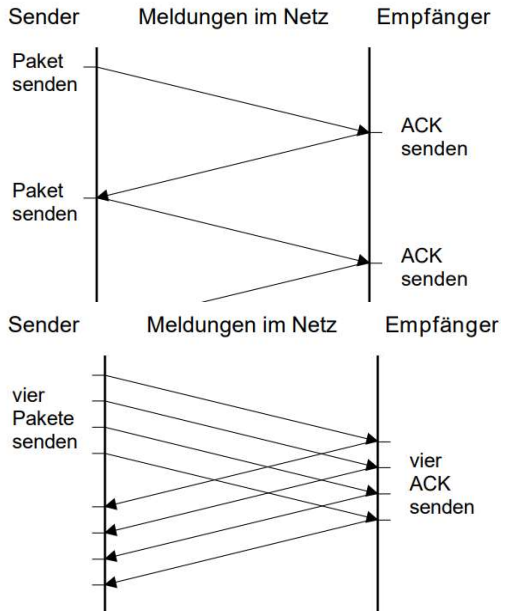
$$SRTT_{neu} = (1 - \alpha) * SRTT_{alt} + \alpha * RTT, \quad \alpha = 0.125$$

$$RTTVAR_{neu} = (1 - \beta) * RTTVAR_{alt} + \beta * |SRTT - RTT|, \quad \beta = 0.25$$

$$RTO = SRTT + 4 * RTTVAR$$

Fluss-Steuerung / Sliding Window

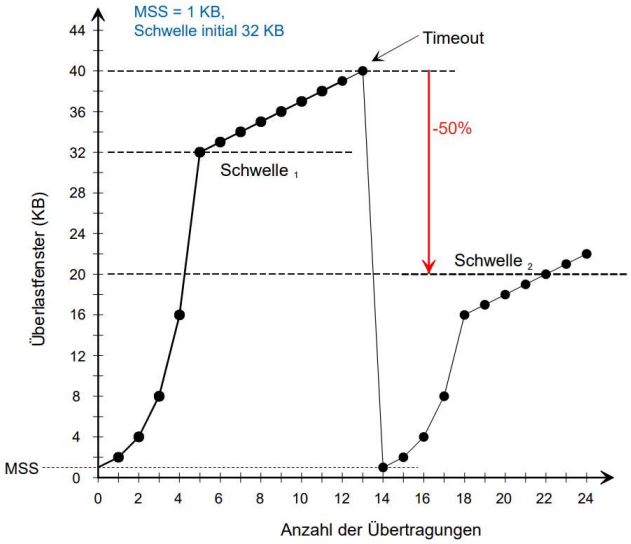
Stop & Wait ist sehr ineffizient, darum ist es sinnvoll mehrere Pakete auf einmal zu schicken.



Wie gross soll nun die TCP Puffergrösse gewählt werden: $\text{BDP (bits)} = \text{RTT (sec)} * \text{Bandbreite (bps)}$

Slow Start

Beim Slow Start wird heran getastet wie gross die einzelnen Frames sein können. Auf dem Diagramm sieht das wie folgt aus:



TCP Header

TCP Source/Destination Port:
Bezeichnet jeweils die Ports auf Sender- & Empfängerseite.

Sequence Nummer:
Wichtig für die Verkehrssteuerung.

Acknowledgement Nummer:
Wichtig für die Verkehrssteuerung.

Header Länge:
In 32-Bit Einheiten → Faktor 4

ECN Flags:
- Bit 8: CWR
- Bit 9: ECE

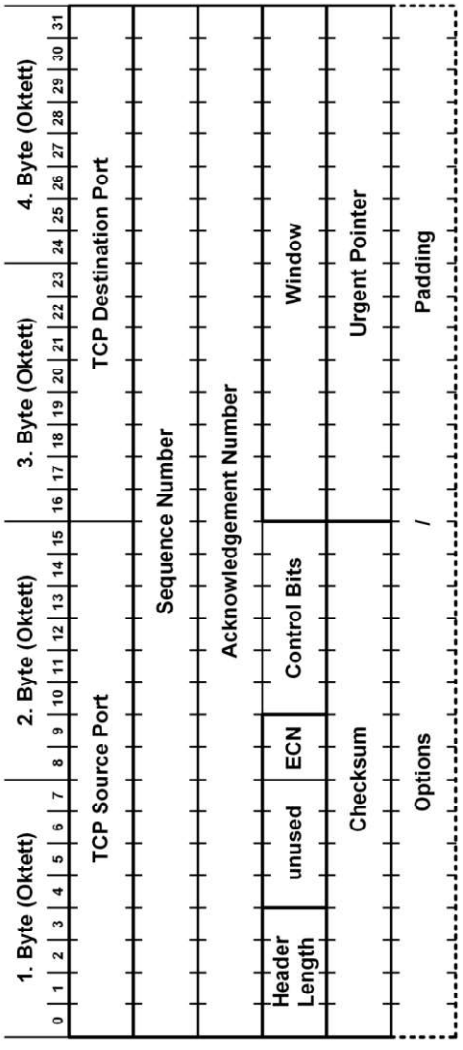
Control Bits:
Bestehend aus 6 Bits, jedes Bit kann einzeln gesetzt werden:

10	11	12	13	14	15
URG	ACK	PSH	RST	SYN	FIN

Window:
Zeigt der anderen Seite die aktuell verfügbare Puffergrösse an.

Checksumme:
16-Bit Prüfsumme über den TCP Header.

Urgent Pointer:
Falls URG-Flag gesetzt wurde: gibt Position in den Daten an wo sich die Urgent Daten befindet.

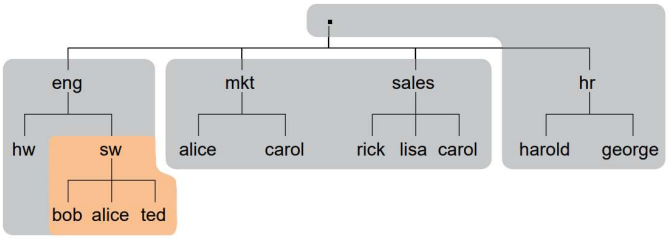


Application Layer

Domain Name System (DNS)

Vereinfacht die Nutzung des Internet für einen Benutzer, da das Internet selbst nur IP-Adressen kennt. Darum muss eine Adresse wie: www.zhaw.ch in die IP-Adresse 160.85.104.112 übersetzten werden können und umgekehrt.

DNS ist dabei eine Verzeichnisstruktur (Baum) und wird von hinten nach vorne gelesen. Dabei hängt alles an der Root (·). Es gilt jeder Name Server kennt sicher seine direkt unterstellten Name Server & deren IP Adresse. Für eine Zone ist immer ein NS zuständig und pro Zone gibt es mindestens zwei NS.



DNS Abfragen

DNS verwendet für Abfragen UDP Port 53. Wenn man nach der Adresse ted.sw.eng sucht wird wie folgt abgefragt:

1. Anfrage an ·: eng? — Wo befindet sich
   Adresse von eng. — Antwort: IP
2. Anfrage an $eng.$: sw? — Wo befindet sich
   Adresse von $sw.eng.$ — Antwort: IP
3. Anfrage an $sw.eng.$: ted? — Wo befindet sich
   Adresse von $ted.sw.eng.$ — Antwort: IP

Dabei wird aber nicht nur eine IP Adresse zurückgegeben. Der Record Type enthält zusätzliche Information wie zum Beispiel:

Type	Beschreibung / Funktion
A	IPv4 Adresse des gesuchten Hosts (32 Bit)
AAAA	IPV6 Adresse des gesuchten Hosts (128 Bit)
MX	Mail Exchange (Mail Server)
NS	Name Server (Name Server Name für eine Zone)
CNAME	Canonical Name (primärer Name) für einen Alias zum Host
TXT	Text Record, in Antworten für verschiedenste Angaben verwendet

DHCP

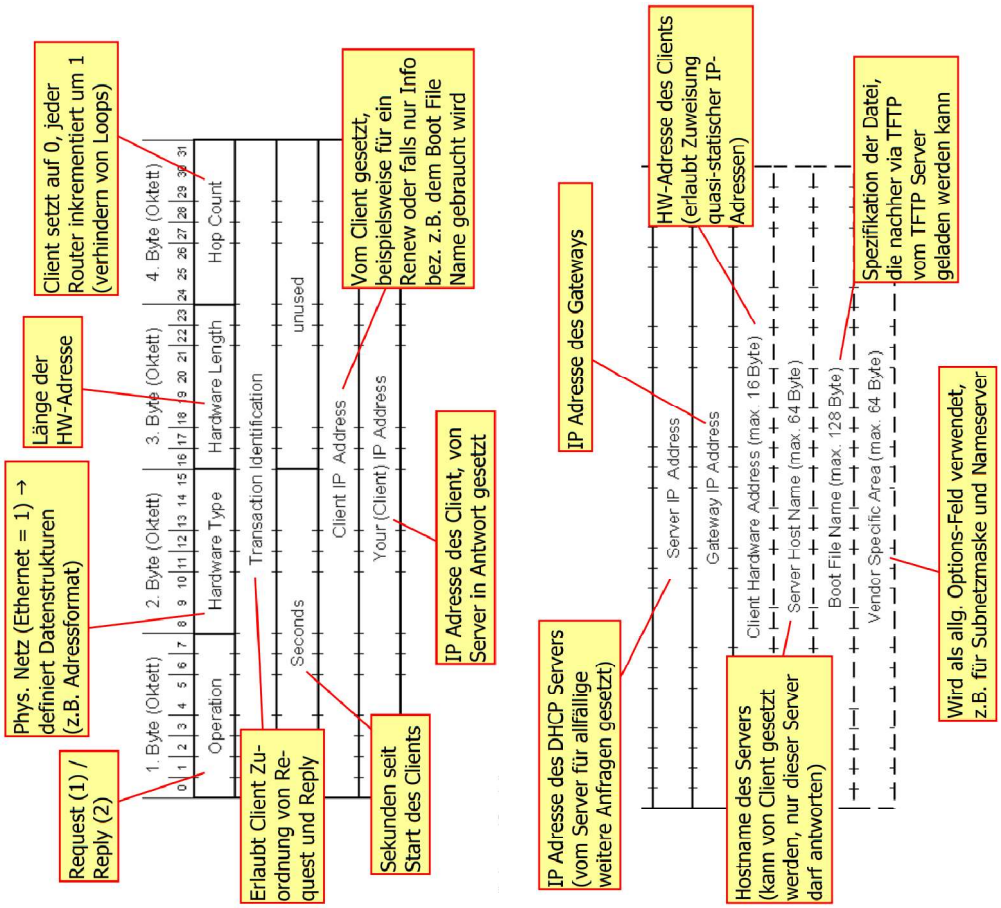
Wie erhält ein Konten seine IP-Adresse?

1. Lokal konfiguriert (statische IP Adressen)
2. Bezug der IP-Adresse über das Netzwerk, dies erlaubt DHCP.

Dynamische Zuweisung von IP-Adressen

1. Client verlangt eine IP-Adresse (DHCP Request)
2. DHCP-Server erteilt eine freie Adresse für definierte Lease Time, oft 10 Minuten.
3. Vor Ablauf der Lease Time muss der Lease (vom Client) erneuert werden.
4. Client, der das Netz verlässt → Lease wird nicht erneuert.

DHCP Paketformat



Network Address Translation (NAT)

Alle Hosts im privaten Netz 192.168.0.0/8 verwenden 192.168.0.1 als Default-Gateway.

Port-basierte NAT (NAPT) hat folgende Funktionen:

- Ersetzt private IP Adresse im IP Header durch eine öffentlich IP des Gateways / Routers.
- Ersetzt die private Port-Nr. des Hosts durch eine freie zulässigen Port-Nr. des Gateways / Routers.
- Erstell ein Mapping von private IP Adresse & Port-Nr. zur öffentlichen Port-Nr.
- Man kann für das Mapping auch statische Werte definieren, hier wird aber nur die Port-Nummer übernommen.

Problem mit NAT

NAT verletzt das Konzept der OSI-Layer. Um einen Port im TCP Header zu ändern muss man eigentlich die Daten im IP-Frame ändern. Bedeutet eine Netzwerk-Funktion greift auf den Transport Header zu.

Von A nach Z

Du schaltest dein PC ein und möchtest www.google.ch aufrufen. Was passiert hier alles?: