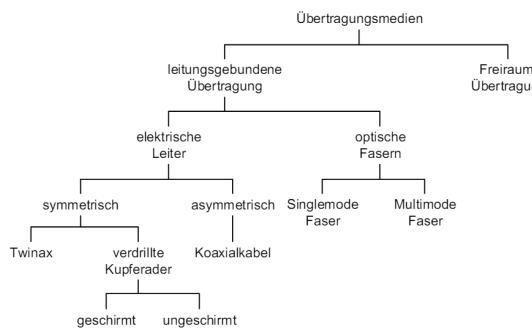


## Übertragungsmedien

### Kategorisierung Übertragungsmedien



### Signale

#### Ausbreitungsgeschwindigkeit

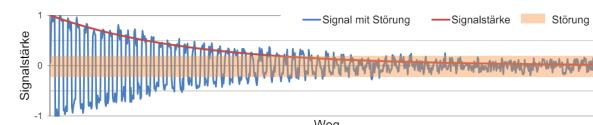
Funk- oder Licht-Signale sind elektromagnetische Wellen, die sich im Vakuum mit Lichtgeschwindigkeit  $c_0 = 299'792'458$  ausbreiten. Die Vakuumgeschwindigkeit kann nicht überschritten werden.

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

#### Signaldämpfung

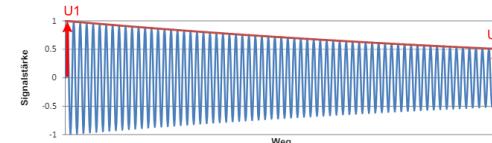
Die Signaldämpfung bezeichnet die Leistungsabnahme eines Signals auf einer Übertragungsstrecke. Die Angabe der Signaldämpfung erfolgt in dB als logarithmische Verhältniszahl von Eingangsleistung  $P_1$  zur Aufgangsleistung  $P_2$ .

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



Halbierung der Leistung entspricht ca. 3dB

#### Berechnung Signaldämpfung



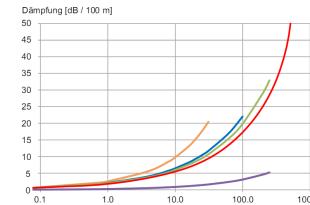
$$U1/U2 = 1/0.5 = 2, \text{ Signaldämpfung} = 20 \cdot \log(2) = 6 \text{ dB}$$

#### SNR Signal to Noise Ratio

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

## Dämpfungsbelag

Für Übertragungsmedien ist die Dämpfung pro Distanz massgebend. Typischerweise in dB pro 100 m angegeben.



#### Leistungslänge Bandbreite und Dämpfungsbelag

- Entscheidend für die maximale Leistungslänge sind Dämpfungsbelag und SNR

$$L_{max} = \frac{SNR_{min} - SNR_{min,empf}}{D_{Belag}}$$

- sinkt man die Bitrate (Bit/s) können grössere Distanzen erreicht werden
- Die Bandbreite (Frequenz) ist in der Grafik abhängig zum Dämpfungsbelag.
- Die höheren Kabelkategorien brauchen, um höhere Dämpfung zu tolerieren, bessere Schirmungen, um das Übersprechen zu minimieren.

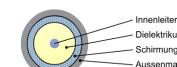
#### Kabeltypen

##### 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

##### Koaxialkabel

Eignen sich für hochfrequente Signale, unempfindlich gegenüber elektromagnetischen Störungen, früher Standard für Netzwerke, aber teuer und mechanisch heikel



##### Paarsymmetrische Kabel (Twisted Pair)

Häufig im Einsatz, auch für breitbandige Datenübertragung nutzbar, Unterscheidung zwischen Shielded(STP)/Unshielded(UTP)

- 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

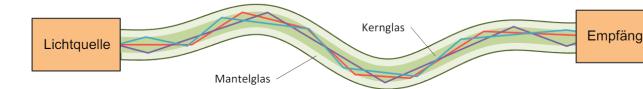
Anfälliger auf Störungen (crosstalk), kapazitiv oder induktiv, Methoden zur Behebung:

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

## Lichtwellenleiter

Hohe Bandbreite, Geringe Dämpfung, Resistent, Dispersion als begrenzender Faktor

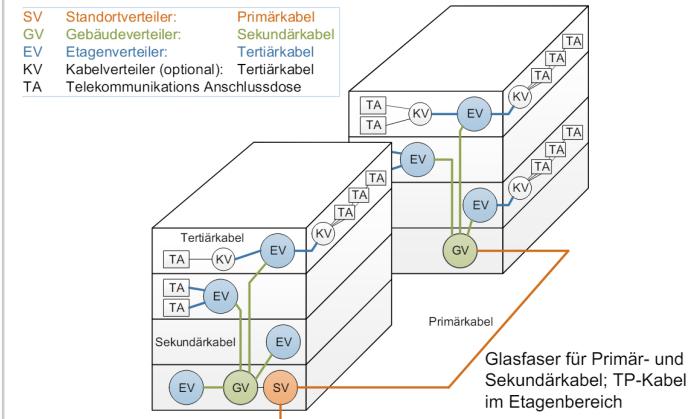
- Zentrum aus Kernglas mit hoher optischer Dichte (Brechungsindex 1.5)
- Vom Mantelglas umschlossen, geringere optische Dichte (Brechungsindex 1.48)



- Multimode: dicker Kern, günstiger, kleinere Datenraten und Übertragungsstrecken
  - Stufenfasern
  - Gradientenfasern
- Singlemode: dünner Kern, teuer!! aber funktionieren super

#### Grundprinzip optischer Fasern beruht auf der Totalreflexion und der Ausbreitung des Lichtes in bestimmten Moden

#### Strukturierte Gebäudeverkabelung nach ISO/IEC 11801



# OSI Referenzmodell

## Schichten, Protokolle und Dienste

**Systeme** Offene Systeme (im Gegensatz zu proprietären Systemen) basieren auf öffentlich verfügbaren Standards für Schnittstellen und Protokolle

**Dienst** sendet und empfängt bestätigte und unbestätigte Daten.

## Klassifizierung von Diensten

### Verbindungsorientiert

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

### Verbindungslos

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

### Zuverlässig

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

### Unzuverlässig

- Möglicher Datenverlust
- Keine Sicherung
- Streaming

**Schicht** hat die Aufgabe der darüberliegenden Schicht bestimmte Dienste zur Verfügung zu stellen. Die Schichten benötigen kein Wissen über die Realisierung der darunterliegenden Schicht.

**Protokoll** eine Sammlung von Nachrichten, Nachrichtenformaten und Regeln zu deren Austausch.

In der Technik ist ein Kommunikationsprotokoll eine Vereinbarung, die festlegt wie eine Datenübertragung zwischen Kommunikationspartnern abläuft.

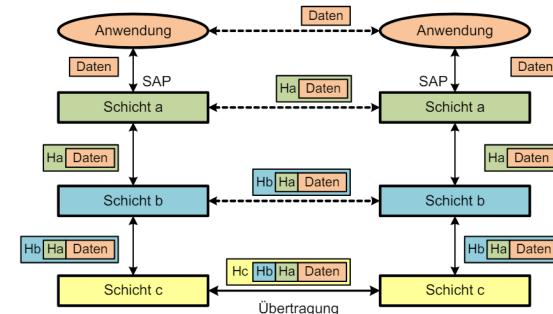
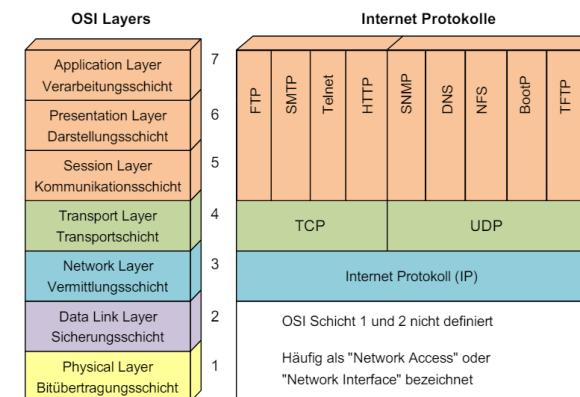
## Datenübertragung und Schichtenmodell

### OSI Modell

- Referenzmodell, beschreibt Kommunikation zwischen räumlich entfernten Kommunikationspartnern
  - 7 Schichten (unabhängige Teilfunktionen)
  - Eine Schicht erfüllt, mit Hilfe der untergeordneten Schichten, bestimmte Aufgaben und bietet der übergeordneten Schicht über eine definierte Schnittstelle einen definierten Dienst an
  - Gleiche Schichten in verschiedenen Systemen kommunizieren nach den Regeln eines standardisierten Protokolls
- Die Kommunikation erfolgt logisch zwischen Protokollimplementierungen auf gleicher Schicht (Kommunikationsbeziehung), physikalisch wandern die Daten beim Sender im Stack «nach unten», werden über ein Medium übertragen, und wandern beim Empfänger wieder «nach oben»

### OSI Layers

- Anwendungsschichten 5-7
  - Lösen allgemeine Aufgaben
- Transportschichten
  - Teil des Betriebssystems und Treiber
  - «Socket»-Schnittstelle für eigene Anwendungsprotokolle



## Physical Layer

### Schicht 1: Bitübertragungsschicht



### Funktionalität

Der Physical Layer sorgt für die ungesicherte Übertragung eines Bitstroms zwischen zwei Systemen

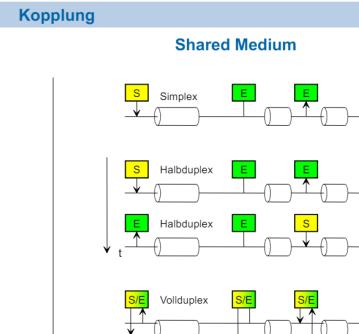
Die Standardisierung umfasst:

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

Verschiedene Übertragungsmedien:

- Koaxialkabel, Twisted Pair, Lichtwellenleiter
- Radiowellen

### Verkehrsbeziehung und Kopplung



### Arten der Kommunikation (Verkehrsbeziehung)

- Simplex: Ein Kanal, eine Richtung
- Halbduplex: Ein Kanal, abwechselnd in 2 Richtungen
- Voll duplex: Ein Kanal pro Richtung

### Arten der Verbindungen (Kopplung)

- Punkt-Punkt: Direkte Verbindung zweier Kommunikationspartner
- Shared Medium: Mehrere Partner verwenden das gleiche Medium

## Übertragungsverfahren: Parallel und Seriell

### Parallel vs Seriell

- Parallel Übertragung: mehrere Bits gleichzeitig über mehrere Leitungen
- Serielle Übertragung (dominierend): einzelne Bits zeitlich gestaffelt

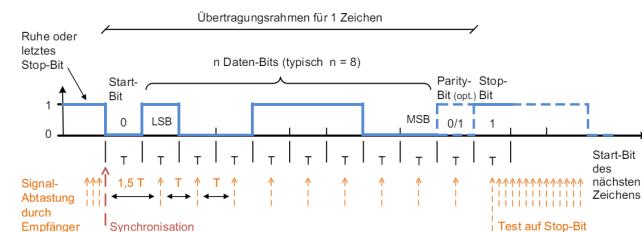
Bei der seriellen Übertragung wird weiter unterschieden zwischen seriell synchroner und seriell asynchroner Übertragung

### Serielle asynchron Übertragung

Zwischen Sender und Empfänger werden folgende Abmachungen benötigt:

- Bitrate
- Anzahl Datenbits (Typisch 1 Byte)
- Anzahl Stoppbits (Typisch 1 Bit)

Taktrückgewinnung ist möglich



Welcher Wert / welches Zeichen wird hier übertragen?

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

Was ist die Genauigkeitsanforderung an die Takte von Sender und Empfänger (geometrisch ausgedrückt)?

- Letzte Abtastung muss noch im Zeitfenster liegen (Stop-Bit bei einem Stop-Bit); also  $\frac{1}{2}T$  auf  $\frac{9}{8}T$

### Clock Drift

Maximale Framegrösse Ethernet: 1'500 Bytes.

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

Sicheres Abtasten von Daten? (im Worst-Case)

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

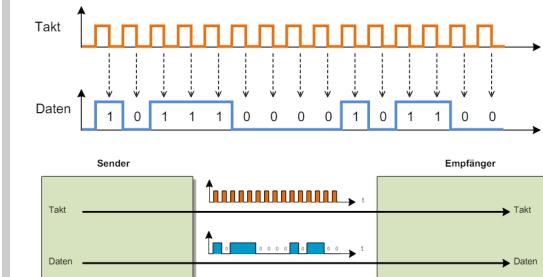
### Serielle synchron Übertragung

Bei der synchronen Übertragung arbeitet der Empfänger mit dem gleichen Takt wie der Sender

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

Die Übertragung des Taks erfolgt über ein Codierungsverfahren oder eine zusätzliche Leitung.

Es ist die Aufgabe vom Data Link Layer die Grenzen der einzelnen Bytes zu ermitteln (Preamble, etc.)

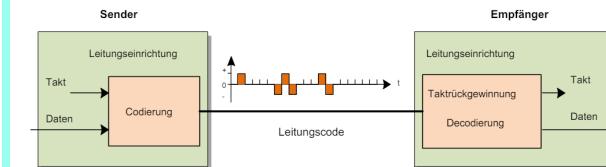


Welches Bit vom obigen Diagramm trifft zuerst beim Empfänger ein (1/0)? "1"

**Vorsicht, wenn Weg und Zeit im selben Bild gezeichnet sind.**

### Synchrone Übertragung ohne separate Taktleitung

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



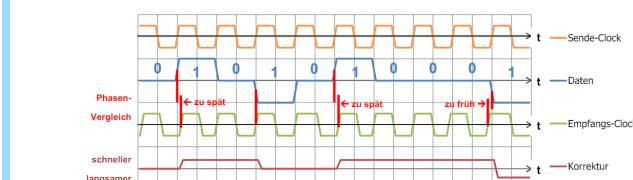
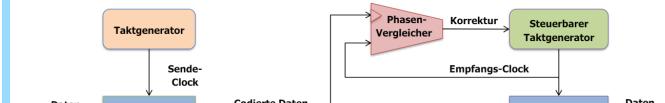
Unter Codierung versteht man hier die Umsetzung der Einsen und Nullen auf eine physikalische Grösse

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

## Leitungscodes

### Leitungscodes und Taktrückgewinnung

Mittels Leitungscode ist es dem Empfänger möglich den Takt heraus zu extrahieren (keine 2. Leitung nötig)



## Anforderungen Leitungscodes

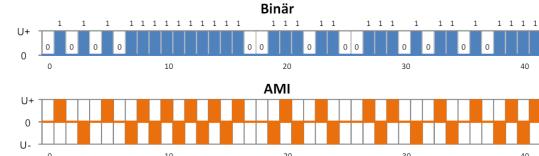
- die physikalisch vorhandene Bandbreite effizient nutzen
- Taktrückgewinnung erlauben (keine separate Taktleitung nötig)
- möglichst gleichspannungsfrei sein, um Sender und Empfänger mit Übertragern (Signaltransformatoren, Magnetics) galvanisch trennen zu können.

Wie könnte man Gleichspannungsfreiheit und galvanische Isolation in einem Schritt erreichen?

- Z.B. durch den Einsatz von Lichtwellenleitern

## AMI Leitungscode

3-wertiger AMI-Code (Alternate Mark Inversion)



Nachteile dieses Leitungscodes:

- Auf der Übertragungsstrecke drei Zustände benötigt → rein binäre Medien genügen nicht
- Bei einer längeren Folge von 0 in den Daten ist keine Taktrückgewinnung mehr möglich

## Manchester Leitungscode

wird z.B. bei 10BASE-T Ethernet verwendet, erlaubt einfache Taktrückgewinnung

- 1 positive Flanke, 0 negative Flanke
- Bei jedem Bit gibt es einen Signalwechsel
- Bandbreite von 10 MHz benötigt ( $2 \times$  theoretisches Minimum)



## NRZI MLT-3 Leitungscodierung

NRZI-Codierung (Non Return to Zero Inverted), kombiniert mit MLT-3

**011001010010001000110**



MLT-3 = Multi-Level Transmit - Ternary

## Datenrate, Bandbreite, Bandrate

### Datenübertragungsrate

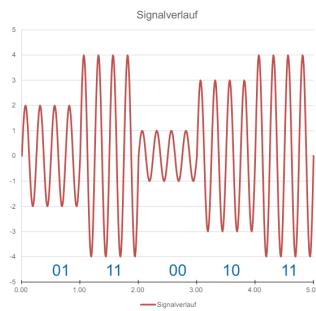
Die maximale Symbolrate  $f_s$  (Baud) ist gleich der doppelten Bandbreite B (Hz) des Übertragungskanals.

$$f_s = 2B$$

### Beispiel Amplitude Shift Keying (ASK-4)

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

- Baudrate: 1 kBaud
- Bit pro Symbol:  $\log_2(4) = 2$
- Bitrate: 2 kBit/s
- Zusatfrage Trägerfrequenz: 4 kHz



### Maximal erreichbare Bitrate

Maximale Bitrate  $R[\text{bit}/\text{s}]$

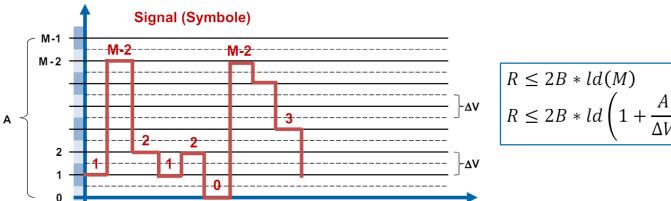
$$R \leq 2B \cdot \log_2(M)$$

Unterscheidbare Signalzustände

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

A = Max. Grösse des Signals

V = Ungenauigkeit des Empfängers



### Kanalkapazität

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

S: Signalleistung

N: Rauschleistung

## Wichtige Kenngrößen

- Bandbreite B – Einheit Hertz (Hz)
  - Eigenschaft des Übertragungskanals und durch das Medium begrenzt
- Symbolrate  $f_s$  – Einheit Baud (Bd)
  - Anzahl der Symbole pro Zeit. Limitiert durch die Bandbreite ( $\leq 2B$ ) (Nyquist)
- Bitrate R – Einheit Bit/s (bps)
  - Produkt von Symbolrate und mittlerem Informationsgehalt der Symbole (Hartley)
- Kanalkapazität C – Einheit Bit/s (bps)
  - Berücksichtigt für einen realen Kanal das Signal-zu-Rausch Leistungsverhältnis S/N (Shannon)
- 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 werden synonym verwendet

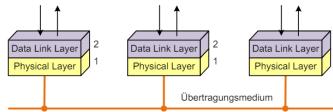
$\log$  = log2,  $\lg$  = log10,  $\ln$  = natürlicher Logarithmus

## Key Takes

- Die physikalische Schicht befasst sich mit der Umwandlung physikalischer Signale (elektrisch, optisch) in einen Bitstrom und umgekehrt.
- Verkehrsbeziehung (Simplex/Duplex), Kopplung (Punkt-Punkt oder Shared Medium) und Übertragungsverfahren (synchron/asynchron) sind bestimmende Eigenschaften.
- Die Leitungscodierung legt fest, wie genau diese Umsetzung erfolgt. Wichtige Anforderungen sind Gleichspannungsfreiheit und Taktrückgewinnung.

## Data Link Layer

Schicht 2: Sicherungsschicht



### Aufgaben

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

### Framing (Rahmenbildung-/erkennung)

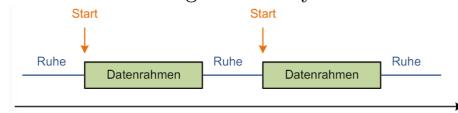


- Senderichtung: Einpacken der zu sendenden Nutzdaten in Datenrahmen (Frames)
- Empfangsrichtung: Erkennung und Auspacken der Datenblöcke aus empfangenen Frames

### Asynchron

Keine Daten → Nichts wird gesendet (Pause zwischen Frames)

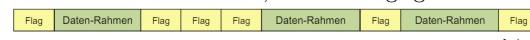
- Zu Beginn eines Frames wird ein Start-Bit gesendet
- Prüfbits am Ende eines Frames!
- Frame-Grenze gibt auch Byte-Grenze



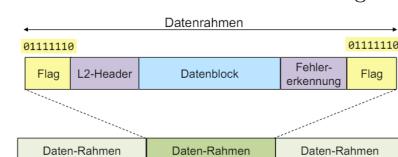
### Synchron

Frames werden ohne Unterbruch gesendet (kontinuierlicher Bitstrom auf Physical Layer)

- Stehen keine Daten an, werden Flags gesendet



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



Maskierung von Sonderzeichen (Flags) nötig!

### Bitstopfen

Wird verwendet um ein Bit-Muster zu garantieren.

- Sender fügt im Datenstrom nach 5 Einsen immer eine Null ein
- Empfänger wirft nach 5 Einsen immer ein Bit weg
- Somit gibt es (ausser bei Flags) die Bitfolge 01111110



## Fehlererkennung/-Korrektur

**Fehlerwahrscheinlichkeit** BER (Bit Error Ratio):

- Eigenschaft des Physical Layers
- wird als Dezimalzahl ausgedrückt:  $BER = 0.5 \rightarrow$  jedes 2. Bit falsch

Weitere Definitionen:

- FER (Frame Error Ratio): Fehlerhaft empfangene Daten (Frames)
- RER (Residual Error Ratio): Unentdeckte, fehlerhaft empfangene Frames

**Frame-Fehlerwahrscheinlichkeit** Wie gross ist die Wahrscheinlichkeit, dass ein Frame der Länge N mindestens einen Bitfehler enthält? Für  $BER = pe << 1$  gilt:  $(1 - pe)^N \approx (1 - N \cdot pe)$ , also:

$$P_{\text{Fehler, Frame}} \approx N \cdot pe (= FER)$$

### Wahl der Framelänge

Die Wahl der optimalen Framegrösse ist ein Kompromiss zwischen Overhead und einer geringen Frame- und Restfehlerwahrscheinlichkeit. Sie wird von der Bitfehlerwahrscheinlichkeit, der Datenrate und Verzögerungen im System beeinflusst.

- Lange Frames:
  - Höhere Nutzdatenrate (höhere Netto-Bitrate, weniger Overhead)
  - Fehlerwahrscheinlichkeit wird grösser
  - Datenverlust bei einem Fehler wird grösser
  - Wahrscheinlichkeit eines unentdeckten Fehlers wird grösser
- Kurze Frames: Tiefe Nutzdatenrate, Zuverlässigkeit

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

### Datenraten

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

$F_R$  = Framerate, B = Bitrate,  $F_L$  = Framelength

$$N = F_R \cdot P \cdot 8$$

N = Nutzbitrate, P = Payload

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

Lösung:

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

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

### Key Take Fehlererkennung/Fehlerkorrektur

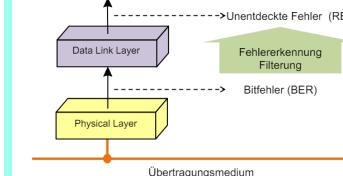
Zu Fehlererkennung wird den Daten Redundanz beigelegt (in Form von zusätzlich übertragener Information).

- Diese erhöht die Hamming Distanz (= Mindestanzahl unterschiedlicher Bits zwischen gültigen Codewörtern).
- Die betrachteten Verfahren gehören zur Familie der Block Codes. Fehlerkorrektur kann rückwärtsgerichtet (erneutes Übertragen der Daten) oder vorwärtsgerichtet (Rekonstruktion von verfälschten Bits beim Empfänger, Forward Error Correction FEC) erfolgen.

## Verfahren zur Fehlererkennung

### Fehlererkennung

Die Zuverlässigkeit der Fehlererkennung ist abhängig von der Framegröße und gewähltem Verfahren

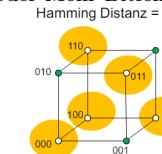


Standards IEEE 802 (LAN-Standards, z.B. Ethernet):

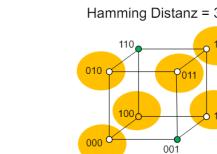
- max.  $5 \cdot 10^{-14}$  unentdeckte Fehler pro Frame-Byte
- $BER pe \leq 10^{-8}$
- CRC32 für Ethernet, mit Generatorpolynom

### Hammingdistanz als Mass für Fehlerdetektion

Codes mit Hamming-Distanz  $\leq 2$  erlauben die Erkennung von Ein- oder Mehr-Bitfehlern



Erkennung eines einzelnen Bitfehlers



Erkennung von zwei Bitfehlern

$$e = h - 1$$

### Fehlererkennung mit einfacher Parity

Ein (1) Prüfbit sichert ein Datenwort (typisch 1 Byte, auch 7 oder 9 Bits werden verwendet)

Even Parity

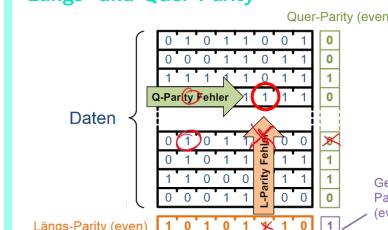


Odd Parity



- Even Parity: Anzahl 1er inkl. Parity-Bit ist gerade
  - Odd Parity: Anzahl 1-Bit inklusive Parity-Bit ist ungerade
- Even und Odd Parity sind gleichwertig

### Längs- und Quer-Parity



Wie viele Fehler können korrigiert/erkannt werden?

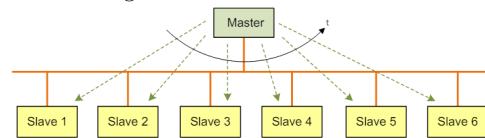
- Korrigieren: 1 Bit-Fehler
- Erkennen: mind. 3 Bit-Fehler

## Zugriffsmechanismen (Media Access)

### Gesteuerter Medium Zugriff

#### Master-Slave Verfahren

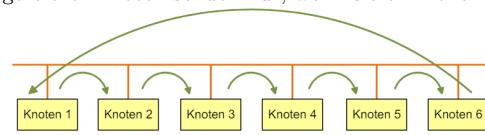
Verwenden mehrere Systeme das gleiche physikalische Medium, so muss der Zugriff auf das Medium koordiniert werden



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

#### Token Verfahren

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

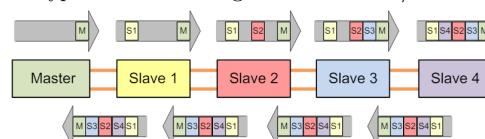


- Vorteil: Deterministisch (man weißt, wann man dran kommt)
- Nachteil: Aufwändig (Startup, Token Verlust, etc.)

#### Kombi Master-Slave und Token

Variante: Anstelle eines Tokens wird ein Frame geschickt

- Knoten fügen ihre Daten an den vorbestimmten Positionen ein (Interbus, Ethercat) oder hängen die Daten hinten am Frame an (PROFINET Dynamic Frame Packing)
- Typische Anwendung in einer Master/Slave-Konfiguration



#### Zeitsteuerung

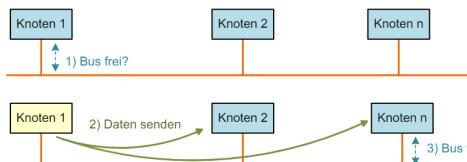
Zeitgesteuertes Zugriff (Netzbetrieb analog Taktfahrplan im Bahnenetz)

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

## Random Medium Zugriff

**Carrier Sense Multiple Access** Vorteil: Alle Stationen sind gleichberechtigt (kein Master) und haben jederzeit Zugriff auf das Übertragungsmedium

- Vor dem Senden wird das geteilte Übertragungsmedium abgehört, ob es frei ist (Carrier Sense), sonst wird bis zu einer Pause gewartet.

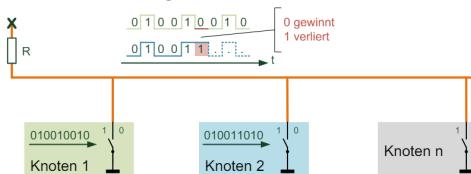


Frage: Was geschieht, wenn 2 Knoten gleichzeitig warten und zum Schluss kommen, dass sie senden können?

#### Kollisionsbehandlung

Für die Kollisionsbehandlung gibt es verschiedene Möglichkeiten:

- CSMA/CD (Collision Detection): Abbrechen und später nochmals Versuchen
  - Originalmechanismus im Ethernet, heute praktisch nicht mehr verwendet
- CSMA/CR (Collision Resolution): Hardware-unterstützte Arbitrierung
  - Arbitrierung kann passiv sein (wie unten) oder aktiv (via Busmaster)
  - Anwendung: CAN-Bus



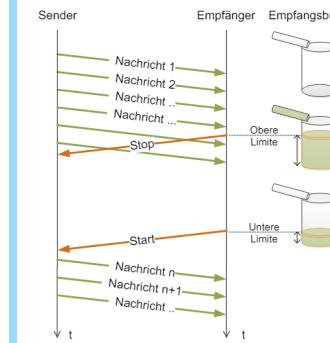
#### Key Takes Zugriffsmechanismen und Flow Control

- Bei mehr als 2 Kommunikationsteilnehmern (Shared Medium) benötigt es auf dem Data Link ein Verfahren zur Adressierung und zur Steuerung der Sendeberechtigung (Medium Access).
  - Master/Slave, Token-, Zeitgesteuert, oder mit Kolissionserkennung und -auflösung.
- Flusskontrolle wird im Zusammenhang mit Ende-zu-Ende Flusskontrolle Schicht 4 behandelt, ist aber auch Aufgabe von Schicht 2 (falls implementiert).

## Flow Control

### Explizite Start-Stop Signalisierung

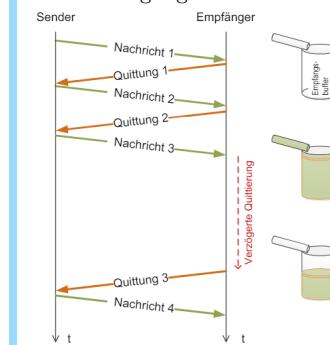
- Flow Control erlaubt einem Empfänger den Sender temporär zu stoppen
- Die Stop-/Start-Meldungen können über Leitungen oder Meldungen im Datenrückkanal erfolgen (siehe Praktikum 2).
- Anwendungen:
  - Empfänger mit langsamer Verarbeitung
  - .. und mit zu wenig Memory um den ganzen Verkehr zu speichern
  - Verstopfungen im Netzwerk (Überlastsituationen)



### Implizit mit Stop and Wait Protokoll

Flow Control ist quasi «gratis», wenn ein Stop-and-Wait-Protokoll für die Backward Error Correction verwendet wird:

- Sender wartet auf Quittung
- Empfänger verzögert seine Quittierung und stoppt damit die Übertragung



## Ethernet und LAN

### Local Area Networks (LAN)

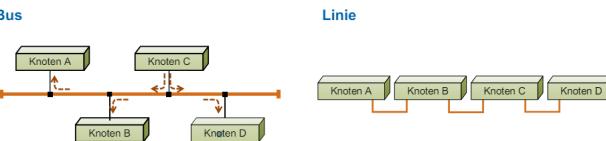
- LAN** Räumlich kleines Netzwerk mit hoher Geschwindigkeit und geringer Latenz
- Auf Schicht 2 (direkte Kommunikation zwischen allen Stationen innerhalb des LAN)

Vorteile/Möglichkeiten: Gemeinsamer Zugriff (Drucker), Datenaustausch (Directory/File Sharing), direkte Kommunikation (VoIP, Gaming), Zugang zum Internet (über Router im LAN)

### Topologien

#### Bus

- passiv angeschlossene Stationen (konstantes abhorchen, werden aktiv wenn sie senden wollen)
- Empfänger erkennt anhand Adresse, ob Daten für ihn sind



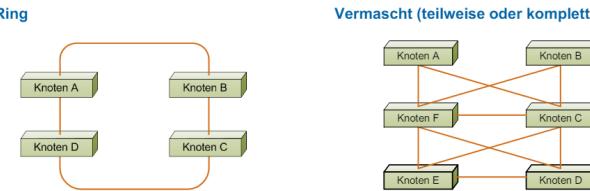
#### Linien

- Punkt-zu-Punkt Verbindungen zwischen benachbarten Knoten
- Daten empfangen, regenerieren und falls nötig weiterleiten
- Ausfall einer Station → Segmentierung des LAN in zwei Teile

#### Ring

- Achtung: «endloser Kreisverkehr» muss verhindert werden
- Gewisse Redundanz: Ausfall einer Station oder Verbindung ok

#### Ring



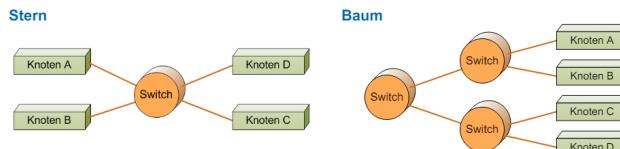
#### Vermascht

- Hohe Redundanz (Ausfälle können toleriert werden)
- hohe Kosten und Aufwand, Achtung: Data duplication

#### Stern

- Jede Station an zentralen Verteiler angeschlossen
- Verteiler entkoppelt Knoten elektrisch und macht LAN weniger störungsanfällig

#### Stern



#### Baum

- Intelligenten Switches → Grossteil der Kommunikation „lokal“
- Verringerung der Last für die einzelnen Switches

### Übertragung und Adressierung

#### Übertragungsarten

In jedem Fall: genau 1 Sender

- Unicast: 1 Empfänger, Adresse dieses Empfängers
  - Multicast: n Empfänger, Multicast-Adresse der Gruppe
  - Broadcast: alle Knoten im LAN, Broadcast-Adresse des LAN
- Diese Begriffe werden nicht nur im LAN, sondern auch allgemein in Netzwerken verwendet



#### Adressierung in LANs

- Jede Station kann Daten von jeder anderen direkt empfangen
- Eindeutige Adressen nötig: erkennen, ob empfangene Daten für die eigene Station bestimmt sind, und wer der Absender ist
- IEEE MAC Adressen
  - werden nicht konfiguriert
  - sind fix einem Interface des Gerätes zugeordnet
  - bestehen aus 6 Bytes
  - Darstellung hexadezimal 1A-2B-3C-4E-5F-67
- Geräte sind möglicherweise mobil und wechseln zwischen LANs, oder LANS werden direkt verbunden → Leitungscode

#### IEEE MAC Adressen

Registrierung bei IEEE:

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

Die ersten beiden Bits des ersten Adress-Bytes klassifizieren die MAC Adresse



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

#### Ethernet Frame Format und MAC-Adresse

Sie senden ein Ethernet Frame über eine 100BASE-TX Schnittstelle und beobachten auf dem Kabel folgende Bit-Sequenz:

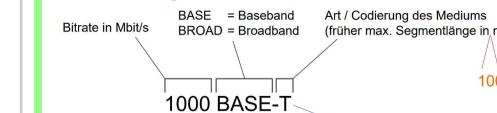
10101010 10101010 10101010 10101010 10101010 10101010  
10101010 10101011 00001000 00000000 01011010 11100011  
10011111 10000110 ...

Wie lautet die MAC-Adresse (in hexadezimaler Darstellung) des Empfängers des Frames und wer ist der Hersteller der Ethernet-Karte dieses Empfängers? (Hinweis: von den einzelnen Bytes des Frames wird zuerst das LSB und am Schluss das MSB übertragen!)

- Zuerst 7 Bytes Präambel (10101010), dann 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

### Ethernet

#### Bezeichnungs-Schema

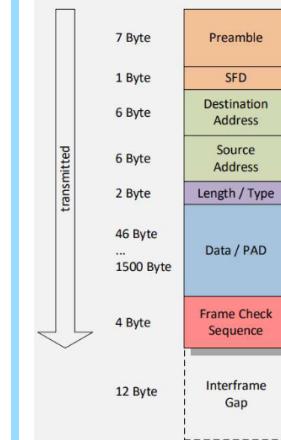


- 10BASE-T: 10 Mbit/s
- 100BASE-TX: 100 Mbit/s
- 1000BASE-T: 1 Gbit/s
- 10GBASE-T: 10 Gbit/s

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

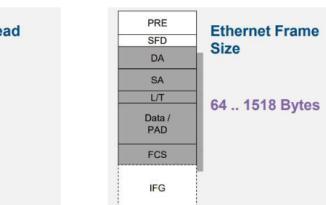
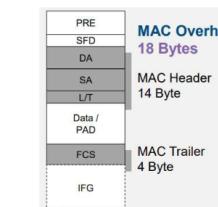
#### Ethernet Frame Format

Pro Byte wird immer das niedrigwertigste Bit zuerst und das hochwertigste Bit zuletzt übertragen. Ausnahme bei Zahlenwerten, z.B. beim Length/Type-Feld.



#### Length/Type (2 Bytes)

- Fall 1: Länge von DATA ohne PAD ( $\leq 1500$ )
  - Fall 2: Typ von Data = Protokoll der nächsten Schicht ( $\leq 1536$ )
- Data / Padding (46 – 1500 Bytes)
- Enthält die eigentlichen Datenbytes
  - Bei weniger als 46 Bytes wird mit PAD Bytes abgefüllt
- Frame Check Sequence, FCS (4 Bytes)
- IEEE CRC-32 Algorithmus
  - Interframe Gap, IFG (12 Bytes)
  - «Zwangspause» zwischen aufeinanderfolgenden Frames



## Ethernet Geräte (Network Gear)

### Übersicht Netzwerkgeräte im LAN

OSI Schicht	2-Port	Multi-Port
Data Link Layer	Dual-Port Switch (*) (IEEE: Bridge)	Ethernet Switch (IEEE: Multiport-Bridge)
Physical Layer	Repeater	Hub

### Repeater/Hubs im OSI Modell

- Verstärkt ankommende Signal auf einem Port und leitet sie «in bester Qualität» weiter
- VERALTED: Keine Kostenvorteile mehr gegenüber Switches

### Switch/Brigde

- Transparent: sollen für Endgeräte unsichtbar sein
- Verwenden «Filtering Database» mit Adress-Learning
  - Aufbau und Update der Filtering Database durch Verkehrsbeobachtung (Absenderadresse, nicht Empfänger)
  - Unbenutzte Einträge werden nach einer gewissen Zeit gelöscht
- Port Mirroring möglich
- Sicht Endgerät: A und B sind direkt verbunden
  - Filtern: nur wenn sicher kein potentieller Empfänger ausgeschlossen ist!!
  - «Flooding» für Broadcast und Multicast Zieladressen
  - Unicast: nur an den richtigen Port weiterleiten

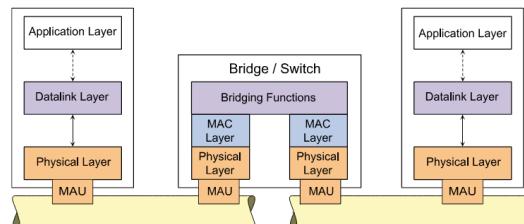
### Multi-Port-Bridges

- Daten werden ausschliesslich an den richtigen Port weitergeleitet.
- Standard-Komponente zur Kopplung von Segmenten
- Werden als Ethernet-Switch bezeichnet

### Switch im OSI Modell

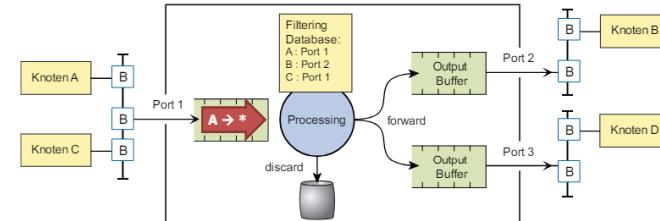
Switch arbeitet auf der Schicht 2: IEEE nennt einen Layer-2 Switch eine Bridge

- Prüft Checksumme und kann Layer-2 Adressen auswerten



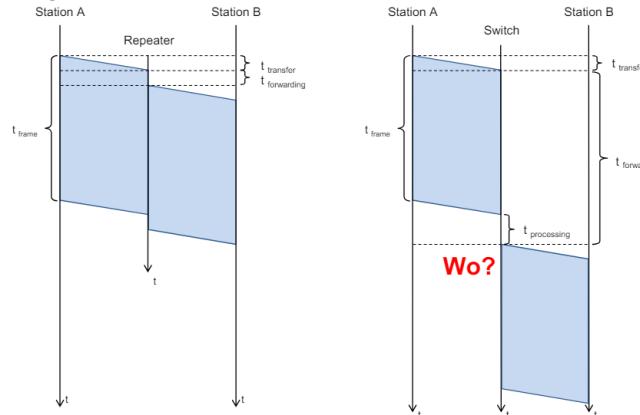
### Filtering Database

Switches verbinden LAN-Segmente



### Weg/Zeit-Diagramm für das Senden eines Frames

Gesamtübertragungszeit (Latenz):  $t_{frame} + t_{transfer}$   
 $t_{forwarding}$  kann verlängert werden um eine Verarbeitungszeit zu ermöglichen

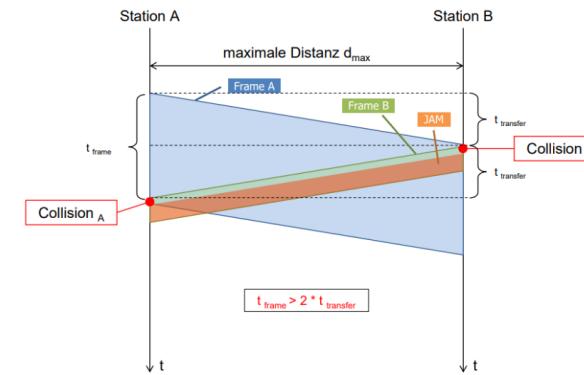


### Senden eines Frames

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

$$t_{transfer} = \frac{d_{max}}{C_{Medium}} = \frac{\text{Distanz}}{2/3 \text{Lichtgeschw.}}$$

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



Bedingungen für Kollisionserkennung:

- Ohne Repeater:  $t_{frame} > 2 \cdot t_{transfer}$
  - 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

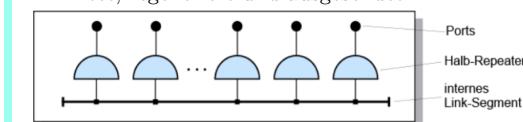
$$d_{max} < \frac{1}{2} \cdot \frac{\text{Framesize}_{min}}{\text{Bitrate}} \cdot C_{Medium}, d_{max} < \frac{1}{2} \cdot \frac{576 \text{Bit}}{10 \cdot 10^6 \cdot \text{Bit/s}}$$

## CSMA/CD

### Repeater and Collision Domain

Eine Collision Domain ist ein Teilbereich eines LANs, in dem die Frames der Stationen miteinander kollidieren können. Besteht aus über einen Repeater verbundenen Segmenten.

- Erkennen von Kollisionen
  - Halfduplex Collision Detection Unit
  - Vollduplex Keine Kollisionen
- Repeater/Hub
  - erkennt Kollisionen wenn gleichzeitig von mehreren Ports Frames empfangen werden
  - Ankommendes Signal wird an alle anderen Ports weitergeleitet, regeneriert und ausgesendet.



### Key Takings LAN/Ethernet Basics

- Alle Ethernet Varianten
  - Definieren die physikalische und Teile der Sicherungsschicht
  - verwenden das gleiche Frame-Format, das auch in allen später entwickelten Ethernet/802.3 -Varianten verwendet wird
  - MAC-Adressen der Länge 6 Bytes identifizieren Ethernet Geräte
- Switches (Bridges) arbeiten transparent (unsichtbar) auf dem Data Link Layer und schliessen mehrere Segmente zu einem LAN zusammen
  - Bridges leiten (Address Learning) Frames nur dorthin weiter, wo sie empfangen werden müssen → Lastreduzierung und Erhöhung der effektiv nutzbaren Kapazität

## Redundanz (Spanning Tree)

### Redundanz

Redundante Pfade schaffen Probleme!  $\Rightarrow$  Alle Segmente in einer loop-freien Topologie verbinden

- Ziel:
- Idee:
  - Root-Bridge auswählen (willkürliche, aber eindeutige Wahl)
  - Ausgehend von der Root einen Baum aufzubauen
  - Redundante Pfade sperren
  - alle Knoten werden genau einmal verbunden

### Spanning Tree Algorithmus

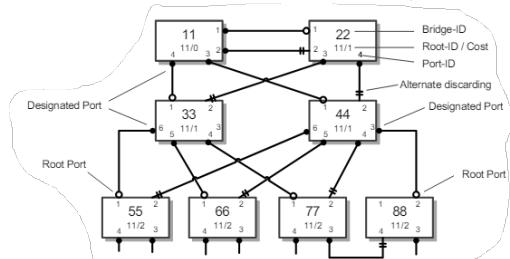
- Alle Ports für Nutzdaten blockiert
- Annahme: «Ich bin Root»
- Austausch BPDUs mit Nachbarn

#### Aufbau des Spanning Tree

- Aufdatieren der Info zu Root (kleinste ID) und Pfadkosten zu dieser
- Austausch aufdatierter BPDUs bis Konvergenz

#### Setzen der Port Roles

- Freigeben für Nutzdaten von
  - Root-Ports (Empfang der «besten» BPDU)
  - Designated-Ports (Gegenstück zu Root-Ports)
- Alle anderen Ports bleiben blockiert (Discarding)

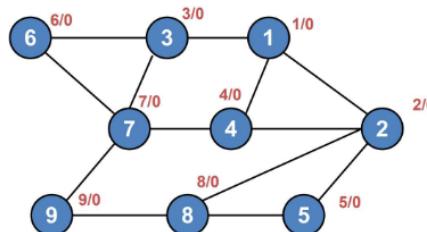


#### BPDUs (Bridge Protocol Data Units) beinhalten:

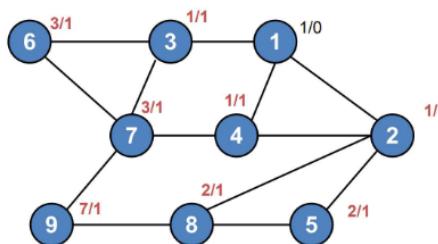
Root-ID (aus lokaler Sicht):	8 Byte
Root-Cost (aus lokaler Sicht):	2 Byte
Bridge-ID («Ich»):	8 Byte
Port-ID (Sendeport):	2 Byte

## Rapid Spanning Tree

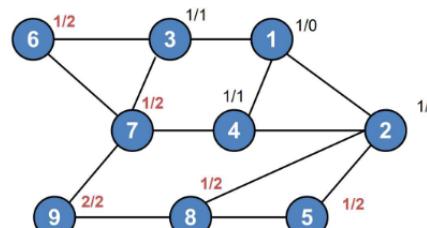
### Initialisierung:



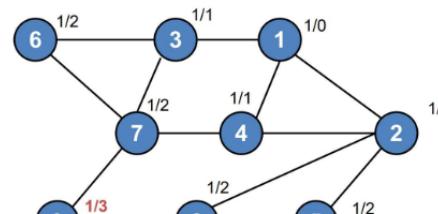
### Iteration 1:



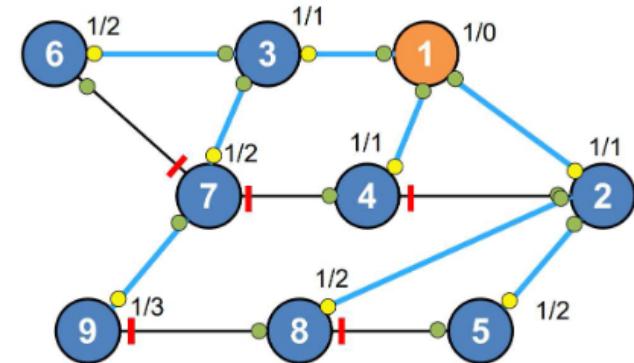
### Iteration 2:



### Iteration 3:

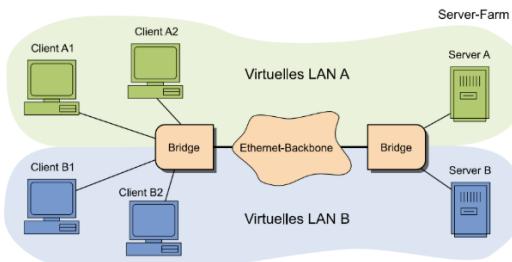


### Final



## Virtuelle LANs

**VLAN** aufteilen eines LANs in mehrere unabhängige logische Netze (Broadcast Domains), unterstützt Prioritäten  
Trunk Links: Teil von mehreren VLANs, Frames müssen eindeutig gekennzeichnet werden

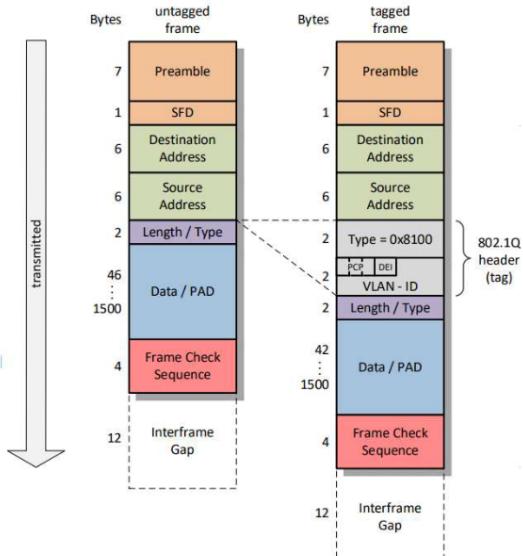


Trunk = Tagged, Access = Untagged

### VLAN Tagging

Erweiterung des Ethernet Headers durch einen VLAN-Tag

- VLAN-ID (VID) im VLAN-Tag wird zur Zuordnung verwendet
- Priority Code Point (PCP) ermöglicht die Priorisierung gewisser Applikationen
- Discard Eligibility Indicator (DEI) 0 → Frame wird bei Engpässen zuerst verworfen
- VLAN Tagging erfolgt oft beim Eintritt/Austritt ins Netz
- Die maximalen Nutzdatenlänge bleibt erhalten, der Ethernet Frame wird 4 Bytes länger
- Vorteile:
  - Transparent (unsichtbar) für Endgeräte
  - VLAN Konfiguration nur im Netz
    - \* Einfache zentrale Konfiguration
    - \* Einfaches Anpassen der Konfiguration



## Switched LANs: Merkmale von Switches/Bridges

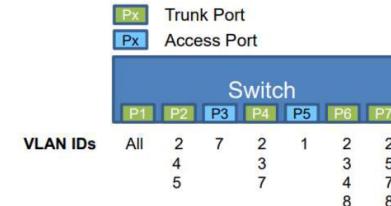
Anzahl Ports	Steckergroesse ist im Extremfall die Limitierung
Adressabelle	Wie viele Stationen können im LAN existieren
Filterrate	Maximale Frames / s / Port (Empfangsrichtung)
Transferrate	Maximale Frames / s / Port (Senderrichtung)
Backplane / Fabric Kapazität	Maximaler Gesamtdurchsatz zwischen allen Ports
Architektur	<b>Store-and-Forward:</b> Frame wird komplett empfangen und dann weitergeleitet <b>Cut-Through:</b> Frame wird schon nach Decodierung der Zieladresse weitergeleitet <b>Leitet auch Korrupte Frames weiter, in der Regel aber kein Problem</b> <b>Adaptive Cut-Through:</b> 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

### VLAN Tagging

Es werden folgende Frames gesendet:

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

Der Switch ist wie folgt konfiguriert:



Welche Frames werden an welchen Ports gesendet und sind diese getagged oder ungetagged? Internet Protokolle des Network Layers Das Internet verbindet mehrere LANs miteinander durch Router.

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

### Key Takeaways Switched LANs

- Port Mirroring ist ein mögliches Verfahren zur Verkehrsbeobachtung und Fehlersuche in Switched Ethernet
- Redundanz wird ermöglicht
  - durch die „künstliche“ Reduktion der Topologie auf eine Baumstruktur durch Spanning Tree Algorithmen
- Kompatibilität (10)/100/1000BASE-T wird erreicht durch
  - Beibehaltung von Frame Format und Schnittstelle zwischen PHY und MAC
  - Autonegotiation mittels FLP bursts / NLP
- PHY Codierung ist unterschiedlich (Scrambled NRZ/MLT-3 mit 4/5 Codierung, ...)
- Höhere Datenrate → höhere Ansprüche an die Signalverarbeitung und Algorithmen im PHY

## Switched LANs

### Switched LAN Monitoring

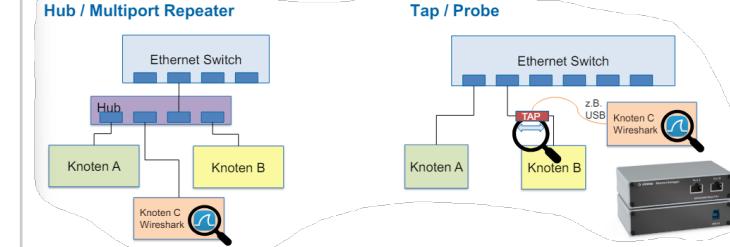
#### Hub/Multiport Reader

- Pro
  - Alle Daten sind auf allen Ports sichtbar
- Con
  - Verfälscht die Situation völlig
  - Nur Half Duplex Betrieb für A und B möglich

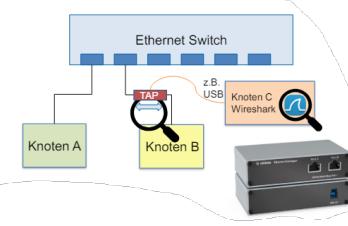
#### Tap/Probe

- Pro
  - Sehr detaillierte low-level Analyse möglich
- Con
  - Kosten
  - Veränderung des Netzwerkes (Latenz)

#### Hub / Multiport Repeater

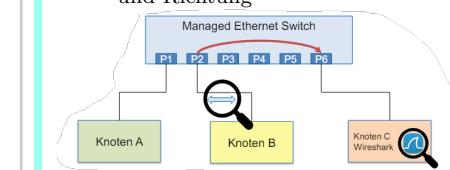


#### Tap / Probe



**Port Mirroring** Managed Switches unterstützen eine Vielzahl Diagnosefunktionen

- Statistik Zähler pro Port: Anzahl Rx und Tx Frames, FCS-Fehler, zu lange Frames, ...
- Port-Mirroring leitet Daten zusätzlich auf einen anderen Port um
- Konfigurationsoptionen herstellerabhängig
  - Kompletter Port (Rx plus Tx), oder selektiv Port Nummer(n) und Richtung



**Autonegotiation** Ermittlung der besten Betriebsart durch Austausch der Leistungsmerkmale zweier Netzwerkkomponenten, beruht auf Fast Link Pulses (FLP)

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

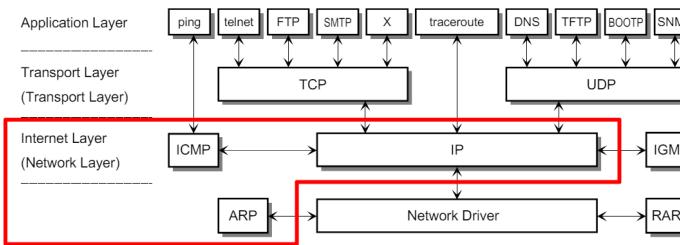
Kabelkategorie	10BASE-T	100BASE-TX	1000BASE-T	10GBASE-T
CAT5e	CAT3 - 16 MHz CAT5 - 100 MHz	CAT5 - 100 MHz CAT6 - 250 MHz	CAT5 - 100 MHz CAT6 - 250 MHz	CAT6A - 500 MHz CAT7 - 600 MHz CAT7A - 1000 MHz
Line Coding	Manchester	MLT-3, 4B5B	PAM-5, 8B/10B	PAM-16, 64B/65B, FEC
Baudrate	2 Adreßpaare simplex 10 MBaud	2 Adreßpaare simplex 125 MBaud	4 Adreßpaare duplex 4 x 125 MBaud	4 Adreßpaare duplex 4 x 800 MBaud
Link Pulses	NLP	FLP	FLP	FLP

#### GBASE-T

Name	Standard	Speed (Mbit/s)	# TP	Coding	Baud rate per lane (Mbps)	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

## Network Layer

Schicht 3: Internet



### Die Netzwerkschicht

- Verbindet verschiedene Netze
  - Er leitet IP-Pakete zwischen zwei beliebigen Hosts weiter; egal in welchen Teilnetzen sich die Hosts befinden.
- Nur Transport der IP-Pakete → höhere Layer übernehmen:
  - Fehlerfreie, komplekte Übertragung
  - Richte Reihenfolge, Flusskontrolle

### Was braucht es damit das Internet "funktioniert"?

Adressierung, Router, Routing, Fragmentierung, ICMP, ARP

### 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

Die vier Grundsätze führen zur Wahl eines paketvermittelnden Netzes auf der Grundlage von vier Layern (siehe OSI Modell)

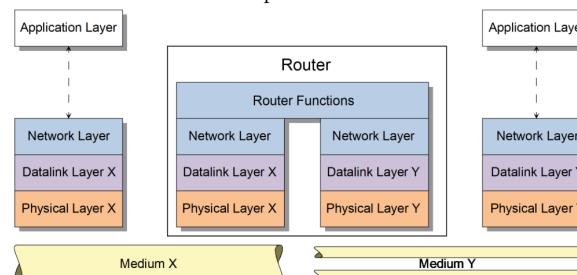
- (Application-)Message/Stream Layer 5-7
- (Transport-)Paket, Datagram Layer 4
- (IP-)Paket (früher Datagram) Layer 3
- (HW-specific) Frame Layer 1-2

## Netzwerk Applikationen und Protokolle

Routing

**Router** verbinden Subnetze (Ethernet, xDSL, WLAN, etc.)

- Router empfangen nur Pakete, die direkt an sie adressiert sind.
- Die Weiterleitung erfolgt anhand der Network Layer Adresse.
- Benutzen immer den optimalen Pfad.



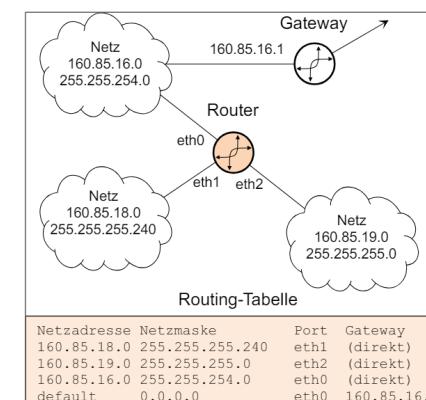
### Routing and Forwarding

- 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
  - Jeder Knoten auf dem Weg, einschliesslich dem Sender, wertet seine Routingtabelle aus und trifft Forwarding-Entscheidungen

### Routing-Tabelle

Enthält Informationen, wie jedes Netz (und damit jedes Interface) erreicht werden kann

- Für Weiterleitungsentscheidung notwendige Informationen
  - Welche Netze (Netzadresse, Subnetzmaske) gibt es? Wichtig für die Skalierbarkeit: Netz-Adressen, nicht Interface Adressen!
  - Über welches Interface sind diese Netze erreichbar?
  - Ist das Zielnetz erreicht, oder muss das Paket an einen nächsten Router (IP-Adresse) weitergegeben werden?
- Sortiert nach der Länge der Netzmasks
- Von oben nach unten durchsucht
- Verglichen werden die Netzadressen
- erster Eintrag der passt wird verwendet, default Eintrag am Schluss passt immer



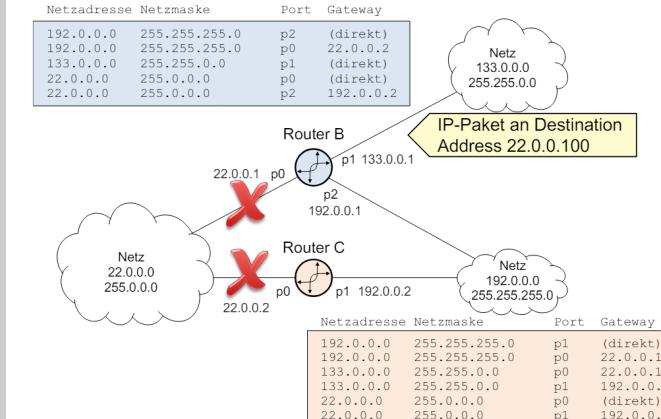
### Flaches Routing

- Router kennt explizite Wege zu jedem einzelnen Zielnetz
  - Pakete an unbekannte Netze werden verworfen
- Redundanz möglich → Speichern mehrerer Wege ins gleiche Netz
  - Wege müssen nicht gleich gut sein
- Einsatz in stark vermaschten Netzen oder im zentralen Bereich (Backbone)
- Sehr grosse Routing-Tabellen

### Flaches Routing Übung

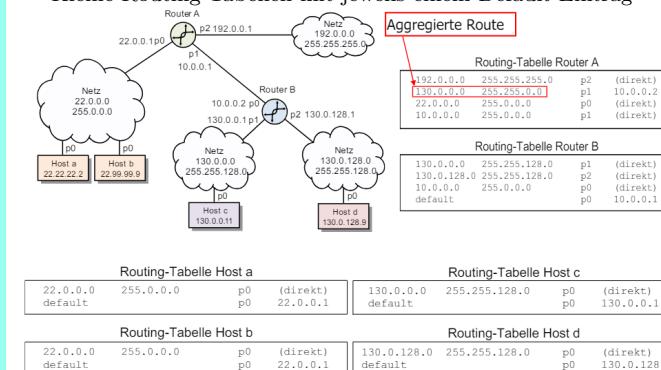
Was geschieht mit dem IP-Paket?

- 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



## Internet Protokolle (IP)

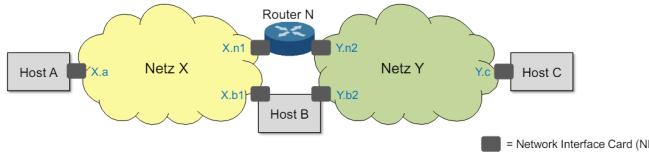
### Hierarchische Adressierung

IP-Adressen sind zweistufig hierarchisch

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

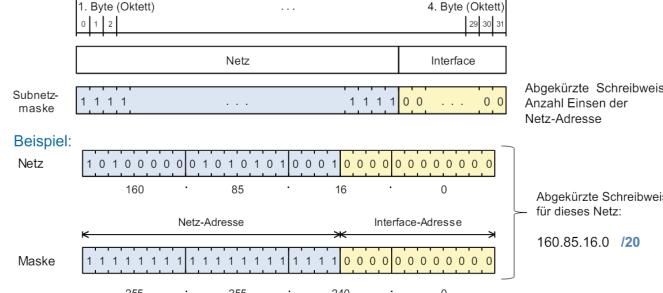
### Terminologie

- Sender und Empfänger werden im TCP/IP Referenzmodell als Hosts bezeichnet
- Der Internet Layer stellt ein virtuelles Netz mit einer einheitlichen Adressierung zur Verfügung → IP-Adressen
  - Die IP-Adresse identifiziert ein Host-Interface (und nicht den Host) eindeutig innerhalb eines Netzwerks
  - Jeder Host hat mindestens eine Adresse
  - Multi-Homed Hosts haben mehrere IP-Adressen
- IP bietet einen unzuverlässigen, verbindungslosen Dienst



### 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 (1000'0000)	/18	16'384 - 2
128 (1000'0000)	/17	32'768 - 2
0 (0000'0000)	/16	65'536 - 2

### 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

### Rechnen mit Netzmasken

Typische Internet-Adressen Aufgabenstellung: Berechnen Sie die fehlenden Informationen

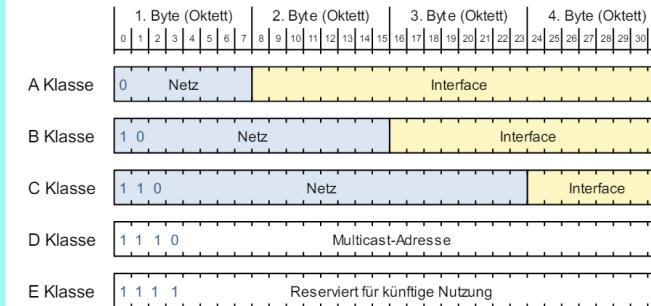
IP-Adresse	Subnetzmaske	Netzadresse	Broadcastadresse	Anzahl Adressen inkl. Netz- und Broadcastadresse
a 17.8.7.8	255.255.0.0 /16	17.8.0.0	17.8.255.255	65'536
b 11.7.177.4	255.255.224.0 /19	11.7.160.0	11.7.191.255	8'192
c 144.3.133.1	255.255.192.0 /24	144.3.128.0	144.3.191.255	16'384
d 31.4.2.166	255.255.255.248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

### Classful Routing: Sub-/Supernetting

#### Classful Routing

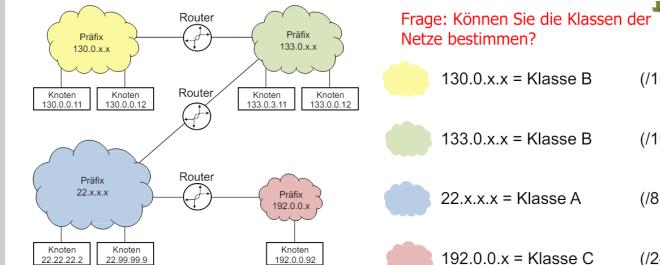
Ursprünglich war der IP Adressbereich in fünf Netzklassen (A - E) eingeteilt

- 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	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	2'097'152	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 Netzmasks 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

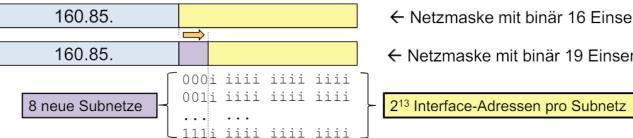
## Subnetting Aufteilung in kleinere Netze

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 Netzmasken-Bits:  $8 = 2^3$ , es werden 3 1en in der binären Netzmaske ergänzt

- 3 Bits identifizieren 8 Subnetze ( $000 \rightarrow 111$ )
- Die Netzmaske verändert sich von /16 zu /19 ( $255.255.0.0 \rightarrow 255.255.224.0$ )
- 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 Netzmaske hat nun 19 statt 16 "1" → Subnetzmaske: 255.255.224.0 oder /19
- Der Host-Anteil der binären Nutzmaske 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

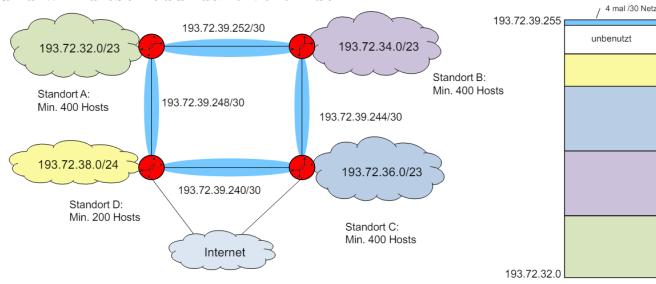
Das fünfte Netz wird wie folgt gekennzeichnet:

- 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

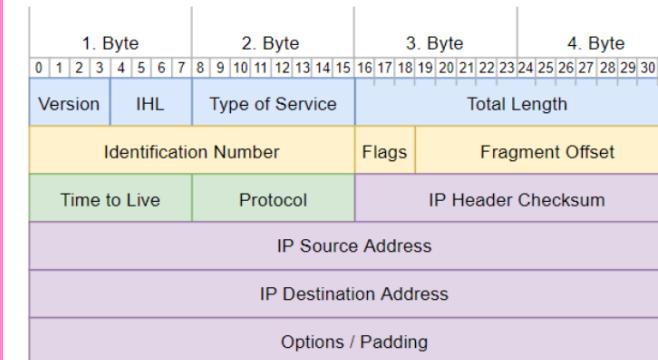
Ein KMU mit 4 Standorten hat von seinem ISP das Netz 193.72.32.0 /21 erhalten. Das KMU hat 3 grössere und einen kleineren Standort und will diese redundant verbinden.



## 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 (nicht Nutzdaten)
- **Source Address** Wer das Paket ursprünglich abgesendet hat
- **Destination Address** Wer das Paket schliesslich erhalten soll
- **Options/Padding** variabel, füllt auf ein Vielfaches von 32Bits auf



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

## Fragmentierung

- Länge der Nutzdaten = Vielfaches von 8 Bytes
- Die Pakete haben die gleiche und grösstmögliche Länge

## Reassembly

- nutze 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

## IP-Fragmentierung in heutigen Systemen

Übertragung durch unterliegendes Netz limitiert (maximale Payload)

- Im IP-Kontext als Maximum Transfer Unit (MTU) bezeichnet
- Unterschiedlich für verschiedene Technologien

Fragmentierung in Routern wird vermieden

- Fragmentierung findet im Sender statt
- Entlastet Router von dieser Aufgabe
- Hierzu muss der Sender die kleinste MTU auf dem gesamten Pfad kennen
- Jedes Fragment ist ein vollständiges IP-Paket inklusive Header und wird an den Empfänger weitergeleitet
- Das Reassemblieren findet erst im Ziel-Host statt
- Pakete nehmen eventuell unterschiedliche Pfade
- Pakete müssen sonst eventuell später wieder fragmentiert werden

## 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

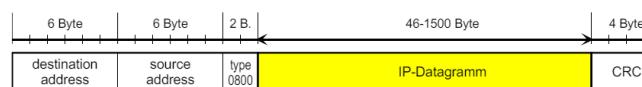
## Key Takes

- 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

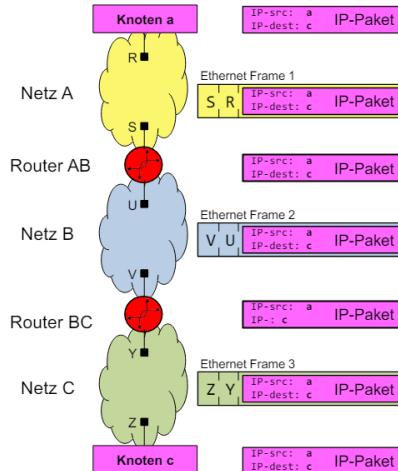
#### ARP (Address Resolution Protocol)

- 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?

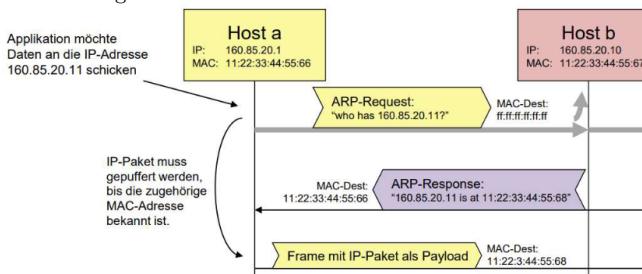
- Knoten a sendet ein IP-Paket an Knoten c
  - das Paket enthält die IP-Adressen von a und c
- Knoten a konsultiert die Routing Tabelle und sieht:
  - dass c über den Router AB erreicht werden kann, und
  - Kennt nun die IP-Adresse von Router AB
- Knoten a generiert ein Ethernet Frame, welches an die 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

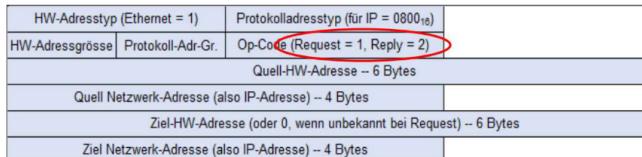
- Ermittlung der Hardwareadresse (MAC) zu einer IP-Adresse
- ARP-Request wird an Broadcast-Adresse gesendet
- ARP-Response wird von Knoten mit angefragter IP-Adresse an Absender gesendet
- Die ARP-Tabelle speichert bekannte <IP-MAC> Kombinationen für eine gewisse Zeit



### 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:

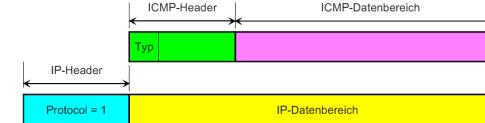
- Erkennung von Adresskonflikten
  - Nach einer Adresszuweisung (manuell oder per DHCP) wird ein ARP-Request an die eigene IP-Adresse gerichtet, um zu prüfen, ob kein anderer Host im LAN die Adresse verwendet
  - Falls eine Antwort kommt, liegt ein Adresskonflikt vor
- Erneuerung von Einträgen im ARP-cache
  - 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

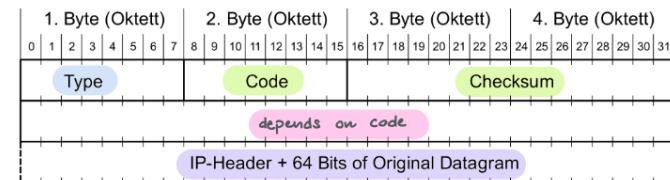


### ICMP Format

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

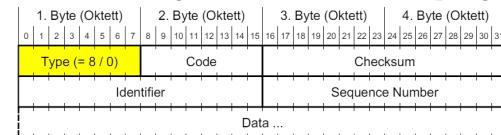
- Destination Unreachable (Fehler)
  - IP-Paket kann nicht zum Ziel gebracht werden
  - Beispiel: Keine Route zum Ziel-Host vorhanden
- Redirect (Optimierung)
  - Ein Host H sendet ein IP-Paket an einen ersten Router R1
  - R1 stellt fest, dass der nächste Router auf dem Weg zum Ziel R2 ist; R2 ist aber 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

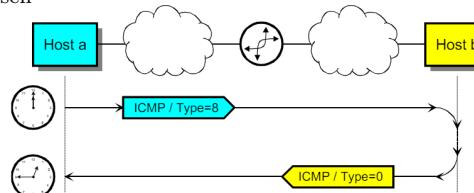
- Test, ob Host erreichbar ist
- Host antwortet auf Echo Request (Type 8) mit Echo Reply (Type 0), mit gleichem Inhalt wie der Echo Request

### 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 of Original Datagram

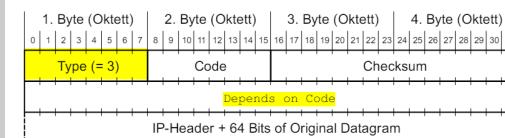
## Path MTU Discovery:

### 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 0038 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?
- Protokoll: 01
- Wie erkennen Sie den ICMP Typ? Type: 03
- Erkennen Sie die "64 Bytes of Original Datagram"? Original Data

## ICMP Time Exceeded

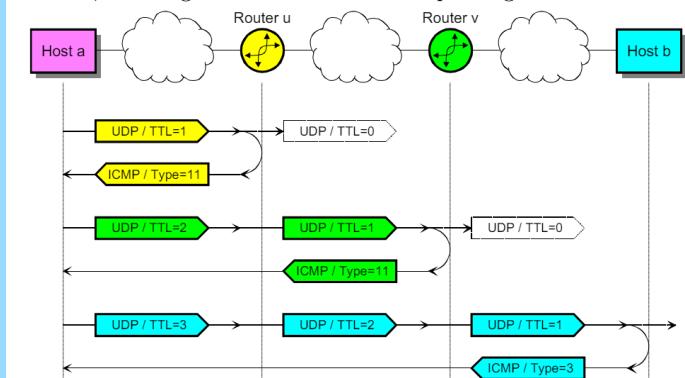
- Type = 11
- unused (must be 0)

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)

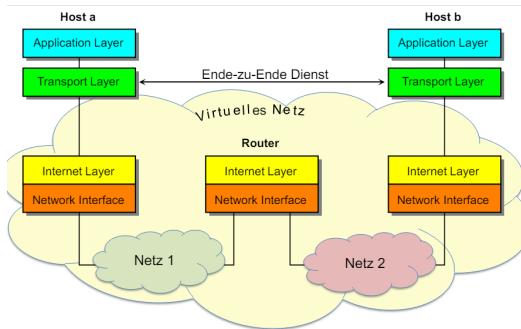
traceroute erlaubt, den Weg zu einem beliebigen Host (oder einem fehlerhaft konfigurierten Router auf diesem Weg) zu finden

- 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 den einzelnen Routern/Zielhosts zu bestimmen, wird zugleich noch die Round-Trip Zeit gemessen



## Transport Layer

Schicht 4: Transportschicht

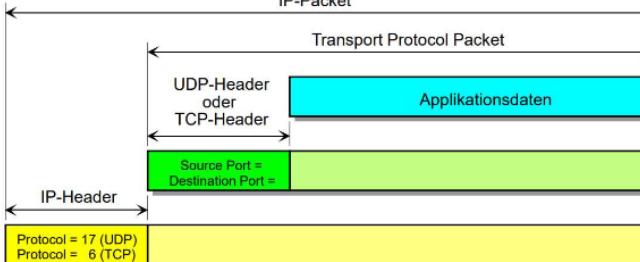


### Transportlayer

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)

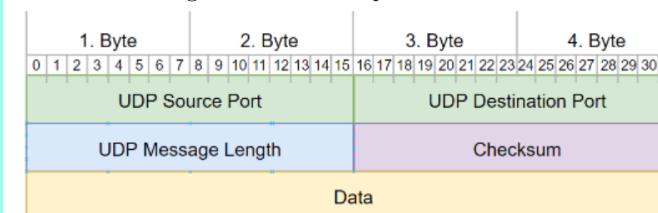
## UDP - User Datagram Protocol

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

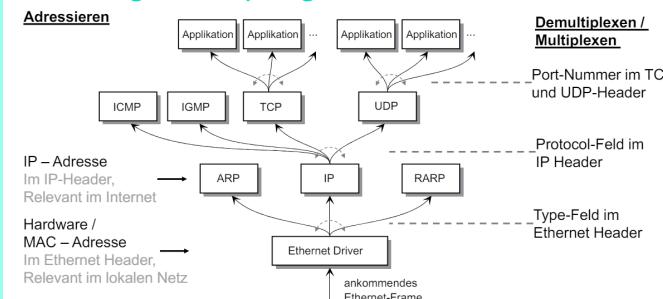
- Verbindungslos
- Unzuverlässig

### 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



### 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

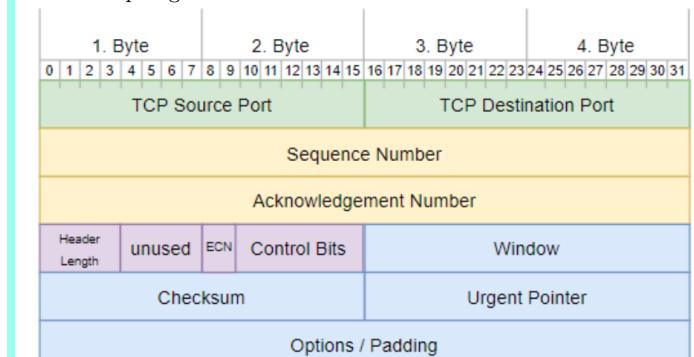
## TCP - Transmission Control Protocol

### TCP Eigenschaften

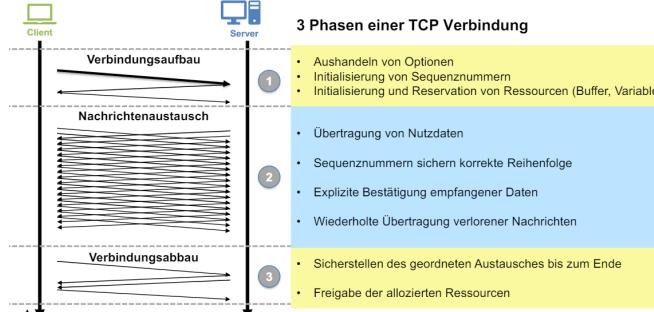
- Verbindungsorientierte Übertragung: Zuerst wird eine Verbindung zwischen Client- und Serveranwendung aufgebaut
- Zuverlässiger Verbindungsauflauf: Bevor eine TCP-Verbindung steht, muss dies von beiden Endpunkten aktiv bestätigt werden
- Hohe Zuverlässigkeit: Die Daten kommen ohne Datenverlust und in 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.**  $n + 1 \rightarrow$  Daten bis und mit  $n$  korrekt und vollständig angekommen
- Data Offset** Gibt an wo Daten beginnen / enden
- ECN-Flags** Explicit Congestion Notification
  - Bit 8: CWR (Congestion Window Reduced)
  - Bit 9: ECE (ECN-Echo)
- Control Bits** 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  $\rightarrow$  Position der wichtigen Daten
- Options** Häufigste Verwendung: MSS (Maximum Segment Size) die empfangen werden kann



## Verbindungsorientierte Kommunikation



## Nachrichtenaustausch

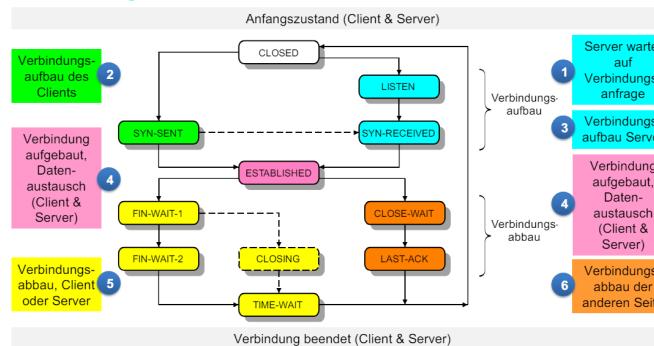
Unabhängig für jede Richtung:

- 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

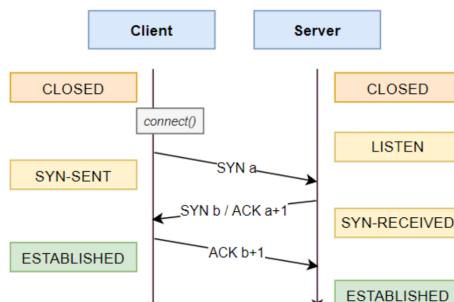
## Zustände

- 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



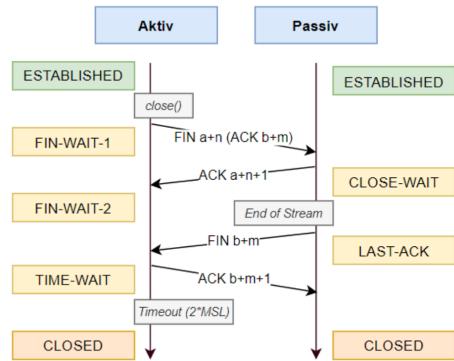
## Verbindungsaufbau



## Datenaustausch

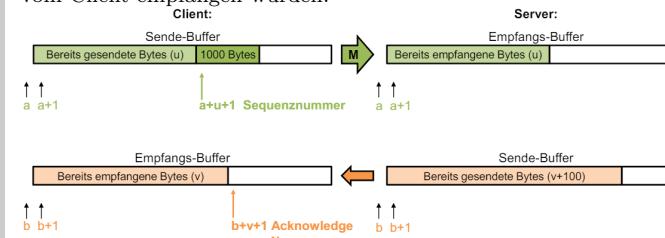
- Nach dem Verbindungsaubau können Daten geschickt werden
- Wenn der Server oder Client Daten schickt muss von dem anderen die Acknowledgement Nummer mit den Anzahl Bits der geschickten Daten aktualisieren

## 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

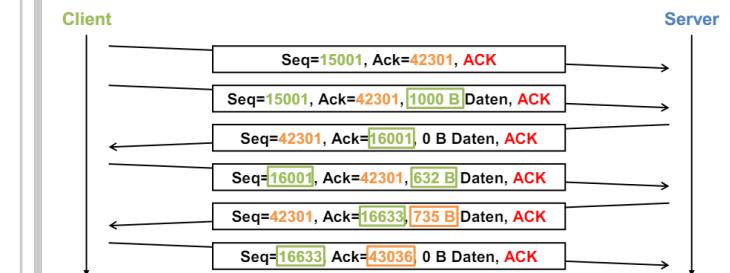
### Verbindungsaubau:

- 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



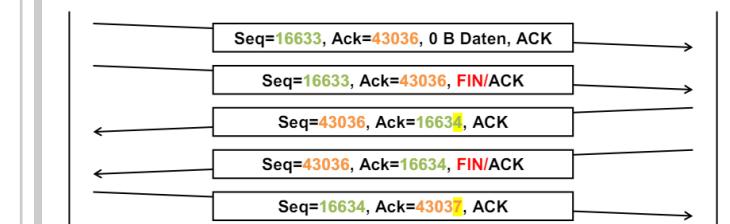
Während des Datenaustausches werden TCP-Nachrichten bi-direktional ausgetauscht

- 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)  

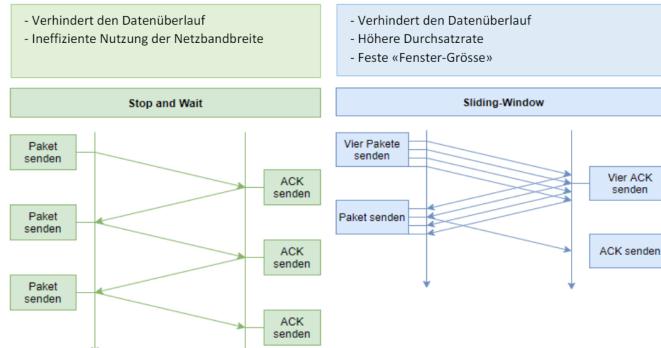
$$\alpha = 0.125 : SRTT_n = (1 - \alpha) \cdot SRTT_{n-1} + \alpha \cdot RTT_n$$
- Streuung **RTTVAR** des SRTT der Abweichungen  

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

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

## Fluss-Steuerung und Congestion Control

### Fluss-Steuerung



### Sliding-Window TCP

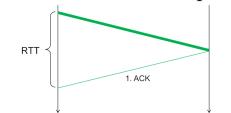
- Beide Richtungen arbeiten unabhängig voneinander
- Fenstergröße wird in Anzahl Bytes angegeben
- Verbindungsaubau: Initiale Fenstergröße wird der anderen Seite mitgeteilt (Typische Werte: 16 / 32 / 64 KB)
- Pufferplatz im Empfänger wird alloziert
- Mit jedem ACK wird der verfügbare Pufferplatz (in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

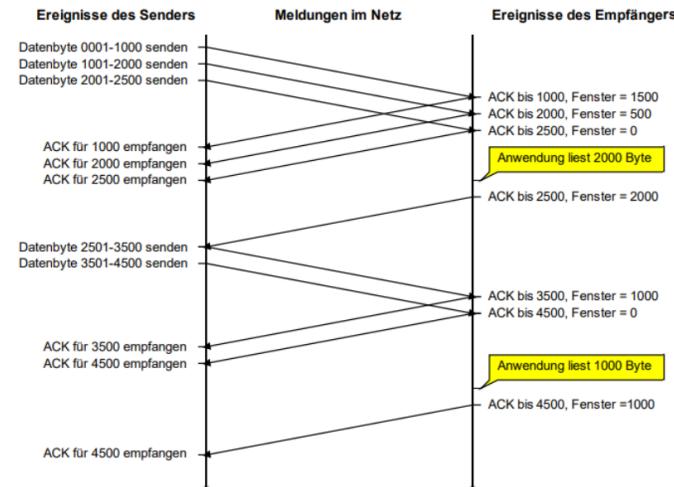
Wie gross sollten die Sende- und Empfangsbuffer gewählt werden, um eine TCP-Verbindung nicht auszubremsen?

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

RTT = Round-Trip-Time



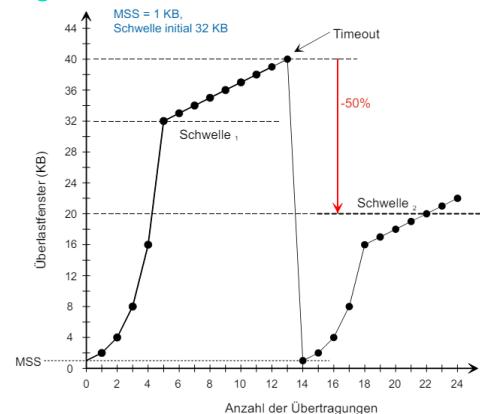
### Fluss-Steuerung bei TCP



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

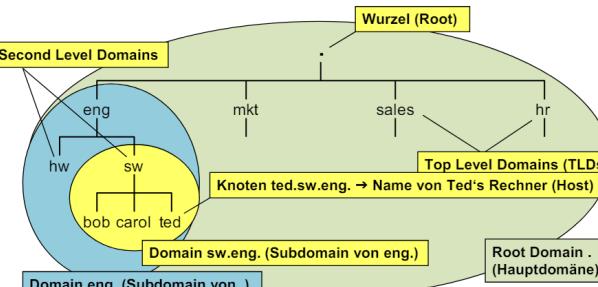
#### Übersicht Applikationsprotokolle

- 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: separater 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

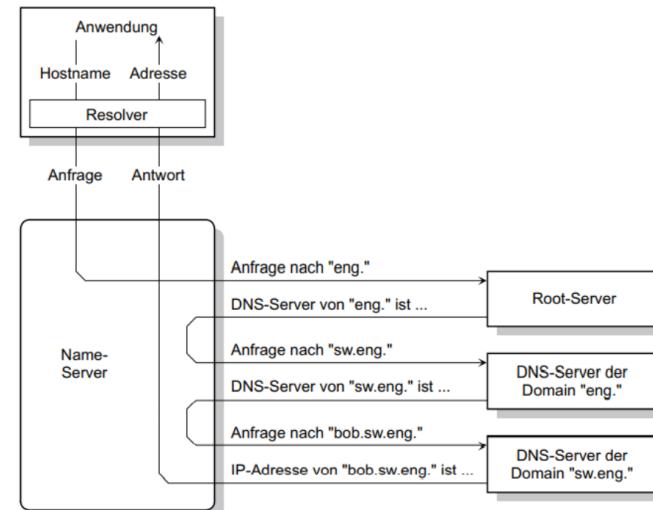
Authentisierung: Ein Server identifiziert/authentifiziert einen Client anhand des Namens, nicht anhand der IP-Adresse

#### 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)

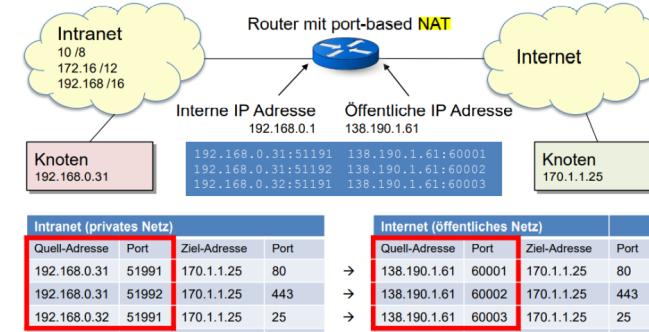
Alle Hosts im privaten Netz 192.168.0.0/8 verwenden 192.168.0.1 als Default-Gateway.

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

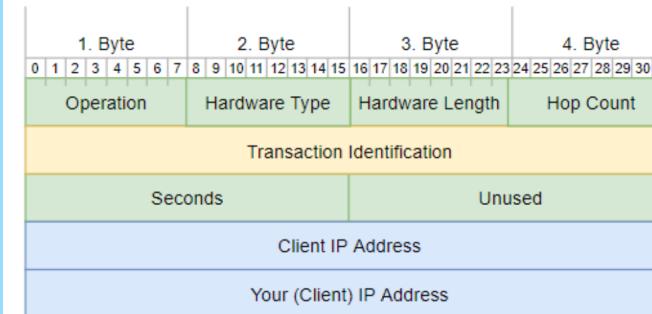
- Client verlangt eine IP-Adresse (DHCP Request)
- DHCP-Server erteilt eine freie Adresse für definierte Lease Time, oft 10 Minuten (DHCP-Response)
- Vor Ablauf der Lease Time muss der Lease (vom Client) erneuert werden
- Client, der das Netz verlässt, wird Lease nicht erneuern → Adresse wieder frei

### 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

