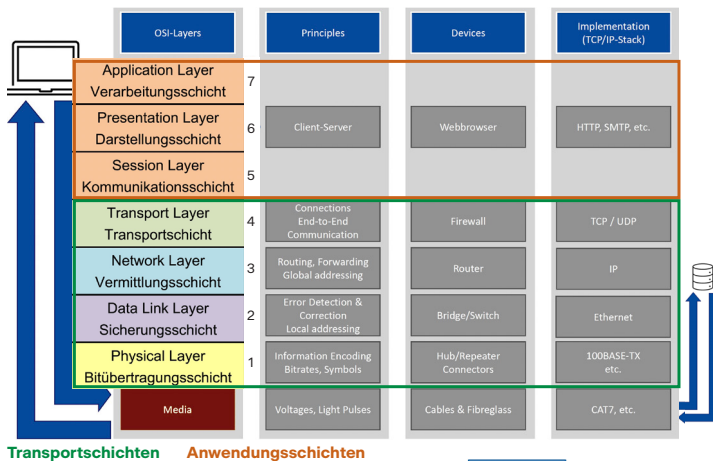


OSI-Modell

Dient nur als Referenzmodell. Konnte sich nicht als Standard durchsetzen.
Fehlt: Verschlüsselung, Zeitsynchronisation und mehr



Schichten, Protokolle und Dienste

Schicht (OSI-Layer): Hat die Aufgabe, der darüberliegenden Schicht bestimmte Dienste anzubieten.

Protokoll: Legt fest, wie eine Datenübertragung zwischen zwei oder mehreren Schichten auf gleicher Ebene (Peer-to-Peer-Protokoll) abläuft. Implementation mittels Hardware oder Software. Kommunikation erfolgt logisch (via Software).

Dienst: Für die Kommunikation zwischen darunter oder darüber liegenden Schichten zuständig (mittels Interface).
Protokoll-Stack (= Protokoll für die übereinanderliegenden Schichten)
Kommunikationübertragung erfolgt physikalisch über ein Medium.

Klassifizierung von Diensten

Zuverlässig / Verbindungsorientiert:

= **Leitungsvermittlung:** Datenaustausch nach Leitunsaufbau, Reihenfolge;
Kein Datenverlust, Statefull; z.B. TCP, Datenbank, Websites, Filetransfer

Unzuverlässig / Verbindungslos:

= **Paketvermittlung:** Senden ohne Leitunsaufbau. Möglicher Datenverlust;
Bei zeitkritischen Übertragungen; z.B. UDP, Videostream, Telefonie, Briefpost

Übertragungsmedien (0)

Für alle Übertragungsmedien gilt:

- Je **größer die Bandbreite** (Übertragungsgeschwindigkeit [Hz]), desto **höhere Datenrate** (Bitrate [Bit/s]).
- Je **kleiner die Dämpfung** (Leistungsverlust), desto **größer Distanzen** können erreicht werden.
- Umso **kleinere die Bitrate** (bei gleicher Dämpfung), desto **grössere Distanzen** können erreicht werden.

Signaldämpfung (Insertion Loss / Attenuation)

= Leistungsabnahme eines Signals

Signaldämpfung / Rauschabstand SNR [dB]

$$SNR = 10 \cdot \log_{10}(S/R) = 20 \cdot \log_{10}\left(\frac{U_1}{U_2}\right) [dB]$$

S/N als Faktor (Verhältnis)

$$S/R = \frac{P_1}{P_2} = \left(\frac{U_1}{U_2}\right)^2 = 10^{\left(\frac{SNR [dB]}{10}\right)} = \left(10^{\left(\frac{SNR [dB]}{20}\right)}\right)^2$$

Signal-Rausch-Verhältnis SNR [dB]

Gibt an wie stark sich das eigentliche Signal vom Rauschen abhebt.

$$SNR = 10 \cdot \log_{10}\left(\frac{P_{\text{Signal}}}{P_{\text{Störung}}}\right) [dB]$$

P = Leistung [W] U = Spannung [V]

70 [dB] = 10^7 = [Faktor], -10 [dB] = 10^{-1} = 0.1 [Faktor]

Gegeben:

Symbol-Dauer: $T_{\text{Sym}} = 66.67 \mu\text{s}$
Bandbreite : $B = 8 \text{ kHz}$
Rauschabstand $SNR = 40 \text{ dB}$

Umrechnen Symboldauer → Symbolrate

$$\text{Baudrate } f_s = \frac{1}{T_{\text{Sym}}} = \frac{1}{66.67 \mu\text{s}} = 15 \text{ kBaud}$$

Rauschabstand in dB → Leistungsverhältnis

$$\frac{S}{N} = 10^{\frac{SNR}{10}} = 10^{\left(\frac{40}{10}\right)} = 10'000$$

1. Prüfe Nyquist:

15 kBaud < $2 \cdot 8 \text{ kHz}$ → OK

2. Maximale Kanalkapazität (Shannon)

$$C_s = B \cdot \log_2\left(1 + \frac{S}{N}\right)$$

$$C_s = 8000 \cdot \log_2(1 + 10'000)$$

$$C_s = 8000 \cdot 13.3 \frac{\text{Bit}}{\text{s}} = 106.3 \frac{\text{kBit}}{\text{s}}$$

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

3. Maximale Anzahl Bits pro Symbol

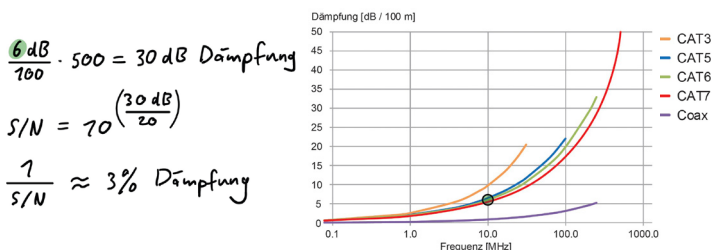
$$\frac{\text{Bitrate}}{\text{Baudrate}} = \frac{106.3 \frac{\text{kBit}}{\text{s}}}{15 \text{ kBaud}} = 7.08 \rightarrow 7 \frac{\text{Bit}}{\text{Symbol}}$$

Wir runden ab

4. Anzahl Symbole

$$\text{Anzahl Symbole} = 2^7 = 128$$

Wie hoch ist die Dämpfung eines 500 m langen CAT6-Kabels bei 10 MHz?



Störungsbehebung (Twisted Pair)

Kapazitive Störung (Ladung)

Lösung: **Komplementäre Signale** und **Folienschirm** (elektrisch leitender Schirm). Der Empfänger subtrahiert die Signale, wobei sich die eingekoppelten gleichgerichteten Störungen weitgehend aufheben.

Induktive Störungen (Magnetfeld)

Lösung: **Verdillung (Twist Pair) von Aderpaaren**

Störungen werden in benachbarten. Schleifen gegenpolig induziert. Störung hebt sich weitgehend auf.

Kabel Übersicht

Twisted Pair: Breitbandige Datenübertragung, anfälliger auf Störungen

Koaxialkabel: Hochfrequente Signalübertragung, unempfindlich gegenüber Störungen

Glasfaser (Lichtwellenleiter): Unempfindlich gegenüber Störungen,

Hohe Bandbreite / grosse Datenrate,

Geringe Dämpfung / grosse Übertragungsstrecken

Die maximale **Ausbreitungsgeschwindigkeit** beträgt etwa

$2/3 c = 20 \text{ cm/ns} = 200'000 \text{ km/s}$

Multimode (Stufenfaser) (für kurze Distanzen)

- Dicker Kern → führt zu **Disperion** (Datenverschmierung)
- Kleine Datenraten und Übertragungsstrecken, dafür billig

Singlemode (Gradientenfaser) (für lange Distanzen)

- Keine Moden-Dispersion
- Hohe Datenraten und grosse Übertragungsstrecken, dafür teuer

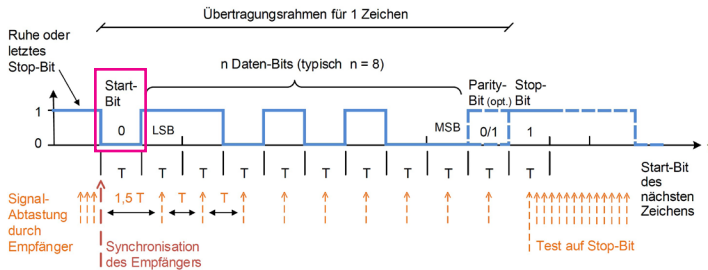
Physical Layer / Bitübertragungsschicht (1)

Aufgaben

Umwandlung physikalischer Signale (elektrisch, optisch) in einen Bitstrom und umgekehrt. Die **Leitungscodierung** legt fest, wie diese Umsetzung erfolgt. Wichtige Anforderungen sind **Gleichspannungsfreiheit** und **Taktzurückgewinnung**. Z.B. RS-232, AMI, PAM3, ...

Asynchrone Übertragung (Seriell)

Übertragung ohne Taktsignal (clk), benötigt Start- (0), Stop- (1) und Paritätsbit (0/1). Langsam. Daten müssen vom Empfänger mit einem Vielfachen der Bitrate abgetastet werden. Während der Übertragung eines Rahmens dürfen die Takte insgesamt nicht mehr als eine halbe Bitzeit T abweichen. Zusätzlich wird ein zweites Medium für Ground benötigt. RS-232:



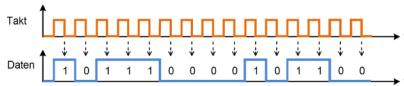
Daten-Bits müssen gespiegelt werden (in Memory) bzw. LSB wird zuerst eingelesen und MSB zuletzt:

LSB 1 1 0 1 0 1 0 0 MSB $\rightarrow$ MSB 0 0 1 0 1 0 1 1 LSB = in memory

Synchrone Übertragung (Seriell)

Erfordert ein Taktsignal (clk) als separate Leitung. Schnell und zuverlässiger. Z.B. Videokonferenzen, Chatrooms, Telefongespräche. Es ist Aufgabe des **Data Link Layers** die Grenzen und das Ende einer Meldung zu ermitteln. Zusätzlich wird ein drittes Medium für Ground benötigt.

Taktzurückgewinnung: Der Empfänger vergleicht sein Takt mit dem des Senders und passt den Takt mit einer Korrektur (Offset) an.



Kopplung (Verbindungsart)

Punkt-Punkt: Direkte Verbindung zweier Kommunikationspartner. Z.B. zwischen PC und lokalem Drucker, Router über Ethernetkabel

Shared Medium: Mehrere Partner verkehren über das gleiche Medium. Z.B. Bus oder klassische Local Area Network (WLAN).

Verkehrsbeziehung (Kommunikation)

Simplex: Kommunikation nur in eine Richtung (Sender Tx $\rightarrow$ Empfänger Rx) über ein Kanal. Z.B. Radio, TV.

Halbduplex: Kommunikation abwechselungsweise (nicht zur selben Zeit) in beide Richtungen über eine Leitung. Z.B. einfaches Funkgerät.

Vollduplex (Duplex): Kommunikation in beide Richtungen gleichzeitig über eine Leitungen. Z.B. Telefon.

Bitrate und Baudrate

Beispiel Amplitude Shift Keying (ASK-4)

4-wertige Symbole, die sich nur in der Amplitude unterscheiden

- Baudrate: $1000 \text{ Baud} = 1 \text{ Symbol/ms}$
- Bit pro Symbol: $\log_2(4) = 2$
- Bitrate: 2000 Bits/s
- Zusatzfrage: $1 \text{ ms} = 10^{-3} \text{ s} = 4 \text{ kHz}$



Leitungscodes

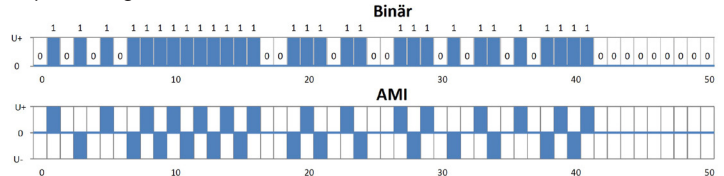
Bei **Synchroner Übertragung ohne separate Taktleitung (clk)**

Aufgabe: **Taktzurückgewinnung** und **Gleichspannungsfreiheit**
Erlaubt den Takt zusammen mit dem Datensignal zu übertragen.
Der Empfänger ist verantwortlich für die **Taktzurückgewinnung**. Bsp:

Gleichspannungsfreiheit und Kanalkodierung

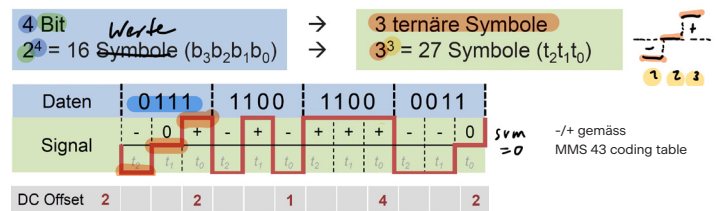
Die elektrischen Potentiale zweier Stromkreise sind voneinander getrennt und untereinander **potentialfrei / neutral** (Leitung ist insgesamt neutral).

Bsp. 3-wertiger AMI-Code (Alternate Mark Inversion):

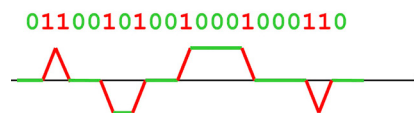


Problem: Wenn für längere Zeit keine Daten übertragen werden kann **keine Taktzurückgewinnung** vorgenommen werden.

PAM3



Ethernet 100BASE-TX: NRZI (Non Return to Zero Inverted) kombiniert mit MLT-3 (Multi-Level Transmit)



Kenngrossen

Bandbreite B [Hz, bit/s]

Eigenschaft des Übertragungskanals und durch Medium begrenzt
Kanalbreite, Übertragungsgeschwindigkeit von Daten.
Je höher die Grundfrequenz, desto mehr Bandbreite nimmt das Signal ein.

Baudrate f_s [Bd]

Symbole/Zeit, Symbolrate, Symboldauer, Baudrate

$$f_s = \frac{1}{\text{Symboldauer } T_s} \text{ [Bd]}$$

Maximale Symbolrate

Für ein ideales Übertragungsmedium ohne Störungen

$$f_s \leq 2B \text{ [Bd]} \text{ (Nyquist)}$$

Bitrate R [bit/s]

Datenrate, Durchsatz (Netto-Datenmenge)

$$R = f_s \cdot I_s \text{ [bit/s]}$$

Maximale Bitrate

$$R \leq 2B \cdot \log_2(M) = C_s \text{ [bit/s]} \text{ (Hartley)}$$

Bitrate R : Rauschen nicht berücksichtigt

$$C_s = B \cdot \log_2(1 + S/N) \text{ [bit/s]} \text{ (Shannon)}$$

Kanalkapazität C_s : Berücksichtigt zusätzlich Rauschen

Symbol

Ein übertragenes physikalisches Signal (Frequenz, Phasenlage, El. Spannung)

$$\text{Symboldauer } T_s = \frac{1}{f_s} \text{ [s/bit]}$$

Anzahl Symbole M

$$\text{Symbole } M = \sqrt{1 + S/N} = 2^{I_s}$$

Informationsgehalt I_s [Bit]

$$I_s = \log_2(M) = \log_2(2^{I_s}) = \frac{C_s}{f_s} \text{ [Bit]}$$

Data Link Layer / Sicherungsschicht (2)

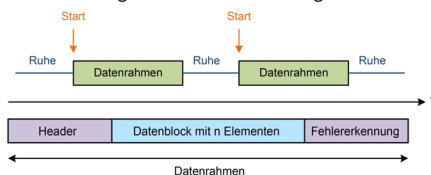
Aufgaben

- **Verpacken** und **Auspacken** von Datenblöcken in Datenrahmen (**Frames**). Erkennen von Framegrenzen aus Bitstrom. Asynchron oder Synchron. Bei synchroner Übertragung ist eine Maskierung (Flags) nötig.
- Zu **Fehlererkennung** wird den Daten **Redundanz** beigefügt (in Form von zusätzlich übertragener Information). Diese erhöht die Hamming Distanz (= Mindestanzahl unterschiedlicher Bits zwischen gültigen Codewörtern). Die betrachteten Verfahren gehören zur Familie der Block Codes.
- Eindeutige **Adressierung** der Teilnehmer (Kommunikationsknoten). Bei >2 Kommunikationspartnern (**Shared Medium**) benötigt es ein Verfahren zur **Adressierung** und Steuerung der **Sendeberechtigung (Medium Access)**.
- Koordination des Medien Zugriff auf gemeinsame Medien.
- **Fehlerkorrektur** kann rückwärtsgerichtet (erneutes Übertragen der Daten) oder vorwärtsgerichtet (Rekonstruktion von verfälschten Bits beim Empfänger, Forward Error Correction FEC) erfolgen.

Framing (Rahmenbildung)

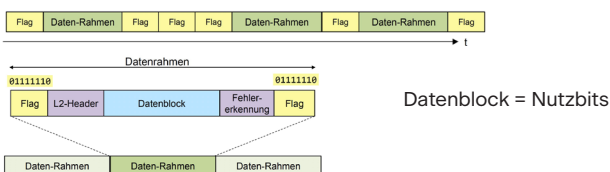
Asynchrone Übertragung

Daten werden mit einer Pause (**Interframe-Gap**) von 96 Bit-Zeiten zwischen den Frames gesendet. Anwendung bei Ethernet.



Synchrone Übertragung

Daten werden ohne Unterbruch gesendet. Stehen keine Daten an werden kontinuierlich **Flags (reservierte Bitfolge)** als **Start- und Endbits** gesendet. Häufiges Bitmuster: 01111110



Bitstopfen (Bit-Stuffing)

Entfernt Start und Endflag aus den Daten im Frame.

Damit das Flag/Bitmuster auch innerhalb eines Frames vorkommen kann, fügt der Sender nach jedem 5. Eins zusätzlich eine 0 ein: 011111010

Der Empfänger entfernt jedes 0 nach jedem 5. Eins. = **Escape**

Fehlerwahrscheinlichkeit

Bit Error Ratio (BER) p_e : Anzahl fehlerhafte Bits im Verhältnis

Frame Error Ratio (FER): Fehlerhaft empfangene Datenrahmen

Residual Error Ratio (RER): Unentdeckte, fehlerhafte, unkorrigierte und ans nächste Layer weitergegeben Frames

Erfolgswahrscheinlichkeit N Bits

$$P_{\text{Erfolg}} = (1 - p_e)$$
$$P_{\text{Fehler, Frame}} = 1 - (1 - p_e)^N$$
$$P_{\text{Erfolg, Frame}} = (1 - p_e)^N$$
$$P_{\text{Fehler, Frame}} \cong N \cdot p_e \quad (=FER)$$

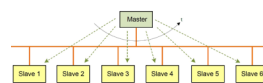
Wahrscheinlichkeit für fehlerhafte Frames $P_{\text{Fehler, Frame}} \cong N \cdot p_e$

12'000 Bit * 10^{-8} Fehler / Bit = **$1.2 \cdot 10^{-4}$**

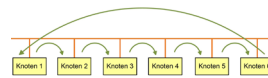
Im Mittel ist jedes **8333**. Frame fehlerhaft

Medium Zugriffsverfahren

Master-Slave: Vorteil: Keine Konflikte, Master koordiniert Zugriff
Nachteil: Ausfall des Masters (Single Point of Failure)



Token-Verfahren («Redestab»): Vorteil: Deterministisch (man weiss, wann man dran kommt); Nachteil: Aufwändig (Startup, Token Verlust, etc.)



Variante: Anstelle eines Tokens wird ein Frame geschickt.

Knoten fügen ihre Daten an den vorbestimmten Positionen ein (Interbus) oder hängen die Daten hinten am Frame an.

Typische Anwendung: Master/Slave-Konfiguration



Zeitsteuerung

Vorteil: Optimierung möglich (nach Auslastung, Durchsatz, «Reisezeit» etc.)

Nachteile: Planung und genaue Zeit in allen Knotenpunkten erforderlich, Konflikte mit unplanbarem Verkehr (SBB Cargo)

Random Medium Zugriff



Carrier Sense Multiple Access (CSMA): Alle Stationen sind gleichberechtigt (kein Master). Vor dem Senden wird das geteilte Übertragungsmedium abgehört, ob es frei ist (Carrier Sense). Wenn 2 Knoten gleichzeitig senden wird das Signal gestört.

Kollisionsbehandlung:

- CSMA/CD (Collision Detection): Abbrechen und nochmals Versuchen
- CSMA/CR (Collision Resolution): Hardware-unterstützte Arbitrierung. Listen-while-talking: Knoten liest und schreibt gleichzeitig.
- CSMA/CA (Collision Avoidance): Prüfen, ob Medium frei ist, dann erst senden (Listen before talk (LBT)). Z.B. WLAN

Fehlererkennung

Data Link Layer (2)

Hamming-Distanz h

Minimale Anzahl Bits, in denen sich zwei gültige Codewörter eines Codes unterscheiden. Eine Hamming-Distanz h erlaubt die Erkennung von h-1 Fehlern.

Korrigierbare Bitfehler $k = \lfloor (h-1) / 2 \rfloor$

Verfahren nach Eignung zu Fehlererkennung

1. CRC-32 (Mehrbitfehler erkennbar)
2. Längs-/Querparität
3. 16-Bit Prüfsumme (Checksum)
4. Odd/Even Parity RS-232

Even-/Odd Parity

1 Fehler sind erkennbar (Hamming Distanz = 2) und 0 korrigierbar.



Even Parity: Anzahl 1er inkl. Parity-Bit ist **gerade**

Odd Parity: Anzahl 1-Bit inklusive Parity-Bit ist **ungerade**

Vorsätze für Masseinheiten (metric prefix)

peta	P	10^{15}	1 000 000 000 000 000	deci	d	10^{-1}	0.1
tera	T	10^{12}	1 000 000 000 000	centi	c	10^{-2}	0.01
giga	G	10^9	1 000 000 000	milli	m	10^{-3}	0.001
mega	M	10^6	1 000 000	micro	μ	10^{-6}	0.000 001
kilo	k	10^3	1 000	nano	n	10^{-9}	0.000 000 001
hecto	h	10^2	100	pico	p	10^{-12}	0.000 000 000 001
deca	da	10^1	10	femto	f	10^{-15}	0.000 000 000 000 001

Local Area Network (LAN)

Data Link Layer (2)

Key-Takings

- LAN: Räumlich (lokales) kleines Netzwerk mit hoher Geschwindigkeit (1 Gbit/s). Verbindet Server, Workstations, PCs, Drucker, etc. Zugang zum Internet über einen Router im LAN.
- LAN-Topologien:** Bus, Linie, Ring, Stern, Baum, vermaschte Strukturen
- Zugriffsverfahren:** Token-Verfahren oder Carrier Sense Multiple Access.
- Alle Ethernet Varianten teilen gleiche Frame-Format
- MAC-Adressen** der Länge 6 Bytes identifizieren Ethernet Geräte
- Switches (Bridges) arbeiten (unsichtbar) auf dem Data Link Layer und schliessen mehrere Segmente zu einem LAN zusammen
- Redundanz:** Mehrere Verbindungen zu anderen Knoten

Übertragungsarten

Unicast: Frame trägt die Adresse dieses Empfängers (Ping, Mail)

Multicast: Frame trägt die Multicast Adresse der Gruppe

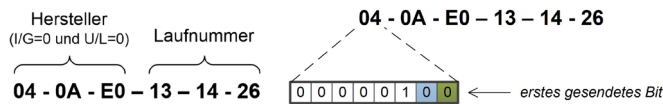
Broadcast: Frame trägt die Broadcast Adresse des LAN

Adressierung

Adressen im LAN müssen eindeutig sein → **MAC Adressen** (IEEE)

Jedes Gerätes hat weltweit eine eindeutige MAC-Adresse.

6 Bytes. LSB (Bit ganz rechts) wird zuerst gesendet.



Ethernet (Frameformat)

Data Link Layer (2)

Ethernet-Technologie und -Systeme

- Port Mirroring** ist ein mögliches Verfahren zur Verkehrsbeobachtung und Fehlersuche in Switched Ethernet
- Redundanz** wird ermöglicht durch die «künstliche» Reduktion der Topologie auf eine Baumstruktur durch Spanning Tree Algorithmen
- Lässt hohe Datenraten einfach zu. Asynchrone Übertragung.

Frame-Format LAN

Präambel (PRE) und SFD: Synchronisation

Destination/Source Address: MAC-Adresse

Length/Type (Payload): 0x0800: IP, 0x0806: ARP, Falls 0x8100 → Frame = VLAN

00 AD 38 17 46 53 08 AF Beispiel Tagged Frame
DE B9 37 22 81 00 08 06 → VLAN-Frame
08 00 45 00 00 20 12 34

Data / PAD: Falls weniger als 46 Bytes werden Nutzdaten mit Nullen aufgefüllt.

Frame Check Sequence (FCS): CRC-32

Interframe Gap: Zwangspause

In einem Byte wird immer das **LSB** zuerst übertragen, d.h. beim Auslesen muss die Bitfolge wieder umgekehrt werden.

Die Byte-Reihenfolge bleibt aber gleich.

LSB MSB LSB MSB MSB LSB MSB LSB
00101110 11110000 → 01110100 00001111

$$N_{\text{Bit}} = (\text{Framesize} + 8) \cdot 8 \text{ [Bit]} + 8 \text{ Bits für Preamble und SFD}$$

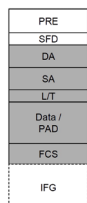
$$T_{\text{Frame}} = \frac{N_{\text{Bit}}}{\text{Bitrate } R} \text{ [s]}$$

$$\text{Framerate} = \frac{\text{Bitrate } R}{(\text{Framesize} + 20) \cdot 8} \text{ [Frame/s]} + 20 \text{ Bits für PRE, SFD und IFG}$$

$$T_{\text{Leitung}} = \frac{N_{\text{Bit}} + 96}{\text{Bitrate } R} \text{ [s]} + 96 \text{ Bits für Interframe Gap}$$

$$T_{\text{Transfer}} = \frac{\text{Distanz [m]}}{\text{Ausbreitungsgeschwindigkeit [m/s]}} \text{ [s]}$$

Framesize
64 ... 1518 Bytes



Virtuelle LANs

Data Link Layer (2)

Frame-Format VLAN

Type 0x8100 (802.1Q): Frame ist getagget.

VLAN-ID (VID): 12 Bits. Welchem VLAN das Frame angehört.

Priority Code Point/PCP: 3 Bit. Priorität des Frames
0 (hohe Prio) bis 7 (tiefe).

Drop Eligibility Indicator/DEI: Bei Überlastung darf dieses Frame zuerst verworfen werden.

Theoretische maximale Anzahl VLANs:

$2^{12} = 4096$ VLAN IDs (**12** Bit VLAN-ID/VID Identifier)

Frame-Size: 64 ... 1522 Bytes (ohne PRE, SFD und IFG)

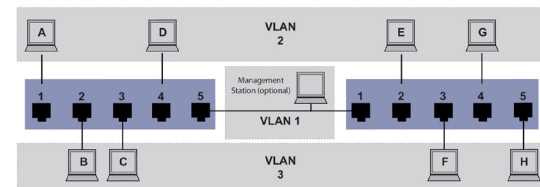
Trunk-Link und Tagged Frames

Trunk-Link = Ports zu anderen Switches.

Switches müssen «ihre» Frames beim Austritt **taggen**, wenn von diesem Port andere VLANs erreichbar sind (Access-Links = Ports zu anderen VLANs). Frames vom Management-VLAN bleiben immer untagget.

Port	Trunk	VLAN 10	VLAN 20	VLAN 30	VLAN 40	VLAN 50	VLAN 60
1		X					
2			X				
7	T	X	X	X			
8	T				X	X	X

Management-VLAN: Eine optionale Management Station kann mit richtiger Konfiguration Zugriff auf alle Netzkomponenten haben. Die Station (e.g. Computer) muss im **VLAN 1** konfiguriert sein und mit allen Switches verbunden sein.



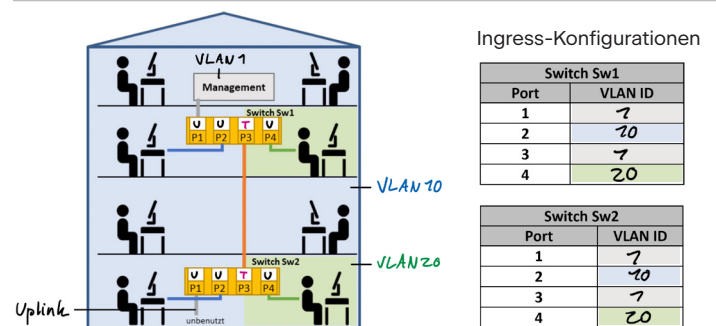
Bsp: Der Switch links muss die Frames beim Port 5 taggen, weil von diesem Port aus mehrere VLANs (VLAN 1 und VLAN 2) erreichbar sind. Gleiches gilt beim Switch rechts bei Port 1.

Eingangs-Tabelle (Ingress): Port → VLAN-ID

Legt fest, welche VLAN-ID ein Port den eingehenden Frames zuweist. Das Endgerät wird über seine Portadresse zu einem VLAN zugeordnet (Port-VLAN-ID = PVID). Z.B: Frame von Port 1 darf an VLAN 2 senden
Getaggte Ports werden immer **VLAN 1** zugewiesen.

Ausgangs-Tabelle (Egress): VLAN-ID → Ports

Legt fest, an welche Ports der Switch die Frames aus diesem VLAN senden darf (VLAN-ID, Port). Definiert auch, ob die Frames getagget werden müssen. Z.B: Frame aus VLAN 3 darf an Ports 1, 2 senden



Ethernet-Geräte (Network Gear)

Data Link Layer (2)

Repeater / Hubs (Legacy)

Verstärkt ankommende Signal auf einem Port und leitet sie weiter.
Bitfehler werden ebenfalls weitergeleitet (keine Fehlererkennung).
Arbeitet auf der ersten Schicht (PHY). Wurde durch Switches verdrängt.

Switch (Bridge)

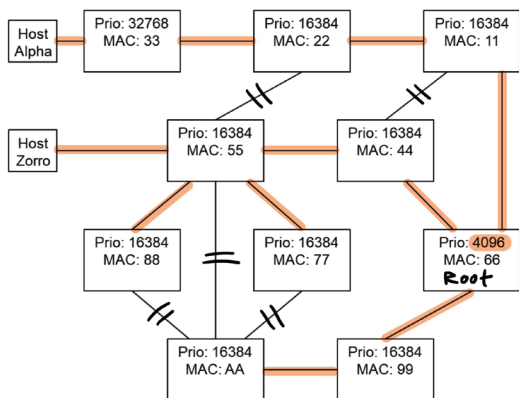
Switches bilden lokale Netzwerke (verbinden Servers miteinander).
– Prüft Checksumme und kann Layer 2 Adressen (MAC) auswerten
– Switches sind im Netzwerk für Endgeräte unsichtbar

Rapid Spanning Tree Protokoll (RSTP)

Redundante Pfade erhöhen die Zuverlässigkeit aber schaffen auch Probleme: Ein LAN darf **keine Loops** (mehrfache Verbindungen zwischen Teilnetzen) enthalten. Der Spanning Tree Algorithmus baut ein Baum (ähnlich Dijkstra) aus dem Netzwerk. Bei Erkennen von Topologieänderungen wird das gesamte Netz blockiert und ein neuer Tree aufgebaut.
RSTP ist eine Weiterentwicklung des Spanning Tree Protokolls (STP). STP benötigt bis zu 30s Konfigurationszeit, RSTP benötigt < 1s

Vorgehen:

1. Node mit **niedrigster Prio-Nr.** (= höchste Prio), ansonsten die **niedrigste Bridge-ID** als Tie-breaker, zum **Root-Node** wählen.
2. Vom Root-Node aus wird nun die **niedrigste Übertragungsgeschwindigkeit (Pfadkosten)** zu allen anderen gesucht. Gewichtung zum nächsten Node = 1, falls nichts anderes angegeben.
3. Redundante Pfade streichen.

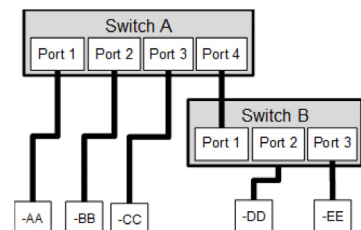


Filtering Database und Address Learning

Aufbau und Update der Filtering Database durch Verkehrsbeobachtung.
Aging Time: Für wie lange gelernte Adressen gespeichert werden.

Vorgehen:

- Es muss nur die Source-Adresse fürs Learning (Adresstabelle) berücksichtigt werden.
- Ist eine Ziel-Adresse noch nicht in der Adresstabelle, wird das Paket einfach an alle (ausser Source) geschickt.



Adresstabelle/FDB Switch A		
Zeit	Port	MAC (-XX)
0	2	BB
1	4	EE
2	4	DD
3	3	CC

Adresstabelle/FDB Switch B		
Zeit	Port	MAC (-XX)
0	1	BB
1	3	EE
2	2	DD
3	1	CC

Sende- und Empfangstabelle						
Zeit	Source	Destination	-AA	-BB	-CC	-DD
T = 0 s	00-00-00-00-00-BB	FF-FF-FF-FF-FF-FF	X	O	X	X
T = 1 s	00-00-00-00-00-EE	00-00-00-00-00-CC	X	X	X	O
T = 2 s	00-00-00-00-00-DD	00-00-00-00-00-AA	X	X	X	O
T = 3 s	00-00-00-00-00-CC	00-00-00-00-00-DD	O	O	O	X
T = 4 s	00-00-00-00-00-EE	00-00-00-00-00-BB	O	X	O	O
T = 5 s	00-00-00-00-00-DD	00-00-00-00-00-AA	X	X	X	O

Ethernet

Schema: 100BASE = 100 Mbit/s Bitrate

Relevante Eigenschaften: Taktzurückgewinnung, Gleichspannungsfreiheit, Optimale Nutzung der Bandbreite, Twisted Pair mit maximale Segmentlänge zwischen Switches von 100 m, RJ45 Stecker

Rückwärtskompatibilität und Autonegotiation

Vereinbarung verwendeter Konfigurationen zwischen Stationen.
Ethernet-Anschlüsse unterstützen von mehreren Bitraten. Wenn zwei Ethernet-Geräte verbunden werden, beginnt der Autonegotiation-Prozess. Jedes Gerät sendet eine Reihe von FLP-Bursts (Fast Link Pulses), die ihre eigenen Fähigkeiten (unterstützte Geschwindigkeiten und Duplex-Modi) kodieren.

Massnahmen für Rückwärtskompatibilität:

Adressaufbau, Autonegotiation, Fast Link Pulse, Frame Format

Wie viele Frames pro Sekunde können auf einem 10Base Ethernet Segment maximal übertragen werden?

$$10 \text{ Base} = 10 \text{ Mbit/s} = 10 \cdot 10^6 \text{ Bit/s} \text{ Bitrate}$$

$$\text{Framesize} = 84 \text{ Byte} \cdot 8 \text{ Bytes (inkl. Interframe Gap)}$$

$$\text{Frames/s} = \frac{\text{Bitrate [bit/s]}}{\text{Framesize [bit]}} = \frac{10 \cdot 10^6}{8 \cdot 84} = \frac{1}{T_{\text{Frame}}} \approx 14887$$

$$T_{\text{Frame}} = \frac{\text{Framesize [bit]}}{\text{Bitrate [bit/s]}} = \frac{8 \cdot 84}{10 \cdot 10^6} = \frac{1}{\text{Frames/s}} = 67.2 \mu\text{s}$$

Network Layer / Vermittlungsschicht (3)

Internetschicht / Netzwerkschicht / Leitungsvermittlung und Paketvermittlung

Aufgaben:

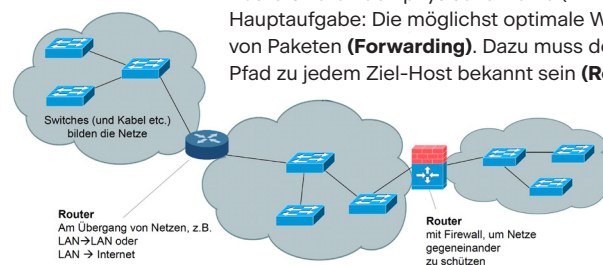
Adressierung (Wohin), Wegfindung (Route), Weiterleitung/Senden der Daten;
IP-Adressen, Routing, IPv4-Pakete, Routing-Tabellen;
Diensprogramme zu IPv4: ARP, ICMP

Key-Takings

- Funktionsweise:
 1. Verbindung von Netzen mit unterschiedlichen Technologien
 2. Die beste Route finden: Kümmt sich nur um den Transport der IP-Pakete (Korrigieren von fehlerhaften Übertragung ist nicht Aufgabe).
- **Netzwerkschicht** verbindet lokale Netze (LANs) mittels Router zu einem grossen virtuellen Netzwerk (= Internet). Zwei **Hauptaufgaben**:
 1. **Routing:** Bestimmung der Wege durch das Netz (Routingtabellen)
 2. **Forwarding:** Weiterleitung von Packets gemäss der Routingtabellen
- **Routingtabellen** definieren, über welche Netzwerk-Interfaces und Router welche Netze erreichbar sind.
- **TCP/IP:** IP bietet einen **verbindungslosen, unzuverlässigen Dienst**, TCP bietet einen **verbindungsorientierten, zuverlässigen Dienst**
- **Sender und Empfänger** werden im TCP/IP Referenzmodell als **Hosts** bezeichnet

Router (Gateway)

Routers verbinden physische Netze (LANs).
Hauptaufgabe: Die möglichst optimale Weiterleitung von Paketen (**Forwarding**). Dazu muss der optimale Pfad zu jedem Ziel-Host bekannt sein (**Routing**).



Der Router hat zu jedem der Netze ein Interface (Anschluss) der entsprechenden Technologie (muss alle Protokolle der Netze verstehen). Ein Router hat **eine Hardwareadresse (MAC)** sowie **eindeutige IP-Adressen** pro Port.

Modem:

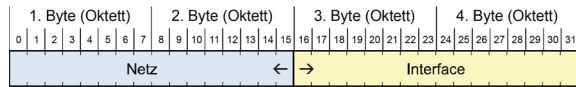
- Verbindet einen Host (Home-Device) via ISP zum Internet.
- «Modelliert» Daten von analog nach digital (und vice versa).
- Enthält oft Router und Switch und Wireless Access Point (WIFI).
- Operiert somit auf OSI-Layern 1 bis 3.

IPv4-Adressen (32 Bit)

Network Layer / Vermittlungsschicht (3)

IP-Adressen müssen **global eindeutig** sein.

2³² (ca. 4.3 Milliarden) Adressen. Bereits seit 2015 aufgebraucht.



Netzadresse = IP-Adresse **AND** Netzmaske
Broadcast = IP-Adresse **OR** Inverse Netzmaske

Anzahl Host-Adressen

2^{Interface-Bits - 2} (-2 für ohne Netz- und Broadcastadresse)

	00000000	00000000	Maske	Hostadressen
1000	128	8	0001 16 1 /24 .0000 0000	254
1001	144	9	0010 32 2 /25 .1000 0000	126
1010	160	10	A 0011 48 3 /26 .1100 0000	62
1011	176	11	B 0100 64 4 /27 .1110 0000	30
1100	192	12	C 0101 80 5 /28 .1111 0000	14
1101	208	13	D 0110 96 6 /29 .1111 1000	6
1110	224	14	E 0111 112 7 /30 .1111 1100	2
1111	240	15	F 0111 112 7 /31 .1111 1110	0

Bit-Wert	2048	1024	512	256	128	64	32	16	8	4	2	1
Bit	12	11	10	9	8	7	6	5	4	3	2	1
Zustände	4096	2048	1024	512	256	128	64	32	16	8	4	2

Wert der Host-Adresse muss zwischen **Netzadresse** und **Broadcast** liegen, sonst fremde Adresse (für das Netzwerk).

160.85.15.130 /25
 10100000.01010101.00001111.10000010 IP-Adresse
 11111111 11111111 11111111 10000000 Netzmaske
 10100000.01010101.00001111.10000000 Netzadresse
 .10000001 Host-Adressen
 .10111111 Broadcast
 10100000.01010101.00001111.11111111 Broadcast

IP-Adresse und Maske → Broadcast und Netzadresse

Geg: 160.85.15.130 /25 (IP-Adresse /Maske)

130 ⇒ 128 + 2 ⇒ 1000 0010

Broadcast: 1000 0010 | 0111 1111 = 1111 1111 = 255
 = 160.85.15.255

Netzadresse: 1000 0010 & 1000 0000 = 1000 0000 = 128
 = 160.85.15.128

IP-Adresse und Netzadresse → Maske und Broadcast

Geg: 11.7.177.4 (IP), 11.7.160.0 (Netz)

160 ⇒ 128 + 32 ⇒ 1010 0000 (Netz)

177 ⇒ 128 + 32 + 16 + 1 ⇒ 1011 0001 (IP)

1010 0000 (IP)
 1011 0001 (Netz) ⇒ Maske = 11 = 255.255.224.0
 1110 0000 (Maske)

Broadcast ⇒ 1011 0001 | 0001 1111 = 1011 1111
 = 11.7.177.255

Routing

Aufbau der Routingtabelle durch (1) **statische Konfiguration** oder (2) **dynamisch durch Routing-Protokolle**.

Die **Routingtabellen** definieren, über welche Netzwerk-Interfaces und Router welche Netze erreichbar sind. Enthält Informationen, wie jedes Netz (über welches Interface) erreicht werden kann. **Forwarding**: Weiterleiten von Paketen aufgrund Informationen in der Routingtabelle.

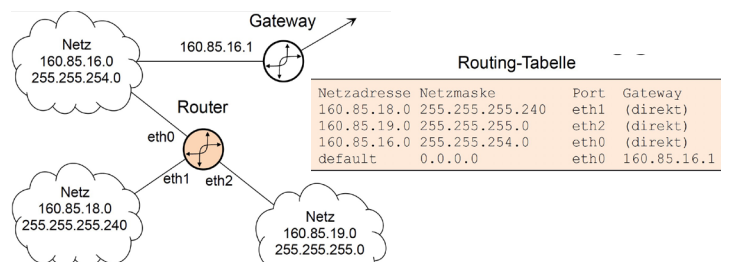
Flaches Routing:

Router kennt **explizite Wege** zu jedem einzelnen Zielnetz. Pakete an unbekannte Netze werden verworfen. Einsatz in stark vermaschten Netzen oder im zentralen Bereich. Grosse Routing-Tabellen.

Hierarchisches Routing (Default Routing):

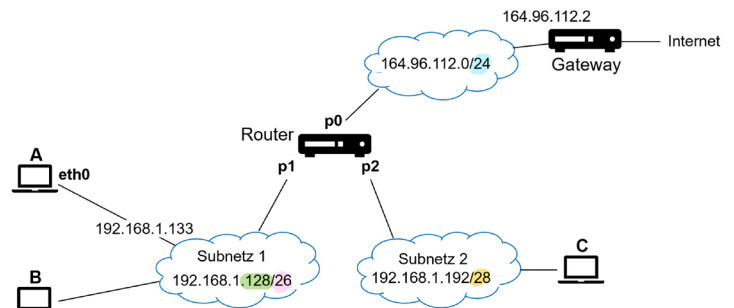
Router kennt **direkt angeschlossenen** Netze. Einsatz am Rand von Netzen (Hosts, Access Router). Kleine Routing-Tabellen mit jeweils einem Default-Eintrag. Der **Default-Eintrag** definiert wohin IP-Pakete geroutet werden sollen, für die kein Eintrag in der Routing Tabelle passt.

Vorgehen: Aus der **Zieladresse** (IP) und **Netzmaske** eine **Ziel-Netzadresse** bestimmen und mit den **allen Netzadressen** in der Tabelle von oben bis unten vergleichen. Der **Default-Eintrag** am Schluss **passt immer**. Hosts sowohl als Router haben Routing-Tabellen. Einträge sind nach Netzmasken absteigend geordnet.



Zieladresse & Maske = Netzadresse ?

Zieladresse Maske Netzadresse
 160.85.19.1 & 255.255.255.240 (/26) ≠ 160.85.18.0 **passt nicht**
 160.85.19.1 & 255.255.255.0 (/24) = 160.85.19.0 **passt!** → eth2



Routing-Tabelle für den Router

Netzadresse	Netzmaske	Port	Gateway
192.168.1.192	/28	p2	direkt
192.168.1.128	/26	p1	direkt
164.96.112.0	/24	p0	direkt
default	/	p0	164.96.112.2

Routing-Tabelle für den Linux-Host A

Netzadresse	Netzmaske	Port	Gateway
192.168.1.128	/26	eth0	direkt
default (0.0.0.0)	/	eth0	192.168.1.129

.129 für Router-Adresse = Subnetzadresse + 1

Classful Routing

Klasse A: /8 Klasse B: /16 Klasse C: /24

Problem: Klassische Netze fixer Grösse (Klassen) sind unflexibel.

Abhilfe schafft CIDR

Classless Inter-Domain Routing (CIDR)

Flexible Verwendung von Netzmasken beliebiger Länge.

Vorteile: Dynamischer Adressbereich

Nachteile: Mühsamer zum berechnen und grössere Routing-Tabelle, Subnetzmaske nicht klar identifizierbar

Supernetting

Zusammenfügen von kleinen Netzen.

Lokales Netz (Interface-Adressen) wird grösser.

Gibt **weniger Einträge** in der Routing-Tabelle.

198.51.0110 0100 0000 0000 = C-Netz 198.51.100.0 /24
198.51.0110 0101 0000 0000 = C-Netz 198.51.101.0 /24
198.51.0110 0110 0000 0000 = C-Netz 198.51.102.0 /24
198.51.0110 0111 0000 0000 = C-Netz 198.51.103.0 /24

198.51.0110 01 00.0000 0000 = Subnetzmaske 255.255.252.0 oder /22

Subnetting

Aufteilung eines Netzes in kleinere Netze.

Lokales Netz (Interface-Adressen) wird kleiner.

Gibt **mehr Einträge** in der Routing-Tabelle.

3 Bits: $2^3 = 8$ Subnetz-Adressen mehr aber 8 LAN-Adressen weniger.

Netzadresse: 160.85.17.0 /24

Adressbereich: .00010001.00000000 bis .11111111 = 256 Adressen (2^8)

Subnetz 1 mit 32 Adressen = 5 Bits ($2^5 = 32$)

160.85.17.0 /27

Adressbereich: .00000000 bis .00011111
.0 .31 (.0+31)

Subnetz 2 mit 16 Adressen = 4 Bits ($2^4 = 16$)

160.85.17.0 /28

Adressbereich: .00100000 (.00011111+1) bis .00101111
.32 (.31+1) .47 (.32+15)

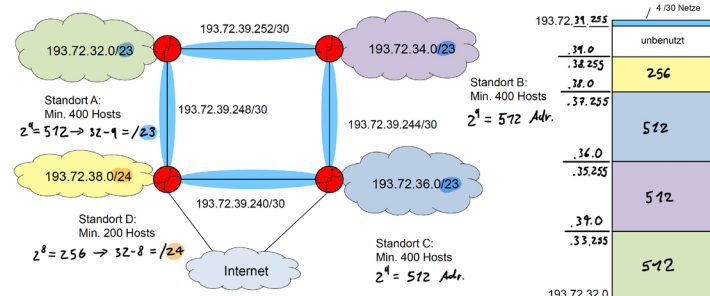
Subnetz 3 mit 16 Adressen: .00110000 (.00101111+1) bis .00111111
.48 (.47+1) .63 (.48+15)

Subnetz 4 mit 16 Adressen: .01000000 (.00111111+1) bis .01001111
.64 (.63+1) .79 (.64+15)

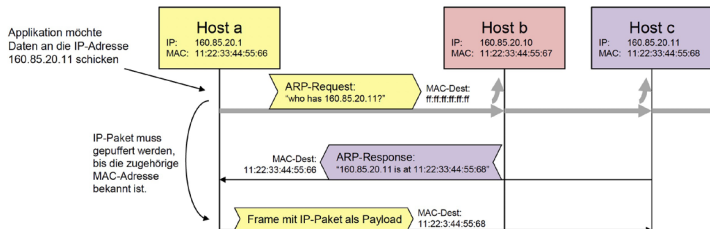
...
Subnetz 8 mit 16 Adressen: .11110000 (.11101111+1) bis .11111111

	Adressen	Mask	Netz	Broadcast
Subnetz 1:	$2^6 = 64$	/26	172.30.10.0	172.30.10.63 (.0+63)
Subnetz 2:	$2^5 = 32$	/27	172.30.10.64	172.30.10.95 (.64+31)
Subnetz 3:	$2^5 = 32$	/27	172.30.10.96	172.30.10.127 (.96+31)

Ein KMU mit 4 Standorten hat von seinem ISP das Netz 193.72.32.0 /21 erhalten.
Das KMU hat 3 grössere und einen kleineren Standort und will diese redundant verbinden.



ARP Grundprinzip (rechts)



IPv4

Network Layer / Vermittlungsschicht (3)

IP-Paket Header

IP-Pakete bestehen aus einem Header (20 Byte) und Nutzdaten (Padding).

IP-Datagramm 46-1500 Byte.

Maximum Transfer Unit (MTU) = Maximale Payload (in Bytes)

1. Byte (Oktett)	2. Byte (Oktett)	3. Byte (Oktett)	4. Byte (Oktett)
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	DiffServ (DS)	Total Length
Identification Number		Flags	Fragment Offset
Time to Live		Protocol	IP Header Checksum
IP Source Address			
IP Destination Address			
Optionen		Padding	

Protocol (8 Bit): Übergeordnetes Protokoll der Nutzdaten

Protocol	Name
1	ICMP – Internet Control Message Protocol
6	TCP – Transport Control Protocol
17	UDP – User Datagram Protocol

Time to Live (TTL) (8 Bit): 1 Byte im Header. Verbleibende Zeit für ein Paket. Router dekrimieren diesen Wert beim Weiterleiten (Hop-Counter).

Verhindert endlose Zirkulation.

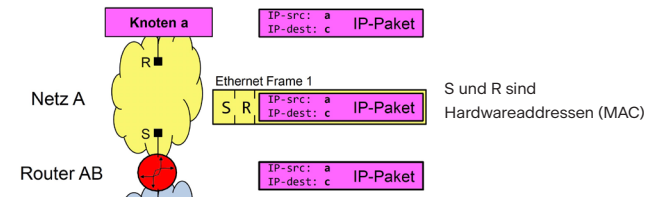
IP Header Checksum (16 Bit): Verhindert fehlgeleitete Pakete.

Options / Padding: Um auf ein Vielfaches von 32 Bits bzw. 4 Byte aufzufüllen.

Kapselung

Das IP-Paket wird im Nutzteil (data/PAD) des **Ethernet-Frames** eingebettet.

Das Type Feld des Ethernet Frames ist auf 0x0800 (für IP) gesetzt.



Damit Knoten a ein Paket versenden kann, muss er zuerst die MAC-Adresse von Router AB kennen → **Adressauflösung**

Adressauflösung / Address Resolution Protocol (ARP)

Gehört zu Data Link Layer (2)

Umsetzung **IPv4-Adresse** ↔ **Hardwareadresse (MAC)**

1. Nachdem die Zielnetzadresse aus der **Routing-Tabelle** bestimmt wurde, wird mittels ARP-Request nach der MAC-Adresse gefragt (Broadcast).

Ein Sender muss zusätzlich zur IP-Adresse des Ziel-Knoten auch seine Hardwareadresse kennen. Ist diese nicht bekannt, werden **alle Knoten per Broadcast** (MAC-Dest: ff:ff:ff:ff:ff:ff) angefragt.

2. **ARP-Request:** Host fragt «Who has 160.85.20.11?» an alle (Broadcast)

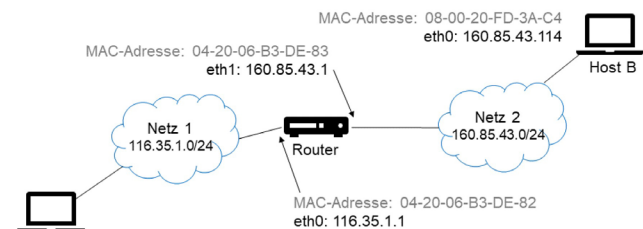
Der Host mit der angefragten IP-Adresse sendet seine MAC zurück:

3. **ARP-Response:** «160.85.20.11 is at 12:34:56:78:98:76»

4. Ethernet-Frame mit IP-Paket als Payload

Die **ARP-Tabelle (Cache)** speichert IP-MAC Kombinationen für einige Minuten.

Bsp: Host A sendet ein Ethernet-Frame an Host B



MAC-Adresse: 60-F2-62-C1-B8-61
IP-Adresse: 116.35.1.41
default Gateway: 116.35.1.1

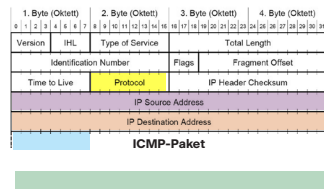
Netzwerk	MAC-source	MAC-destination	IP-source	IP-destination
Netz 1	B8-61	DE-82	116.35.1.41	160.85.43.114
Netz 2	DE-83	3A-C4	116.35.1.41	160.85.43.114

Internet Control Message Protocol (ICMP)

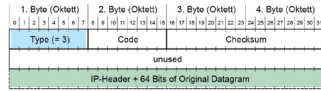
Übtragung für Fehlermeldungen oder Informationsaustausch.

ICMP Meldungen werden in IP-Datenbereich gekapselt.
Es gibt keine Garantie, dass die Meldung je ankommt.

IP-Paket



ICMP-Paket



«Anhang» wird durch **Protocol** bestimmt.

```
4500 0038 8038 0000 fd01 5bc0 a055 821e IP-Header
a055 1f03 0301 4bf7 0000 0000 4500 0028 ICMP-Paket
8b10 0000 0711 a8a4 a055 1f03 a055 1d63
8b0d 829d 0014 a348
```

Meldungstypen:

ICMP-Typ	Bedeutung (Fehler)	ICMP-Typ	Bedeutung (Information)
3	Destination Unreachable	0	Echo Reply
5	Redirect	8	Echo
11	Time Exceeded	13	Timestamp
12	Parameter Problem: Bad IP Header	14	Timestamp Reply

} ping

Ping: Prüft die Erreichbarkeit und Round-Trip Zeit eines Routers/Hosts.

Traceroute: Erlaubt den Weg zu einem beliebigen Host zu finden:

- Es werden UDP Pakete an den Zielhost gesendet, meist an Port 33434
- Das TTL Feld wird, von 1 ausgehend, jeweils um 1 erhöht
- Jeder Router sendet eine ICMP Time Exceeded Message, weil TTL (Time to live, Type = 11) 1 = 0
- Der Zielhost sendet eine ICMP Destination Unreachable Message, weil keine Applikation an diesem Port horcht

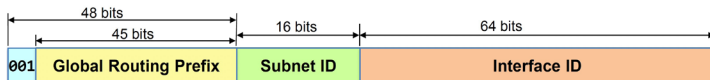
IPv6

Network Layer / Vermittlungsschicht (3)

IPv6 Adresse (128 Bit)

$2^{128} = 3.4 \cdot 10^{38}$ Adressen

Maximal 2^{64} Geräte im lokalen Netzwerk (64 Interface Bits).



Darstellung der 128 Bit in **Doppel-Bytes** (1 Stelle/Hex-Wert = 1 Nibble):

2001:0620:0000:0004:0A00:20FF:FE9C:7E4A

Verkürzte Notation:

In Zahlen können **links stehenden Nullen** weggelassen werden:

2001:0620:0000:0004:0A00:20FF:FE9C:7E4A und

2001:620:0:4:A00:20FF:FE9C:7E4A sind gleichwertig

Eine Nullfolge kann durch **2 Doppelpunkte** dargestellt werden

1023:0000:0000:0000:1736:a673:88a0:a620 und

1023::1736:a673:88a0:a620 sind gleichwertig

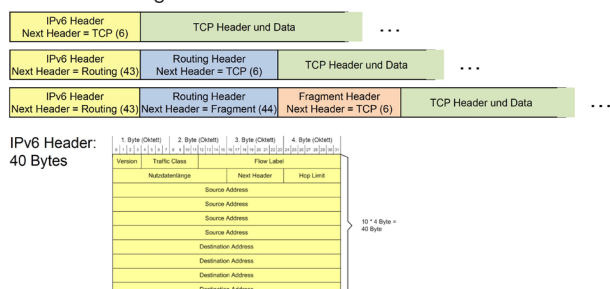
Anzahl Nullstellen muss anhand der Byte-Anzahl vor und nach dem Doppelpunkt berechnet werden.

NDP (Neighbor Discovery Protocol) anstelle **ARP**.

IPv6 ermittelt immer zuerst die MTU des gesamten Pfades.

Extension Headers

Variable Headergröße. «Verkettete Liste» von Headers.



Transport Layer / Vermittlungsschicht (4)

Aufgabe: Verbindung des Betriebssystems (Kernel Space) mit Applikation/ Dienst (User Space) mittels Schnittstelle **TCP/UDP Sockets** unter der Verwendung von **Port Nummern** um Applikationen zu identifizieren.

TCP (6): Verbindungsorientiert

Zuverlässig

Leitungsvermittlung,
Voll duplexübertragung möglich,
Multicast/Broadcast gibt es nicht

Kommunikationsphasen:

- Verbindungsaufbau
- Datenaustausch
- Verbindungsabbau

Aushandeln von Optionen

beim Verbindungsaufbau:

Puffergröße, Verschlüsselung

Reihenfolge der Daten bleibt erhalten

Z.B. Email, Banktransaktion

UDP (17): Verbindungslos

Unzuverlässig

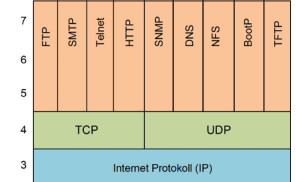
Paketvermittlung/Paketorientiert

Kommunikationsphasen:

Datenaustausch (**send & forget**)

Effizient, keine Informationen über Fähigkeit des Empfängers.

Z.B. Videostreaming



Multiplexing / Demultiplexing

Kommunikation zwischen Applikationen. Der Vorgang, lokal Daten an verschiedene Instanzen höherer Protokollschichten zu verteilen. Findet auf allen Schichten statt:

Type im Ethernet-Header (0x0800 = IP)

Protocol im IP-Header (6 = TCP, 17 = UDP)

Port bei TCP/UDP-Header (z.B. 443 = HTTPS)

Port Nummer

Identifizierung und Adressierung der (Server-) Applikation durch Destination Port. Source Port wird meist zufällig vom Betriebssystem vergeben. Logische Verbindung zum Austausch von Daten zwischen Services/Apps.

- Well-known Ports** (1 – 1023): Verwaltet und standardisiert durch IANA
- Registered Ports** (1024 – 49'151): Für herstellerspezifische Applikationen
- Dynamic / Private Ports** (49'152 – 65'535): Für private Zwecke

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

Eindeutige Kommunikationsbeziehung durch:

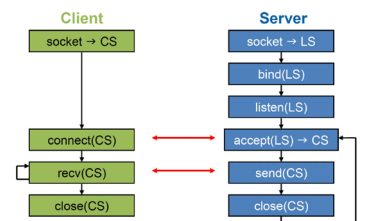
- Source- und Destination IP-Adressen
- Source- und Destination Port-Nummer
- Protocol (UDP oder TCP)

Server: Rechner, stellt Dienste (Server-Anwendungen / Applikationen), z.B. Webserver, FTP, etc., zur Verfügung. **Wartet** bis von Client kontaktiert.

Client: Rechner, nimmt Dienste mittels Client-Anwendung, z.B. Webbrowser, Email, etc.) eines Servers, in Anspruch. Ist immer **Initiator** einer Kommunikationsbeziehung.

Socket-Interface

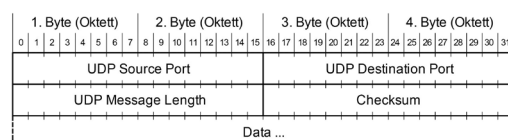
Socket = Ein vom Betriebssystem bereitgestelltes Objekt (File), das als Kommunikationsendpunkt dient. Ein Programm verwendet Sockets, um Daten mit anderen Programmen über ein Netzwerk auszutauschen.



Ein geschlossener Port kann erst wieder nach ca. 2 Minuten verwendet werden. Es stehen $65'536 (2^{16})$ Ports zur Verfügung ≈ 650 Ports/s

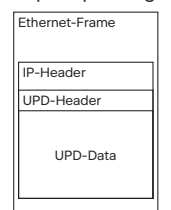
UDP

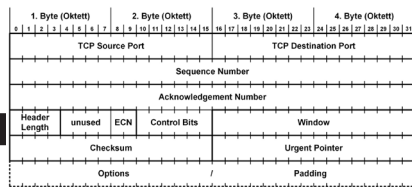
Transport Layer / Vermittlungsschicht (4)



Message Length: Länge des Daten (inkl. Header) in Bytes. Max. Länge: 65'535 Bytes

Bsp. Kapselung:





TCP

Transport Layer (4)

Gewährleistung einer **sicheren Kommunikation**:

- Jedes Byte ist eindeutig identifizierbar mit einer **32-Bit Sequenznummer**
- Zufällige Sequenznummer bei Initialisierung
- 3-Wege Handshake** beim Verbindungsaufbau
- Adaptive Timeouts** basierend auf Wert und Varianz der gemessenen Round-Trip Zeiten (RTT)
- Schutz des Empfängers durch dynamische Anpassung der Fenstergröße durch **Sliding Window Protocol** und **Advertised Window**
- Schutz des Netzwerks vor Überlastung durch **Congestion Control** (beim Sender) mit dem **Slow Start Algorithmus**

Probleme:

- Verbindungsaufbau- und Abbau
- Korrekte Reihenfolge → **Sequenz-Nr., Adaptiver Timeout, Sliding Window**
- Empfänger soll nicht mit Daten überschwemmt werden, der Sender soll sich an die Möglichkeiten des Empfängers anpassen
→ **Flow Control** mit **Advertized Window Size**
- Vernünftiger Durchsatz, Netz soll nicht überlastet werden
→ **Congestion Control** mit **Slow Start Algorithmus**

TCP Verbindungsphasen:

Control Bits (Flags):

SYN/FIN: Verbindungsaufbau- und Abbau

ACK: Acknowledge Nummer (empfangenes Segment ist gültig)

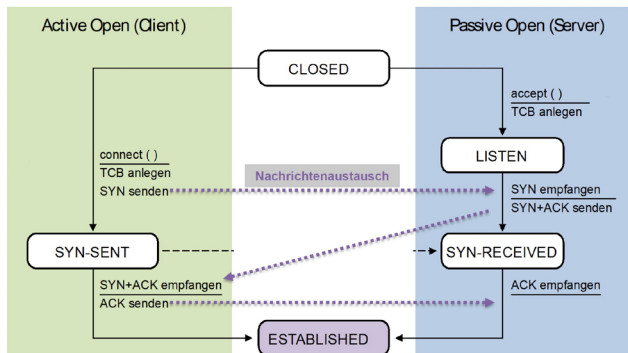
PSH: Daten sollen schnellstmöglich an Dienst weitergegeben werden.

- Verbindungsaufbau:** Aushandeln von Optionen, Initialisierung von Sequenznummern und Reservation von Ressourcen (Buffer, Variablen)
 - Datenaustausch:** Sequenznummern sichern korrekte Reihenfolge, explizite Bestätigung empfangener Daten
 - Verbindungsabbau:** Freigabe der Ressourcen
- Beide Seiten können den Verbindungsabbau einleiten.

TPC Zustandsdiagramme

Verbindungsaufbau:

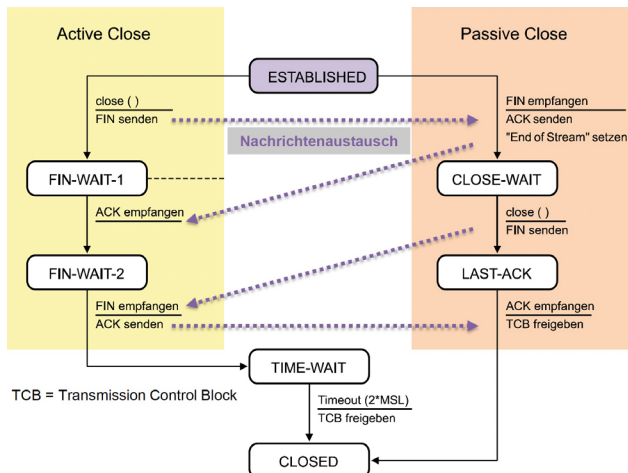
3- Wege Handshake



Verbindungsabbau:

Client oder Server

Server oder Client



TCB = Transmission Control Block

Time-Wait: Der «Closing-Host» wartet eine gewisse Zeit, im Fall eines Übertragungsfehlers (ACK nicht erhalten) die Verbindung regulär zu schließen.

Seq = [letzte Seq + Datenbytes (oder +1 für SYN/FIN)] des selben Knoten
Ack = [letzte Seq + Datenbytes (oder +1 für SYN/FIN)] des anderen Knoten

Die ersten Sequenznummern werden zufällig gewählt.

Sender	Flags	SeqNr	AckNr	Datenlänge	StateMachine Zustand (Sender)
Client	Syn	3000	-	0	Syn-Sent
Server	Syn+Ack	5000	3001	0	Syn-Received
Client	Ack	3001	5001	0	Established
Server	Ack	5001	3001	28	Established
Client	Ack	3001	5029	0	Established
Server	Fin+Ack	5029	3001	0	Fin-Wait 1
Client	Ack	3001	5030	0	Close-Wait
Client	Fin+Ack	3001	5030	0	Last-Ack
Server	Ack	5030	3002	0	Time-Wait

TCP: Adaptive Elemente

Transport Layer / Vermittlungsschicht (4)

- Round Trip Time (RTT):** Erkennung verloren gegangene Paketen
- Flow Control (Fluss-Steuerung):** Überlast der Empfänger schützen
- Congestion Control (Überlast-Steuerung):** Überlast des Netzes schützen

Flow Control (Fluss-Steuerung)

Schützt den Empfänger vor Überlast. Problem: Daten werden schneller geschickt, als diese vom Empfänger verarbeitet werden können:

Empfangsbuffer überläuft und Daten gehen verloren.

→ Sender wartet Retransmission Timeout (RTO) und sendet Daten erneut.

a) Stop and Wait:

Sender wartet auf Bestätigung des Empfängers. Empfänger sendet Bestätigung erst nach Verarbeitung der Pakete. Problem: Ineffizient

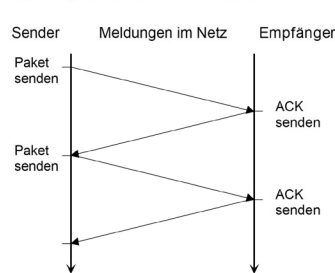
Bsp: **Paketgröße 1 kByte, Bitrate 2 Mbit/s** (= 2000 kbit/s),
 Ende-zu-Ende Verzögerung (RTT) 50 ms

- Höchstens (= RTT) alle $2 \cdot 50 \text{ [ms]} = 100 \text{ ms}$ kann ein Paket gesendet
- Durchsatz:** $1000 \cdot 8 \text{ [bit]} / 0.1 \text{ [s]} = 80 \text{ kbit/s}$, **2000 kbit/s** wären möglich
- Auslastung:** $80 \text{ [kbit/s]} / 2000 \text{ [kbit/s]} \sim 0.04 = 4\%$

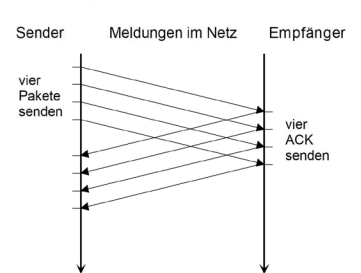
b) Sliding Window:

Wird im Window-Feld des TCP-Headers übermittelt. Mit jedem ACK wird die verfügbare Fenstergröße (= Buffer in Bytes) vom Empfänger dynamisch angepasst und dem Sender mitgeteilt. **0 Bytes** bedeutet, dass vorerst keine weiteren Daten gesendet werden dürfen.

a) Stop and Wait Flow Control



b) Sliding Window Flow Control



$$\text{Fenstergröße BDP [byte]} = \frac{\text{Bandbreite BB [bit/s]} \cdot \text{RTT [s]}}{8}$$

$$\text{Bandbreite BB [bit/s]} = \frac{\text{Fenstergröße BDP [byte]} \cdot 8}{\text{RTT [s]}}$$

$$\text{RTT [s]} = \frac{\text{Fenstergröße BDP [byte]} \cdot 8}{\text{Bandbreite BB [bit/s]}}$$

BB = Bandbreite, Datenrate, Anz. Pakete unterwegs bis ACK erhalten [bit/s, Hz]

BDP = Bandwidth-Delay-Product, Fenstergröße, Buffergröße, Advertised Window Size, Pakete unterwegs auf einmal [byte]

RTT = Round Trip Time [s]

Bsp: Flow Control (Fluss-Steuerung)

Geg: **Fenstergrösse 500 kByte**, **Datenrate 100 MBit/s**,
Round TripTime (RTT) 100 ms

1. Welche **Bandbreite** können Sie ausnutzen ?

$$BB [bits/s] = \frac{BDP [byte] \cdot 8}{RTT [s]} \Rightarrow \frac{500 \cdot 10^3 \cdot 8 [bits]}{100 \cdot 10^{-3} [s]} = \underline{40 \text{ MBit/s}}$$

2. Wie gross müsste die **Fenstergrösse** optimal sein?

$$\text{Fenstergrösse } BDP [byte] = \frac{\text{Datenrate } BB [bit]}{8} \cdot RTT [s]$$
$$\Rightarrow \frac{100 \cdot 10^6 [bits]}{8} \cdot 100 \cdot 10^{-3} [s] = \underline{7.25 \text{ Mbyte}}$$

Congestion Window/Control (Überlastfenster)

= Fenstergrösse des Senders

Sendefenster = min { Congestion Window, Advertised Window }

Schützt vor Überlast des Netzes. Funktion im Sender.
Limitiert die Grösse des Sendefensters.

Slow Start: Wahl der Fenstergrösse. Sender testet die
Fenstergrösse exponentiell, danach linear. Deprecated.

Auf Empfängerseite (Server): Empfänger teilt dem
Sender die **Advertised Window Size**

Auf Senderseite (Client): Ermittlung der Sendefenster:

Sendefenster = min { Congestion Window, Advertised Window }

Retransmission Timeout (RTO)

Nachricht wird vom Absender erneut geschickt wenn innerhalb
Timeout keine Bestätigung (ACK) vom Empfänger erhalten.

Timeout wird **dynamisch** bestimmt / berechnet → **adaptive** Anpassung
Wegen der Umlaufverzögerung macht ein allgemein gültiger Wert keinen Sinn.

■ TCP misst bei jeder aktiven Verbindung die **Round-Trip Time (RTT)**

■ **Gewichteter Mittelwert SRTT (Smoothed Round-Trip Time):**

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

■ **Streuung RTTVAR** ist gewichteten Mittelwert der Abweichungen:

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

■ **Retransmission Time-Out RTO:**

$$RTO = SRTT + 4 * RTTVAR$$

Application Layer (7)

Key-Takings

- Anwendungsprotokolle (HTTP, Mail, DNS) erlauben den Datenaustausch für bestimmte Anwendungen.
- Dienste können unabhängig von Betriebssystem und Client-Programmen verwendet werden.

Domain Name System (DNS)

Namensauflösung (Name Resolution). Übersetzt Hostnamen (z.B. zhaw.ch) in IP-Adressen und umgekehrt (mittles TCP/UDP): **Hostname ↔ IP-Adresse**

Ein Name-Server (NS) ist jeweils für eine Zone verantwortlich. Ein NS kennt: IP-Adressen zu den Hostnamen in seiner Zone, IP-Adressen seiner Subdomäne, IP-Adressen von Root und Top-level Domain (TLD) Name Server um beliebige Abfragen zu erlauben

Eine Domain muss eindeutig sein = Fully Qualified Domain Name (FQDN).
Top-Level Domains: .ch, .com, etc.

Domain → IP: Eine Domain wird rückwärts von der Root her aufgelöst

\$ dig **ch**. NS oder z.B. dig ch. NS @1.1.1.1
ch. NS **a.nic.ch**.

\$ dig **zhaw.ch**. NS @**a.nic.ch**.
zhaw.ch. NS **ns1.zhaw.ch**.

\$ dig **www.zhaw.ch**. A @**ns1.zhaw.ch**.
www.zhaw.ch. CNAME srv-clst-301-data168.zhaw.ch.
srv-clst-301-data168.zhaw.ch. A **160.85.104.112**

Nameserver (NS): Server, welche die Namensauflösung verwaltet.

Bsp: 1.1.1.1 (Cloudflare), 8.8.8.8 (Google), /etc/hosts (lokal)

DNS Record-Typen:

A	IPv4 Adresse des gesuchten Hosts (32 Bit)
AAAA	IPv6 Adresse des gesuchten Hosts (128 Bit)
MX	Mail Exchange (Mail Server)
CNAME	Canonical Name (primärer Name) für einen Alias zum Host.
NS	Name Server (Name Server Name für eine Zone)
TXT	Text Record, in Antworten für verschiedenste Angaben verwendet

Dynamic Host Configuration Protocol (DHCP)

Erlaubt einem Rechner, seine IP-Adresse von einem Server/Router **dynamisch** zu beziehen (mittles UDP). Der Router vergibt die IP-Adressen im lokalen Netzwerk.

Lease-Time: DHCP-Server erteilt ein freie Adresse für einen definierten Zeitraum, of 10 Minuten (DHCP-Response). **Ablauf:**

1. **DHCP discover:** Client sucht eine DHCP-Server mittels Broadcast.
2. **DHCP offer:** Ein oder mehrere DHCP-Server antworten.
3. **DHCP request:** Der Client wählt einen Server und fordert eine Auswahl der angegebenen Paramter an.
4. **DHCP acknowledge:** Der Server bestätigt die endgültigen Paramter
5. Vor Ablauf der **Lease Time** erneuert der Client die Adresse mittels Unicast.

Port Mapping (NAT, Network Address Translation)

Router mappet die lokale IP/Port zu seiner externen IP mit neuem Port.
Erlaubt den Zugriff vom Internet auf lokale Hosts/Dienste. Nur für IPv4 nötig.
Erlaubt die Wiederverwendung privater IP-Adressen.

<lokale-IP>:<lokaler-Port> ↔ <externe-IP>:<externer-Port>

Bsp: 192.168.1.100:6667 ↔ 160.85.16.2:12345

Ist kein NAT aufgesetzt, verwendet der Router die lokale IP → Die Antwort des Empfängers geht verloren!

Dynamisch: Für ausgehende Pakete. Der Router erstellt die NAT-Tabelle dynamisch. Routing vor Mapping (**Post-Routing**).

Statisch (Port Forwarding): Für eingehende Pakete. Benutzerdefiniertes Mapping. Routing erst nach Mapping (**Pre-Routing**).

HTTP

Darstellungsschicht / Presentation Layer (6)

Erlaubt den Zugriff auf verteilte Dokumente. Dokumente und Ressourcen werden mittels URL (Uniform Resource Locator) eindeutig adressiert.

Funktionsweise von HTTP

- Basiert auf **TCP, Port 80** (HTTPS auf Port 443)
- ASCII-basiert, **MIME**-Typen (Multipurpose Internet Mail Extensions), MIME schaffen Kompatibilität für zusätzliche Zeichen sowie für Multimedia, erlaubt dem Sender beliebige Daten zu senden
- Transaktion: **HTTP Request** (des Clients) → **HTTP Response** (des Servers)
- **Stateless** (zustandslos): Kein Bezug zwischen Transaktionen

Request Methoden

GET: Fordert vom Server Daten

POST: Schickt Input-Daten an einen Prozess auf dem Server

HEAD: Wie GET, der Server schickt aber nur den Header der Response ohne Daten. Oft zum Testen verwendet.

OPTIONS: Welche Request Methoden der Server akzeptiert.

Weitere: **DELETE, PUT, PATCH**

\$ curl -X GET https://en.wikipedia.org/wiki/HTML \\\n-H "Accept-Language: de, en"

HTTP/2 200

Date: Mon, 20 May 2024 13:08:59 GMT

server: Apache/9.6.3 (unix)

Content-Type: text/html

Content-Length: 1264

Wrap-Up

Ich mach eine Anfrage an Google stellen. Zuerst schalte ich den PC ein:

Ethernet-Link

1. FLP: Autonegotiation, Fast Link Pulse, **Rechner** binden sich an **Switch**.

Netzwerkconfiguration

2. DHCP: **Rechner** fragt **Router** nach seiner IP-Adresse.
3. ARP: Adressauflösung (MAC vom Zielhost)
4. DNS: IP von Google wird abgefragt.
5. HTTP: Pakete werden mittels TCP verschickt