

Übertragungsmedien

Signale

Ausbreitungsgeschwindigkeit

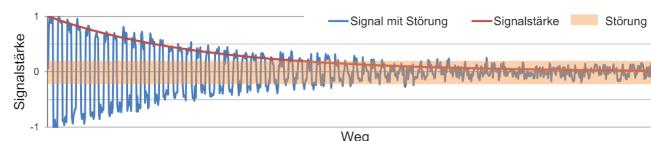
Lichtgeschwindigkeit im Vakuum: $c_0 = 299'792'458 \text{ m/s}$

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

Signaldämpfung

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

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



Halbierung der Leistung entspricht ca. 3dB

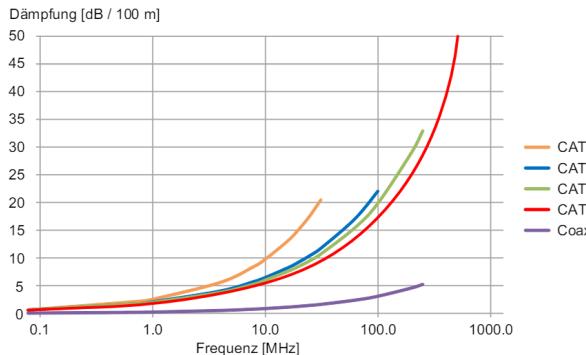
SNR

Signal to Noise Ratio

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

Dämpfungsbelag

Dämpfung pro Distanz - dB pro 100m



Leistungslänge Bandbreite und Dämpfungsbelag

- maximale Leistungslänge L_{max} :

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

SNR_{min} : Minimales benötigtes SNR für korrekte Datenübertragung

- tiefere Bitrate → grösse 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

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

Verbindungslos

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

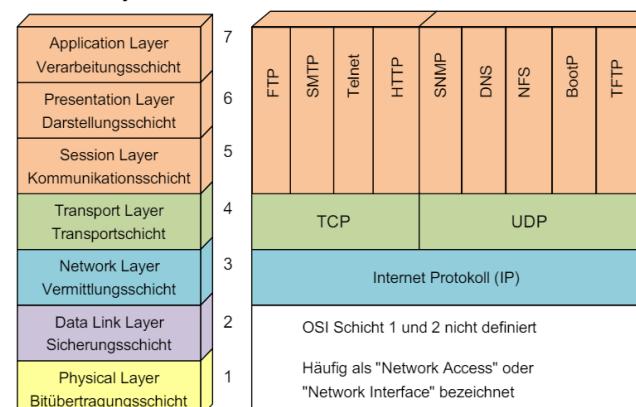
Zuverlässig

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

Unzuverlässig

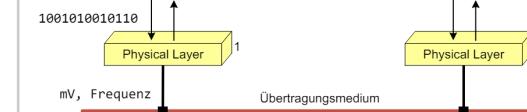
- Möglicher Datenverlust
- Keine Sicherung
- Streaming

OSI Layers



Physical Layer

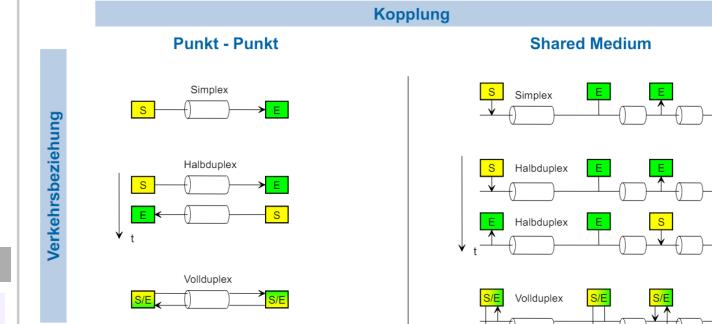
Schicht 1: Bitübertragungsschicht



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, abwechselnde 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

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

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

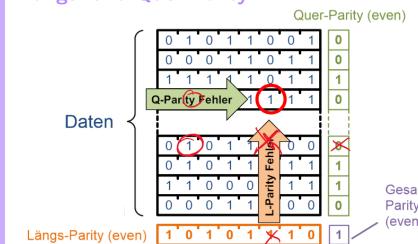
Einfache Parity

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

Zeitsteuerung

Zeitgesteuerte Zugriff (wie Taktfahrplan im Bahnhof)

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

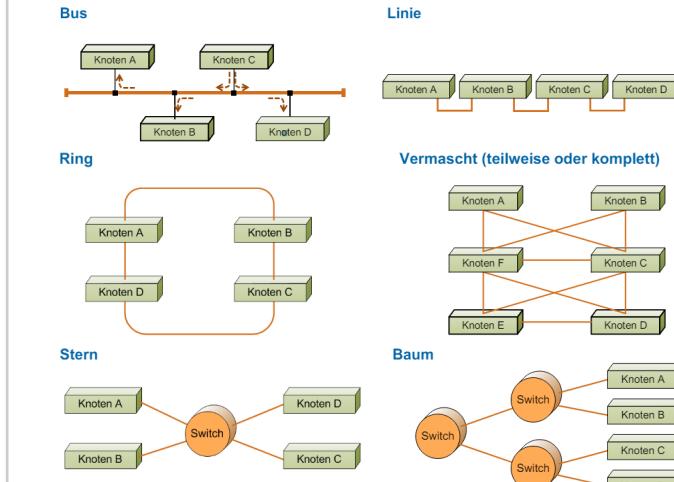
Flow-Control

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

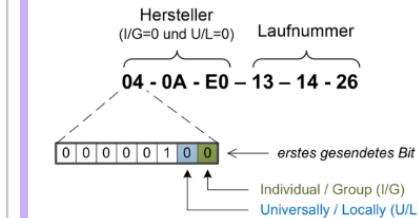


- Unicast: 1 E
- Multicast: n E (Gruppe)
- Broadcast: alle Knoten im LAN

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 ...

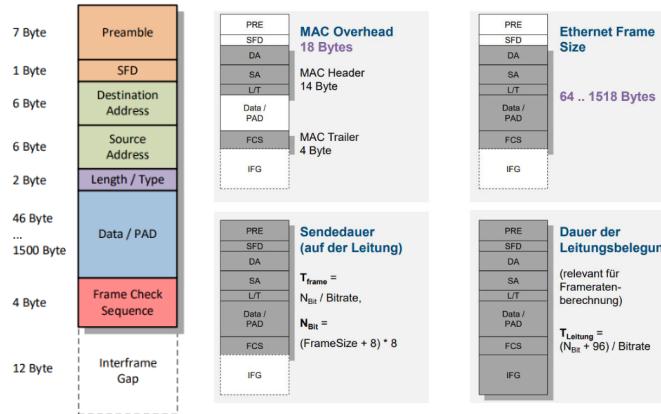
MAC-Adresse und Hersteller des Empfängers:

- 7 Bytes Präambel (10101010), 1 Byte SFD (10101011)
- 6 Bytes Destination Address: 00001000 (=08) 00000000 (=00)
 01011010 (=5A) 11000111 (=C7) 11110011 (=F9) 01100001 (=61)
- ⇒ MAC-Adresse: 08-00-5A-C7-F9-61, Hersteller (08-00-5A) IBM

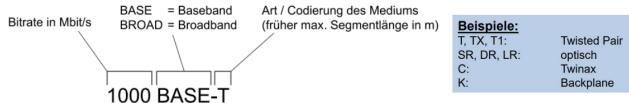
Pro Byte zuerst LSB, dann MSB (Ausnahme Zahlenwerte, z.B. Length/Type-Feld)

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	CAT5 - 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 Adreppaare simplex	MLT-3, 4B5B 4 Adreppaare simplex	PAM-5, 8B/10B 4 Adreppaare duplex	PAM-16, 64B/48B, FEC 4 Adreppaare 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:

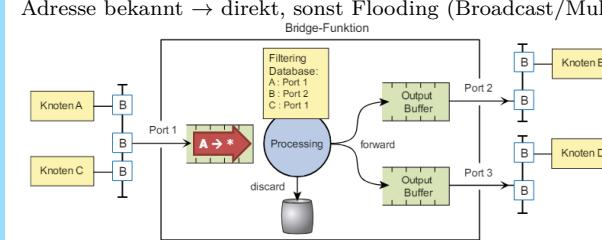
- Prüft Checksumme und kann Layer-2 Adressen auswerten
- Transparent: sollen für Endgeräte unsichtbar sein
- Verwendet Filtering Database (Adress-Learning)

Merkmale von Switches und Bridges

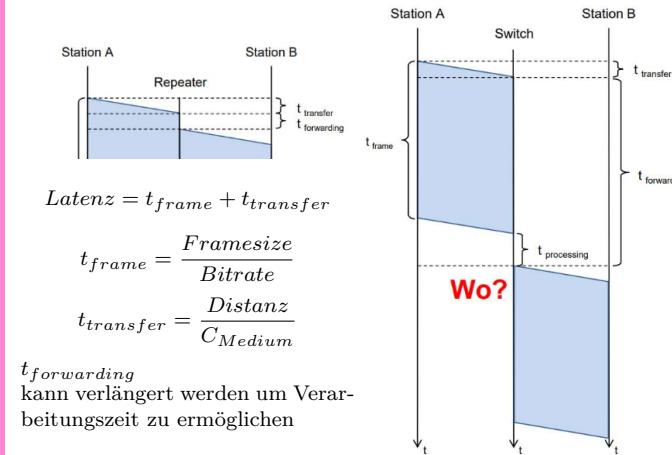
Anzahl Ports	Steckergöße ist im Extremfall die Limitierung
Adressabelle	Wie viele Stationen können im LAN existieren
Filtertate	Maximale Frames/s/Port (Empfangsrichtung)
Transferrate	Maximale Frames/s/Port (Senderichtung)
Backplane / Fabric Kapazität	Maximaler Gesamtdurchsatz zwischen allen Ports
Architektur	Store-and-Forward: Frame wird komplett empfangen und dann weitergeleitet Cut-Through: Frame wird schon nach Decodierung der Zieladresse weitergeleitet 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

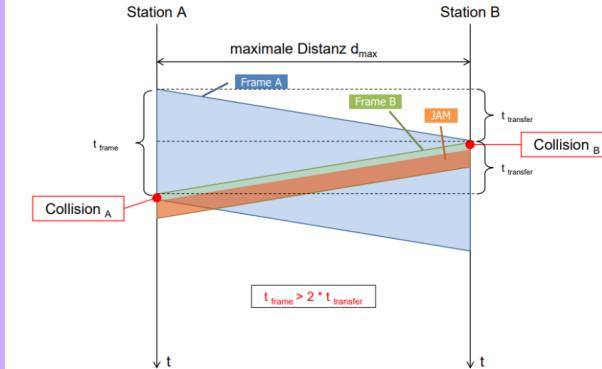
Adresse bekannt → direkt, sonst Flooding (Broadcast/Multicast)



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: $t_{frame} > 2 \cdot (\sum t_{transfer} + \sum t_{forwarding})$

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

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

Redundanz (Spanning Tree)

Spanning Tree Algorithmus

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

Initialisierung

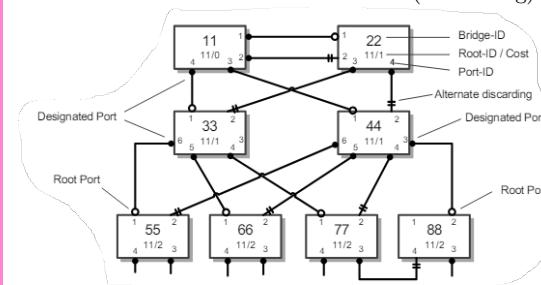
- Alle Ports für Nutzdaten blockiert
- Annahme: «Ich bin Root»
- Austausch BPDUs mit Nachbarn (Root ID, Root Cost, Bridge ID, Port ID)

Aufbau des Spanning Tree

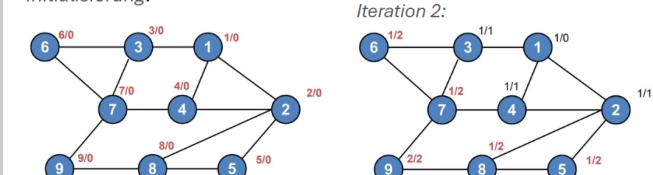
- «kleinster» Nachbar als Root gesetzt → Anzahl Hops + 1 (Beachte Prioritätswert)
- wiederholen bis alle dieselbe Root ID haben

Setzen der Port Roles

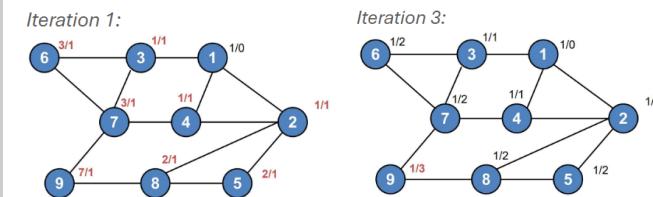
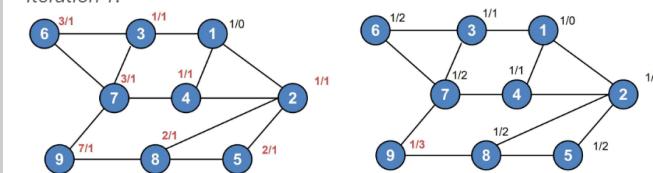
- Root-Ports (Empfang der «besten» BPDU)
- Designated-Ports (Gegenstück zu Root-Ports)
- Weg zum «kleinsten» Nachbar wird bevorzugt (ID, Anzahl Hops)
- Alle anderen Ports bleiben blockiert (Discarding)



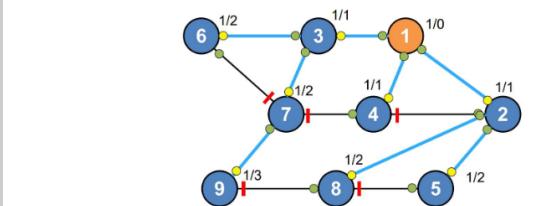
Initialisierung:



Iteration 1:

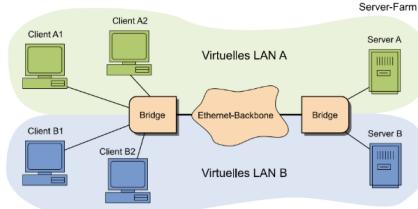


Final



Virtuelle LANs

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

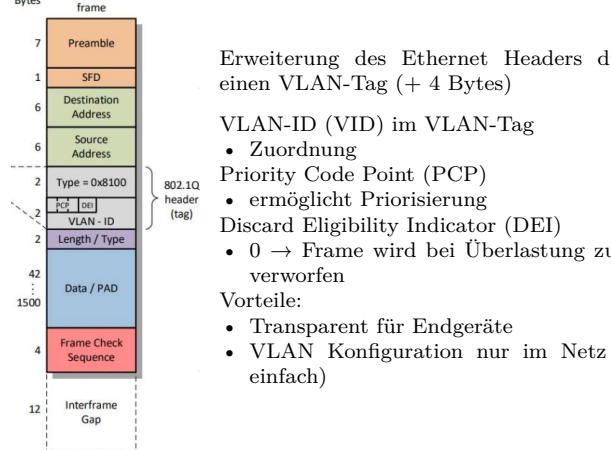


Trunk Links:
Teil von mehreren VLANs →
Frames eindeutig kennzeichnen!

Trunk = Tagged
Access = Untagged

VLAN Tagging

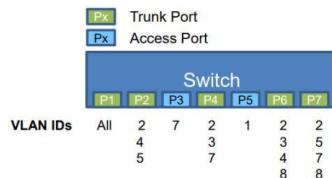
Bytes



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:

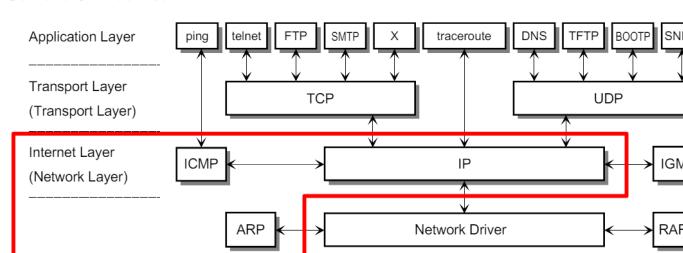


Welche Frames werden an welchen Ports gesendet und sind diese getagged oder ungetagged?

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

NUR Transport der IP-Pakete → höhere Layer übernehmen:

- Fehlerfreie, komplette Übertragung
- Richtige Reihenfolge, Flusskontrolle

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

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 (x 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

1. Byte	2. Byte	3. Byte	4. Byte
0	1	2	3 4 5 6 7 8
Version	IHL	Type of Service	Total Length
9	10 11 12 13 14 15	16 17 18 19 20 21	22 23 24 25 26 27 28 29 30 31
Identification Number	Flags	Fragment Offset	
Time to Live	Protocol	IP Header Checksum	
IP Source Address			
IP Destination Address			
Options / Padding			

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

Fragmentierung

- Länge der Nutzdaten = Vielfaches von 8 Bytes
- Die Pakete haben die gleiche und grösstmögliche Länge
- Identification Number, Flags und Fragment Offset (siehe gelbe Felder in Grafik oben) wichtig für Fragmentierung
- früher von Router durchgeführt, heute im Sender

Reassembly

nutzt Flags (0/DF/MF) und Fragment Offset

- Zusammensetzen beim Zielhost
- Letztes Fragment: MF = 0

Feld	Position	Werte	Funktion
0		0	Reserved, must be Zero
DF	1	0 / 1	May / Don't Fragment
MF	2	0 / 1	Last / More Fragments

Kombination mit DF und MF erlaubt vollständige Rekonstruktion ohne explizite Übertragung der ursprünglichen Paketgrösse

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

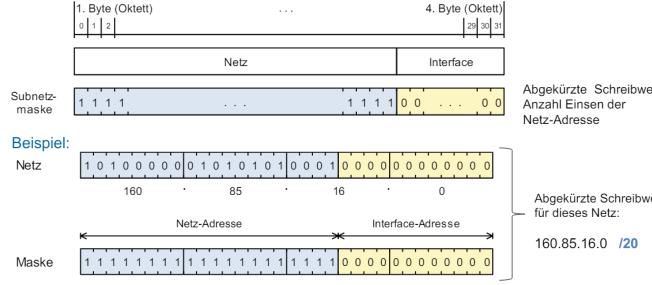
- Reserviert: Darf nicht für Interfaces verwendet werden!
- Tiefste Adresse im Subnetz (Interface-Adressbits alle 0)
- Berechnet durch: Interface-Adresse AND Subnetzmaske

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 die Grenze zwischen Netz- und Interface-Adressbits:



Netzmasken

Wert (dezimal/binär) alternative Schreibweise: Anzahl adressierbare Interfaces:

255 (1111'1111)	/24	256 - 2
254 (1111'1110)	/23	512 - 2
252 (1111'1100)	/22	1'024 - 2
248 (1111'1000)	/21	2'048 - 2
240 (1111'0000)	/20	4'096 - 2
224 (1110'0000)	/19	8'192 - 2
192 (1100'0000)	/18	16'384 - 2
128 (1000'0000)	/17	32'768 - 2
0 (0000'0000)	/16	65'536 - 2

Rechnen mit Netzmasken

Typische Internet-Adressen Aufgabenstellung: Berechnen Sie die fehlenden Informationen

	IP-Adresse	Subnetzmakske	Netzadresse	Broadcastadresse	Anzahl Adressen inkl. Netz- und Broadcastadresse
a	17.8.7.8	255.255.0.0 /16	17.8.0.0	17.8.255.255	65'536
b	11.7.177.4	255.255.224.0 /19	11.7.160.0	11.7.191.255	8'192
c	144.3.133.1	255.255.192.0 /24	144.3.128.0	144.3.191.255	16'384
d	31.4.2.166	255.255.255.248 /29	31.4.2.160	31.4.2.167	8

Flaches und Hierarchisches Routing

Flaches Routing

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

Flaches Routing Übung

Was geschieht mit dem IP-Paket?

- Kein Unterbruch?

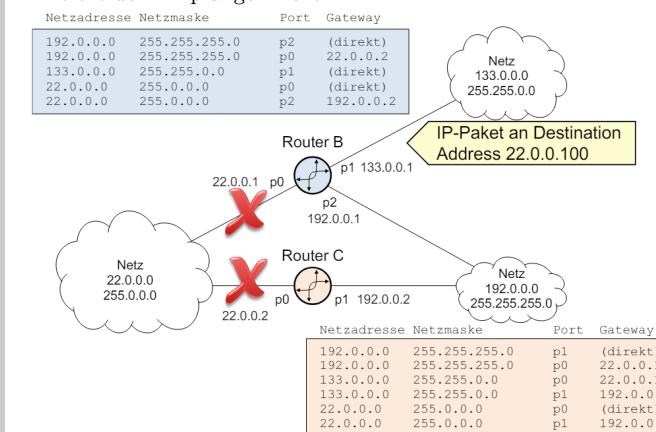
Es wird nach gemäss dem 4. Eintrag der Routingtabelle von Router B an p0 weitergeleitet

- Unterbruch von p0 / Router B ?

Es wird gemäss Eintrag 5 in der Routingtabelle von Router B an p2 weitergeleitet.

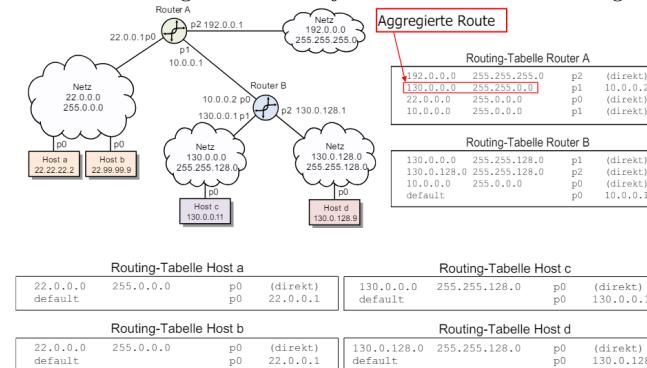
- zusätzlicher Unterbruch p0 / Router C ?

Router C kann das IP-Paket nicht weiterleiten, es IP-Paket 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, access Router
- Kleine Routing-Tabellen mit jeweils einem Default-Eintrag

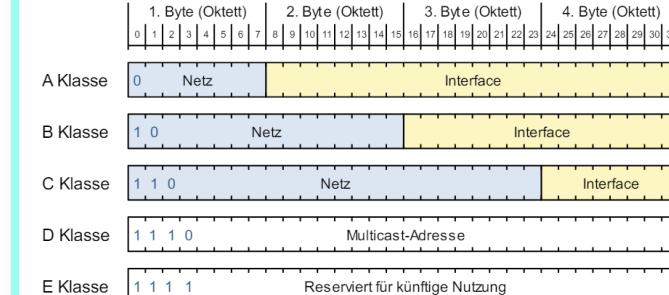


Classful Routing: Sub-/Supernetting

Classful Routing

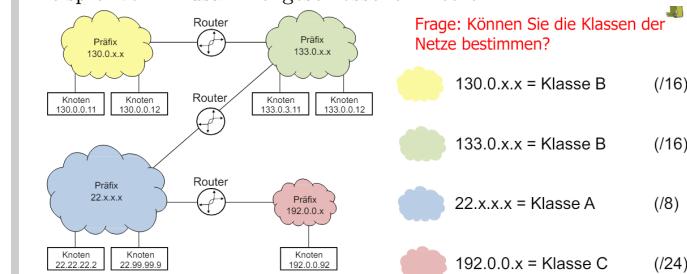
Ursprünglich war der IP Adressbereich in fünf Netzklassen (A - E) eingeteilt

- Eine Prefix (die ersten 4 Adress-Bits) erlaubt die Bestimmung der Klasse



Classful Routing

Beispiel von 4 zusammengeschlossenen Netzen:



Internet-Adressierung (IPv4 Netz-Klassen)

Klasse	Adressbereich	Anzahl Netze	Interfaces pro Netz
A	1.0.0.0 – 127.255.255.255	127	16777214
B	128.0.0.0 – 191.255.255.255	16'384	65534
C	192.0.0.0 – 223.255.255.255	2'097152	254
D	224.0.0.0 – 239.255.255.555	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

- Die klassischen Netze fixer Grösse sind unflexibel
 - Klasse C Netze sind für Unternehmen zu klein
 - Klasse A Netze sind zu gross
 - Klasse B Netze sind zu wenig
- Abhilfe schafft CIDR – Classless Inter-Domain Routing
 - Flexible Verwendung von Netzmasken beliebiger Länge
 - Aufteilung grosser Netze in kleinere Subnetze, Zusammenfassen mehrerer kleiner Netze zu einem gemeinsamen grösseren Netz

localhost

Loopback-Adressen

- Das gesamte A-Netz 127.0.0.0/8 ist für Loopback-Test reserviert
- Daten werden an ein emuliertes Loopback-Gerät geschickt, das sie direkt zurück gibt (kein Netzwerk-/Interface nötig).

Sub- und Supernetting

Supernetting

Zusammenfügen von kleinen Netzen
Hintereinanderliegende Class C Netze können zu einem Netz zusammengefügt werden.

Kann ebenfalls helfen, Routingtabellen in Routern zu verkleinern (Aggregate Routes)

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

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

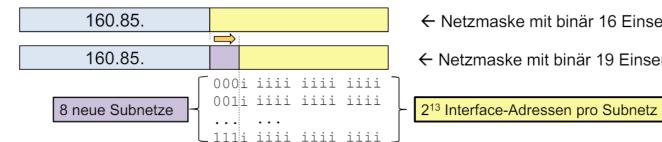
198.51.0110 01 00.0000 0000 = Subnetzmaske 255.255.252.0 oder /22

198.51.0110 01 00.0000 0000 ← Netz-Adresse 198.51.100.0, Netz 198.51.100.0 /22
192.51.0110 01 11.1111 1111 ← Broadcast-Adresse

Subnetting

Aufteilung in kleinere Netze
Die ZHAW besitzt das Klasse B Netz 160.85.0.0

- Total $2^{16} \cong 65000$ Hosts
- Die ZHAW möchte dieses in 8 kleinere Subnetze aufteilen → Subnetting
- Verschieben der Netzmasks-Bits: $8 = 2^3$, es werden 3 1en in der binären Netzmase ergänzt
- 3 Bits identifizieren 8 Subnetze (000 → 111)
- Die Netzmase verändert sich von /16 zu /19 (255.255.0.0 → 255.255.224.0)
- Der Interface-Anteil verändert sich von 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

- Der Netz-Anteil der binären Netzmase hat nun 19 statt 16 "1" → Subnetzmaske: 255.255.224.0 oder /19
- Der Host-Anteil der binären Nutzmase hat nun 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

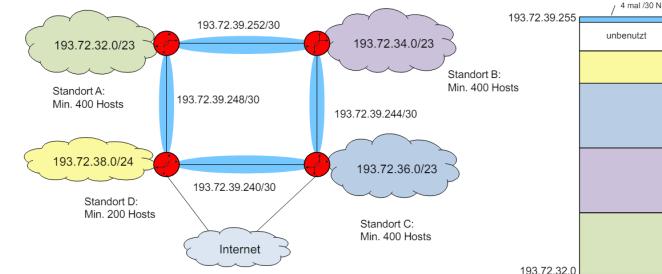
Das fünfte Netz wird wie folgt gekennzeichnet:

- 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

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



IPv6

IPv6

- IPv6 ist in RFC 2460 spezifiziert
- 128-bit Adressen; diese werden mit je zwei Bytes in Hex-Darstellung notiert und durch Doppelpunkte getrennt
- IPv6 verwendet Extension Headers, um den Basic Header zu vereinfachen
- Ein Interface kann mehr als eine IPv6 Adresse haben
 - Ein Interface hat in der Regel eine lokale und zwei globale IPv6 Adressen:
 - Eine MAC-basierte und eine nicht von der Hardware abhängige.
- verwendet zur Abfrage der Layer-2 Adressen NDP statt ARP
- Domain Name Service (DNS)
 - IPv4 stellt an den Resolver Anfragen nach A-Records
 - IPv6 stellt an den Resolver Anfragen nach AAAA-Records
- hat sich nicht durchgesetzt weil:
 - nicht so einfach lesbar wie IPv4
 - Viele Probleme von IPv4 konnten gelöst werden
 - IPv6 ist nicht rückwärtskompatibel

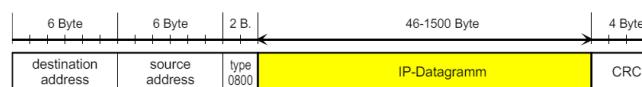
Key Takes

- Der IP-Header besteht aus 20 Bytes (ohne Optionen)
- Um über Netze mit verschiedenen Maximum Transfer Units (MTU) arbeiten zu können, unterstützt IP Fragmentierung und Reassembly
 - Heute wird in der Regel beim Sender fragmentiert und im Ziel-Host reassembliert
 - Path MTU discovery mittels ICMP kann verwendet werden, um die kleinste MTU auf dem Weg zum Ziel-Host zu identifizieren
- IP-Pakete werden in Ethernet Frames gekapselt und von jedem Router wieder ausgepackt und erneut gekapselt.
 - Dazu muss der Router die Layer-2 Adresse (MAC-Adresse) des nächsten Routers/Hosts kennen (ARP-Cache) oder erfragen (ARP-Request)
- ICMP wird verwendet, um Fehler innerhalb der Netzwerkschicht zu behandeln (keine Retransmissions)
 - ICMP-Nachrichten werden in IP-Pakete gekapselt, werden aber dennoch der Netzwerkschicht zugeordnet

Kapselung und Adressauflösung

Kapselung eines IP-Pakets im Ethernet Frame

- Meist wird heute Ethernet-Encapsulation verwendet
- Das IP-Paket wird direkt im Nutzdatenteil des Frames übertragen
- Das Type Feld des Ethernet Frames erhält den Wert 0800 (hex)
- Die MTU ist damit 1500 Bytes



Kapselung und Adressauflösung

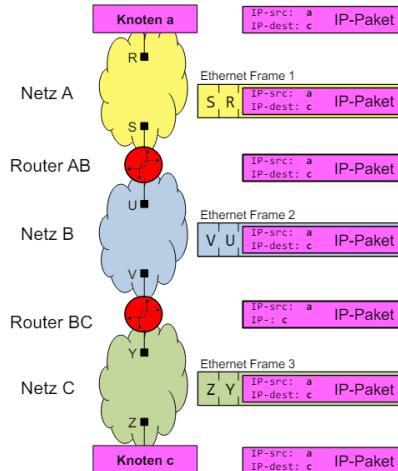
ARP (Address Resolution Protocol)

- Ermittelt HW-Adresse (MAC) zu einer IP-Adresse

Internet Control Message Protocol (ICMP)

- Übertragungen von Fehlermeldungen oder Informationsaustausch

Übertragung eines IP-Pakets mit Encapsulation



Was geschieht bei der Übertragung genau?

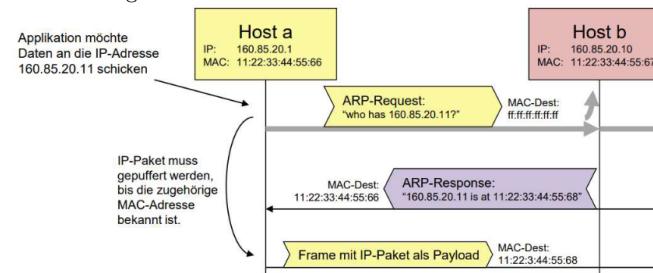
- Knoten a sendet ein IP-Paket an Knoten c
 - das Paket enthält die IP-Adressen von a und c
- Knoten a konsultiert die Routing Tabelle und sieht:
 - dass c über den Router AB erreicht werden kann, und
 - Kennt nun die IP-Adresse von Router AB
- Knoten a generiert ein Ethernet Frame, welches an die Hardwareadresse S von Router AB gesendet wird
 - a muss aus der IP-Adresse von Router AB die Hardware-Adresse S herausfinden
 - Adressauflösung**
- Router AB empfängt das Ethernet Frame, packt das IP-Paket aus und modifiziert den Header (TTL)
- Router AB konsultiert die Routing Tabelle und sieht:
 - dass c über den Router BC erreicht werden kann, und
 - Kennt nun die IP-Adresse von Router BC

Die IP-Adressen a und c bleiben während der gesamten Übertragung unverändert

Address Resolution Protocol (ARP)

ARP Grundprinzip

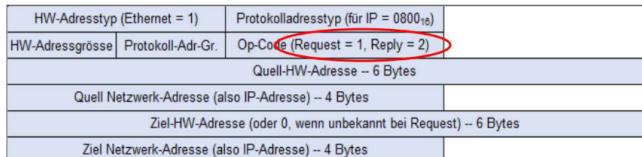
- Ermittlung der Hardwareadresse (MAC) zu einer IP-Adresse
- ARP-Request wird an Broadcast-Adresse gesendet
- ARP-Response wird von Knoten mit angefragter IP-Adresse an Absender gesendet
- Die ARP-Tabelle speichert bekannte <IP-MAC> Kombinationen für eine gewisse Zeit



ARP Nachrichtenstruktur

ARP-Request und ARP-Response sind je in genau einem Ethernet Frame enthalten mit Type x0806

- Beim Request ist die Destination Address FF-FF-FF-FF-FF-FF (Broadcast Frame) und die Hardware Address of Target ist 0



ARP Implementierung und Verwendung

- Ein ARP-Request/Response für jedes IP-Paket wäre sehr ineffizient
 - Jeder Knoten führt eine Tabelle (ARP-Cache) mit bekannten HW-Adressen
- Aufgelöste (bekannte) Mappings IP Adresse → Hardwareadresse werden im ARP-Cache für gespeichert
 - Erneuerung nach Ablauf eines Timers, typisch: einige Minuten
- Abfrage/Modifizieren des ARP-Cache mit arp (Windows):
 - arp -a: Anzeigen aller Einträge
 - arp -d ip_addr: Löschen eines Eintrags
 - arp -s ip_addr hw_addr: Setzen eines Eintrags
- Neue / empfohlene Befehle für Linux:
 - ip neigh { add | del | show}

Weitere Verwendung:

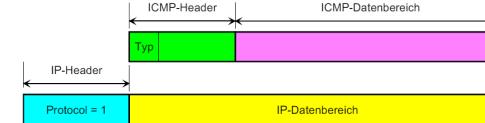
- Erkennung von Adresskonflikten
 - Nach einer Adresszuweisung (manuell oder per DHCP) wird ein ARP-Request an die eigene IP-Adresse gerichtet, um zu prüfen, ob kein anderer Host im LAN die Adresse verwendet
 - Falls eine Antwort kommt, liegt ein Adresskonflikt vor
- Erneuerung von Einträgen im ARP-cache
 - Linux Systeme senden in diesem Fall einen ARP-Request als Unicast
 - Reduziert Broadcast-Last im Netz

Internet Control Message Protocol (ICMP)

Internet Control Message Protocol (ICMP)

Übertragung von Fehlermeldungen oder Informationsaustausch auf Internet Layer, z.B.

- Time to live (TTL) hat den Wert 0 erreicht
- Ein Host möchte testen, ob ein anderer Host „up“ ist ICMP Meldungen werden in IP Paketen gekapselt
- Sieht aus wie ein Protokoll eines höheren Layers, welches den Internet Layer verwendet
- ICMP ist aber so eng mit IP verbunden, dass es zum Network Layer gezählt wird

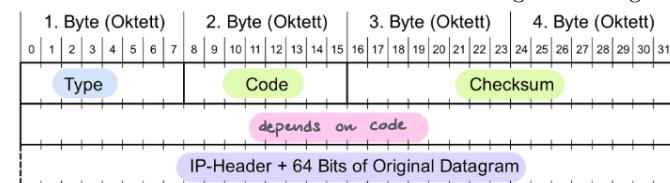


ICMP Format

Header:

- Type ICMP Typ
- Code Message Details
- Checksum Prüfsumme über die ICMP Meldung
- depends on code Unterschiedliche Werte und Verwendung je nach ICMP Typ

Datenbereich IP-Header und 64 Bits of Original Datagram



ICMP Meldungstypen

- ICMP benutzt direkt IP - keine Garantie, dass die Meldungen je ankommen
- Meldungen sind NUR informativ gedacht

ICMP-Typ	Bedeutung (Fehler)
3	Destination Unreachable
5	Redirect
11	Time Exceeded
12	Parameter Problem: Bad IP Header

ICMP-Typ	Bedeutung (Information)
0	Echo Reply
8	Echo
13	Timestamp
14	Timestamp Reply

Codes:

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

ICMP Meldungstypen - Details

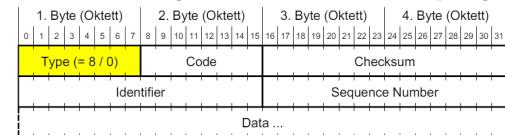
- Destination Unreachable (Fehler)
 - IP-Paket kann nicht zum Ziel gebracht werden
 - Beispiel: Keine Route zum Ziel-Host vorhanden
- Redirect (Optimierung)
 - Ein Host H sendet ein IP-Paket an einen ersten Router R1
 - R1 stellt fest, dass der nächste Router auf dem Weg zum Ziel R2 ist; R2 ist aber im gleichen Netz wie H und R1 (möglicherweise unvollständige Routingtabelle im Host H)
 - R1 sendet an H eine Redirect-Meldung, damit H Pakete fortan direkt an R2 sendet
- Time Exceeded (Fehler)
 - Router ändert das TTL-Feld im IP-Header von 1 auf 0
 - Host hat nicht alle Fragmente erhalten, bevor der Timer abläuft
- Parameter Problem: Bad IP Header (Fehler)
 - IP Packet Header enthält ungültigen Wert, der nicht verarbeitet werden kann (z.B. nicht existierende IP-Option)
- Echo Request/Reply (Information)
 - Host sendet Echo-Request, der adressierte Host antwortet mit Echo-Reply; Reply enthält die gleichen Daten wie Request
- Timestamp Request/Reply (Information)
 - Wie Echo, aber zusätzlich wird die aktuelle Zeit der Hosts ausgetauscht (32-Bit Wert, Millisekunden seit Mitternacht GMT)

Echo Request/Reply Messages

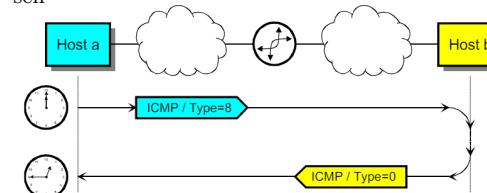
- Test, ob Host erreichbar ist
- Host antwortet auf Echo Request (Type 8) mit Echo Reply (Type 0), mit gleichem Inhalt wie der Echo Request

Format

- Identifier: Erlaubt Zuordnung von Reply zu Echo-Request
- Sequence Number: Wird innerhalb eines Identifiers jeweils um 1 erhöht
- Data: Beliebige Daten, werden vom Empfänger gespiegelt



ping verwendet Echo und Echo Reply, um die Erreichbarkeit eines Routers/Hosts zu prüfen; ebenfalls wird die Round-Trip Zeit gemessen



ICMP Destination Unreachable

Vom Router/Zielhost an Absender gesendet, 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

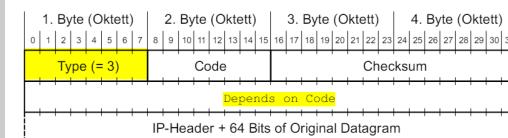
Path MTU Discovery:

Ziel

- Erkennung der kleinsten MTU auf Pfad zwischen Sender und Empfänger (Path-MTU, PMTU)
- RFC 1191 → Path MTU discovery

Zweck

- Vermeidung von Fragmentierung «unterwegs»



Welche Codes werden von einem Router (0,1,4) und welche vom Zielhost (2,3) generiert? Welche vermutlich von einer Firewall (13)?

Path MTU discovery

Annahme, dass die PMTU gleich der lokalen MTU ist

- Senden von IP-Paketen mit Länge=PMTU und mit DF=1
- Empfang von «Destination Unreachable» mit Code 4 «fragmentation needed and DF set»
- PMTU reduzieren auf «Next-Hop MTU»

Die «Next-Hop MTU» erkennt man: Enthalten in Octet 5.8 («must be zero» stimmt nur, wenn wirklich «unused»)

ICMP Destination Unreachable

Host 160.85.31.3 versucht, das folgende Paket an Host 160.85.29.99 zu senden (Farben siehe IP-Header def.):

- 4500 0028 8b10 0000 0711 a8a4 a055 1f03 a055 1d63 8b0d 829d 0014 a348 030a 0000 7504 1137 407c 0800
- Erkennen Sie in diesem Paket die IP Adressen von Sender und Destination?

– Sender : a055 1f03, Destination : a055 1d63

Ein Router kennt keinen Weg und sendet diese Destination Unreachable Message zurück (Farben siehe ICMP-Header def.):

- 4500 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
- Wie erkennen Sie, dass es sich um eine ICMP Message handelt?
Protocol: 01
- Wie erkennen Sie den ICMP Typ? **Type: 03**
- Erkennen Sie die "64 Bytes of Original Datagram"? **Original Data**

ICMP Time Exceeded

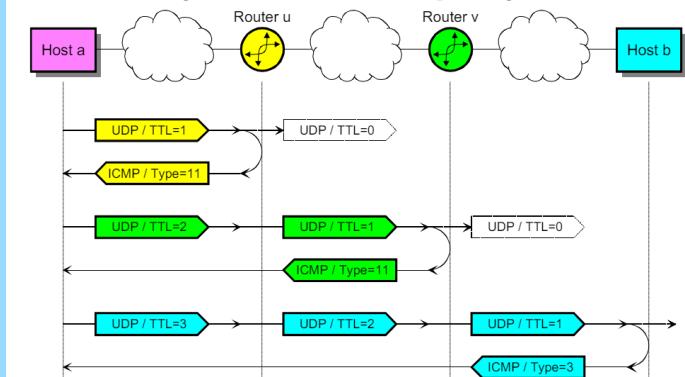
- Type = 11
- unused (must be 0)

Wird in diesen 2 Fällen gesendet:

- Router setzt TTL-Feld von 1 auf 0
 - Paket wird verworfen und der Absender informiert (Code = 0)
- Zielhost kann ein fragmentiertes Paket nicht innerhalb nützlicher Zeit reassemblieren
 - Fragmente werden verworfen und der Absender informiert (Code = 1)

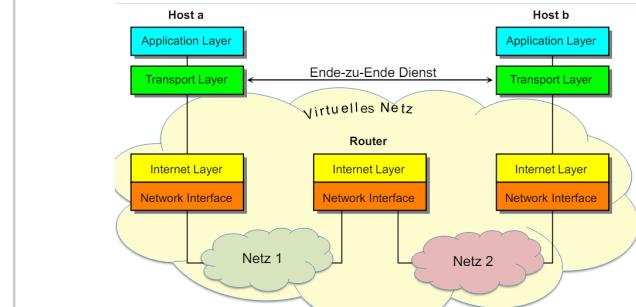
traceroute erlaubt, den Weg zu einem beliebigen Host (oder einem fehlerhaft konfigurierten Router auf diesem Weg) zu finden

- UDP Datagramme an hohe Destination Portnummer (zufällig gewählt, default 33434)
- Erstes Datagramm: TTL := 1
 - Erster Router setzt TTL auf 0, verwirft Paket und sendet Time Exceeded Message zurück
 - Erste Router ist bekannt
- Nächstes Datagramm: TTL := 2
 - Zweiter Router ist bekannt etc...
-
- Zielhost kann Zielport nicht erreichen
 - Destination Unreachable Message (Code = 1) an Absender
 - Zielhost ist erreicht
- Um die Entfernung bei den einzelnen Routern/Zielhosts zu bestimmen, wird zugleich noch die Round-Trip Zeit gemessen



Transport Layer

Schicht 4: Transportschicht

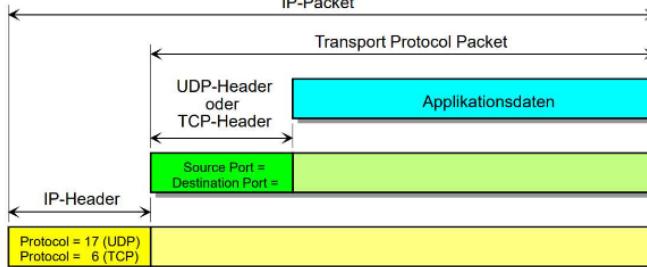


Transportlayer

Der Transport Layer bildet auch die Schnittstelle zwischen dem Betriebssystem (Kernel Space) und den Anwendungen (User Space). Der Zugriff auf die Funktionen des Transport Layers erfolgt via einer klar definierten Schnittstelle (Sockets).

Kapselung

- Die Applikationsdaten werden von den Protokollen des Transport Layers in ein IP-Paket gekapselt
- Das "Protocol"-Feld unterscheidet UDP und TCP Daten



Adressierung der Applikation durch Port Nummern

Der Client adressiert mit der Destination Port Nummer die gewünschten Server-Applikation

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

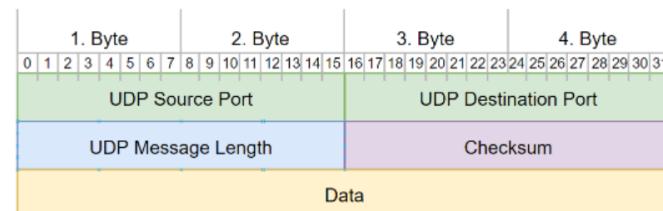
UDP - User Datagram Protocol

UDP dient dem Multi- und Demultiplexen der Datagramme zu den Applikationen.

- Verbindungslos
- Unzuverlässig

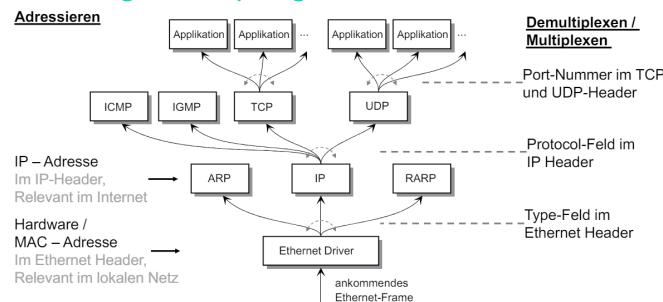
UDP-Header

- 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
 - z.B. aufgrund eines Bit-Flip



Adressierung und Multiplexing

Adressieren



Port-Nummern

- System Ports (Well-Known)** Feste Port-Nummern, für bekannte Appl. reserviert
- User Ports (Registered)** Reservierter Bereich für herstellerspezifische Appl.
- Dynamic / Private Ports** Frei verfügbare Ports

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

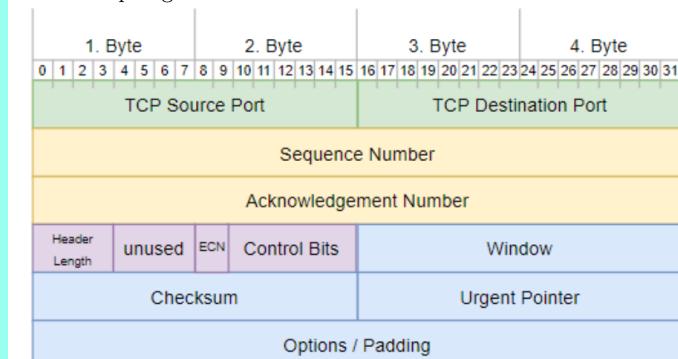
TCP - Transmission Control Protocol

TCP Eigenschaften

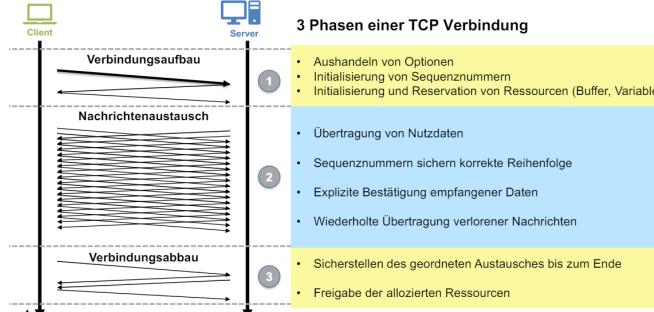
- Verbindungsorientierte Übertragung: Zuerst wird eine Verbindung zwischen Client- und Serveranwendung aufgebaut
- Zuverlässiger Verbindungsauflauf: Bevor eine TCP-Verbindung steht, muss dies von beiden Endpunkten aktiv bestätigt werden
- Hohe Zuverlässigkeit: Die Daten kommen ohne Datenverlust und in der richtigen Reihenfolge auf der anderen Seite an
- Vollduplexübertragung: Gleichzeitige, voneinander unabhängige, Übertragung in beiden Richtungen möglich
- Stream-Schnittstelle: Die Anwendung sendet/empfängt eine unstrukturierte Byte-Folge
- Graceful Termination (Verbindungsabbau): TCP gewährt die Zustellung aller Daten auch beim Verbindungsabbau
- Punkt-zu-Punkt Kommunikation: Zwei Applikationen tauschen Daten aus. Konzepte wie Multicast oder Broadcast existieren nicht.

TCP-Header Format

- Sequence-Nr.** Nummer zur Ordnung der Segmente
- Acknowledgement-Nr.** n + 1 → 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** Verbindungsauflauf- und -abbau (Bits 10-15) URG: Urgent Pointer ACK: Acknowledgement Number PSH: Push (sofort ohne buffern weiterleiten) RST: Reset (Verbindung zurücksetzen oder geschlossenen Port signalisieren) SYN: Synchronize (Verbindung aufzubauen) FIN: Verbindung abbauen
- Window** Verfügbare Puffergrösse (so viele Bytes dürfen noch gesendet werden)
- Urgent Pointer** URG = 1 → Position der wichtigen Daten
- Options** Häufigste Verwendung: MSS (Maximum Segment Size) die empfangen werden kann



Verbindungsorientierte Kommunikation



Nachrichtenaustausch

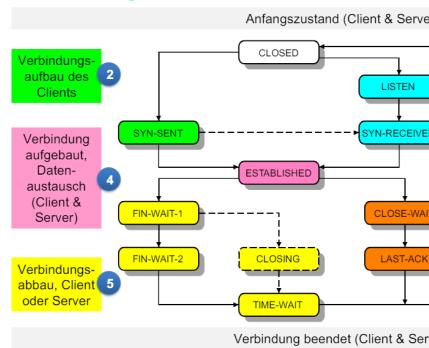
Unabhängig für jede Richtung:

- Sequence Numbers (Senderichtung)
 - Sicherstellen der richtigen Reihenfolge der Daten
 - Erkennen verloren gegangener Daten
- Acknowledge Numbers (Empfangsrichtung)
 - Bestätigung korrekt empfangener Daten
 - Erkennen verloren gegangener Daten
- Flags steuern Verbindungsau- und -abbau, signalisieren Gültigkeit von Informationen im Header und besondere Situationen.
 - SYN/FIN: Verbindungsau- und -abbau
 - ACK: Acknowledge Number ist gültig
 - PSH: Daten sollen schnellstmöglich weitergegeben werden

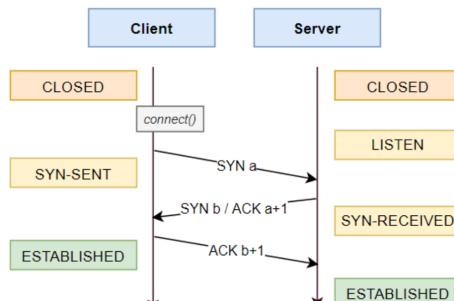
Zustände

- LISTEN Auf Anforderung warten
- SYN-SENT Anforderung geschickt
- SYN-RECEIVED Anforderung erhalten
- ESTABLISHED Verbindung besteht
- FIN-WAIT-1 Abbauanforderung geschickt
- FIN-WAIT-2 Abbauanforderung bestätigt
- CLOSE-WAIT Auf Lokale Verbindung warten
- LAST-ACK Verbindungsabbau bestätigt
- TIME-WAIT Letzte Bestätigung gesendet

Zustandsdiagramm



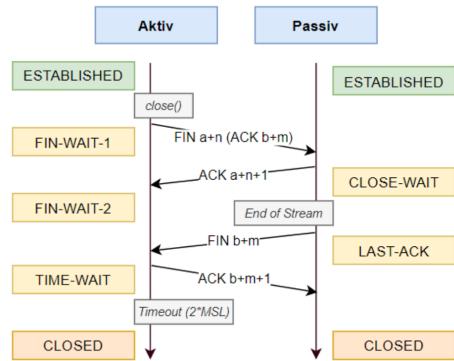
Verbindungsaufbau



Datenaustausch

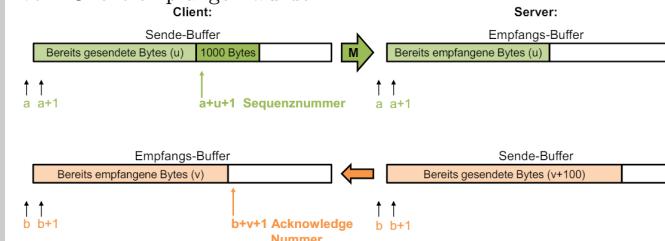
- Nach dem Verbindungsaubau können Daten geschickt werden
- Wenn der Server oder Client Daten schickt muss von dem anderen die Acknowledgement Nummer mit den Anzahl Bits der geschickten Daten aktualisieren

Verbindungsabbau



Datenaustausch

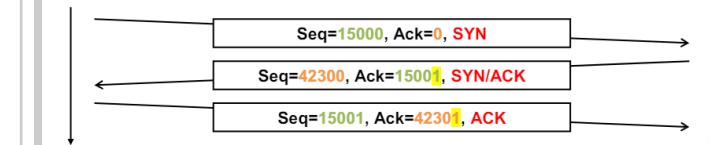
Geben Sie die Seq- und Ack-Nummern der Meldung M (1000 Bytes von Client zum Server) an und zeichnen Sie die entsprechenden Positionen ein. Beachten Sie, dass 1000 Bytes vom Server noch nicht vom Client empfangen wurden.



Vollständiges Beispiel

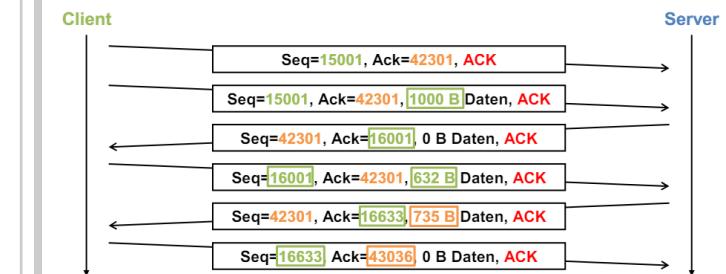
Verbindungsaubau:

- Server „horcht“ (LISTEN) auf einer bestimmten Port Nummer (z.B. 80 für einen HTTP Server)
- Client sendet Segment mit SYN=1 und zufälliger initialer Sequenznummer a (z. Bsp. 15'000) (ACK=0, weil Acknowledgement Nummer ungültig)
- Server bestätigt Sequenznummer mit Acknowledgement Nummer a+1 (15'001) und ACK=1 und wählt zufällige initiale Sequenznummer b (z. Bsp. 42'300) und setzt SYN=1
- Client bestätigt b mit Acknowledgement Nummer b+1 (42'301)
 - Erstes Byte vom Client zum Server hat Sequenznummer a+1
 - Erstes Byte vom Server zum Client hat Sequenznummer b+1



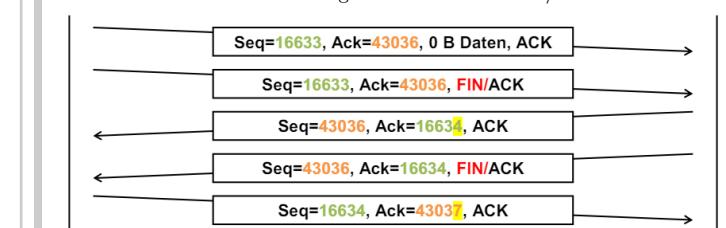
Während des Datenaustausches werden TCP-Nachrichten bi-direktional ausgetauscht

- Sequenznummer: Position des ersten Bytes der Daten im gesamten TCP-Datenstrom
- Acknowledgement Nummer: Sequenznummer des nächsten erwarteten Bytes
- ACK Flag: immer gesetzt



Beide Seiten können den Verbindungsabbau einleiten

- Ist eine Richtung geschlossen (FIN, ACK), so können in die andere Richtung immer noch Daten gesendet werden; dieser Verbindungsstatus wird als Half-Closed bezeichnet.
 - 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



TCP Adaptive Elemente

Herausforderungen zur Zuverlässigkeit zwischen Ethernet (Schicht 2) und TCP (Schicht 4):

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, $ms \dots s$	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.

Umgang mit dynamischen Situationen

- Erkennung von verlorenen Telegrammen (Round Trip Time)
- Überlast des Empfängers (Fluss-Steuerung, Flow Control)
- Überlast des Netzes (Überlast-Steuerung, Congestion Control)

RTO - Round Trip Time Out

Round Trip Time dynamische Anpassung der Wartezeit bis zum senden des nächsten Pakets (Überlastung des Netzes). TCP misst bei jeder aktiven Verbindung die RTT und passt den RTO an.

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

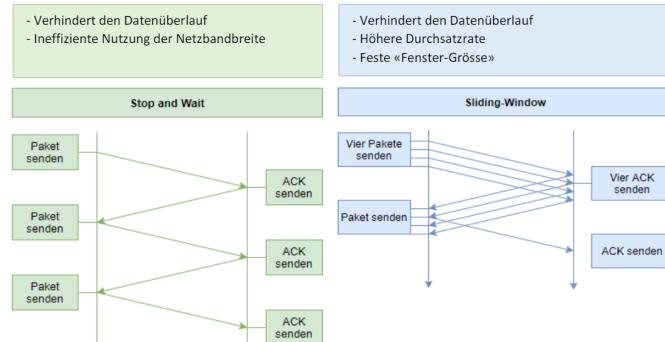
$$\alpha = 0.125 : SRTT_n = (1 - \alpha) \cdot SRTT_{n-1} + \alpha \cdot RTT_n$$
- Streuung **RTTVAR** des SRTT der Abweichungen

$$\beta = 0.25 : RTTVar_n = (1 - \beta) \cdot RTTVar_{n-1} + \beta \cdot |SRTT_n - RTT_n|$$
- Retransmission Time-Out RTO

$$RTO_n = SRTT_n + 4 \cdot RTTVar_n$$

Fluss-Steuerung und Congestion Control

Fluss-Steuerung



Sliding-Window TCP

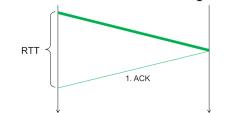
- Beide Richtungen arbeiten unabhängig voneinander
- Fenstergröße wird in Anzahl Bytes angegeben
- Verbindungsaubau: Initiale Fenstergröße wird der anderen Seite mitgeteilt (Typische Werte: 16 / 32 / 64 KB)
- Pufferplatz im Empfänger wird alloziert
- Mit jedem ACK wird der verfügbare Pufferplatz (in 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)

Bandwidth Delay Product (TCP-Puffergrößen)

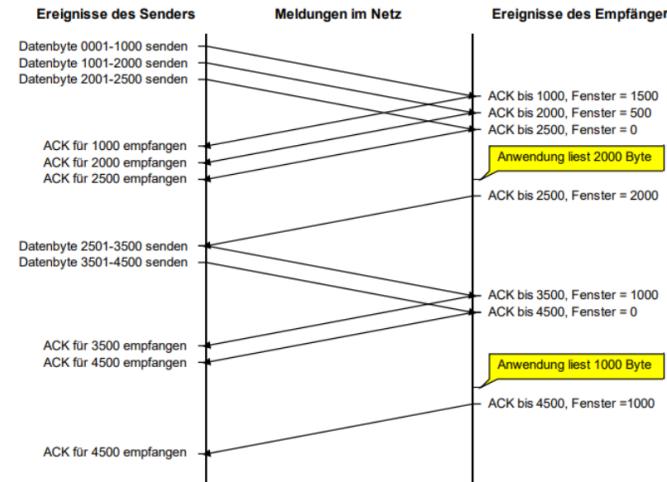
Wie gross sollten die Sende- und Empfangsbuffer gewählt werden, um eine TCP-Verbindung nicht auszubremsen?

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

RTT = Round-Trip-Time



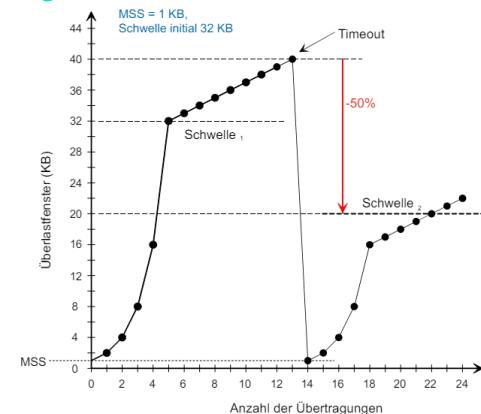
Fluss-Steuerung bei TCP



Annahmen:

- 2'500 Byte Empfangspuffer
 - 5'000 Bytes Daten
- Ablauf:
- Fenstergröße des Empfängers wird im WindowFeld des TCP-Headers übermittelt
 - Wireshark gibt dieses als Advertized Window Size an
 - Sender-Applikation benötigt nur einen Aufruf von send() für die gesamten 5'000 Bytes

Congestion Control - Slow Start



Beim Slow Start wird heran getastet wie gross die einzelnen Frames sein können.

Wichtig: Der Sender kombiniert das Congestion Window mit den Informationen zur Flow Control vom Empfänger und schickt unbestätigte Daten bis zum Erreichen von: min {Congestion Window, Advertised Window}

Application Layer

Netzwerk-Applikationen und Protokolle

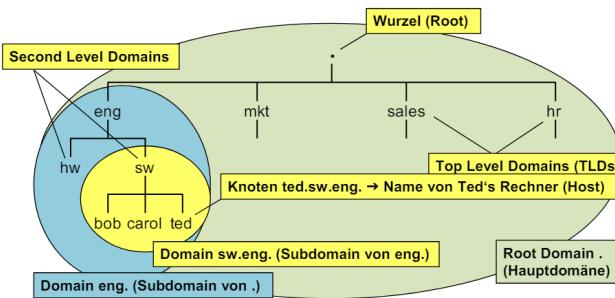
Übersicht Applikationsprotokolle

- Das Domain Name System (DNS) erlaubt übersetzt Hostnamen in IP Adressen und umgekehrt
 - Besteht aus einem hierarchischen DNS Name Space
 - Das DNS wird auf einer grossen Anzahl Name Server verteilt betrieben, ein Name Server ist jeweils für eine Zone verantwortlich (z.B. zhaw.ch)
- DHCP erlaubt einem Rechner, seine IP Konfiguration von einem Server zu beziehen
- TFTP ist ein einfaches, aber zuverlässiges File Transfer Protocol, welches z.B. diskless Systemen dazu dient, das Betriebssystem Image vom Server zu beziehen
- HTTP erlaubt den Zugriff auf verteilte Dokumente, die mittels Uniform Resource Locator (URL) eindeutig adressiert werden
- Network Address Translation (NAT) erlaubt die Wiederverwendung privater IP-Adressen

DNS - Domain Name System

DNS - Domain Name Space

- Leserliche Darstellung von IP-Adressen (Name Resolution)
- Hauptdomäne = Root
- Der Fully Qualified Domain Name (FQDN) muss eindeutig sein, Beispiel: sw.eng.
- Geschwisterknoten dürfen nicht den gleichen Namen haben



Verwaltung von Domains

- Das DNS wird verteilt betrieben (verteilt, nicht repliziert)
- Ein Name Server ist meist für eine Zone verantwortlich
 - Zone: separat administrierter Subtree des DNS
 - Ein Name Server kennt
 - * die IP-Adressen zu den Hostnamen in seiner Zone
 - * die IP-Adressen der Name Server seiner Subdomänen, falls diese nicht in seiner Zone liegen
 - * die IP-Adressen von Root und TLD Name Server, um beliebige Abfragen zu erlauben
- Aus Redundanzgründen min. zwei Name Server für eine Zone
 - Primary (Master) und Secondary (Slave)
- Ein NS kann eine Unterzone seiner Zone weiter delegieren

DNS Record Types

Der "Record Type" enthält Information, welche Daten angefragt beziehungsweise in einer Antwort vom Name Server mitgeteilt werden

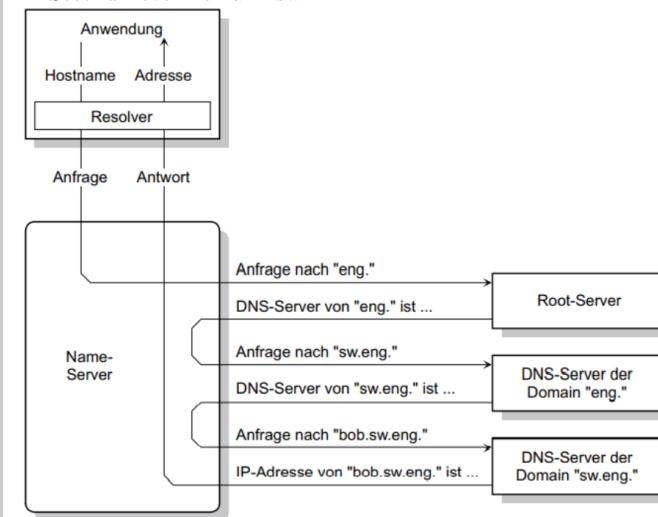
Type	Beschreibung / Funktion
A	IPv4 Adresse des gesuchten Hosts (32 Bit)
AAAA	IPv6 Adresse des gesuchten Hosts (128 Bit)
MX	Mail Exchange (Mail Server)
NS	Name Server (Name Server Name für eine Zone)
CNAME	Canonical Name (primärer Name) für einen Alias zum Host
TXT	Text Record, in Antworten für verschiedenste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

Reverse DNS Lookup

Authentisierung: Ein Server identifiziert/authentifiziert einen Client anhand des Namens, nicht anhand der IP-Adresse

DNS-Abfragen auswerten

- DNS verwendet Port 53 (UDP)
 - Resolver: lokale Software, die mit dem Name Server kommuniziert
- Beispiel: Anwendung benötigt die IP-Adresse von bob.sw.eng.
- FQDN: bob.sw.eng.
 - Root: .
 - Top Level Domain: eng
 - Second Level Domain: sw



NAT - Network Address Translation

NAT (Port Mapping)

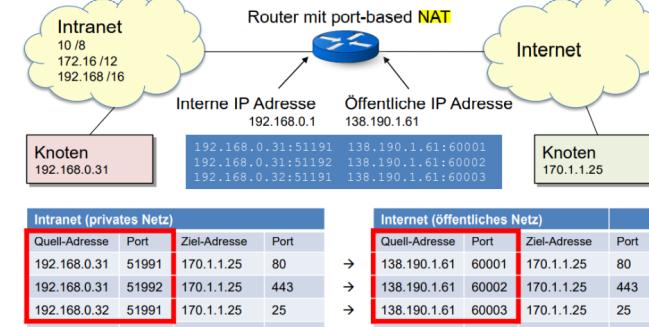
Alle Hosts im privaten Netz 192.168.0.0/8 verwenden 192.168.0.1 als Default-Gateway.

Port-basierte NAT (NAPT) hat folgende Funktionen:

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

Problem mit NAT:

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



DHCP - Dynamic Host Configuration Protocol

Bezug IP-Adresse

Wie erhält ein Knoten seine IP-Adresse?

- Lokal konfiguriert (static IP)
- Bezug der IP-Adresse über das Netzwerk
 - DHCP – erlaubt dynamische Zuteilung aus dem lokalen Adressbereich

Dynamische Zuweisung von IP-Adressen

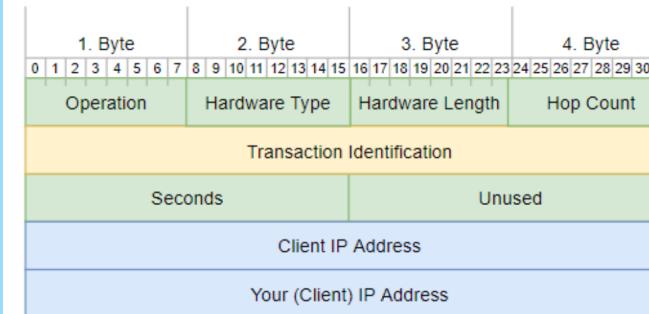
- Client verlangt eine IP-Adresse (DHCP Request)
- DHCP-Server erteilt eine freie Adresse für definierte Lease Time, oft 10 Minuten (DHCP-Response)
- Vor Ablauf der Lease Time muss der Lease (vom Client) erneuert werden
- Client, der das Netz verlässt, wird Lease nicht erneuern → Adresse wieder frei

Ablauf DHCP

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

DHCP - Dynamic Host Configuration Protocol

- Dynamische Zuweisung von IP-Adressen
- Reserviert nur IP's von aktiven Geräte



DHCP Paketformat

