

Übertragungsmedien

Signale

Ausbreitungsgeschwindigkeit

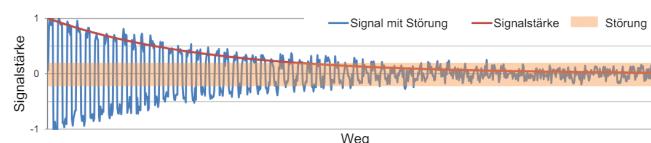
Lichtgeschwindigkeit im Vakuum: $c_0 = 299'792'458 \text{ m/s}$

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

Signaldämpfung

Leistungsabnahme auf Übertragungsstrecke
 P_1 : Anfangsleistung, P_2 : Leistung am Ende der Strecke

$$\text{Signaldämpfung [dB]} = 10 \cdot \log\left(\frac{P_1}{P_2}\right) = 10 \cdot \log\left(\left(\frac{U_1}{U_2}\right)^2\right) = 20 \cdot \log\left(\frac{U_1}{U_2}\right)$$



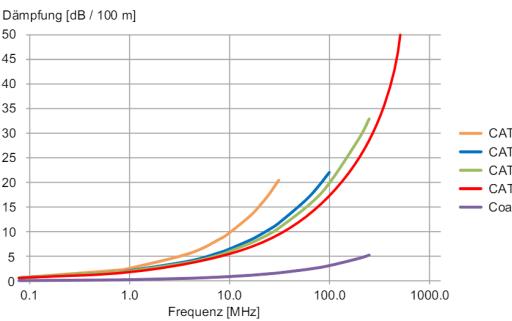
Halbierung der Leistung entspricht ca. 3dB

SNR

Signal to Noise Ratio

$$SNR_{\text{dB}} = 10 \cdot \log\left(\frac{P_{\text{Signal}}}{P_{\text{Störung}}}\right) = 20 \cdot \log\left(\frac{U_{\text{Signal}}}{U_{\text{Störung}}}\right)$$

Dämpfungsbelag Dämpfung pro Distanz - dB pro 100m Dämpfungsbelag = $\frac{\text{Dämpfung}_{\text{dB}}}{\text{Länge}_m} \cdot 100$



Leistungslänge Bandbreite und Dämpfungsbelag

- maximale Leistungslänge L_{max} :

$$L_{\text{max}} = \frac{SNR_{\text{min}}}{\text{Dämpfungsbelag}}$$

SNR_{min} : Minimales benötigtes SNR für korrekte Datenübertragung

- tiefere Bitrate → grössere Distanzen können erreicht werden
- Die Bandbreite (Frequenz) ist abhängig zum Dämpfungsbelag.
- höhere Kabelkategorien haben bessere Schirmung → tolerieren höhere Dämpfung

Kabeltypen

Overview

- Koaxialkabel: Geeignet für hochfrequente Signale
- Twinaxialkabel: Hoher Schutz (double koax)
- Twisted Pair (TP): Häufig im Einsatz (Shielded/Unshielded)
- Glasfaser: Hohe Bandbreite, Geringe Dämpfung, resistent
 - Multimode, Singlemode (besser)

Paarsymmetrische Kabel (Twisted Pair)

- Schirmeigenschaften
 - Drahtgeflecht: niederfrequente Einstreuungen
 - metallisch beschichtete Folien: hochfrequente Störungen
- Bezeichnungsschema ISO/IEC 11801
xx/yTP worin TP für Twisted Pair steht:

xx steht für die Gesamtschirmung:

U = ungeschirmt
 F = Folienschirm
 S = Geflechtschirm
 SF = Schirm aus Geflecht und Folie

y steht für die Aderpaarschirmung:

U = ungeschirmt
 F = Folienschirm
 S = Geflechtschirm

Behebung von Störungen (crosstalk):

- Kapazitiv: Komplementäres Signal, elektrisch leitenden Schirm
- Induktiv: Verdrillte Aderpaare

OSI Referenzmodell

Klassifizierung von Diensten

Verbindungsorientiert

- Verbindungsaufbau nötig
- Informationen vom Empfänger - Optionen aushandeln
- Reihenfolge der Daten bleibt erhalten

Verbindungslos

- Jederzeit (send and forget)
- Ziel muss nicht bereit sein
- einfacher umzusetzen

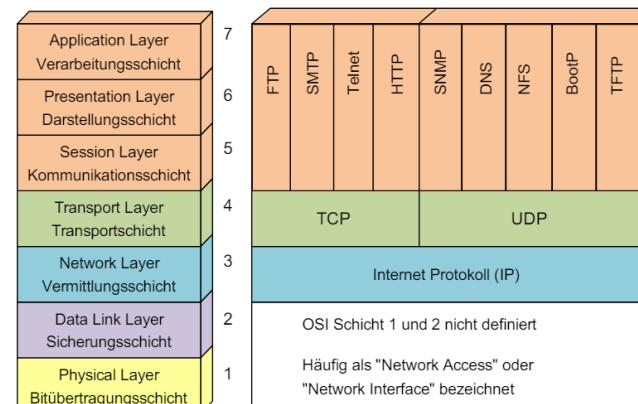
Zuverlässig

- Kein Datenverlust
- Sicherung durch Fehler-Erkennung/-Korrektur
- Text-Nachrichten

Unzuverlässig

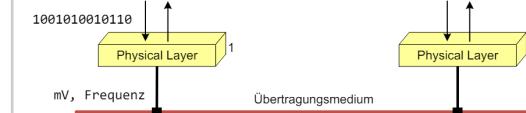
- Möglicher Datenverlust
- Keine Sicherung
- Streaming

OSI Layers



Physical Layer

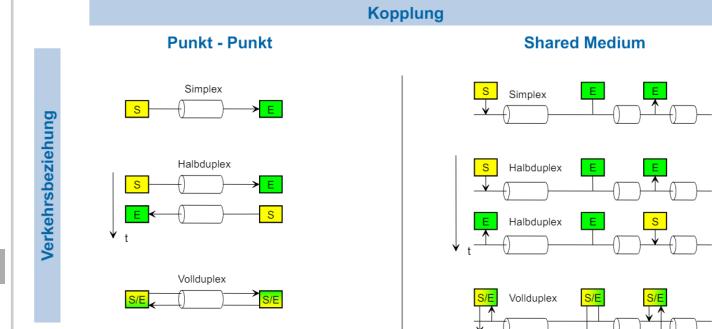
Schicht 1: Bitübertragungsschicht



Funktionalität ungesicherte Übertragung eines Bit-Stroms

- Elektrische Eigenschaften (Signalform, Amplituden, etc.)
- Codierung (Abbildung der Daten auf elektrische Signale)
- Mechanische Eigenschaften (Stecker, Pinbelegung etc.)

Verkehrsbeziehung und Kopplung



Arten der Kommunikation (Verkehrsbeziehung) und Kopplung

- Simplex: Ein Kanal, eine Richtung
- Halbduplex: Ein Kanal, abwechselungsweise in 2 Richtungen
- Vollduplex: Ein Kanal pro Richtung
- Punkt-Punkt: Direkte Verbindung 2 Kommunikationspartnern
- Shared Medium: Mehrere Partner verwenden das gleiche Medium

Einheiten und Kenngrößen

Wichtige Kenngrößen

- Bandbreite B – Einheit Hertz (Hz)
 - Maximal übertragbare Frequenz, durch Medium limitiert
- Symbolrate f_s – Einheit Baud (Bd)
 - # Symbole pro Zeit, limitiert durch Bandbreite ($\leq 2B$)
- Bitrate R – Einheit Bit/s (bps)
 - R = Symbolrate \times Anzahl Bits pro Symbol
- Kanalkapazität C – Einheit Bit/s (bps)
 - Berücksichtigt realen Kanal SNR $\frac{S}{N}$

Bitrate nominell gleich wie Symbolrate wenn: Informationsgehalt pro Symbol = 1 Bit → z.B. bei binärer Codierung (2 Zustände)

- In der Kommunikation stehen k, M, G etc. SI-konform für die exakten Zehnerpotenzen:
 - kBit = 10^3 Bit, MBit = 10^6 Bit, GBit = 10^9 Bit
- Bitrate/Datenübertragungsrate/Durchsatz = Synonyme
- Bandbreite = maximale Symbolrate

$\lg = \log_2$, $\lg = \log_{10}$, $\ln =$ natürlicher Logarithmus

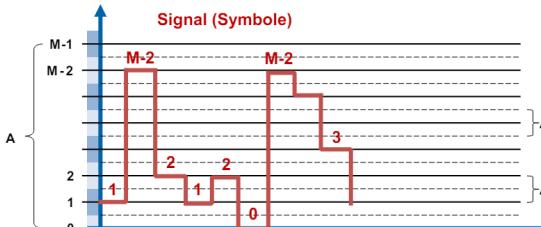
Datenrate, Bandbreite, Bandrate

Datenübertragungsrate $f_s \leq 2B$

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

Unterscheidbare Signalzustände: $M = 1 + \frac{A}{\Delta V}$

A = Max. Grösse Signals, V = Ungenauigkeit Empfänger



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

S: Signalleistung, N: Rauschleistung

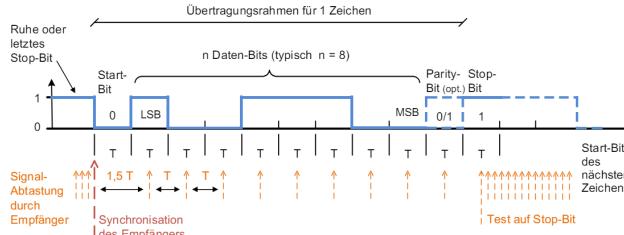
Übertragungsverfahren: Parallel und Seriell

Serielle asynchron Übertragung

Benötigte Abmachungen zwischen Sender und Empfänger:

- Bitrate, # Datenbits (typ. 1 Byte), # Stoppbits (typ. 1 Bit)
- Parität (gerade, ungerade, keine)

Taktrückgewinnung möglich



- Empfangen wird 1001 1100 – LSB first $\rightarrow 0011 1001$ (binär); 0x39 (hex); ASCII Code 57 = «9»

Genauigkeitsanforderung an Takte von Sender und Empfänger:

- Letzte Abtastung muss noch im Zeitfenster liegen (Stop-Bit bei einem Stop-Bit); also $\frac{1}{2}T$ auf $\frac{9}{2}T$

Clock Drift

Maximale Framegrösse Ethernet: 1'500 Bytes.

- Standard: Oszillatoren brauchen Genauigkeit von ± 50 ppm
- 50 ppm (parts per million) \rightarrow Fehler von 0.00005
- Worst-Case: Sender Fehler = -50 ppm, Empfänger Fehler = +50 ppm (oder umgekehrt)

Sicheres Abtasten von Daten? (im Worst-Case)

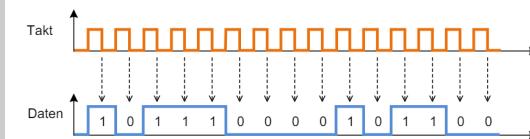
- 1'500 Bytes = 12'000 Bit; $T_{Bit} = 1$ Bit-Zeit
- 100ppm Differenz Sender/Empfänger $\rightarrow 100 \cdot 10^{-6} = 1 \cdot 10^{-4}$
- Fehler pro Bit: $10^{-4}T_{Bit}$
- 1'500 Bytes sind 12'000 = $1.2 \cdot 10^4$ Bit
- Die Abweichung ist somit $1.2 \cdot 10^4 \text{ Bit} \cdot 10^{-4}T_{Bit}/\text{Bit} = 1.2T_{Bit}$
- fehlerfreie Abtastung nicht möglich (ohne weitere Massnahmen)

Serielle synchron Übertragung

Empfänger und Sender arbeiten mit gleichem Takt (synchronisiert)

- Keine Start- und Stoppbits benötigt
- Takt muss zusätzlich übertragen werden

Taktübertragung: Codierungsverfahren oder zusätzliche Leitung.
Aufgabe vom Data Link Layer: Grenzen der einzelnen Bytes zu ermitteln (Preamble, etc.)

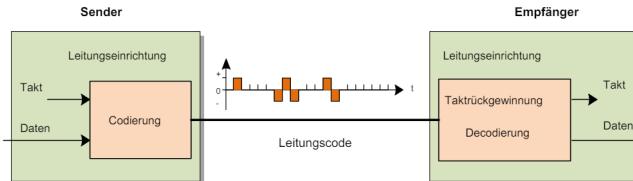


Welches Bit trifft zuerst beim Empfänger ein (1/0)? $\rightarrow 1$
Vorsicht, wenn Weg und Zeit im selben Bild gezeichnet sind.

Leitungscodes und Taktrückgewinnung

Synchrone Übertragung ohne separate Taktleitung

Geeignete Codierverfahren erlauben den Takt zusammen mit dem Datensignal zu übertragen (Leitungscode)



Unter Codierung versteht man hier die Umsetzung der Einsen und Nullen auf eine physikalische Grösse

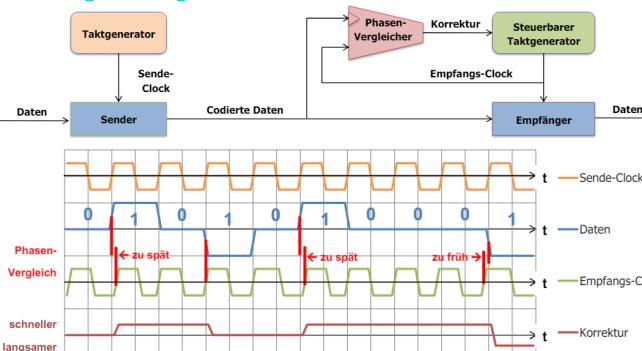
- Vorteil: Es wird nur eine Leitung benötigt
- Nachteil: Zusätzlich 2 x Leitungseinrichtung

Anforderungen an Leitungscodes

- Effiziente Nutzung der physikalisch vorhandenen Bandbreite
- Taktrückgewinnung erlauben (keine separate Taktleitung nötig)
- Gleichspannungsfreiheit (keine langen Folgen von 0 oder 1) \rightarrow Galvanische Isolation von Sender und Empfänger

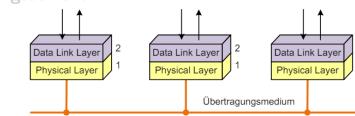
Bekannte Leitungscodes: NRZ, NRZI, Manchester, MLT-3, AMI

Taktrückgewinnung



Data Link Layer

Schicht 2: Sicherungsschicht



Aufgaben

- Realisieren einer zuverlässigen Verbindung zwischen Systemen
- Framing und Flow Control
- bei >2 Teilnehmern: Adressierung, Media Access, Timing

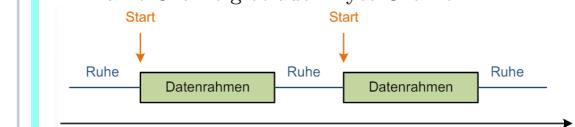
Framing (Rahmenbildung/-erkennung)

- Senderichtung: Einpacken der zu sendenden Nutzdaten in Datenrahmen (Frames)
- Empfangsrichtung: Erkennung und Auspacken der Datenblöcke aus empfangenen Frames

Asynchron

Keine Daten \rightarrow Nichts wird gesendet (Pause zwischen Frames)

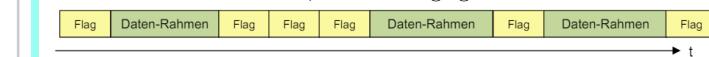
- Zu Beginn eines Frames wird ein Start-Bit gesendet
- Prüfbits am Ende eines Frames!
- Frame-Grenze gibt auch Byte-Grenze



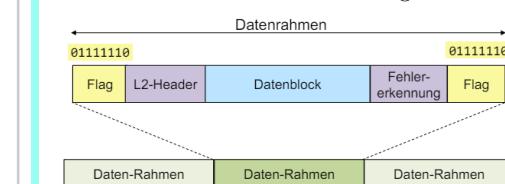
Synchron

Frames werden ohne Unterbruch gesendet
 \rightarrow kontinuierlicher Bitstrom auf Physical Layer

- Stehen keine Daten an, werden Flags gesendet



Frames werden durch ein Start-Flag und ein End-Flag begrenzt:

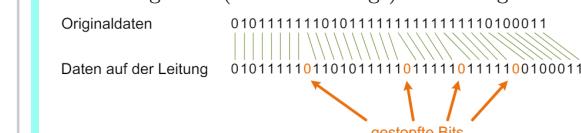


Maskierung von Sonderzeichen (Flags) nötig!

Bitstopfen

Wird verwendet um ein Bit-Muster zu garantieren.

- Sender fügt im Datenstrom nach 5 Einsen immer eine Null ein
- Empfänger wirft nach 5 Einsen immer ein Bit weg
- Somit gibt es (ausser bei Flags) die Bitfolge 01111101



Framelänge und Fehlerwahrscheinlichkeit

Fehlerwahrscheinlichkeit BER (Bit Error Ratio):

- $BER = 0.5 \rightarrow$ jedes 2. Bit falsch

Weitere Definitionen:

- FER (Frame Error Ratio): Fehlerhaft empfangene Frames
- RER (Residual Error Ratio): Unentdeckte fehlerhafte Frames

Frame-Fehlerwahrscheinlichkeit

Wahrscheinlichkeit dass Frame der Länge N min. 1 Bitfehler enthält:
 $BER = p_e << 1 \rightarrow (1 - p_e)^N \approx (1 - N \cdot p_e)$

$$\Rightarrow P_{Fehler, Frame} \approx N \cdot p_e (= FER)$$

Wahl der Framelänge Kompromiss zwischen Overhead und geringer Fehlerwahrscheinlichkeit

- Lange Frames:
 - Höhere Nutzdatenrate (\uparrow Netto-Bitrate, \downarrow Overhead)
 - \uparrow Fehlerwahrscheinlichkeit und Datenverlust pro Fehler
 - \uparrow Wahrscheinlichkeit eines unentdeckten Fehlers
- Kurze Frames: Tiefere Nutzdatenrate, Zuverlässigkeit

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

Datenraten

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

F_R = Framerate, B = Bitrate, F_L = Framelength,
 IFG = Interframe Gap, N = Nutzbitrate, P = Payload

Fehlererkennung und -korrektur

Fehlererkennung

Prinzip: Daten Redundanz hinzufügen \rightarrow erhöht Hammingdistanz

Zuverlässigkeit: abhängig von Framelänge und Verfahren

Standards IEEE 802 (LAN-Standards, z.B. Ethernet):

- max. $5 \cdot 10^{-14}$ unentdeckte Fehler pro Frame-Byte
- $BER p_e \leq 10^{-8}$
- CRC32 für Ethernet, mit Generatorpolynom

Fehlerkorrektur - Error Correction (EC)

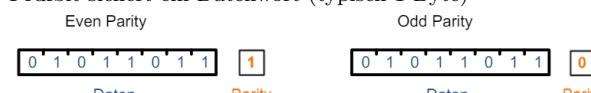
- Backward (BEC): erneutes Übertragen der Daten
- Forward (FEC): Rekonstruktion von verfälschten Bits beim Empfänger

Hamming-Distanz (h)

- Fehlererkennung: $(h - 1)$ Fehler erkennbar
- Fehlerkorrektur: max. $\frac{h-1}{2}$ Fehler korrigierbar

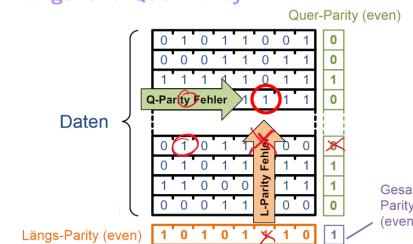
Einfache Parity

Prüfbit sichert ein Datenwort (typisch 1 Byte)



Even Parity: # 1er inkl. Parity-Bit ist gerade (Odd analog)

Längs- und Quer-Parity



Korrigieren:
 1 Bit-Fehler
 Erkennen:
 2 Bit-Fehler

Zugriffsmechanismen (Media Access)

Gesteuerter Medium Zugriff

Master-Slave Verfahren

Verwenden mehrere Systeme das gleiche physikalische Medium, so muss der Zugriff auf das Medium koordiniert werden

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

Token Verfahren

Die Sendeberechtigung wird in einer festgelegten Reihenfolge weitergereicht: Knoten senden nur, wenn sie ein Token halten

- Vorteil: Deterministisch (man weißt, wann man dran kommt)
- Nachteil: Aufwändig (Startup, Token Verlust, etc.)

Zeitsteuerung

Zeitgesteueter Zugriff (wie Taktfahrplan im Bahnenetz)

- Vorteil: Optimierung möglich (nach Auslastung, Durchsatz, etc.)
- Nachteile:
 - Planung und genaue Zeit in allen Knotenpunkten erforderlich
 - Konflikte mit unplausibarem Verkehr (SBB Cargo)
- Anwendungen: PROFINET IRT, Time Sensitive Networks

Random Medium Zugriff

Carrier Sense Multiple Access

- Vor dem Senden geteiltes Übertragungsmedium abhören ob frei (Carrier Sense), sonst bis zu Pause warten
- Vorteil: Alle Stationen gleichberechtigt (kein Master) \rightarrow jederzeit Zugriff auf Übertragungsmedium
- Nachteil: Kollisionen möglich (Collision Detection)

Kollisionsbehandlung - CSMA

- CD (Collision Detection): Kollision \rightarrow abbrechen, später nochmals (ALT)
- CR (Collision Resolution): Hardware-unterstützte Arbitrierung (aktiv/passiv)
 - Kollisionen werden erkannt und kontrolliert aufgelöst
- CA (Collision Avoidance): Kollisionen vermeiden
 - Request to Send / Clear to Send

Flow-Control

Explizite Start-Stopp Signalisierung:

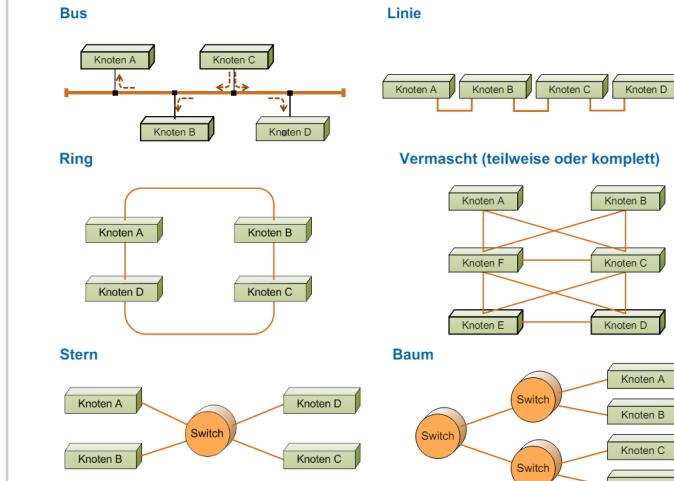
- Obere und untere Limite, stopp wenn oben, start wenn unten

Implizites Stop-and-Wait:

- Sender wartet auf Bestätigung (ACK) bevor er weiter sendet

Ethernet und LAN

Local Area Networks (LAN)



Übertragung und Adressierung

Übertragungsarten Immer genau 1 Sender, $E = \#$ Empfänger

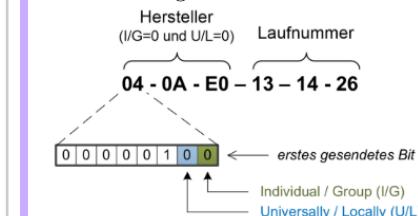


- Unicast: 1 E
- Multicast: n E (Gruppe)
- Broadcast: alle Knoten im LAN

IEEE MAC Adressen

- 3-Byte «OUI» identifiziert Hersteller
- 3-Byte Laufnummer durch Hersteller verwaltet

Klassifizierung der MAC Adresse:



Individual/Group Bit:

- 0 = individual address
- 1 = group address

Universally/Locally Bit:

- 0 = universally administered address
- 1 = locally administered address

Ethernet Frame Format und MAC-Adresse

Sende Ethernet-Frame über 100BASE-TX Schnittstelle, Bit-Sequenz auf Kabel:

10101010 10101010 10101010 10101010 10101010
 10101010 10101011 00010000 00000000 01011010 11100011
 10011111 00000110 ...

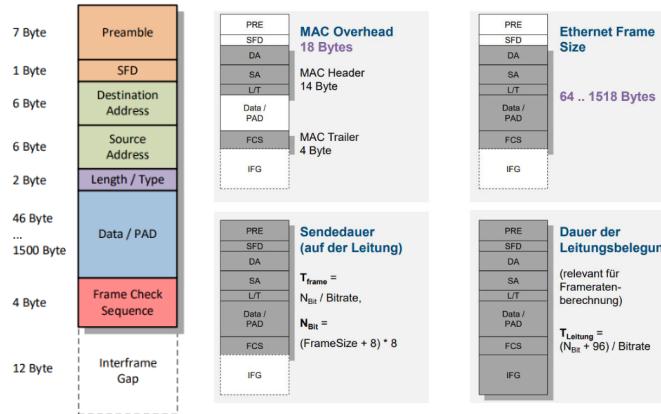
MAC-Adresse und Hersteller des Empfängers:

- 7 Bytes Präambel (10101010), 1 Byte SFD (10101011)
- 6 Bytes Destination Address: 00001000 (=08) 00000000 (=00)
 $01011010 (=5A) 11000111 (=C7) 1111001 (=F9) 01100001 (=61)$
- ⇒ MAC-Adresse: 08-00-5A-C7-F9-61, Hersteller (08-00-5A) IBM

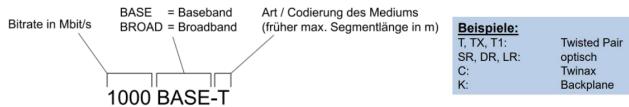
Pro Byte zuerst LSB, dann MSB (Ausnahme Zahlenwerte, z.B. Length/Type-Feld)

Ethernet

Ethernet Frame Format



Bezeichnungsschema und Datenraten



Autonegation Ermittlung der besten Betriebsart durch Austausch der Leistungsmerkmale zweier Netzwerkkomponenten

Link Pulses:

- NLP = Link Presence Detection
- FLP = Autonegotiation, Autopolarity

	10BASE-T	100BASE-TX	1000BASE-T	10GBASE-T
Kabelkategorie	CAT3 - 16 MHz CAT5 - 100 MHz	CAT5 - 100 MHz CAT6 - 250 MHz	CAT5 - 100 MHz CAT6 - 250 MHz	CAT6A - 500 MHz CAT7 - 600 MHz CAT7a - 1000 MHz
Line Coding	Manchester 2 Adreipare simplex	MLT-3, 4B5B 4 Adreipare simplex	PAM-5, 8B/10B 4 Adreipare duplex	PAM-16, 64B/65B, FEC 4 Adreipare duplex
Baudrate	10 MBaud	125 MBaud	4 x 125 MBaud	4 x 800 MBaud
Link Pulses	NLP	FLP	FLP	FLP

Ethernet Geräte (Network Gear)

Switch/Brigde Signale weiterleiten und verstärken, zusätzlich:

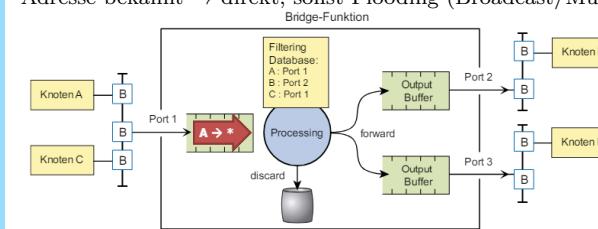
- Prüft Checksumme und kann Layer-2 Adressen auswerten
- Transparent: sollen für Endgeräte unsichtbar sein
- Verwendet Filtering Database (Adress-Learning)

Merkmale von Switches und Bridges

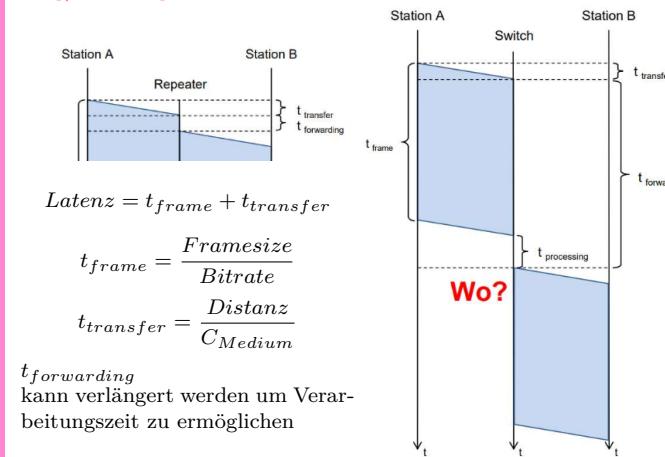
Anzahl Ports	Steckergrösse ist im Extremfall die Limitierung
Adressstabelle	Wie viele Stationen können im LAN existieren
Filtern	Maximale Frames/s/Port (Empfangsrichtung)
Transferrate	Maximale Frames/s/Port (Senderichtung)
Backplane / Fabric Kapazität	Maximaler Gesamt durchsatz zwischen allen Ports
Architektur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

Filtering Database Mapped MAC-Adressen auf Ports, lernt nur Absenderadresse

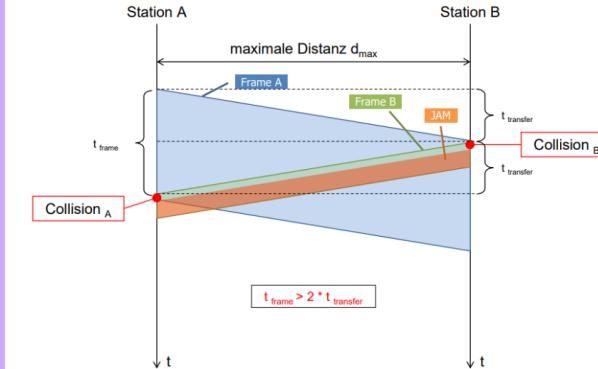
Adresse bekannt → direkt, sonst Flooding (Broadcast/Multicast)



Weg/Zeit-Diagramm für das Senden eines Frames



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



Bedingungen für Kollisionserkennung:

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

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

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

Redundanz (Spanning Tree)

Spanning Tree Algorithmus

Ziel: Redundante Pfade → Probleme! ⇒ Alle Segmente loop-frei verbinden

Initialisierung

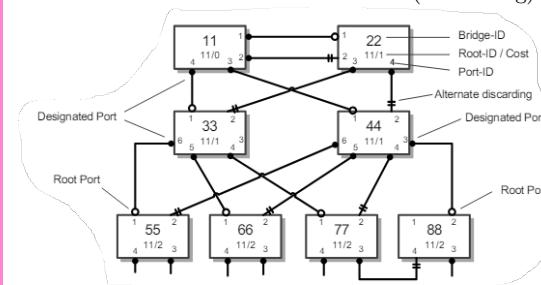
- Alle Ports für Nutzdaten blockiert
- Annahme: «Ich bin Root»
- Austausch BPDUs mit Nachbarn (Root ID, Root Cost, Bridge ID, Port ID)

Aufbau des Spanning Tree

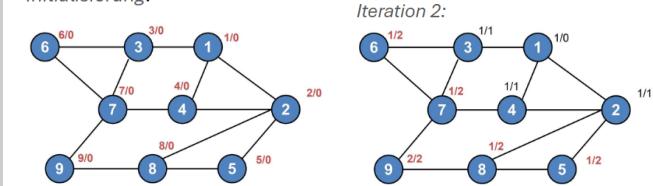
- «kleinster» Nachbar als Root gesetzt → Anzahl Hops + 1 (achte Prioritätswert)
- wiederholen bis alle dieselbe Root ID haben

Setzen der Port Roles

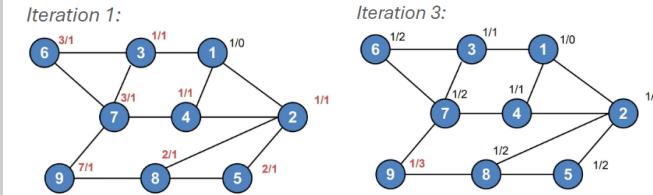
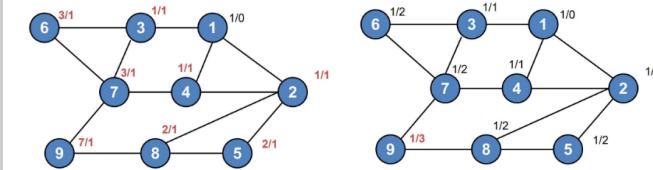
- Root-Ports (Empfang der «besten» BPDU)
- Designated-Ports (Gegenstück zu Root-Ports)
- Weg zum «kleinsten» Nachbar wird bevorzugt (ID, Anzahl Hops)
- Alle anderen Ports bleiben blockiert (Discarding)



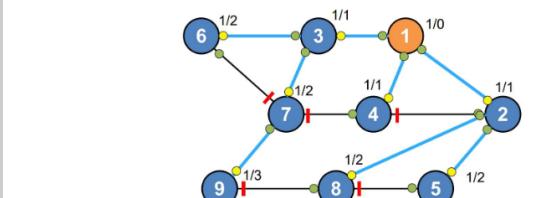
Initialisierung:



Iteration 1:

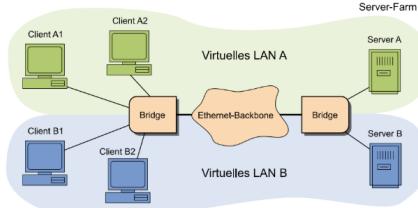


Final



Virtuelle LANs

VLAN Aufteilen eines LANs in mehrere unabhängige logische Netze (Broadcast Domains)



Trunk Links:
Teil von mehreren VLANs →
Frames eindeutig kennzeichnen!

Trunk = Tagged
Access = Untagged

VLAN Tagging

Bytes	tagged frame
7	Preamble
1	SFD
6	Destination Address
6	Source Address
2	Type = 0x8100
2	PCP DEI
2	VLAN ID
2	Length / Type
42	Data / PAD
1500	
4	Frame Check Sequence
12	Interframe Gap

Erweiterung des Ethernet Headers durch einen VLAN-Tag (+ 4 Bytes)

VLAN-ID (VID) im VLAN-Tag

- Zuordnung

Priority Code Point (PCP)

- ermöglicht Priorisierung

Discard Eligibility Indicator (DEI)

- 0 → Frame wird bei Überlastung zuerst verworfen

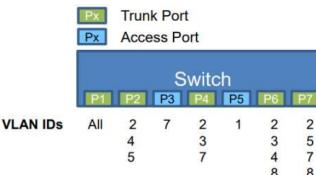
Vorteile:

- Transparent für Endgeräte
- VLAN Konfiguration nur im Netz (dh einfach)

Gesendete Frames:

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

Switch Konfiguration:

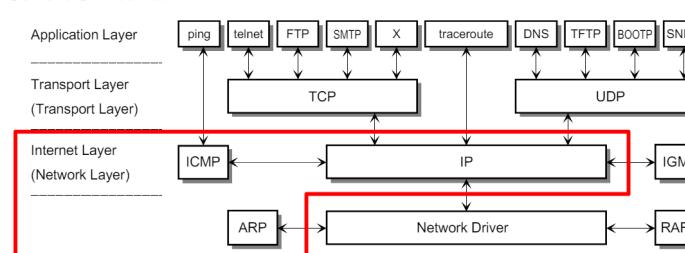


Welche Frames werden an welchen Ports gesendet und sind diese getagged oder ungetagged?

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

Network Layer

Schicht 3: Internet



Die Netzwerkschicht

Verbindet Hosts in verschiedenen Netzen und ermöglicht den Transport der IP-Pakete → NUR Transport
höhere Layer übernehmen: Fehlerkorrektur, Flusskontrolle, etc.

Grundsätze des Internets

- Jedes Netzwerk soll für sich selbst funktionsfähig sein
- Die Kommunikation basiert auf «best effort»
- Die Verbindung der Netze erfolgt durch Black Boxes
- Keine zentrale Funktionssteuerung wird benötigt

Kommunikationsobjekte

- OSI Layern zugeordnet
- (Application-)Message/Stream Layer 5-7
 - (Transport-)Paket, Datagram Layer 4
 - (IP-)Paket (früher Datagram) Layer 3
 - (HW-specific) Frame Layer 1-2

Netzwerk Applikationen und Protokolle

Routing

- Router** verbinden Subnetze (Ethernet, xDSL, WLAN, etc.)
- empfangen nur Pakete, die direkt an sie adressiert sind
 - Weiterleitung erfolgt anhand der Network Layer Adresse
 - Benutzen immer den optimalen Pfad.

Routing and Forwarding

- Routing: Aufbau und Update der Routingtabellen in den Knoten
 - Router müssen optimalen Pfad zu jedem Host kennen
 - kleine oder Teilnetze: Statische Konfiguration
 - grössere Netze: Dynamisch durch Routing-Protokolle: Topologie des Netzes ermitteln → ideale Pfade bestimmen
- Forwarding: Weiterleiten der Daten
 - Aufgrund von Routingtabellen Datenpakete weiterleiten

Routing-Tabelle

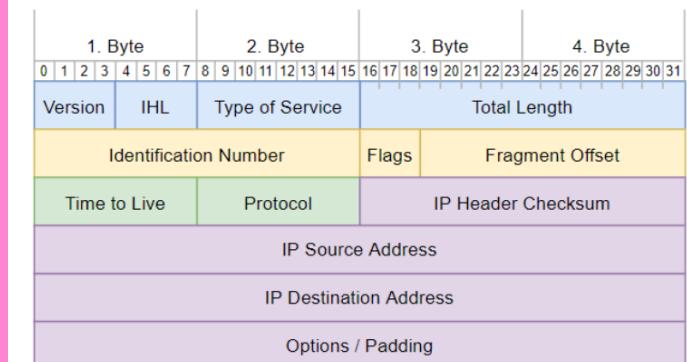
- Info wie jedes Netz/Interface erreicht werden kann
- Für Weiterleitungsentscheidung notwendige Informationen:
 - Eintrag für jedes erreichbare Netz (Netzadresse, Netzmaske)
 - Interfaces, über die die Netze erreicht werden können
 - IP-Adresse des nächsten Routers, wenn Zielnetz nicht direkt erreicht werden kann
 - Eigenschaften:
 - sortiert nach Länge der Netzmaske, von oben nach unten durchsucht
 - erster Eintrag der passt wird verwendet, default Eintrag am Schluss passt immer

IPv4

IP-Header Format

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

- Version IPv4 / IPv6
- IHL Header Length in 4-Byte (20 Byte → IHL = 5)
- Type of Service neu Differentiated Services (DS), Erlaubt Priorisierung, Einteilung der Daten in Verkehrsklassen
 - DSCP: spez. Verhalten bzgl. Weiterleitung
 - ECN: kann drohende Überlast markieren
- Total Length Länge des IP-Packets (Header + Nutzdaten)
- ID Number Identifikation des IP-Pakets / Fragmente, erlaubt Identifikation zusammengehöriger Fragmente
- Flags Kontroll-Flags für Fragmentierung (0/DF/MF)
- Fragment Offset Gibt an, wo ein Fragment hingehört
- Time to Live anz. Sek, Hop-Counter, 0 → Paket wird verworfen
- Protocol Übergeordnetes Protokoll
- Header Checksum verhindert fehlgeleitete Pakete (× Nutzdaten)
- Source Address Wer das Paket ursprünglich abgesendet hat
- Destination Address Wer das Paket schliesslich erhalten soll
- Options/Padding variabel, füllt auf ein Vielfaches von 32Bits auf



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

Internet Protokolle (IP)

Hierarchische Adressierung

IP-Adressen sind zweistufig hierarchisch

- IP-Adresse eines Hosts = Netzadresse + Interface-Adresse

Terminologie

- Sender und Empfänger → Hosts
- IP bietet einen unzuverlässigen, verbindungslosen Dienst
 - IP-Adr. identifiziert Host-Interface (nicht den Host) eindeutig innerhalb des Netzwerks
 - Jeder Host hat min. eine Adresse, Multi-Homed Hosts mehrere

Netzadresse

- Reserviert: Darf nicht für Interfaces verwendet werden!
- Tiefste Adresse im Subnetz (Interface-Adressbits alle 0)
- Berechnet durch: Interface-Adresse AND Subnetzmaske

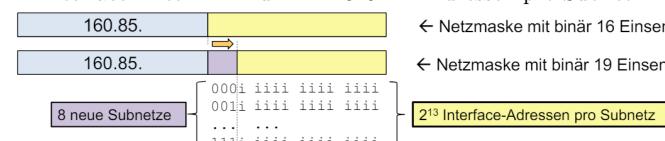
Subnetting Aufteilung in kleinere Netze

ZHAW besitzt B Netz 160.85.0.0 → total $2^{16} \cong 65000$ Hosts

- in 8 kleinere Subnetze aufteilen → Subnetting

Verschieben der Netzmasken-Bits:

- $8 = 2^3$, 3 Bits identifizieren 8 Subnetze (000 → 111) in binärer Netzmaske
- Netzmaske: /16 zu /19 ($255.255.0.0 \rightarrow 255.255.224.0$)
- Interface-Anteil: 2^{16} zu $2^{13} = 8192$ IP Adressen pro Subnetz



Damit haben wir 8 neue Subnetze mit den folgenden Netzadressen:

- 160.85.0000 0000 0000 0000 = 160.85.0.0
- 160.85.0010 0000 0000 0000 = 160.85.32.0
- 160.85.0100 0000 0000 0000 = 160.85.64.0
- 160.85.0110 0000 0000 0000 = 160.85.96.0
- ...
- 160.85.1110 0000 0000 0000 = 160.85.224.0

- Netz-Anteil: 19 statt 16 "1" → Subnetzmaske: 255.255.224.0 oder /19
- Host-Anteil: 13 statt 16 "0" → Anzahl Hostadressen = 8'192

Das zweite Netz oben wird deshalb korrekt wie folgt gekennzeichnet:

- 160.85.32.0 / 255.255.224.0 oder 160.85.32.0 /19

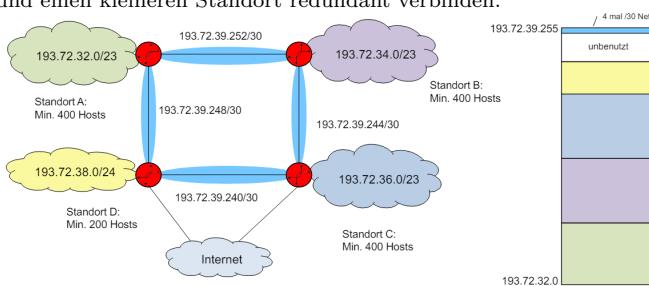
Das fünfte Netz wird wie folgt gekennzeichnet:

- 160.85.128.0 / 255.255.224.0 oder 160.85.128.0 /19

Wichtige Regel: Eine Netzwerkadresse ist immer ein Vielfaches der Netzgrößen!

Flexible Aufteilung eines Netzbereiches

4 Standorte, von ISP Netz 193.72.32.0 /21 erhalten. Ziel: 3 grössere und einen kleineren Standort redundant verbinden.



Kapselung und Adressauflösung

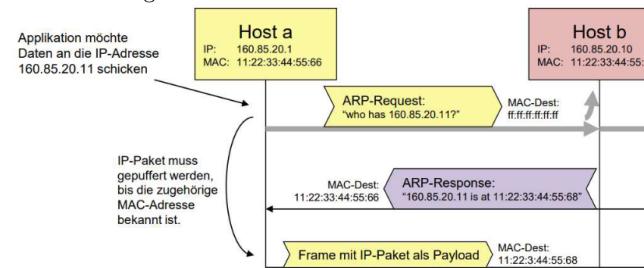
Kapselung eines IP-Pakets im Ethernet Frame von Type 0x0806

Präambel	SFD	Destination MAC-Adresse	Source MAC-Adresse	Protocol Type (= 0806 ₁₆ für ARP)	Daten (eingegebettete ARP Daten)	FCS
7	1	6	6	2	max. 1500 Daten (eingegebettete ARP Daten)	4

Address Resolution Protocol (ARP)

ARP Ermittlung der Hardwareadresse (MAC) zu einer IP-Adresse

- ARP-Request wird an Broadcast-Adresse gesendet
- ARP-Response wird von Knoten mit angefragter IP-Adresse an Absender gesendet



Erkennung von Adresskonflikten: ARP Request an eigene IP-Adresse

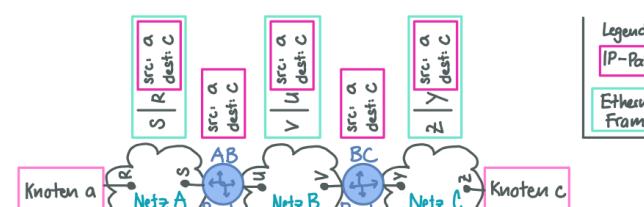
ARP Nachrichtenstruktur

HW-Adresstyp (Ethernet = 1)	Protokolladresstyp (für IP = 0800 ₁₆)
HW-Adressgröße	Protokoll-Adr.-Gr.
Op-Code (Request = 1, Reply = 2)	
Quell-HW-Adresse – 6 Bytes	
Quell Netzwerk-Adresse (also IP-Adresse) – 4 Bytes	
Ziel-HW-Adresse (oder 0, wenn unbekannt bei Request) – 6 Bytes	
Ziel Netzwerk-Adresse (also IP-Adresse) – 4 Bytes	

Request: Destination Address = Broadcast, HW Address of Target = 0

ARP Cache mit bekannten HW-Adressen

ARP für jedes IP-Paket ineffizient → Jeder Knoten führt ARP-Cache (speichert bekannte IP-MAC Kombinationen für gewisse Zeit)



- a sendet IP-Paket c (Enthält Adressen a und c)
- a konsultiert Routing Tabelle → c kann über Router AB erreicht werden und a kennt nun IP-Adresse von Router AB
- a generiert Ethernet Frame, welches an HW-Adresse S von Router AB gesendet wird
 - a muss aus IP-Adresse von Router AB die HW-Adresse S herausfinden → Adressauflösung
- Router AB empfängt Ethernet Frame, packt IP-Paket aus und modifiziert den Header (TTL)
- Router AB konsultiert Routing Tabelle → c kann über Router BC erreicht werden und AB kennt nun IP-Adresse von BC
- IP-Adressen a und c bleiben unverändert!

Internet Control Message Protocol (ICMP)

Internet Control Message Protocol (ICMP)

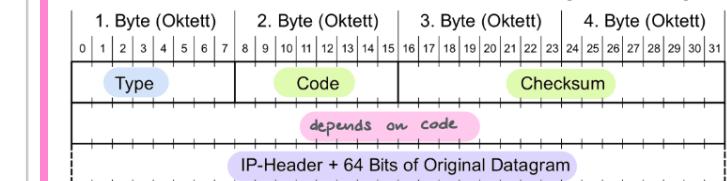
Übertragung von Fehlermeldungen oder Informationsaustausch

- nutzt direkt IP - keine Garantie, dass Meldungen ankommen
- Meldungen sind NUR informativ gedacht

ICMP Format Header:

- Type ICMP Typ
- Code Message Details
- Checksum Prüfsumme über die ICMP Meldung
- depends on code Wert/Verwendung je nach ICMP Typ

Datenbereich IP-Header und 64 Bits of Original Datagram



ICMP Meldungstypen

- | | |
|------------------------------|-------------------------|
| • 0: Echo Reply | • 11: Time Exceeded |
| • 3: Destination Unreachable | • 12: Parameter Problem |
| • 5: Redirect | • 13: Timestamp Request |
| • 8: Echo Request | • 14: Timestamp Reply |

ICMP Destination Unreachable Router/Zielhost → Absender wenn Paket nicht weitergeleitet werden kann

Feld	Wert/Semantik
Type	3
Code	0 = net unreachable, 1 = host unreachable, 2 = protocol unreachable, 3 = port unreachable, 4 = fragmentation needed and DF set, 13 = communication administratively prohibited
Checksum	Prüfsumme über die ICMP Meldung
IP Header + 64 Bits of Original Datagram	Information für den Empfänger zur Zuordnung der Meldung zu einem gesendeten IP Paket of Original Datagram

Path MTU discovery Vermeidung von Fragmentierung «unterwegs» Dazu: Erkennung der kleinsten MTU auf Pfad zwischen Sender und Empfänger (Path-MTU, PMTU)

Vorgehen: (Annahme PMTU = lokale MTU)

- Sende IP-Pakete mit Länge=PMU und mit DF=1
- Empfange «Destination Unreachable» mit Code 4 «fragmentation needed and DF set»
- PMU reduzieren auf «Next-Hop MTU» (enthalten in Octet 5..8)

ICMP Destination Unreachable Farben siehe IP-Header def.

Host 160.85.31.3 sendet an Host 160.85.29.99:

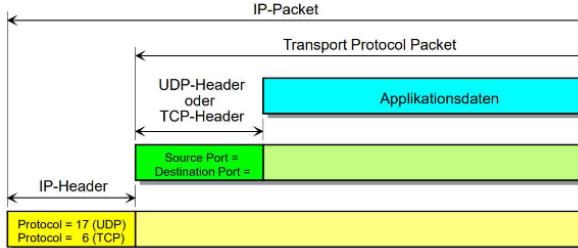
- 4500 0028 8b10 0000 0711 a8a4 a055 1f03 a055 1d63 8b0d 829d 0014 a348 030a 0000 7504 1137 407c 0800
- Senderadr.: a055 1f03, Destinationadr.: a055 1d63
- Router kennt keinen Weg: sendet Destination Unreachable Message zurück:
 - 4500 0038 8038 0000 fd01 5bc0 a055 821e a055 1f03 0301 4bf7 0000 0000 4500 0028 8b10 0000 0711 a8a4 a055 1f03 a055 1d63 8b0d 829d 0014 a348
 - Erkennen dass dies ICMP Message ist: **Protocol: 01**
 - ICMP Typ: **Type: 03**
 - 64 Bytes of Original Datagram: **Original Data**

Transport Layer

Schicht 4: Transportschicht

Transportlayer Schnittstelle zwischen Betriebssystem (Kernel Space) und Anwendungen (User Space)
Zugriff auf Funktionen des Transport Layers erfolgt via klar definierten Schnittstelle (Sockets)

Kapselung "Protocol" Feld unterscheidet UDP und TCP Daten



Adressierung

Client adressiert Server-Applikation mit Destination Port Nr.

- sonst weiss TCP/UDP-Modul im Empfänger nicht, welche Applikation gemeint ist
- für Source Port Nummer verwendet Client (meist) zufällige Port Nummer >1'023 (vom Betriebssystem vergeben)

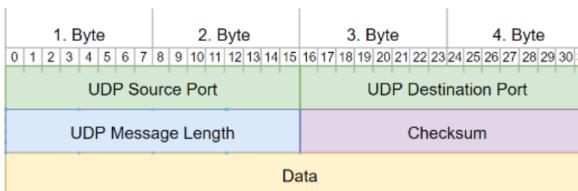
UDP - User Datagram Protocol

UDP Multi-/Demultiplexen der Datagramme zu Applikationen

- Verbindungslos und unzuverlässig

UDP-Header

- Source Port** Sendende Applikation
- Destination Port** Applikation des Empfängers
- Message Length** Länge des Datagramms
- Checksum** Prüfsumme über einen Pseudo-Header, UDP-Header und Daten (kann Null sein)
 - Pseudo-Header: IP Source- und Destination Address, Protocol Feld, Länge des Datagramms
 - * so können fehlgeleitete Datagramme erkannt werden



Port-Nummern

- System Ports (Well-Known)** Fix, für bekannte Appl. reserviert
- User Ports (Registered)** Reserviert für herstellerspez. Appl.
- Dynamic / Private Ports** Frei verfügbare Ports

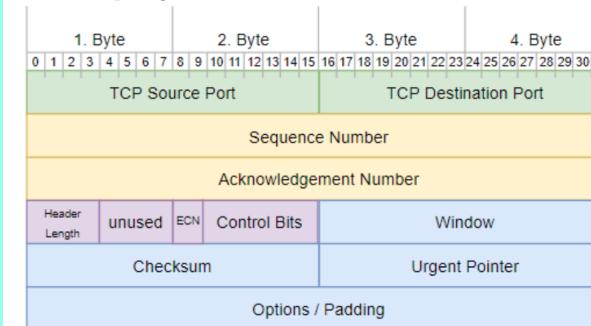
System Ports	User Ports	Dynamic Ports
0 - 1023	1024 - 49'151	49'152 - 65'535

TCP - Transmission Control Protocol

TCP Eigenschaften Verbindungsorientierte Übertragung, zuverlässiger Verbindungsaufbau, hohe Zuverlässigkeit, Vollduplexübertragung, Stream-Schnittstelle, Graceful Termination, Punkt-zu-Punkt Kommunikation

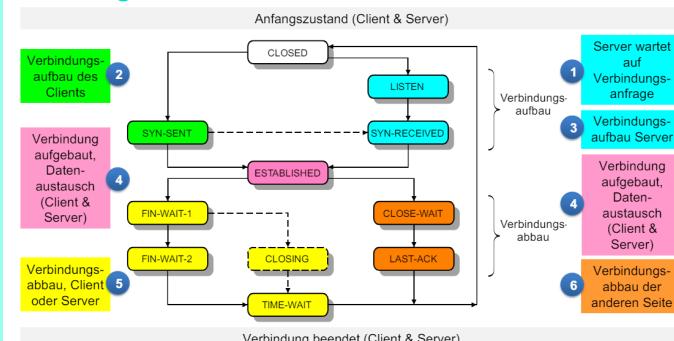
TCP-Header Format

- Sequence-Nr.** Sicherstellung Reihenfolge der Daten, Erkennung verlorener Daten
- Acknowledgement-Nr.** $n + 1 \rightarrow$ Daten bis und mit n korrekt und vollständig angekommen
- Data Offset** Gibt an wo Daten beginnen / enden
- ECN-Flags** Explicit Congestion Notification
 - Bit 8: CWR (Congestion Window Reduced)
 - Bit 9: ECE (ECN-Echo)
- Control Bits** Verbindungsauf- und -abbau (Bits 10-15)
URG: Urgent Pointer
ACK: Acknowledgement Number (Bestätigung empfangener Daten, Erkennung verlorener Daten)
PSH: Push (sofort ohne buffern weiterleiten)
RST: Reset (Verbindung zurücksetzen oder geschlossenen Port signalisieren)
SYN: Verbindungsaufbau, FIN: Verbindungsabbau
- Window** Verfügbare Puffergrösse
- Urgent Pointer** URG = 1 \rightarrow Position der wichtigen Daten
- Options** Häufigste Verwendung: MSS (Maximum Segment Size) die empfangen werden kann

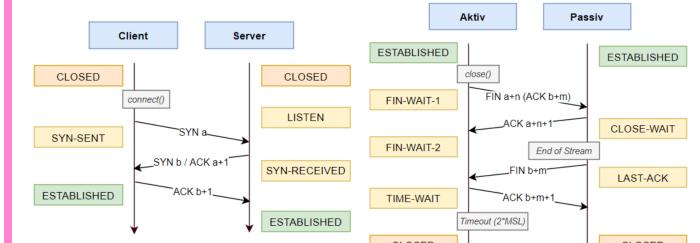


TCP Verkehrssteuerung

Verbindungsorientierte Kommunikation



Verbindungsaufbau und Verbindungsabbau

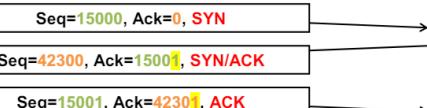


ACK nr. muss mit der Anzahl der Bits der empfangenen Daten aktualisiert werden.

Vollständiges Beispiel

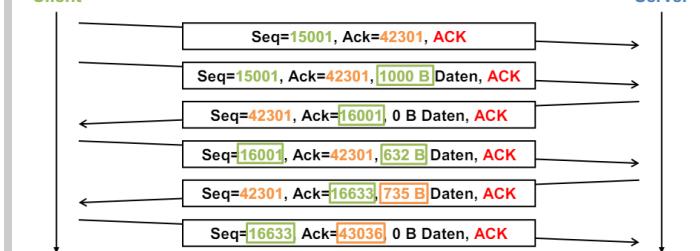
Verbindungsaufbau:

- Server „horcht“ (LISTEN) auf einer bestimmten Port Nummer
- Client sendet Segment mit SYN=1 und zufälliger init. Sequenznummer a (ACK=0, weil ACK nr. ungültig)
- Server bestätigt Sequenznummer mit ACK nr. $a+1$ und ACK=1, wählt zufällige initiale Sequenznummer b , setzt SYN=1
- Client bestätigt b mit ACK nr. $b+1$
 - Erstes Byte vom Client zum Server hat Sequenznummer $a+1$
 - Erstes Byte vom Server zum Client hat Sequenznummer $b+1$



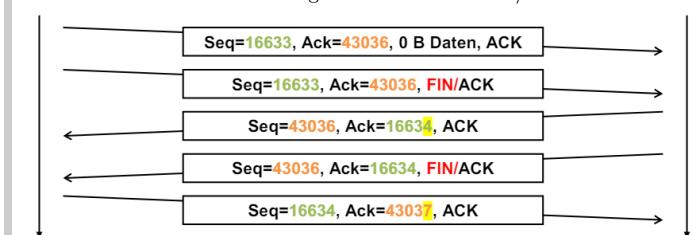
Datenaustausch: TCP-Nachrichten werden bi-direktional ausgetauscht

Client



Beide Seiten können den Verbindungsabbau einleiten

- Ist eine Richtung geschlossen (FIN, ACK), so können in die andere Richtung immer noch Daten gesendet werden (Half-Closed)
 - In Richtung der "geschlossenen" Verbindung wird nicht mehr kommuniziert (Acknowledge number mismatch)
- Falls die zweite Seite die Verbindung auch schliesst, können die 3. und die 4. Nachricht zusammengefasst werden \rightarrow FIN/ACK



TCP Adaptive Elemente

Herausforderungen zur Zuverlässigkeit zwischen Ethernet/TCP:

Problem	Schicht 2	Schicht 4	Massnahmen bei TCP
Nachrichtenverlust	$P_{Verlust} = FER$	$P_{Verlust} >> FER$	Positives ACK
Telegramm-Reihenfolge	fix	kann variieren	Sequenznummern
Round Trip Time	konstant, $\mu s \dots ms$	variabel, $ms \dots s$	Adaptive Retransmission Timeout
Überlast des Empfängers	kommt vor	kommt vor	Sliding Window mit dynamischer Fenstergröße
Überlast des Netzwerks	direkt beobachtbar (Medium)	nur indirekt beobachtbar	Slow Start (Congestion Window)
Neustart von Hosts	direkt beobachtbar	nur indirekt beobachtbar	3 Weg Handshake, Initialisierung Sequenznr.

Timed Delays

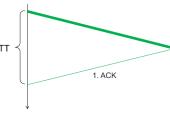
Round Trip Time (RTT) dynamische Anpassung der Wartezeit

- $SRTT_n = (1 - \alpha) \cdot SRTT_{n-1} + \alpha \cdot RTT_n$
 - $RTTVar_n = (1 - \beta) \cdot RTTVar_{n-1} + \beta \cdot SRTT_n - RTT_n$
 - $RTOn = SRTT_n + 4 \cdot RTTVar_n$
- $\alpha = 0.125, \beta = 0.25$ sind Standardwerte

Bandwidth Delay Product (TCP-Puffergrößen)

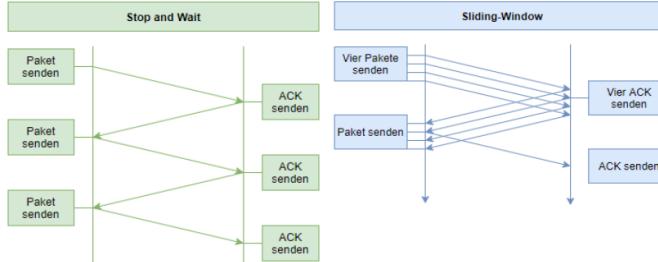
Wahl der Grösse von Sende- und Empfangspuffer, um Verbindung nicht auszubremsen

$$BDP(\text{bits}) = RTT(\text{sec}) \cdot \text{Bandbreite}(bps)$$

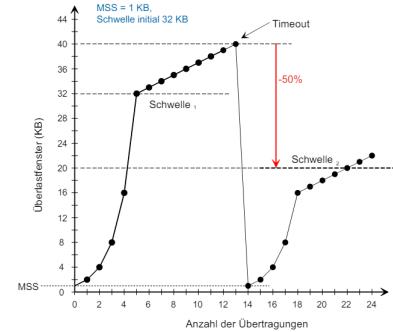


Fluss-Steuerung und Congestion Control

Fluss-Steuerung



Congestion Control - Slow Start

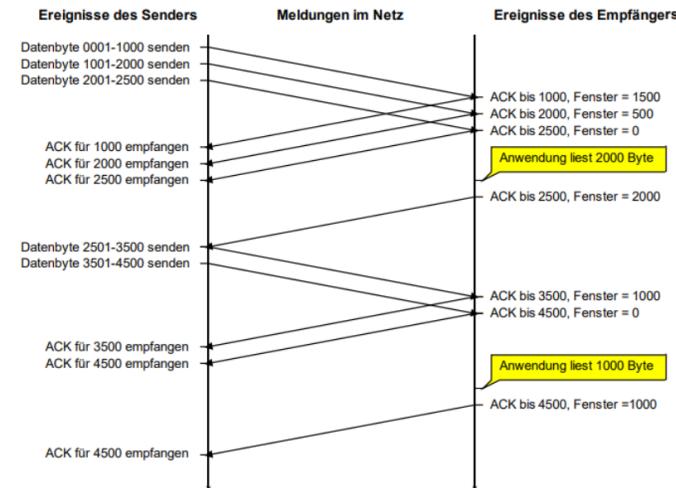


Slow Start: herantasten wie gross die einzelnen Frames sein können.

Wichtig: Sender kombiniert Congestion Window mit Informationen zur Flow Control vom Empfänger → schickt unbestätigte Daten bis $\min\{\text{Congestion Window}, \text{Advertised Win.}\}$ erreicht

Sliding-Window TCP

- Fenstergrösse dynamisch anpassen
- Beide Seiten haben ein Fenster, das die Anzahl der Bytes angibt, die gesendet werden können
 - Verbindungsauftbau: Initiale Fenstergrösse wird der anderen Seite mitgeteilt (Typische Werte: 16 / 32 / 64 KB)
 - Pufferplatz im Empfänger wird alloziert
 - Mit jedem ACK wird der verfügbare Pufferplatz (in Bytes) mitgeteilt und damit die Fenstergröße dynamisch angepasst
 - Fenstergröße von 0 Bytes → keine Daten mehr senden
 - Ist im Empfangsbuffer wieder Pufferplatz vorhanden, wird erneut eine Bestätigung mit diesem Pufferplatz an die andere Seite gesendet (= aktuelle Fenstergröße)
- beide Richtungen arbeiten unabhängig voneinander



- Annahmen: 2'500 Byte Empfangspuffer, 5'000 Bytes Daten
- Fenstergröße des Empfängers: WindowFeld des TCP-Headers
 - Wireshark: Advertized Window Size
 - Sender: nur einen Aufruf von send() für die gesamten 5'000 Bytes

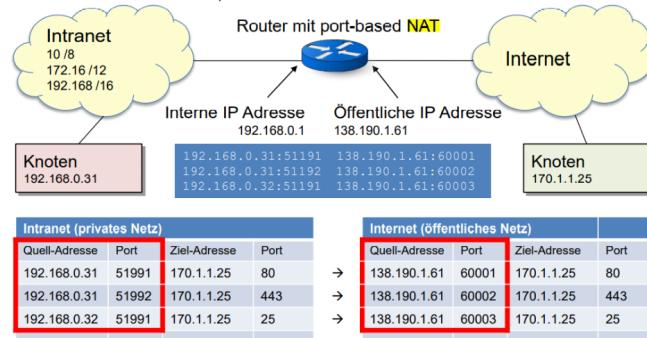
NAT - Network Address Translation

NAT (Port Mapping)

Port-basierte NAT (NAPT) (Boomer Paranoia)

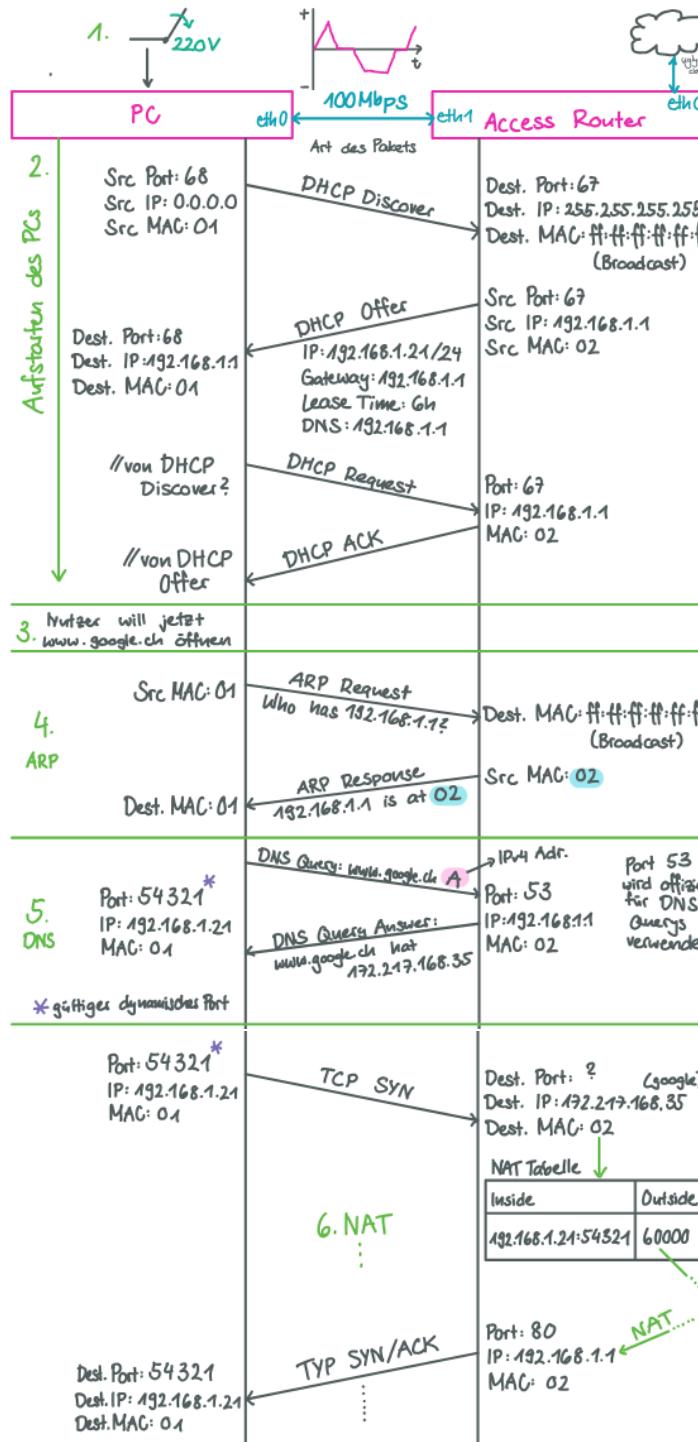
- Ersetzt private IP-Adr. durch public IP des Gateways/Routers
- Ersetzt private Port-Nr. des Hosts durch freie zulässige Port-Nr. des Gateways/Routers
- Mapping privater IP-Adr. und Port-Nr. zur öffentlichen Port-Nr. auch statisch möglich, aber nur Port-Nr wird übernommen

Probleme mit NAT: Verletzung des OSI-Layer-Konzepts
Um Port im TCP Header zu ändern müssen Daten im IP-Frame verändert werden → Netzwerk-Funktion greift auf den Transport Header zu, IP-Adresse/Portnummer werden dabei verändert



∀ Hosts im privaten Netz 192.168.0.0/8: Default-Gateway 192.168.0.1

Von A bis Z: Aufruf einer Webseite



Ports, Adressen und Routingtabellen für A-Z Beispiel!

PC: UDP-Port = 67, IP = ?, MAC = 01:01:01:01:01:01

Routingtabelle:

Netzadresse + Maske	Port	Gateway
192.168.1.0/24	eth0	direkt
default	eth0	192.168.1.1

Access Router: DNS Port = 53, UDP-Port = 67

- Interne IP = 192.168.1.1
- Externe IP = 195.1.2.1
- MAC = 00:02:02:02:02:02

Routingtabelle:

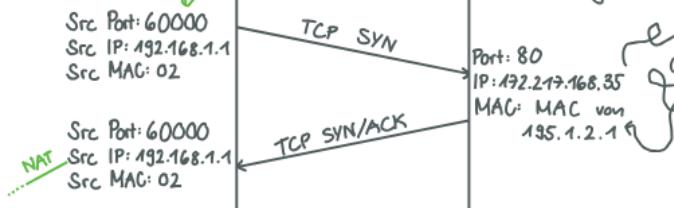
Netzadresse + Maske	Port	Gateway
192.168.1.0/24	eth1	direkt
default	eth0	195.1.2.1

NAT Tabelle

Inside	Outside
192.168.1.21:54321	60000

6. NAT:

Port 80 ist der offizielle Port für HTTP TCP/UDP Verbindungen.



6. NAT

TCP SYN → Port: 80, IP: 192.168.1.1, MAC: 02. NAT Tabelle:

Inside	Outside
192.168.1.21:54321	60000

Dest. Port: 54321
Dest. IP: 192.168.1.21
Dest. MAC: 01

TCP SYN/ACK → Port: 80, IP: 192.168.1.1, MAC: 02. NAT: