

Übertragungsmedien

Signale

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

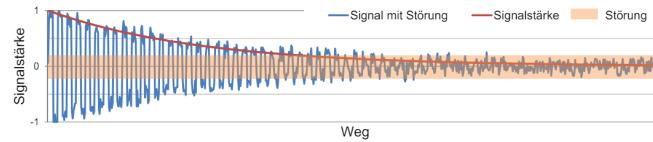
c_0 : Lichtgeschwindigkeit im Vakuum ($3 \cdot 10^8 \text{ m/s}$)

Signaldämpfung Leistungsabnahme auf Übertragungsstrecke

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

P_1 : Anfangsleistung, P_2 : Leistung am Ende der Strecke

U_1 : Anfangsspannung, U_2 : Spannung am Ende \Rightarrow Amplitude des Signals



Halbierung der Leistung entspricht ca. 3dB

Signal to Noise Ratio $SNR_{dB} = 20 \cdot \log\left(\frac{U_{Signal}}{U_{Störung}}\right)$

Dämpfungsbelag Dämpfungsbelag $= \frac{\text{Dämpfung}_{dB}}{\text{Länge}_m} \cdot 100$

Dämpfung pro Distanz - dB pro 100m (Coax < TP CAT7 < TP CAT3)

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

SNR_{min} : Minimales benötigtes SNR für korrekte Datenübertragung

- tiefere Bitrate \rightarrow grössere Distanzen können erreicht werden
- Die Bandbreite (Frequenz) ist abhängig zum Dämpfungsbelag.
- höhere Kabelkategorien haben bessere Schirmung \rightarrow 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, resistent, Geringe Dämpfung - aber Dispersion (Lichtgeschwindigkeit abhängig von Wellenlänge)
 - 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

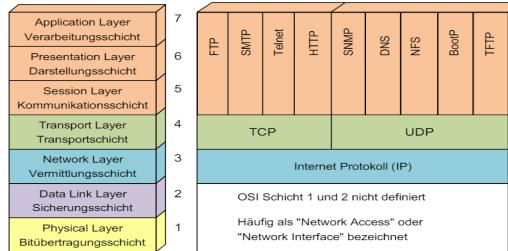
- Verbindungsaufbau nötig
- Informationen vom Empfänger \rightarrow Optionen aushandeln
- Reihenfolge der Daten bleibt erhalten

Zuverlässig

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

| | Verbindungsorientiert | Zuverlässiger Dienst |
|----------|-----------------------|----------------------|
| Ethernet | O | O |
| IP | 0 | 0 |
| TCP | X | x |
| UDP | 0 | 0 |

OSI Layers

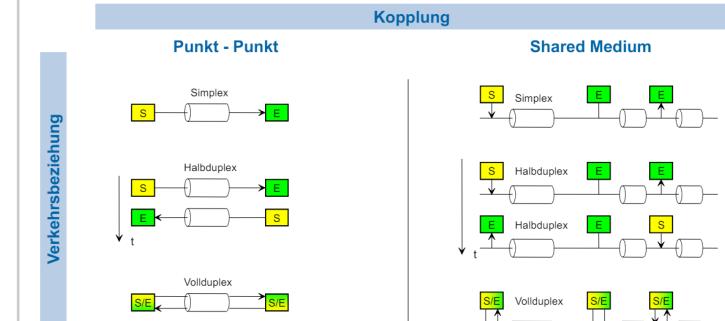


Aufgaben der Schichten:

- (5 und 6 nicht prüfungsrelevant)
1. **Physical Layer:** ungesicherte Übertragung eines Bitstroms
 - Bitübertragung, Signalisierung, Bit-Synchronisation
 - Definiert technische Details der Übertragung: Elektrische Eigenschaften, Codierung, Mechanische Eigenschaften
 - Standards: USB, Ethernet, WLAN, Bluetooth, etc.
 2. **Data Link Layer:** Realisieren einer zuverlässigen Verbindung
 - Fehlererkennung und -korrektur (z.B. CRC, Hamming-Code)
 - Framing/Frame Delineation (Präambel/SFD)
 - Flow-Control (Vermeidung von Überlastung)
 - Adressierung, Media Access Control (CSMA/CD), Timing
 - Protokolle: Ethernet, VLAN, Spanning Tree Protocol (STP), etc.
 3. **Network Layer:** Verbindet Netze, ermöglicht Kommunikation
 - Adressierung, Routing, Weiterleitung von IP-Paketen
 - Fragmentierung, Reassembly, Kapselung und Adressauflösung (ARP), Übertragung von Fehlermeldungen (ICMP)
 - Protokolle: IP, ICMP, ARP, etc.
 4. **Transport Layer:** Schnittstelle zwischen Betriebssystem (Kernel Space) und Anwendungen (User Space)
 - Zugriff via klar definierten Schnittstelle (Sockets)
 - Flusskontrolle, Reihenfolge, Fehlererkennung und -korrektur
 - Quality of Service (QoS), Multiplexing, Demultiplexing
 - Protokolle: TCP, UDP
 5. **Application Layer:** Anwendungsprotokolle
 - Kommunikation zwischen Anwendungen
 - NAT (Network Address Translation) - Übersetzung von IP-Adressen, Port Mapping (NATP)
 - Anwendungsprotokolle: HTTP, FTP, SMTP, DNS, DHCP, etc.

Physical Layer

Schicht 1: Bitübertragungsschicht

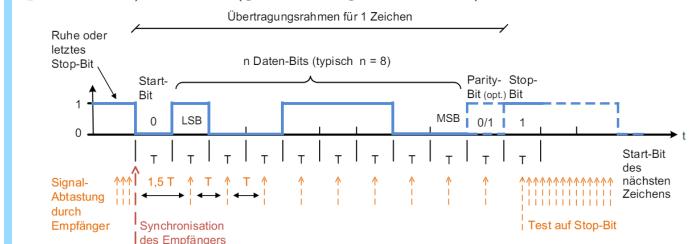


Übertragungsverfahren

Seriell Asynchron

\rightarrow Signalabtastung
Die Daten werden einfach geschickt, der Empfänger ist zuständig für das richtige Abschätzen des Taktes. Benötigte Abmachungen zwischen Sender und Empfänger:

Bitrate, Anzahl Datenbits (typisch 1 Byte), Anzahl Stoppbits (typisch 1 Bit), Parität (gerade, ungerade, keine)



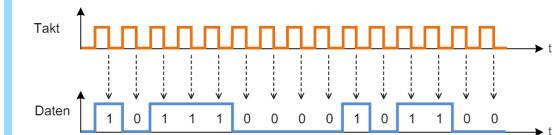
Achtung:

- Startbit/Stopbit gehören nicht zum Nutzdatenbyte
- Startbit: 0, Stopbit: 1, Parity-Bit optional
- Empfangen wird 1001 1100 – LSB first \rightarrow 0011 1001
- Genauigkeitsanforderung an Takte von Sender und Empfänger:
- Letzte Abtastung muss noch im Zeitfenster liegen (Stop-Bit bei einem Stop-Bit): hier also $1/2T$ auf $9\frac{1}{2}T$

Seriell Synchron

\rightarrow Taktübertragung/-rückgewinnung
Empfänger und Sender arbeiten mit gleichem Takt (synchronisiert): Keine Start- und Stoppbits benötigt
Takt muss zusätzlich übertragen werden (Codierungsverfahren oder zusätzliche Leitung)

Aufgabe vom Data Link Layer: Grenzen der einzelnen Bytes ermitteln



Die Taktrückgewinnung ist möglich, solange regelmässig Zustandsänderungen auftreten. Nachteil: Wenn viele Nullen geschickt werden, kann der Takt beim Empfänger nicht rekonstruiert werden.

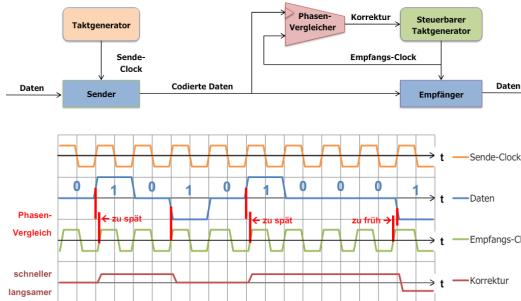
Taktrückgewinnung und Leitungscodes

Synchrone Übertragung ohne separate Takteleitung

Geeignete Codierverfahren erlauben den Takt zusammen mit dem Datensignal zu übertragen (Leitungscode)
Codierung Sender, Taktrückgewinnung/Decodierung Empfänger
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

Taktrückgewinnung

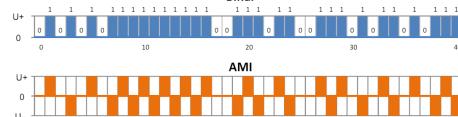


Anforderungen an Leitungscodes

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

bekannte Leitungscodes

AMI Leitungscode Alternate Mark Inversion (3-wertig)



Nachteil: drei Zustände benötigt → binäre Medien genügen nicht

Manchester Leitungscode 10BASE-T, 10Base2

erlaubt einfache Taktrückgewinnung: 1 = 0-1, 0 = 1-0

- 1 positive Flanke, 0 negative Flanke
- bei jedem Bit Signalwechsel



Nachteil: Bandbreite von 10 MHz benötigt (2x das theoretische Minimum)

NRZI MLT-3 Leitungscodierung 100BASE-TX

- NRZI: Non Return to Zero Inverted
- MLT-3:
Multi-Level Transmit-Ternary
- 1 = Wechsel, 0 = kein Wechsel



Kapazität, Bandbreite und Übertragungsrate

$$\text{Kanalkapazität } C_s = B \cdot \log_2(1 + \frac{S}{N}) \text{ Bit/s (bps)} \quad (\frac{S}{N} = SNR)$$

$$\text{Bandbreite } B = f_{\max} - f_{\min} \text{ Hertz (Hz)} \quad \text{maximale Symbolrate}$$

f_{\max} : Max. übertragbare Frequenz, f_{\min} : Min. übertragbare Frequenz

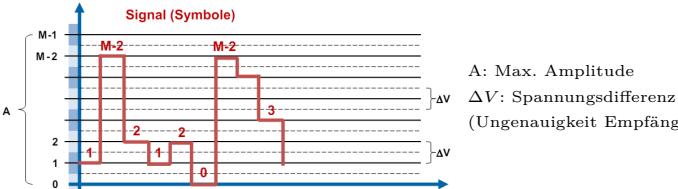
$$\text{Symbolrate } f_s = \frac{1}{T_s} = \text{Symbole pro Zeiteinheit} \quad f_s \leq 2B$$

T_s : Dauer eines Symbols, Einheit: Baud (Bd)

$$\text{Bitrate } R = f_s \times \text{Anzahl Bits pro Symbol} = \text{Bit/s (bps)}$$

$$\text{Maximal erreichbare Bitrate: } R \leq 2B \cdot \log_2(M)$$

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



Clock Drift

Maximale Framegrösse Ethernet: 1'500 Bytes.

- Standard: Oszillatoren brauchen Genauigkeit von ± 50 ppm
- 50 ppm (parts per million) → Fehler von 0.00005
- Worst-Case: Fehler = -50 ppm und +50 ppm (100 ppm Differenz)

Sicheres Abtasten von Daten? (im Worst-Case)

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

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
 $ld = \log 2$, $lg = \log 10$, ln = natürlicher Logarithmus

Data Link Layer

Schicht 2: Sicherungsschicht

Framing (Rahmenbildung-/erkennung)

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

Asynchron

- Keine Daten → Nichts 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

- ohne Unterbruch → kontinuierlicher Bitstrom auf Phy. Layer
- Stehen keine Daten an, werden Flags gesendet (anstatt Pause)
 - Frames werden durch ein Start-Flag und ein End-Flag begrenzt:
- Datenrahmen
Maskierung von
01111110 01111110
Flag L2-Header Datenblock Fehlererkennung Flag
Sonderzeichen (Flags) nötig!

Bitstopfen

- um 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) nie die Bitfolge 01111110

Framelänge und Fehlerwahrscheinlichkeit

Fehlerwahrscheinlichkeit

- BER (Bit Error Ratio) = 0.5 → jedes 2. Bit falsch
- 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:
 $BER = p_e \ll 1 \rightarrow (1 - p_e)^N \approx (1 - N \cdot p_e)$

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

Wahl der Framelänge

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

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

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

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

Fehlererkennung und -korrektur

Fehlererkennung

Redundanz → erhöht Hammingdistanz

Zuverlässigkeit: abhängig von Framelänge/Verfahren

Standards IEEE 802 (LAN-Standards, zb Ethernet): max. $5 \cdot 10^{-14}$ unentdeckte Fehler pro Frame-Byte, $BER p_e \leq 10^{-8}$ (1 Bitfehler pro 100 Mio. Bit) → CRC32 für Ethernet, mit Generatorpolynom (erkennt Fehler nur, korrigiert nicht)

Fehlerkorrektur - Error Correction (EC)

- Backward (BEC): erneutes Übertragen der Daten
- Forward (FEC): Rekonstruktion verfälschter Bits bei 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: Anzahl 1en inkl. Parity-Bit ist gerade (Odd analog)

Längs- und Quer-Parity

3 Paritätsbits: 1 für jedes Byte, 1 für jedes Bit (Längs- und Quer-Parity), Gesamtparity

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

Zugriffsmechanismen (Media Access)

Gesteuerter Medium Zugriff

Master-Slave Verfahren

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

Token Verfahren

Sendeberechtigung in fixer Reihenfolge

Knoten senden nur, wenn sie ein Token halten

- Vorteil: Deterministisch (man weiß, wann man dran kommt)
- Nachteil: Aufwändiges Token Management (Startup, Token Verlust, etc.)

Zeitgesteuerter Zugriff

- Vorteil: Optimierung möglich (nach Auslastung, Durchsatz, etc.)
- Nachteile: Planung und genaue Zeit in allen Knotenpunkten erforderlich, Konflikte mit unplanbarem Verkehr

Random Medium Zugriff

Carrier Sense Multiple Access

Vor Senden geteiltes Übertragungsmedium abhören ob frei (Carrier Sense), sonst bis Pause warten

- Vorteil: Alle Stationen gleichberechtigt (kein Master) → jederzeit Zugriff auf Übertragungsmedium
- Nachteil: Kollisionen möglich (Collision Detection)

Kollisionsbehandlung - CSMA

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

Flow-Control

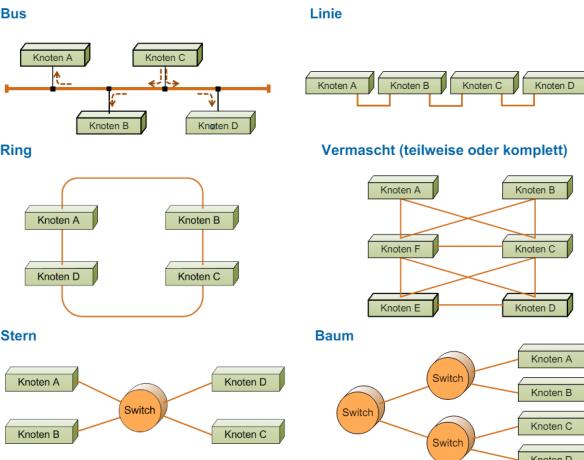
Explizite Start-Stopp Signalisierung:

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

Implizites Stop-and-Wait: Sender wartet auf ACK vor Senden

Ethernet und LAN

Local Area Networks (LAN) Topologien



Übertragung und Adressierung

Übertragungsarten

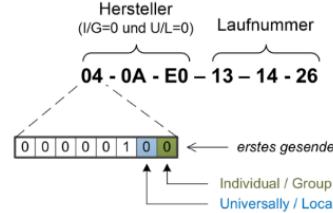
Immer genau 1 Sender, E = # Empfänger



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

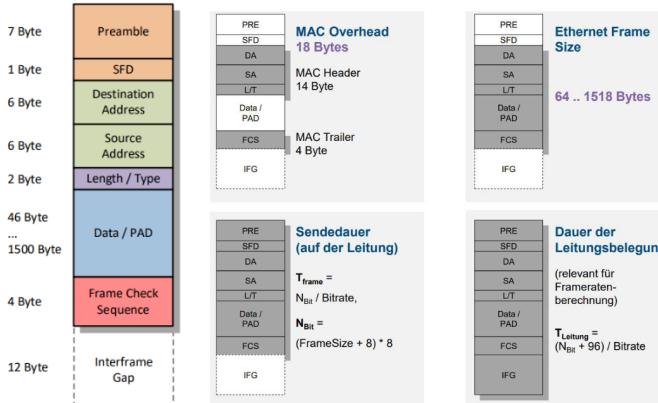
IEEE MAC Adressen

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



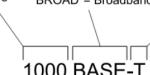
- Individual/Group Bit:**
- 0 = individual address
 - 1 = group address
- Universally/Locally Bit:**
- 0 = universally administrated address
 - 1 = locally administrated address

Ethernet Frame Format



Bezeichnungsschema und Datenraten

Bitrate in Mbit/s BASE = Baseband BROAD = Broadband Art / Codierung des Mediums (früher max. Segmentlänge in m)



| | |
|-------------------|------------------------------------------------|
| Beispiele: | T, TX, T1: SR, DR, LR; C: K: |
| | Twisted Pair optisch Twinax Backplane |

Ethernet Frame Format und MAC-Adresse

Sende Ethernet-Frame über 100BASE-TX Schnittstelle

Bit-Sequenz auf Kabel:

10101010 10101010 10101010 10101010 10101010 10101010
10101010 10101011 00010000 00000000 01011010 11100011
10011111 10000110 ...

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) 11111001 (=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 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 (Address-Learning)

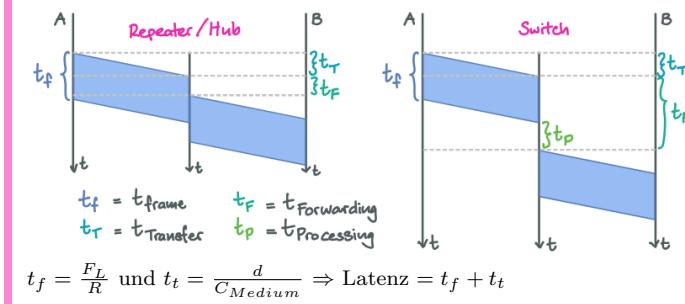
Leistungskennzeichen von Switches und Bridges

| | |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Anzahl Ports | Steckergroesse ist im Extremfall die Limitierung |
| Adressstabelle | Wie viele Stationen können im LAN existieren |
| Filtratoren | Maximale Frames / s / Port (Empfangsrichtung) |
| Transfertate | Maximale Frames / s / Port (Senderichtung) |
| Backplane / Fabric Kapazität | Maximaler Gesamtdurchsatz zwischen allen Ports |
| Architektur | <p>Store-and-Forward: Frame wird komplett empfangen und dann weitergeleitet</p> <p>Cut-Through: Frame wird schon nach Decodierung der Zieladresse weitergeleitet</p> <p>Adaptive Cut-Through: Schaltet bei hoher Fehlerrate automatisch auf Store-and-Forward um</p> |
| 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

Maps MAC-Adressen auf Ports (lernt nur Absenderadressen)
Wenn Adresse bekannt → direkt an diese senden!
Sonst: Flooding (Broadcast/Multicast)
Vergisst Einträge nach gewisser Zeit (Aging Time)

Weg/Zeit-Diagramm für das Senden eines Frames



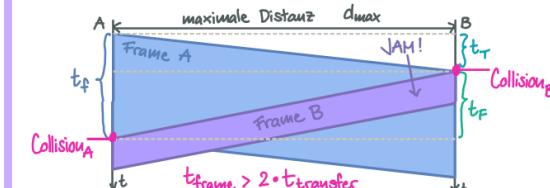
$$t_f = \frac{F_L}{R} \quad \text{und} \quad t_t = \frac{d}{C_{Medium}} \Rightarrow \text{Latenz} = t_f + t_t$$

t_f verlängern ⇒ Verarbeitungszeit ermöglichen

$$F_L = \text{Frame Länge/Framesize}, R = \text{Bitrate}, d = \text{Distanz}, C_{Medium} = 2/3c_0$$

Kollisionserkennung

Überlagerung von Signalen



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

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

Bedingung für Kollisionserkennung mit Repeater: $t_{frame} > 2 \cdot (\sum t_{transfer} + \sum t_{forwarding})$

Redundanz (Spanning Tree)

Spanning Tree Algorithmus Redundante Pfade → Probleme!

⇒ Ziel: Alle Segmente loop-frei verbinden

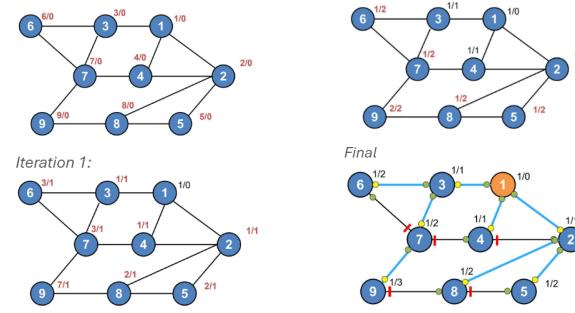
Initialisierung: Alle Ports für Nutzdaten blockiert, Annahme: «Ich bin Root», Austausch BPDUs mit Nachbarn (Root ID, Root Cost, Bridge ID, Port ID)

Aufbau des Spanning Tree: «kleinster» Nachbar als Root gesetzt
→ Anzahl Hops + 1 (Beachte Prioritätswert)

wiederholen bis alle dieselbe Root ID haben

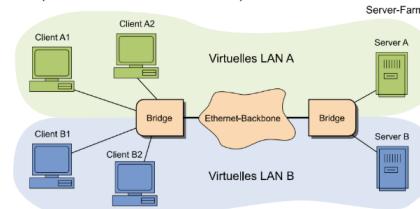
Setzen der Port Roles: Root-Ports (Empfang der «besten» BPDU), Designated-Ports (Weg zum «kleinsten» Nachbar), Blockierte Ports (Discarding)

Initialisierung:



Virtuelle LANs

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

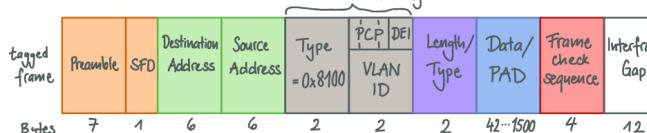


Trunk Links: Teil mehrerer VLANs
→ Frames eindeutig kennzeichnen!

Trunk = Tagged Access = Untagged

VLAN Tagging

Erweiterung Ethernet Header (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 relativ simpel)

Switch Konfiguration:

| | Px | Trunk Port |
|----------|-----|-------------|
| | Px | Access Port |
| Switch | | |
| P1 | P2 | P3 |
| P4 | P5 | P6 |
| P7 | | |
| VLAN IDs | All | 2 7 2 1 2 2 |
| | 4 | 3 3 3 5 |
| | 5 | 7 4 7 |
| | | 8 8 |

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

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 |

| 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

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

- (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

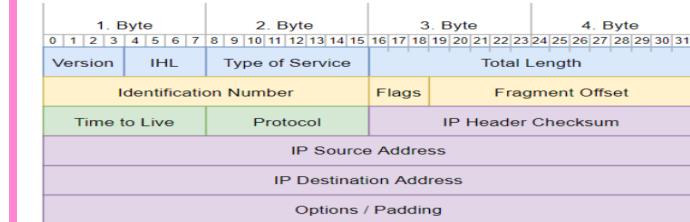
- Info wie jedes Netz/Interface erreicht werden kann
- Für Weiterleitungsentscheidung notwendige Informationen:
 - Eintrag für jedes erreichbare Netz (Netzadresse, Netzmaske)
 - Interfaces, über die die Netze erreicht werden können
 - IP-Adresse des nächsten Routers, wenn Zielnetz nicht direkt erreicht 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

IP-Packet = Header (min. 20 Byte) + Nutzdaten

- Version IPv4 / IPv6
- IHL Header Length in 4-Byte (20 Byte → IHL = 5 (Minimum)) wenn Optionen vorhanden, 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 (bei jeder Station -1) wenn 0 → Paket wird verworfen
- Protocol Übergeordnetes Protokoll (TCP, UDP, ICMP, etc.)
- 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



IP-Header wird von jedem Router neu berechnet!

Achtung TTL: Router verwirft Paket wenn nach dekrementieren = 0

Protocol: 1=ICMP, 2=IGMP, 3=GGP(Gateway-to-Gateway), 4=IP in IP(encapsulation), 5=Stream, 6=TCP, 16=UDP, 17=UDL, 125=FILE

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

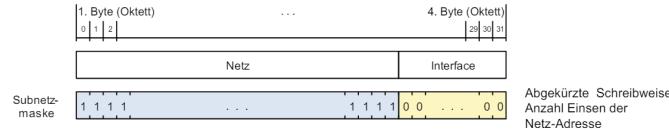
Netzadresse

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

Broadcast-Adresse

- Reserviert: adressiert alle Interfaces in einem Subnetz
- Höchste Adresse im Subnetz (All Ones Broadcast)
- Berechnet durch: Interface-Adresse OR Invertierte Subnetzmaske

Subnetzmaske bestimmt Grenze zwischen Netz-/Interface-Adressbits:



Netzmasken bestimmen Anzahl adressierbarer Interfaces im Netz
Auflistung: Wert (dezimal/binär), alternative Schreibweise, Anzahl der adressierbaren Interfaces

| | | | | | |
|----------------|-----|------|----------------|-----|-------|
| 255 (11111111) | /24 | 256 | 240 (11110000) | /20 | 4096 |
| 254 (11111110) | /23 | 512 | 224 (11100000) | /19 | 8192 |
| 252 (11111100) | /22 | 1024 | 192 (11000000) | /18 | 16384 |
| 248 (11111000) | /21 | 2048 | 128 (10000000) | /17 | 32768 |
| | | | 0 (00000000) | /16 | 65536 |

Achtung: bei der Anzahl der adressierbaren Interfaces immer noch -2 rechnen (Netz- und Broadcastadresse)!

Rechnen mit Netzmasken Typische Aufgabenstellung

| IP-Adresse | Netzmaske | Netzadresse | Broadcastadr. | #Adr. |
|-------------|---------------------|-------------|---------------|-------|
| 17.8.7.8 | 255.255.0.0 /16 | 17.8.0.0 | 17.8.255.255 | 65536 |
| 11.7.177.4 | 255.255.224.0 /19 | 11.7.160.0 | 11.7.191.255 | 8192 |
| 144.3.133.1 | 255.255.192.0 /18 | 144.3.128.0 | 144.3.191.255 | 16384 |
| 31.4.2.166 | 255.255.255.248 /29 | 31.4.2.160 | 31.4.2.167 | 8 |

Anzahl (#) der adressierbaren Interfaces inkl. Netz- und Broadcastadresse

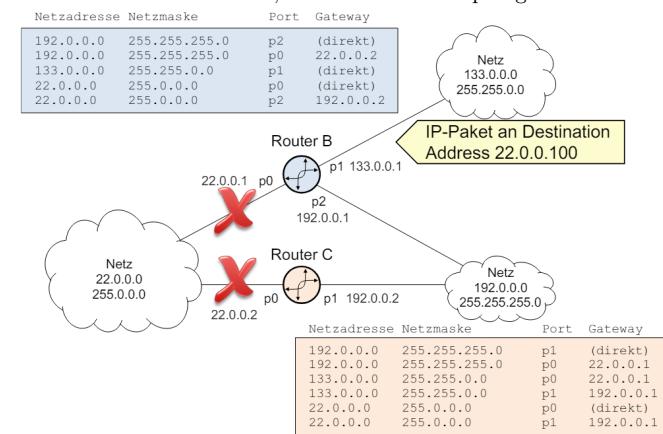
Flaches, Hierarchisches und Classful Routing

Flaches Routing

- Router kennt (evtl. mehrere) explizite Wege zu jedem Zielnetz – Pakete an unbekannte Netze werden verworfen
- Einsatz: stark vermaschte Netze / zentraler Bereich (Backbone)
- Nachteil: Sehr grosse Routing-Tabellen

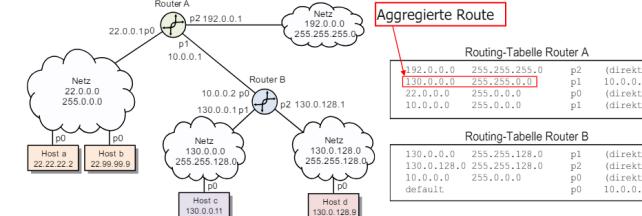
Flaches Routing Übung

- Was geschieht mit dem IP-Paket?
- Kein Unterbruch: Es wird nach gemäss dem 4. Eintrag der Routingtabelle von Router B an p0 weitergeleitet
 - Unterbruch von p0/Router B: Es wird gemäss Eintrag 5 in der Routingtabelle von Router B an p2 weitergeleitet
 - zusätzlicher Unterbruch p0/Router C: Router C kann das IP-Paket nicht weiterleiten, es erreicht den Empfänger nicht



Hierarchisches Routing (Default)

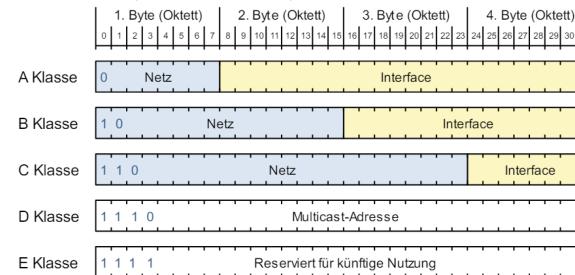
- Router kennt die direkt angeschlossenen Netze seiner Interfaces und genau einen anderen Router, an den er alles schickt, was für andere Netze bestimmt ist
 - Der nächste Router geht genau gleich vor
- Einsatz am „Rand“ von Netzen Hosts, ccess Router
- Kleine Routing-Tabellen mit jeweils einem Default-Eintrag



| Routing-Tabelle Router A | Routing-Tabelle Router B | Routing-Tabelle Router C |
|-------------------------------------|-------------------------------------|-------------------------------------|
| 192.0.0.0 255.255.255.0 p2 (direkt) | 130.0.0.0 255.255.128.0 p0 (direkt) | 130.0.0.0 255.255.128.0 p1 (direkt) |
| 10.0.0.0 255.0.0.0 p0 (direkt) | 10.0.0.1 22.0.0.1 p2 | 130.0.0.1 22.0.0.1 p0 |

| Routing-Tabelle Host a | Routing-Tabelle Host b | Routing-Tabelle Host c | Routing-Tabelle Host d |
|--------------------------------|--------------------------------|-------------------------------------|-------------------------------------|
| 22.0.0.0 255.0.0.0 p0 (direkt) | 22.0.0.0 255.0.0.0 p0 (direkt) | 130.0.0.0 255.255.128.0 p0 (direkt) | 130.0.0.0 255.255.128.0 p0 (direkt) |
| default | 22.0.0.1 | default | 130.0.0.1 |

Classful Routing ursprünglich fünf Netzklassen (A - E) für IPv4
Eine Prefix (Adress-Bits 0-3) erlaubt die Bestimmung der Klasse:



Internet-Adressierung (IPv4 Netz-Klassen)

| Klasse | Adressbereich | Anzahl Netze | Interfaces pro Netz |
|--------|-----------------------------|-----------------------------------|---------------------|
| A | 1.0.0.0 – 127.255.255.255 | 127 | 16'777'214 |
| B | 128.0.0.0 – 191.255.255.255 | 16'384 | 65'534 |
| C | 192.0.0.0 – 223.255.255.255 | 20'971'52 | 254 |
| D | 224.0.0.0 – 239.255.255.255 | Multicast Adressen | |
| E | 240.0.0.0 – 255.255.255.255 | Reserviert für zukünftige Nutzung | |

Private Adressbereiche (werden im Internet nicht weitergeleitet):

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

Adressbereiche für Classful Routing

- klassische Netze fixer Grösse sind unflexibel (ungeeignet für Unternehmen) → C zu klein, A zu gross, B zu wenig
- Abhilfe schafft CIDR – Classless Inter-Domain Routing
 - Flexible Verwendung von Netzmasks beliebiger Länge
 - Sub- und Supernetting

localhost Loopback-Adressen

Das gesamte A-Netz 127.0.0.0/8 ist für Loopback-Test reserviert

Sub- und Supernetting

Supernetting

Zusammenfügen von kleinen Netzen
Hintereinanderliegende C Netze zu einem Netz zusammenfügen
Bonus: Routingtabelle in Routern verkleinern (Aggregate Routes)

Zusammenfassen von 4 Class C Netzen (22 = 2 Bits der Subnetzmaske)

| | | |
|-----------------|-----------|---------------------------|
| 198.51.0110010 | 0000 0000 | = C-Netz 198.51.100.0 /24 |
| 198.51.01100101 | 0000 0000 | = C-Netz 198.51.101.0 /24 |
| 198.51.01100110 | 0000 0000 | = C-Netz 198.51.102.0 /24 |
| 198.51.01100111 | 0000 0000 | = C-Netz 198.51.103.0 /24 |

198.51.01100101 00000000 = Subnetzmaske 255.255.252.0 oder /22

198.51.01100101 00000000 ← Netz-Adresse 198.51.100.0, Netz 198.51.100.0 /22
192.51.01100111 11.1111 1111 ← Broadcast-Adresse

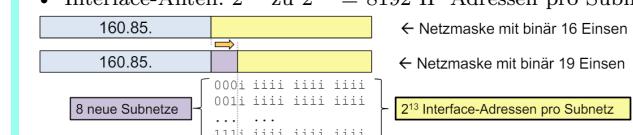
Subnetting

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 Netzmasks-Bits:

- $8 = 2^3$, 3 Bits identifizieren 8 Subnetze (000 → 111)
- Netzmaske: /16 zu /19 (255.255.0.0 → 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
- 169.85.0110 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 /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

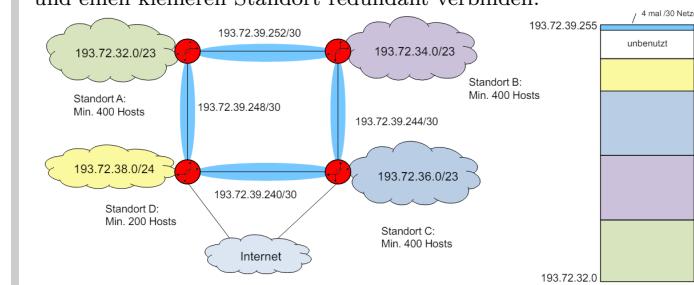
Das fünfte Netz wird wie folgt gekennzeichnet:

- 160.85.128.0 / 255.255.224.0 oder 160.85.128.0 /19

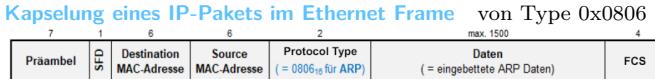
Wichtige Regel: Eine Netzwerkadresse ist immer ein Vielfaches der Netzgrösse!

Flexible Aufteilung eines Netzbereiches

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

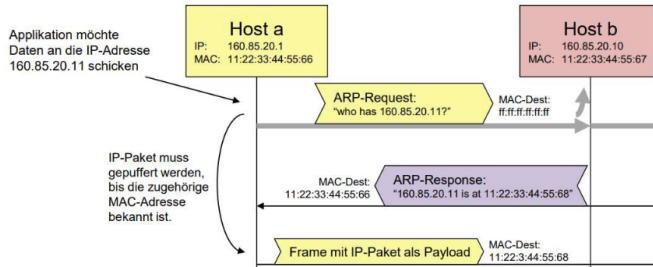


Kapselung und Adressauflösung



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

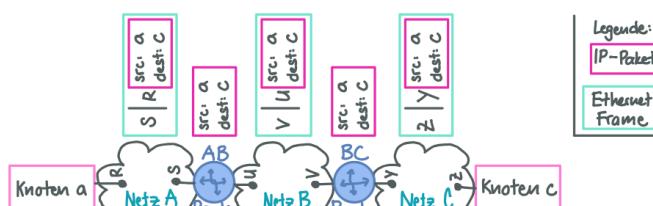
ARP Nachrichtenstruktur

| | | |
|-----------------------------|---------------------------------------------------|----------------------------------------------------------------|
| 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!

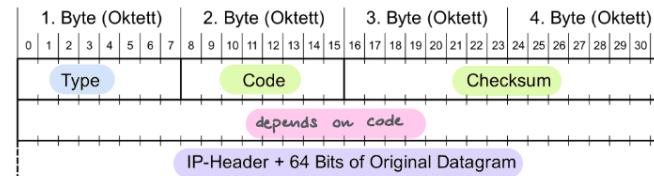
Internet Control Message Protocol (ICMP)

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

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)

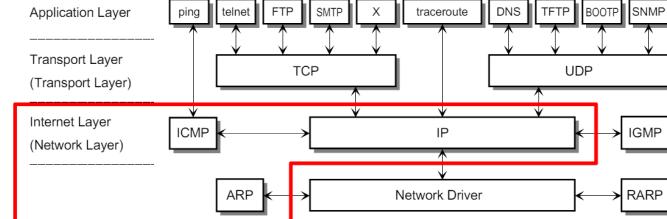
- Sende IP-Pakete mit Länge=PMTU und mit DF=1
- Empfange «Destination Unreachable» mit Code 4 «fragmentation on needed and DF set»
- PMTU reduzieren auf «Next-Hop MTU» (enthalten in Octet 5..8)

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**

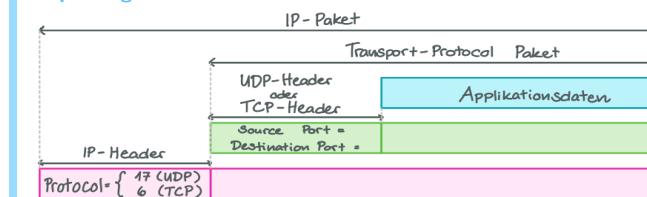
Zusammenhänge:



Transport Layer

Schicht 4: Transportschicht

Kapselung



Adressierung

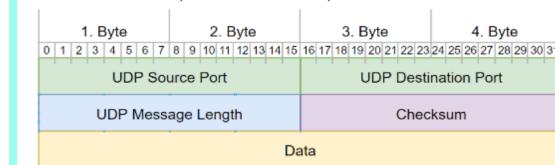
- Client adressiert Server-Appl. mit Destination Port Nr.
- sonst weiss TCP/UDP-Modul im Empfänger nicht, welche Applikation gemeint ist
 - für Source Port Nummer verwendet Client (meist) zufällige Port Nummer >1023 (vom Betriebssystem vergeben)

UDP - User Datagram Protocol

UDP Multi-/Demultiplexen: anhand Portnummern Daten den richtigen Applikationen zuweisen ⇒ Verbindungslos und unzuverlässig Vorteile: simpel, wenig Overhead, kein Verbindungsauflauf (schnell)

UDP-Header

- Source/Destination Port Sendende/Empfänger Applikation
- 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 → fehlgeleitete Datagramme können erkannt werden

Port-Nummern

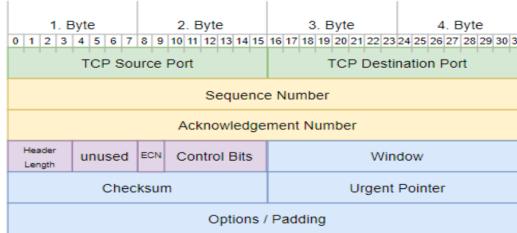
- System Ports (Well-Known) Fix, für bekannte Appl. reserviert
- User Ports (Registered) Reserviert für herstellerspez. Appl.
- Dynamic/Private Ports Frei verfügbare Ports

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

TCP Eigenschaften Verbindungsorientierte Übertragung, zuverlässiger Verbindungsauflaufbau, hohe Zuverlässigkeit, Voll duplexübertragung, Stream-Schnittstelle, Graceful Termination, Punkt-zu-Punkt Kommunikation

TCP-Header Format

- Sequence-Nr. Sicherstellung Reihenfolge, Erkennung lost Data
- Acknowledgement-Nr. n+1: Daten bis & mit n korrekt/vollständig
- 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 Verbindungsauflauf- und -abbau (Bits 10-15)
- Window Verfügbare Puffergrösse
- Urgent Pointer URG = 1 → Position der wichtigen Daten
- Options häufigste Verwendung: MSS (Maximum Segment Size)



Control Bits: URG: Urgent Pointer, ACK: Acknowledgement Number (Bestätigung empfangener Daten, Erkennung verlorenen Daten), PSH: Push (sofort ohne buffern weiterleiten), RST: Reset (Verbindung zurücksetzen oder geschlossenen Port signalisieren), SYN: Verbindungsauflaufbau, FIN: Verbindungsabbau

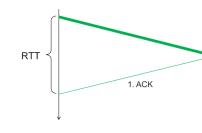
Herausforderungen zur Zuverlässigkeit zwischen Ethernet/TCP:

| Problem | Schicht 2 | Schicht 4 | Massnahmen bei TCP |
|-------------------------|-----------------------------|--------------------------|----------------------------------------------|
| Nachrichtenverlust | PVerlust = FER | PVerlust >> FER | Positives ACK |
| Telegramm-Reihenfolge | fix | kann variieren | Sequenznummern |
| Round Trip Time | konstant, $\mu s \dots ms$ | variabel, $ms \dots s$ | Adaptiver 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

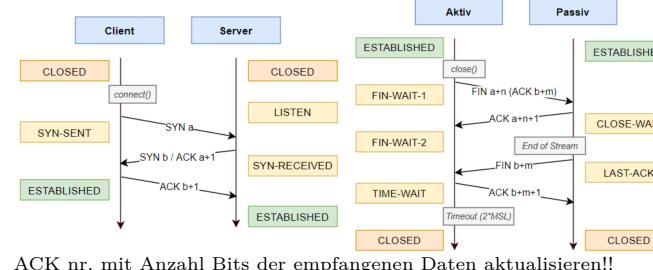
Round Trip Time (RTT) dynamische Anpassung der Wartezeit

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



Bandwidth Delay Product (TCP-Puffergrößen)
Wahl der Grösse von Sende- und Empfangspuffer, um Verbindung nicht auszubremsen
 $BDP(bits) = RTT(sec) \cdot Bandbreite(bps)$

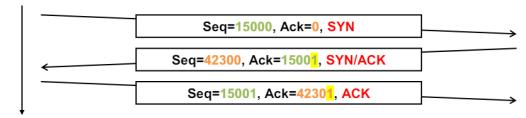
Verbindungsauflaufbau und Verbindungsabbau



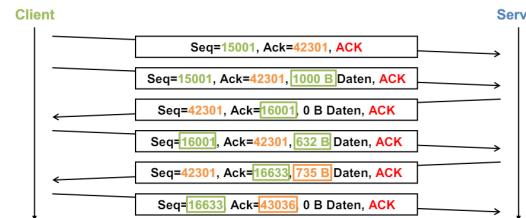
Vollständiges Beispiel

Verbindungsauflaufbau:

- Server: (LISTEN) auf 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

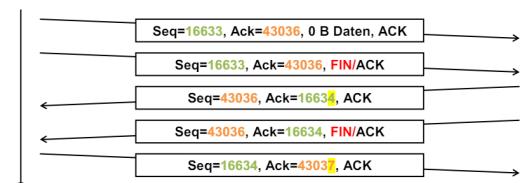


Datenaustausch: TCP-Nachrichten werden bi-direktional ausgetauscht



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 → FIN/ACK



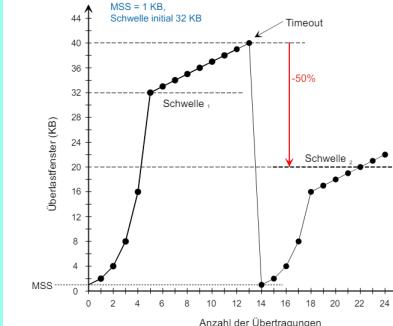
Mehrere Clients wollen gleichzeitig auf denselben Server zugreifen? Verbindungsanfragen (SYN) kommen nacheinander zum Server. 1. SYN → Verbindungsauflaufbau zum entsprechenden Client. Weiteren Anfragen werden gequeued. Nach Schliessen der 1. Verbindung wird nächste Anfrage der Queue entnommen/behandelt, usw.

Fluss-Steuerung und Congestion Control

Fluss-Steuerung aus Perspektive Sender

- Stop-and-Wait: wartet auf Bestätigung bevor er weiter sendet
- Sliding Window: sendet mehrere Frames bevor er auf Bestätigung wartet

Congestion Control - Slow Start



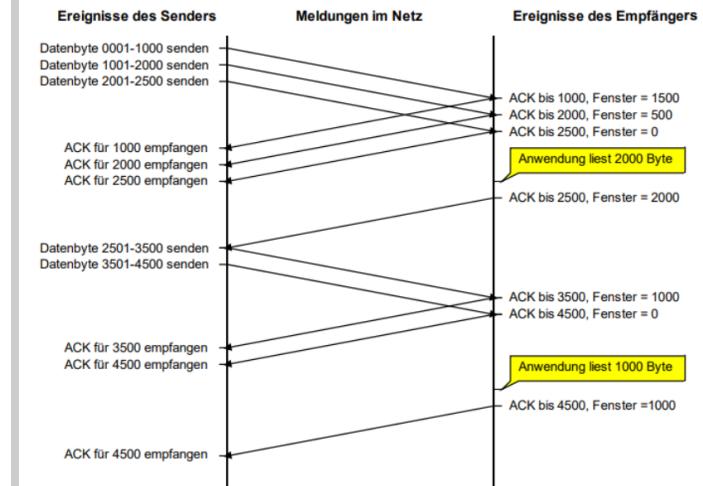
Slow Start: herantasten wie gross die einzelnen Frames sein können.

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

Sliding-Window TCP

- Beide Seiten haben ein Fenster, das die Anzahl der Bytes angibt, die gesendet werden können
- Verbindungsauflaufbau: 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

Von A bis Z: Aufruf einer Webseite

Ports, Adressen und Routingtabellen für A-Z Beispiel!

PC:

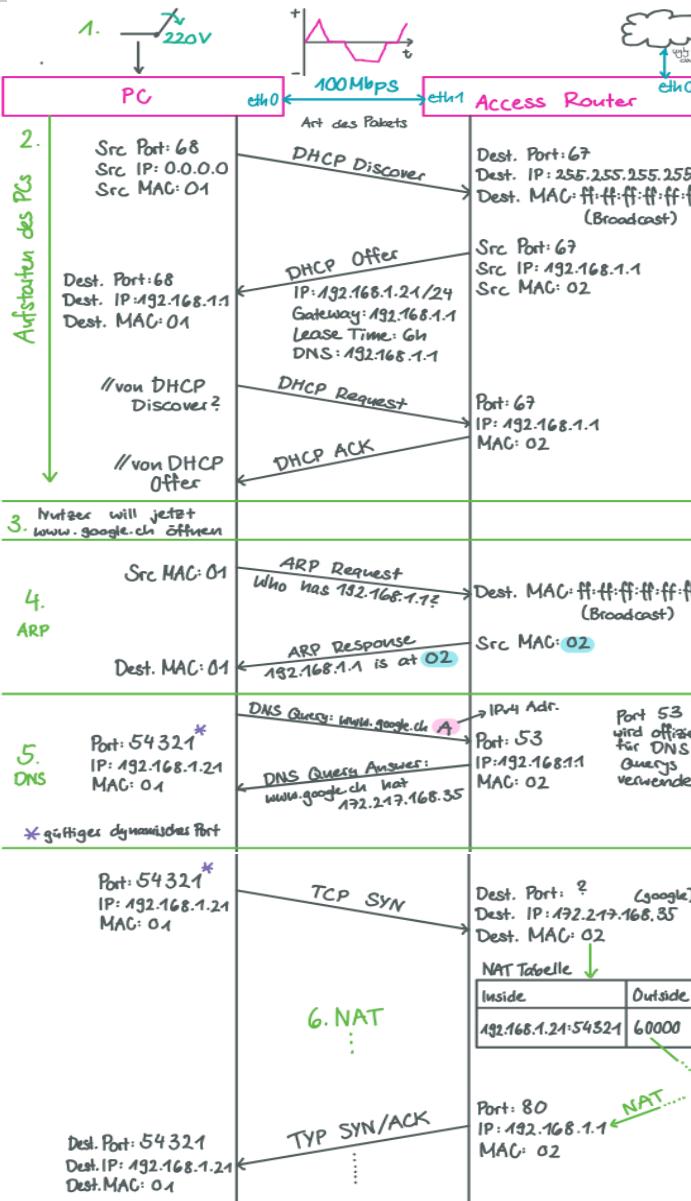
- UDP-Port = 67
- IP = ?
- MAC = 01:01:01:01:01:01

| 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

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



Application Layer

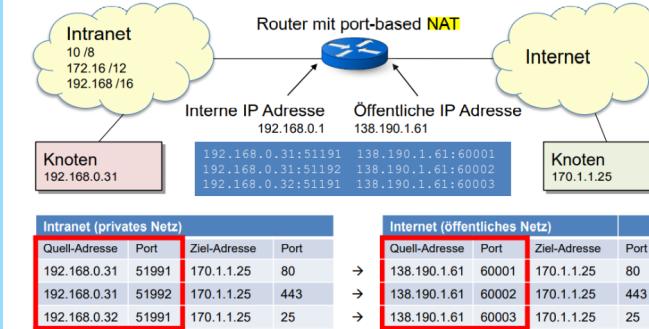
NAT - Network Address Translation

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

Problem mit NAT: Verletzung des OSI-Layer-Konzepts

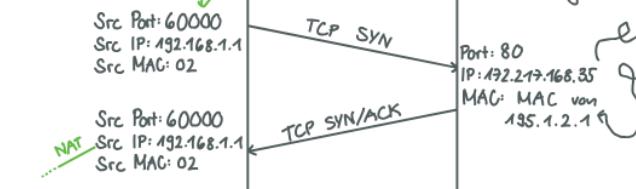
Um Port im TCP Header zu ändern müssen Daten im IP-Frame verändert werden → Netzwerk-Funktion greift auf den Transport Header zu, IP-Adresse/Portnummer werden dabei verändert



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

Funktionalität NAT in Worte gefasst: Der NAT-Router muss... Verbindungsinformationen speichern, um Antwortpakete an richtigen Rechner weiterleiten zu können Port-Nummer speichern, um Antwortpakete an richtigen Prozess auf Rechner weiterleiten zu können öffentliche IP-Adresse in lokale IP-Adresse übersetzen, damit Rechner Antwortpakete empfangen kann lokale IP-Adresse in öffentliche IP-Adresse übersetzen, damit Server Antwortpakete an Rechner senden kann

| NAT Tabelle | |
|--------------------|---------|
| Inside | Outside |
| 192.168.1.21:54321 | 60000 |



Beispiele

Übertragungsmedien

Bandbreite von Twisted-Pair Kabeln

Bei allen TP Kabelkategorien steigt der Dämpfungsbelag mit zunehmender Frequenz beinahe gleich an

Was ist der Grund, dass CAT5 Kabel mit einer Bandbreite von 100 MHz verwendet werden, während CAT7 Kabel bis zu 600 MHz eingesetzt werden?

Antwort: Die Dämpfung ist nicht die einzige Größe, welche die effektiv nutzbare Bandbreite eines Kabels bestimmt. Höherwertige Kabel haben eine viel besseres Übersprechverhalten, so dass eine höhere Dämpfung akzeptiert werden kann.

Optische Übertragungsmedien

Die optische Sendeeinrichtung sendet mit 10^{-3} Watt Leistung. Für eine genügende Übertragungsqualität muss der Leistungspegel beim Empfänger mindestens 10^{-5} Watt betragen. Der Lichtwellenleiter hat einen Dämpfungsbelag von 0.5 dB/km. Welche Distanz kann überbrückt werden?

Antwort: Die erlaubte Abschwächung der Leistung entspricht dem Faktor 100, also 20 dB. Demzufolge darf die Leitung maximal 40 km lang sein.

OSI Modell

Klassifizierung von Diensten Zwei Schichten M und N haben zur jeweils höheren Schicht die folgenden Interfaces. Klassifizierung der Dienste?

M:

connect_M (dest_addr, options, &handle)
send_M (handle, options, numBytes, &data)

..

close_M (handle)

Antwort: verbindungsorientiert, Connect/Close der Verbindung

N:

send_telegramm_N (remote_addr, numBytes, &data)
receive_telegram_N (&remote_addr, &local_address, &buffer, buf-siz)

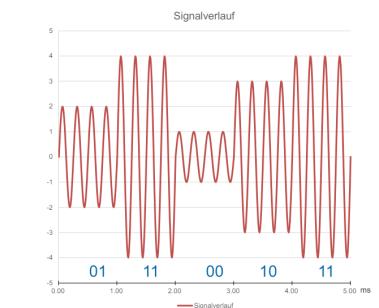
Antwort: verbindungslos, keine Funktion zum Auf-/Abbau einer Verb.

Physical Layer

Beispiel Amplitude Shift Keying (ASK-4)

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

- Baudrate: 1 kBaud
- Bit pro Symbol: Id {4} = 2
- Bitrate: 2 kBit/s
- Zusatzfrage Trägerfrequenz: 4 kHz



Maximale Bitrate Ein Übertragungssystem nutzt eine Bandbreite von 100MHz und codiert den Bitstrom mit 4wertigen Symbolen. Welches ist die theoretisch erreichbare maximale Bitrate?

Antwort: $\max_{Symbolrate} = 200\text{MBd}$
2 Bit/Symbol $\rightarrow \max_{Bitrate} = 400\text{Mbit/s}$

Zur Steigerung der Bitrate wird ein 8-wertiger Code in Erwägung gezogen. Unter welchen Voraussetzungen ist das möglich?

Antwort: Der Signal/Rauschabstand S/N muss genügend gross sein. Das ergibt sich aus der Sendeleistung, der Leitungsdämpfung und der Störleistung

Maximale Zeichenrate Wir betrachten eine asynchrone Schnittstelle, welche mit folgenden Parametern betrieben wird: Bitdauer T ist 1 ms, 8 Bit/Zeichen, 1 Stopp-Bit.

a) Welche maximale Zeichenrate lässt die Schnittstelle zu?

Antwort: $1000\text{ Bit/s} / 10\text{ Bit/Zeichen} = 100\text{ Zeichen/s}$

b) Um wieviel darf die Frequenz des Empfängertaktgebers von dem des Senders maximal abweichen, ohne dass das einen Übertragungsfehler bewirkt? Relative Angabe in Prozent.

Antwort: Zeitmessung startet mit der fallenden Flanke des Start-Bits nach 9.5^*T ist man im Idealfall in der Mitte des letzten Datenbits, Fehlablesung entsteht dann, wenn man um 0.5^*T daneben liegt
 $\rightarrow 0.5 * T / 9.5 * T = 1/19 = 5.26\%$

c) Wir betrachten den Fall, bei dem die Frequenz des Empfängertaktgebers geringfügig höher ist, als der unter b) errechnete Wert. Es wird das Zeichen 10101010 gesendet. Welches Zeichen detektiert der Empfänger?

Antwort: Das zuletzt übertragene Bit wird nicht abgetastet, dafür das vorhergehende zweimal. Weil das letzte Bit das MSB ist, wird das Zeichen 00101010 empfangen.

Codierung auf Leitungen

a) Welche Eigenschaft muss ein Leitungscode aufweisen, dass der Empfänger den Takt aus dem Datenstrom extrahieren kann?

Antwort: Der Bitstrom muss so codiert werden, dass er - unabhängig von den übertragenen Daten - genügend häufige Pegeländerungen (Signalflanken) aufweist.

b) Nennen Sie zwei Codes, welche die Bedingung unter a) erfüllen.

Antwort: Manchester (10Base2, 10BASE-T), dreiwertiger NRZI mit 4B5B Codierung (100Base-TX), 4B3T (10BASE-T1L)

c) Aus welchen Gründen kann es notwendig sein, dass ein Leitungscode gleichstromfrei ist?

Antwort: Wird das Signal galvanisch getrennt über einen Transistor geführt wird, dann geht der Gleichstromanteil verloren.

d) Nennen Sie zwei Codes, welche gleichstromfrei sind. **Antwort:** AMI, 4B3T (10BASE-T1L), dreiwertiger NRZI (100Base-T)

Ein Übertragungsverfahren soll 320 MBit/s übertragen können. Die im Übertragungsmedium nutzbare Bandbreite B beträgt 40MHz. Wie viele Zustände pro Symbol muss das Codierungsverfahren bieten, damit bei maximaler Symbolrate die geforderte Bitrate erreicht werden kann?

Antwort: Hartley: Bitrate = $2 * B * \text{Id}(N)$
 $\rightarrow 320\text{Mb/s} \leq 80\text{MHz} * \text{Id}(A)$

$\rightarrow \text{Id}(A) \geq 4$ Damit werden mindestens $2 * 4 = 16$ Zustände benötigt.

Data Link Layer

Datenübertragung und Framing

Gegeben: Bitrate 100 Mbit/s, Frame-Länge 1000 Byte, IFG 96 Bit, Payload 800 Byte. Berechnen Sie die Nutzdatenrate.

Lösung:

$$F_R = \frac{100 \cdot 10^6}{8 \cdot (1000 + 96)} = 1.19 \cdot 10^6 \text{ Frames/s}$$

$$N = 1.19 \cdot 10^6 \cdot 800 \cdot 8 = 7.6 \cdot 10^9 \text{ Bit/s} = 7.6 \text{ Gbit/s}$$

Bitstuffing

Synchrone Datenübertragung und Codes
Bei der synchronen Datenübertragung werde das Flag 01111110 und Bit-Stuffing (Bitstopfen) verwendet.

a) Wozu verwendet man hier Bit-Stuffing?

Antwort: Start-/Ende-Flags dürfen nicht in den eigentlichen Daten vorkommen, da dies vom Empfänger als Flag detektiert würde

b) Wie sieht der folgende gesendete Bitstrom auf der Leitung aus?

Antwort:

1010111110100111111111000000101011111011111101
1010111101010011111011100000010101111100111110101

Fehlerkorrektur beim Stop-and-Wait Protokoll

Der Sender erkennt durch einen Timeout eine fehlende Bestätigung und wiederholt daraufhin das zuletzt gesendete Telegramm.

LAN und Ethernet

Gbase-T

| Name | Standard | Speed (Mbit/s) | # TP | Coding | Baud rate per lane (MBd) | Bandwidth | Max distance (m) | Cable | Cable rating (MHz) |
|------------|--------------|----------------|------|----------------------|--------------------------|-----------|------------------|--------|--------------------|
| 10GBASE-T | 802.3an-2006 | 10000 | 4 | 64B6B PAM-16 128-DSQ | 800 | 400 | 100 | Cat 6a | 500 |
| 5GBASE-T | 802.3bz-2016 | 5000 | 4 | 64B6B PAM-16 128-DSQ | 400 | 200 | 100 | Cat 6 | 250 |
| 2.5GBASE-T | 802.3bz-2016 | 2500 | 4 | 64B6B PAM-16 128-DSQ | 200 | 100 | 100 | Cat 5e | 100 |

100Base-TX Kabel und Stecker

Charakterisieren Sie das für 100 Base-TX verwendete Kabel (inkl. Stecker). Worin können sich solche Kabel unterscheiden?

- UTP (Kabel, das ungeschirmte verdrillte Aderpaare enthält). Das universell verwendbare Kabel enthält 8 Adern, von denen bei 100 Base-TX nur 4 verwendet werden.

- Üblicher Stecker ist vom Typ RJ45.

- Unterschiede: Es gibt 1:1 und Crossover - Kabel. Die Übertragungseigenschaften können sehr verschieden sein und werden durch die Kabelkategorie charakterisiert.

100Base-TX Oszilloskop

Sie sehen auf einem Oszilloskop das folgende Bild eines 100Base-TX Signals:



Ein Kästchen entspricht einem Bit. Wie lauten die ersten 10 Bits?

Antwort: 1100010101

Ethernet Segment Rate Wie viele Frames pro Sekunde können auf einem 100 Base-TX Ethernet Segment pro Richtung maximal übertragen werden? Hinweis: Denke an Inter-Frame Gap, die mindestens so lange wie die Übertragung von 12 Bytes dauert.

Antwort: Die grösste Frame Rate wird erreicht, wenn alle Frames die minimale Länge von 64 Oktette aufweisen. Solche Frames belegen folgende Anzahl von Oktett Einheiten von 80 ns:

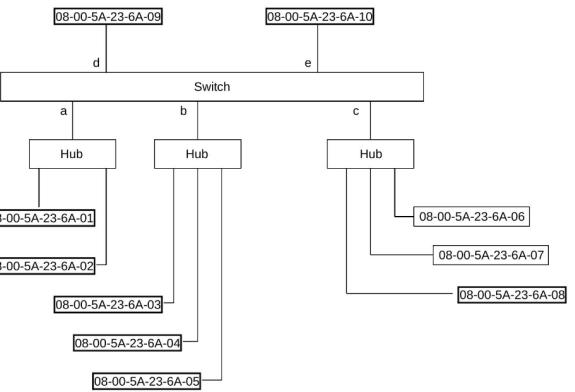
- 8 für Präambel und Start Frame Delimiter (SFD)
- 64 für das eigentliche Frame (mit bis zu 46 Oktette Nutzdaten)
- 12 für Interframe Gap (960 ns)

Framerate = $10^8 \text{ bit per second} / 8 * (8 + 64 + 12) \text{ bit per frame} = 148'800 \text{ frames per second}$

Transit Delay Wir betrachten ein sehr schwach belastetes Ethernet (d.h. praktisch keine Kollisionen und keine Wartezeiten in Switches). Worin besteht der wesentliche Unterschied, ob man nun ein 100Mb/s oder ein Gigabit-Ethernet verwendet? Wo ist dieser Unterschied wichtig?

Antwort: Transit Delay ist im 1000 Mb/s-Ethernet 10 mal geringer, was bei kaskadierten Store and Forward Switches erheblich sein kann. Die resultierende Round Trip Time kann die Performance stark beeinflussen. 1000Mb/s-Ethernets haben nicht nur einen höheren Durchsatz, sie sind auch reaktionsfreudiger!
Beispiel: 3 Switches und maximal langes Frame $\rightarrow 4 * 12.3\mu\text{s}$ versus $4 * 123\mu\text{s}$

Filtering Database Ein Netzwerk besteht aus einem Ethernet Switch, drei Ethernet Hubs und 10 Endgeräten. Anfangsbedingung: Der Ethernet Switch kennt zu Beginn keine einzige MAC-Adresse. Annahmen: Die Aging-Time betrage 300 Sekunden. Die Aging-Time für eine MAC-Adresse wird nur zurückgesetzt, wenn diese MAC-Adresse als Quelladresse erscheint.



Auf welchen Switchports (a ... e) sind die Meldungen sichtbar?

| Zeit t | Nr. | Quelladresse | Zieladresse | a | b | c | d | e |
|--------|-----|-------------------|-------------------|---|---|---|---|---|
| 0 | 1 | 08-00-5A-23-6A-01 | 08-00-5A-23-6A-10 | x | x | x | x | x |
| 10 s | 2 | 08-00-5A-23-6A-05 | 08-00-5A-23-6A-01 | x | x | | | |
| 20 s | 3 | 08-00-5A-23-6A-09 | 08-00-5A-23-6A-08 | x | x | x | | |
| 30 s | 4 | 08-00-5A-23-6A-07 | 08-00-5A-23-6A-04 | x | x | x | x | |
| 120 s | 5 | 08-00-5A-23-6A-09 | 08-00-5A-23-6A-05 | x | | | | |
| 250 s | 6 | 08-00-5A-23-6A-06 | 08-00-5A-23-6A-01 | x | | | | |
| 260 s | 7 | 08-00-5A-23-6A-03 | 08-00-5A-23-6A-09 | x | | | | |
| 320 s | 8 | 08-00-5A-23-6A-03 | 08-00-5A-23-6A-07 | x | x | | | |
| 350 s | 9 | 08-00-5A-23-6A-09 | 08-00-5A-23-6A-07 | x | x | x | x | x |
| 400 s | 10 | 08-00-5A-23-6A-02 | 08-00-5A-23-6A-06 | x | x | x | x | x |

Interpretation eines Hex-Dumps (MAC-Frame) Hex-Dump eines MAC-Frames, wie er mit Wireshark aufgezeichnet worden ist:

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

a) Markieren und benennen Sie die einzelnen Felder.

Oktett 0-5: Destination MAC-Address (08-00-2B-C3-AC-A5)

Oktett 6-11: Source MAC-Address (00-00-F8-1A-84-1A)

Oktett 12/13: Length / Type → hier Type = 0×0800

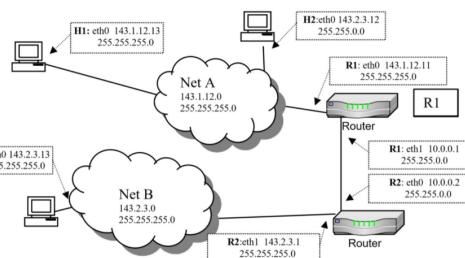
Oktett 14-59: Data / padding

Oktett 60-63: Frame Check Sequence

b) Was lässt sich über den Inhalt des Datenfeldes aussagen? **Antwort:** ein IP-Paket (wegen Type = 0x0800)

Network Layer

Routingtabelle Gegeben ist folgendes Netzwerk:



Routingtabelle von Host H1:

| Netzadresse | Netzmaske | Interface | Gateway |
|-------------|---------------|-----------|-------------|
| 143.1.12.0 | 255.255.255.0 | eth0 | (direkt) |
| default | | eth0 | 143.1.12.11 |

a) Host H1 pingt Host H2 mit ping 143.2.3.12, bekommt jedoch keine Antwort, obwohl alle Router richtig konfiguriert sind. Warum nicht? H2 hat falsche Subnetzmaske und hat eine Adresse vom Netz B. Daraus folgt: H1 sendet das Paket zum Router anstatt ins angrenzende Netz. Das Paket wird zu Router R2 weitergeleitet, der in Netz B keinen entsprechenden Host findet.

b) Routing-Tabelle von Router R1:

| Netzadresse | Netzmaske | Interface | Gateway |
|-------------|---------------|-----------|----------|
| 143.1.12.0 | 255.255.255.0 | eth0 | (direkt) |
| 10.0.0.0 | 255.255.0.0 | eth1 | (direkt) |
| 143.2.3.0 | 255.255.255.0 | eth1 | 10.0.0.2 |

c) Warum kann der Host H3 den Host H2 nicht erreichen? In welchem Subnetz sucht H3 den Zielhost? Welchen Weg beschreitet das „Ping-Paket“ (Gehen Sie davon aus, dass die Routingtabellen in allen Komponenten vollständig gemäß den Einträgen aus obigem Netzdiagramm generiert wurden).

H3 sucht H2 im Netz B, d.h. der Ping wird nicht über den Router gesendet. H2 erhält somit die Anfrage nicht.

IP Subnetz Aufteilung

Sie bekommen von Ihrem Internet Service Provider (ISP) ein privates Klasse-C Netz zugewiesen. In Ihrem Haus befinden sich 4 Parteien, welche sich den Internet-Anschluss teilen. Sie geben jeder Partei ein gleich großes Subnetz, indem sie das Klasse-C Netz 192.168.1.0/24 in 4 Subnetze aufteilen. Geben Sie für alle 4 Subnetze die Netzadresse, die Netzmaske, die Broadcast-Adresse, den Default Gateway sowie die Anzahl adressierbarer Hosts an.

| Subnetz 1 | Subnetz 3 |
|-------------------|---------------|
| Netzadresse | 192.168.1.0 |
| Broadcast-Adresse | 192.168.1.63 |
| Subnetz 2 | Subnetz 4 |
| Netzadresse | 192.168.1.64 |
| Broadcast-Adresse | 192.168.1.192 |
| Netzadresse | 192.168.1.127 |
| Broadcast-Adresse | 192.168.1.255 |

Für alle Subnetze gilt:

Netzmaske 255.255.255.192 /26 → letztes Byte: 11000000

Anzahl adressierbarer Hosts: $2^6 - 2 = 64 - 2 = 62$

Adressauflösung/Kapselung (ARP)

Ein Rechner möchte eine Anfrage an Google (IP 8.8.8.8) senden. Der Rechner ist neu am Netz und alle ARP-Caches sind noch leer. Nachdem der Rechner die Route zum gewünschten Ziel in der Routing-Tabelle gefunden hat, sendet er seinen Request. Schreiben Sie in der untenstehenden Tabelle die ersten drei Request auf.

MAC: 08:00:B4:3D:64:22
eth0: 160.85.24.1

MAC: 08:00:B4:3D:64:23
eth1: 160.85.36.1

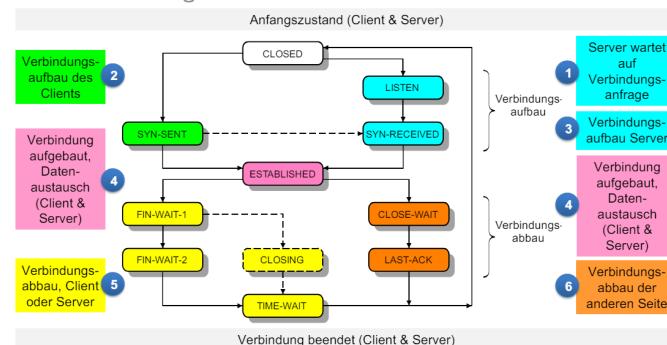


Geben Sie in der Tabelle jeweils die letzte HEX-Zahl der MAC-Adresse oder die letzten zwei Dezimalzahlen der IP-Adresse an.

| Request Typ | MACsrc | MACdest | IPsrc | IPdest | What |
|-------------|--------|---------|-------|---------|-----------------|
| ARP Request | 11 | FF | - | - | Who has IP 24.1 |
| ARP Reply | 22 | 11 | - | - | 24.1 has 22 |
| IP Paket | 11 | 22 | 24.5 | 8.8.8.8 | - |

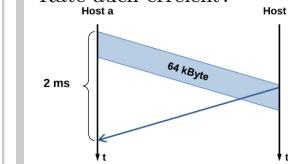
Transport Layer

TCP Zustandsdiagramm



TCP Sliding Window Zwei Hosts sind mit einem Duplex-Übertragungskanal von 1 GBit/s verbunden. Welche Übertragungsrate kann man mit einer TCP-Verbindung maximal erreichen, falls die Window Size auf 64 kByte begrenzt ist und die Round Trip Time 2 ms beträgt.

Der Overhead der Protokoll-Header kann bei dieser Betrachtung vernachlässigt werden. Unter welchen Bedingungen wird diese maximale Rate auch erreicht?

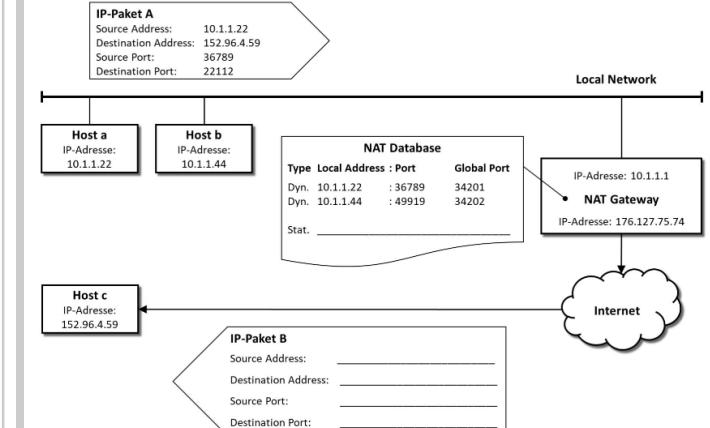


Antwort: Übertragungsrate = 64 kByte alle 2 ms = $64 \text{ kByte} * 8 \text{ Bit[Byte}/2 \text{ ms} = 256 \text{ Mbit/s}$ Damit ist der Kanal nur zu ca. 1/4 der Zeit genutzt.

Diese theoretische Obergrenze wird nur dann erreicht, falls - im Netz keine Paketverluste auftreten (würde als Congestion interpretiert und die Datenrate bremsen) - Sender und Empfänger über genügend Leistung verfügen

Application Layer

NAT und Port Mapping Gegeben sei die Konfiguration gemäß untenstehendem Bild: Ein lokales IP-Netz 10.0.0.0/8 mit Host a und Host b, das über einen NAT Gateway und das Internet mit dem Host c verbunden sei:



Host a will das obige IP-Paket A an den Host c im Internet senden. Der NAT Gateway empfängt das IP-Paket B an den Host c weiter.

a) Geben Sie oben die Source- und Destination-Adressen sowie die Source- und Destination-Ports des IP-Pakets B an.

IP Paket B:

Source Address: _____

Destination Address: _____

Source Port: _____

Destination Port: _____

b) Der Host b soll dem Host c als TFTP Server dienen. Wie muss dazu der statische Eintrag in der NAT Database lauten? Hinweis: TFTP nutzt das UDP Port 69.

| Type | Local Address: | Port | Global Port |
|--------|----------------|------|-------------|
| Static | 10.1.1.44:69 | | 69 |