

Systeme II / Rechnernetze

1. Organisation, Literatur, Internet, TCP/IP-Schichtenmodell, ISO/OSI-Schichten

Christian Schindelhauer

Technische Fakultät

Rechnernetze und Telematik

Albert-Ludwigs-Universität Freiburg

Version 24.04.2017

Organisation

- Vorlesungen
 - Dienstag, 11 - 12 Uhr, Hörsaal 101-00-036
 - Donnerstag, 10 - 12 Uhr, Hörsaal 101-00-036
- Übungen
 - Dienstags und donnerstags 12-13 Uhr
- Web-Seite
 - <http://cone.informatik.uni-freiburg.de/lehre/aktuell/systeme2-ss17>
- ILIAS
 - https://ilias.uni-freiburg.de/goto.php?target=crs_772363&client_id=unifreiburg

Übungen

- Bitte in ILIAS in Ihre gewünschte Übungsgruppe eintragen
 - Innerhalb der ersten Woche werden Sie evtl. neu geordnet
- Gruppe 1 – Jan Ole von Hartz
 - Dienstag, 12-13 Uhr, Geb. 051, Hörsaal 00-006
- Gruppe 2 – Francine Wagner
 - Dienstag, 12-13 Uhr, Geb. 051 Seminarraum 00-031
- Gruppe 3 – Justin Pearse-Danker
 - Dienstag, 12-13 Uhr, Geb. 052 Seminarraum 02-017
- Gruppe 4 – Sven Köhler
 - Donnerstag, 12-13 Uhr, Geb. 051 Hörsaal 00 006
- Gruppe 5 – Leonie Feldbusch
 - Donnerstag, 12-13 Uhr, Geb. 051 Seminarraum 00-031
- Gruppe 6 – Julia Abels
 - Donnerstag, 12-13 Uhr, Geb. 051 Seminarraum 00-034

Übungsaufgaben

- Erscheinen jeden Mittwoch in ILIAS
 - Abgabe als PDF bis Montag 23.59 Uhr (GMT+1) der Folgewoche
 - Abgabe über ILIAS
 - Namenskonvention beachten:
 - <BlattNr>-<Gruppennummer>-<Matrikelnummer>.pdf
 - 01-G1-726818.pdf
- Grundlage für schriftliche Klausur
- Besprechung am Tag nach der Abgabe
 - Korrektur durch den Tutor
 - Rückgabe eine Woche nach Abgabe
- Lösungspräsentation durch die Studenten

Prüfung

- **Klausur**
 - schriftlich, 90 Minuten
- **Prüfungsanmeldung**
 - erfolgt on-line über das Campus-Online
- **Fristen beachten!**
- **Erlaubte Hilfsmittel**
 - Keine außer einer Auswahl eigener Übungsabgaben
 - Diese werden in gedruckter Form zur Klausur bereitgestellt
 - ohne Korrekturen der Tutoren
 - nur sinnvolle Abgaben
 - keine Plagiate

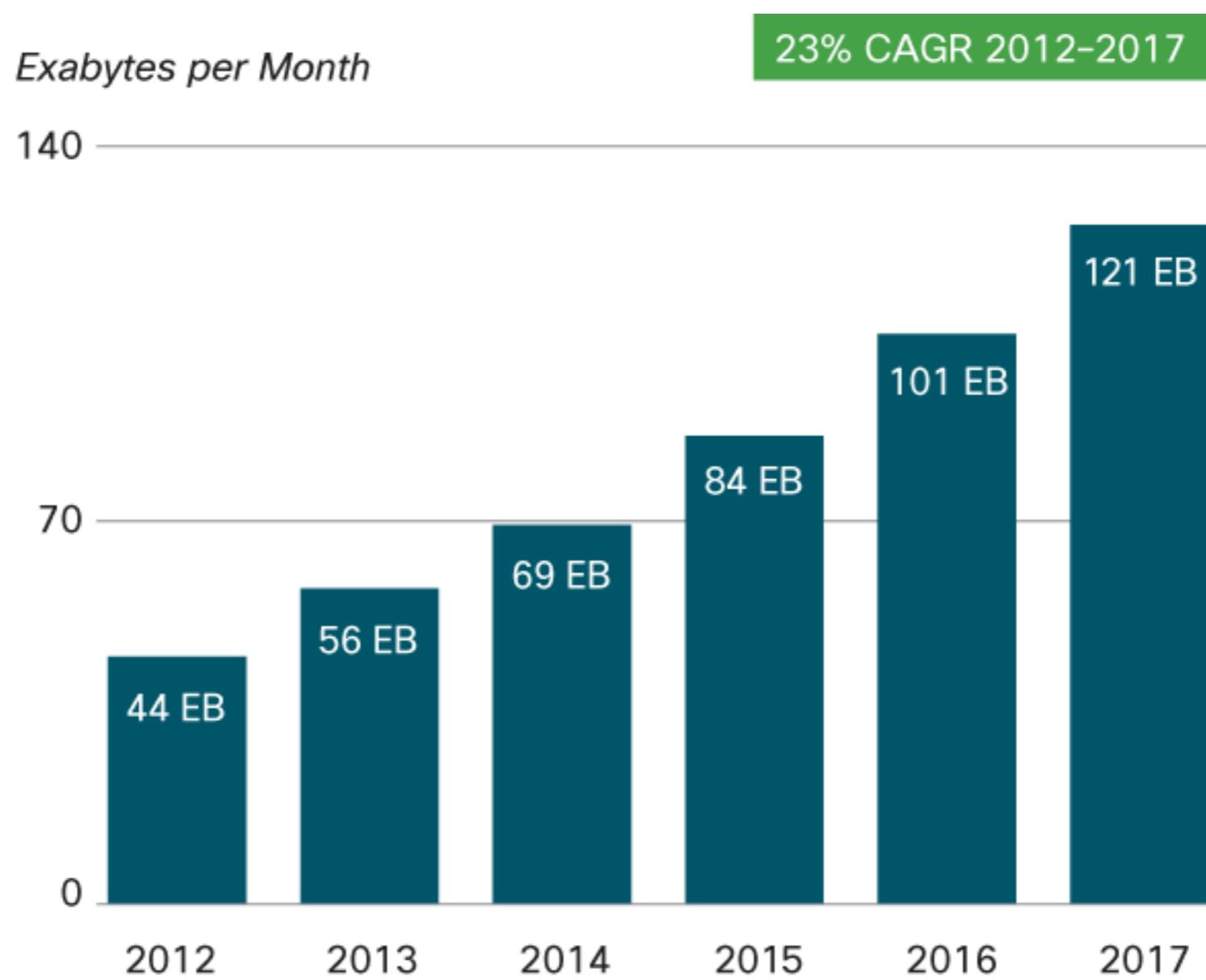
- PDF-Foliensätze
 - vor der Vorlesung auf Ilias
 - mit/ohne Notizen
- Aufzeichnung von den Vorjahren
- Literaturhinweise
 - gleich und auf der Webseite/Ilias
- Forum
 - in Ilias
 - zur Diskussion
 - sonstige Organisation

Inhalte

1. Organisation
2. Schichtenmodelle
3. Bitübertragungsschicht (Physical Layer)
4. Sicherungsschicht (Data Link Layer)
5. Mediumzugriffs-Steuerung
(Medium Access Control Sub-Layer - MAC)
6. Vermittlungsschicht (Network Layer)
7. Transportschicht (Transport Layer)
8. Anwendungsschicht (Application Layer)
9. Sicherheit

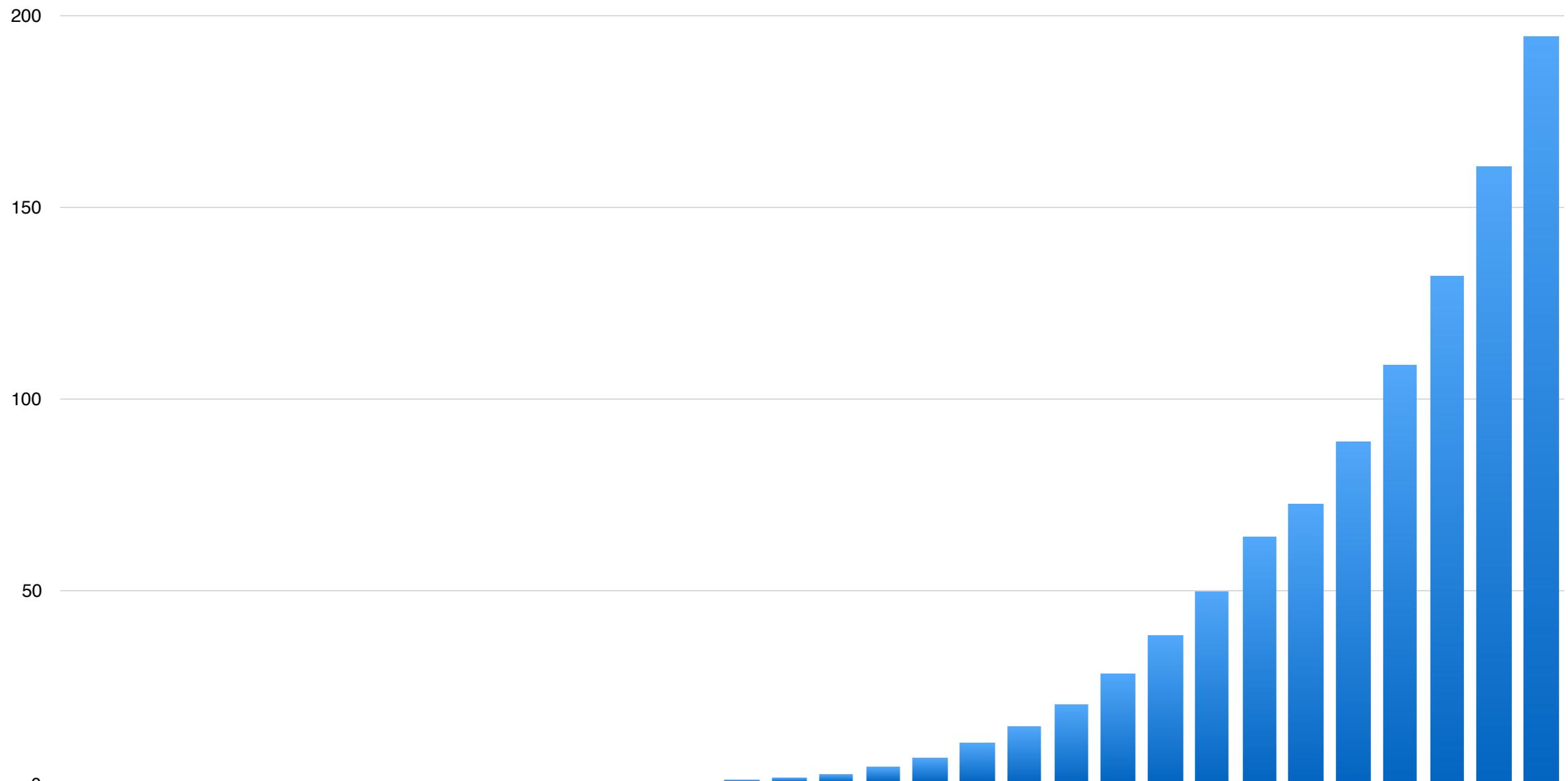
Veranstaltungen im Bereich Netzwerke

Netzwerke I	=	Systeme II	jeden Sommer	Einführung in Netzwerke Ethernet Grundlagen des Internets
Netzwerke II	=	Communication Systems	Winter	WLAN, Mobiltelefon, VoIP, u.v.a.
Vertiefung Netzwerke	z.B.	Distributed Systems Peer-to-Peer-Netzwerke Network Algorithms	Sommer/ Winter	
Verwandtes		Graphentheorie	Winter Sommer	
Praktika, Projekte, Teamprojekte	z.B.	Wireless Sensor Systems	jedes Semester	
Seminare Bachelor-/ Master- Arbeiten		je nach Lehrstuhl, individuell	jedes Semester	forschungsnahe Arbeit



Internet Verkehr

EB/month



Source: CISCO VNI 2009,2010,2012,2014,2016

Internet Verkehr

Von Kilo bis Yotta

■ Datenmengen

- 1 Byte = 1 B = 8 Bit = 8b
- 1 kilobyte = 1 kB = 1000 Bytes
- 1 megabyte = 1 MB = 1000 kB = 1 E6 Bytes
- 1 gigabyte = 1 GB = 1000 MB = 1 E9 Bytes
- 1 terabyte = 1 TB = 1000 GB = 1 E12 Bytes
- 1 petabyte = 1 PB = 1000 TB = 1 E15 Bytes
- 1 exabyte = 1 EB = 1000 PB = 1 E18 Bytes
- 1 zettabyte = 1 ZB = 1000 EB = 1 E21 Bytes
- 1 yottabyte = 1 YB = 1000 ZB = 1 E24 Bytes

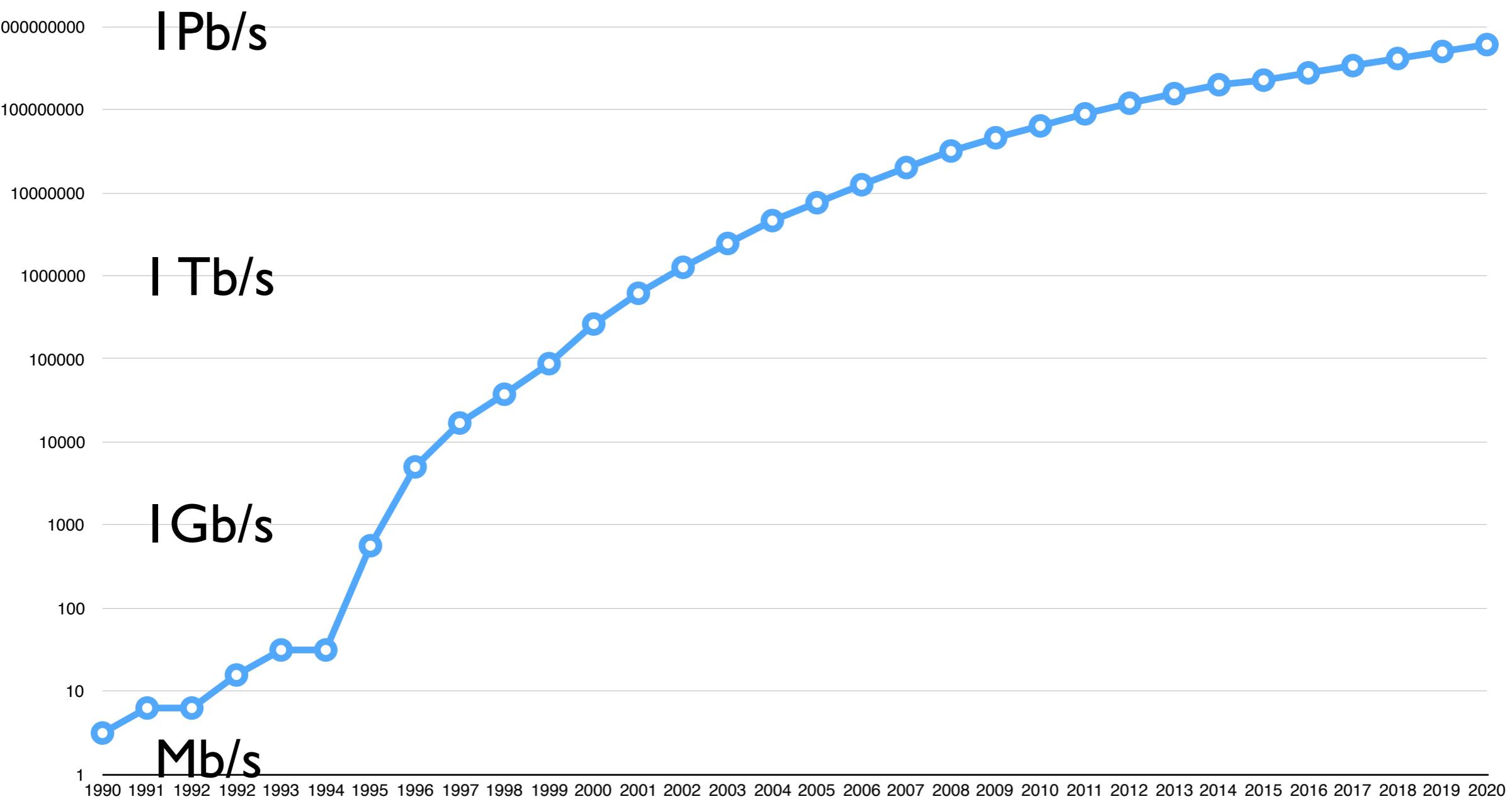
■ Speichergrößen

- 1 Byte = 1 B = 8 Bit = 8b
- 1 kibibyte = 1 kB = 1024 Bytes
- 1 mebibyte = 1 MiB = 1024 kiB = 1.04 E6 Byte
- 1 gibibyte = 1 GiB = 1024 MiB = 1.07 E9 Bytes
- 1 tebibyte = 1 TiB = 1024 GiB = 1.10 E12 Bytes
- 1 pebibyte = 1 PiB = 1024 TiB = 1.12 E15 Bytes
- 1 exbibyte = 1 EiB = 1024 PiB = 1.15 E18 Bytes
- 1 zebibyte = 1 ZiB = 1024 EiB = 1.18 E21 Bytes
- 1 yobibyte = 1 YiB = 1024 ZiB = 1.21 E24 Bytes

Datenraten und Speicherplatz

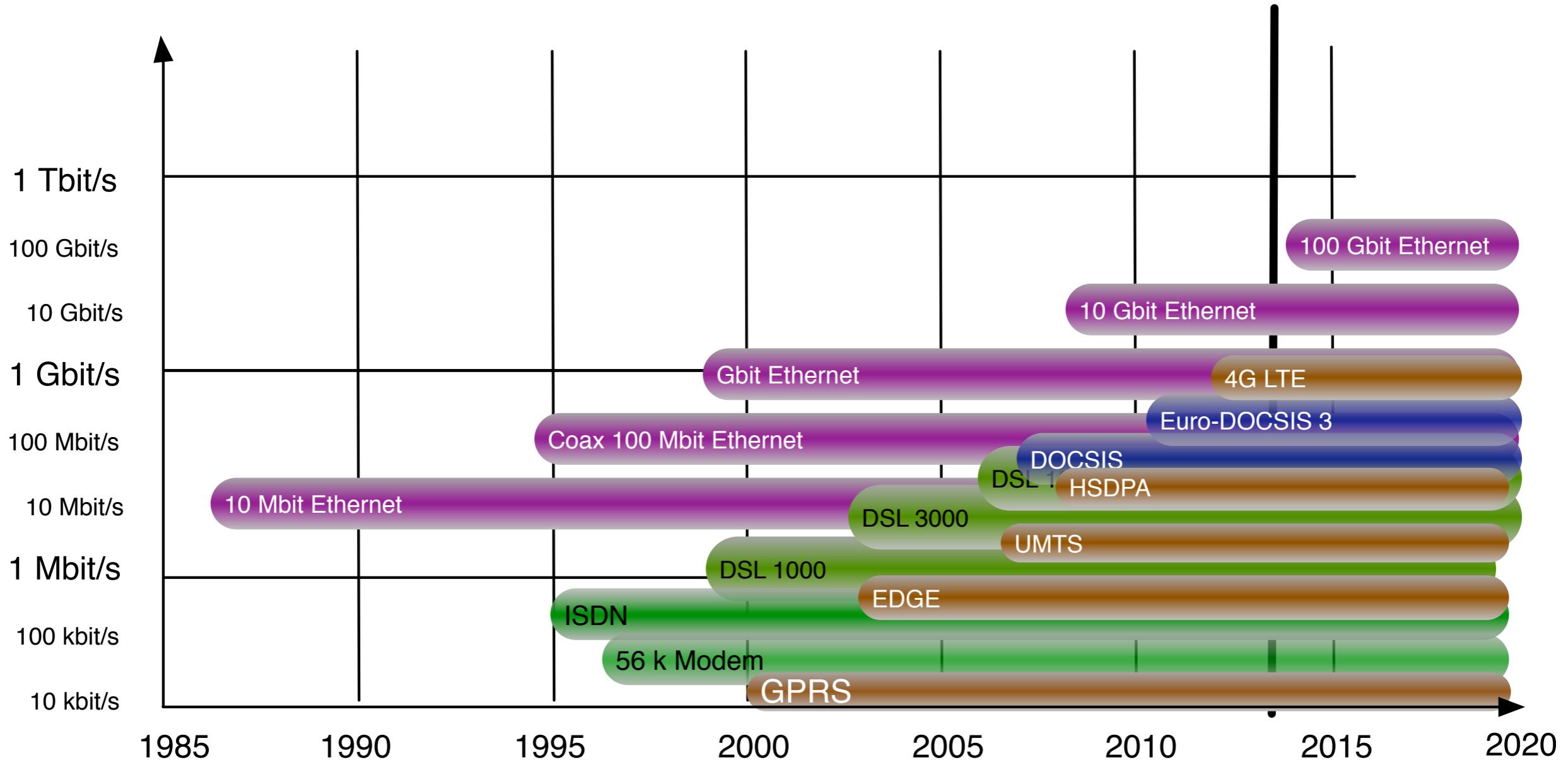
- Datenraten
 - werden in bit/s angegeben
 - oder Baud = Symbole/s
 - $\text{kbit/s} = 10^3 \text{ Bit/s}$, etc
- Speicher wird in Byte = 8 Bit angegeben
 - Größe meist in kibibyte, mibibyte
 - wird aber (fälschlich) als kilobyte, megabyte angegeben
- 1 Mb/s
 - $= 0,128 \text{ MB/s} = 7,68 \text{ MB/min} = 460 \text{ MB/h}$
 - $= 11\text{GB/d} = 330 \text{ GB/mo} = 3,9 \text{ TB/y}$
- 69 EB/mo
 - $= 27 \text{ Tb/s}$

Internet Verkehr



Source: CISCO VNI 2009,2010,2012,2014,2016

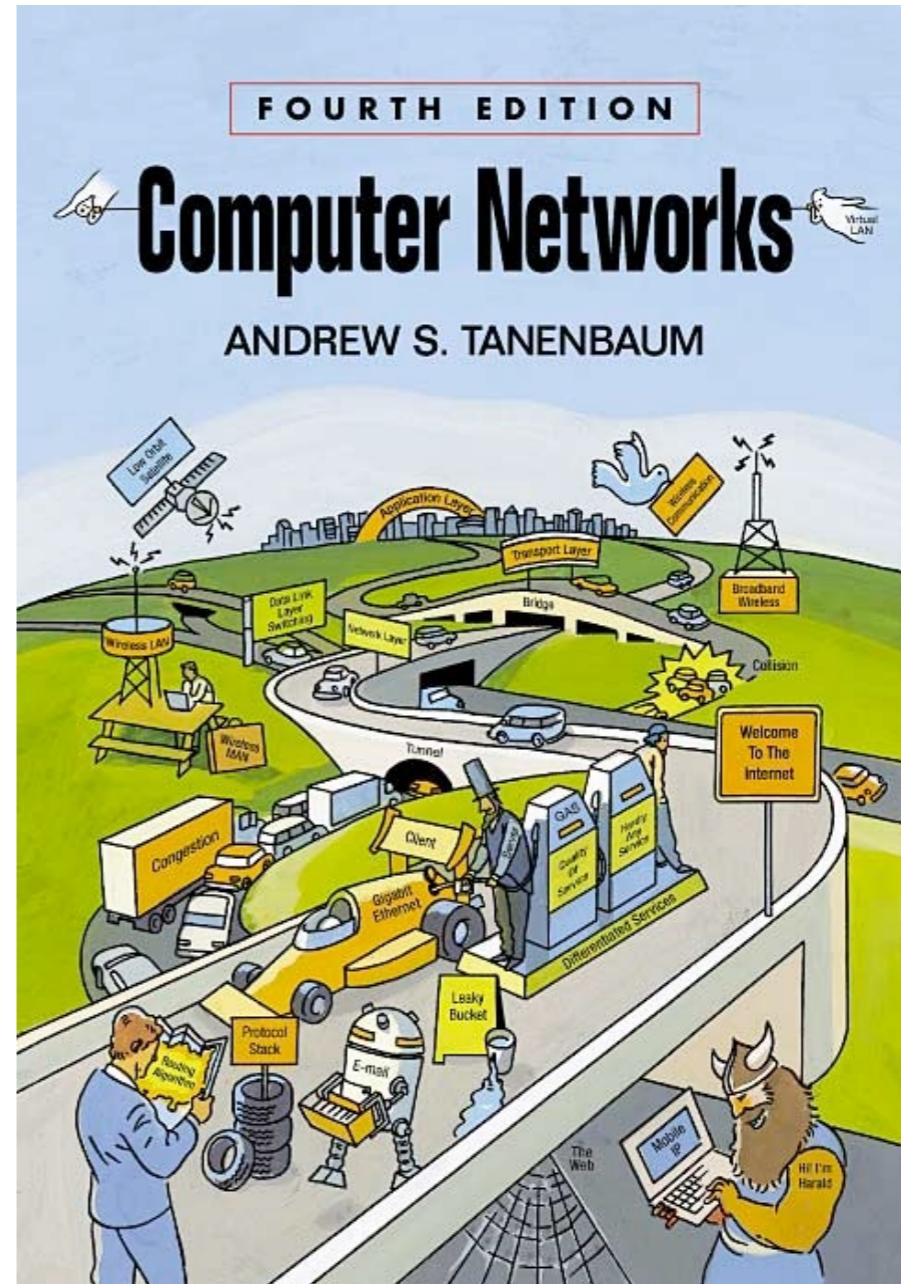
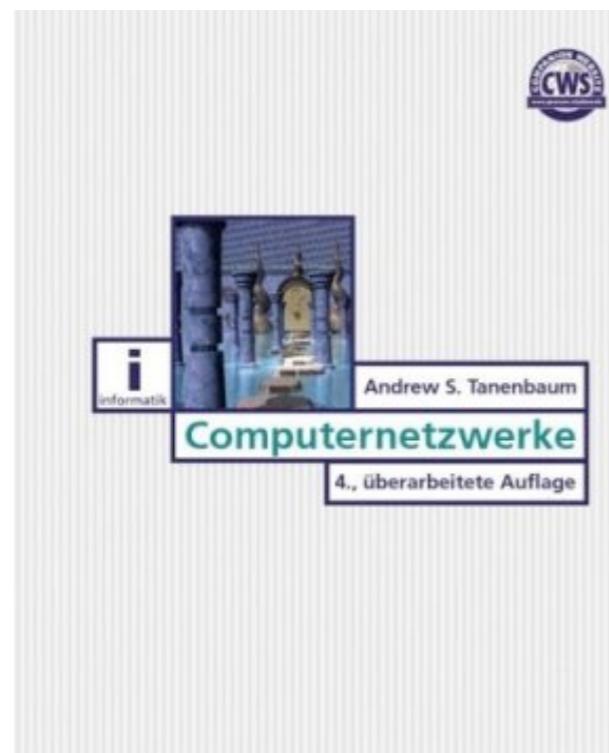
Die letzte Meile



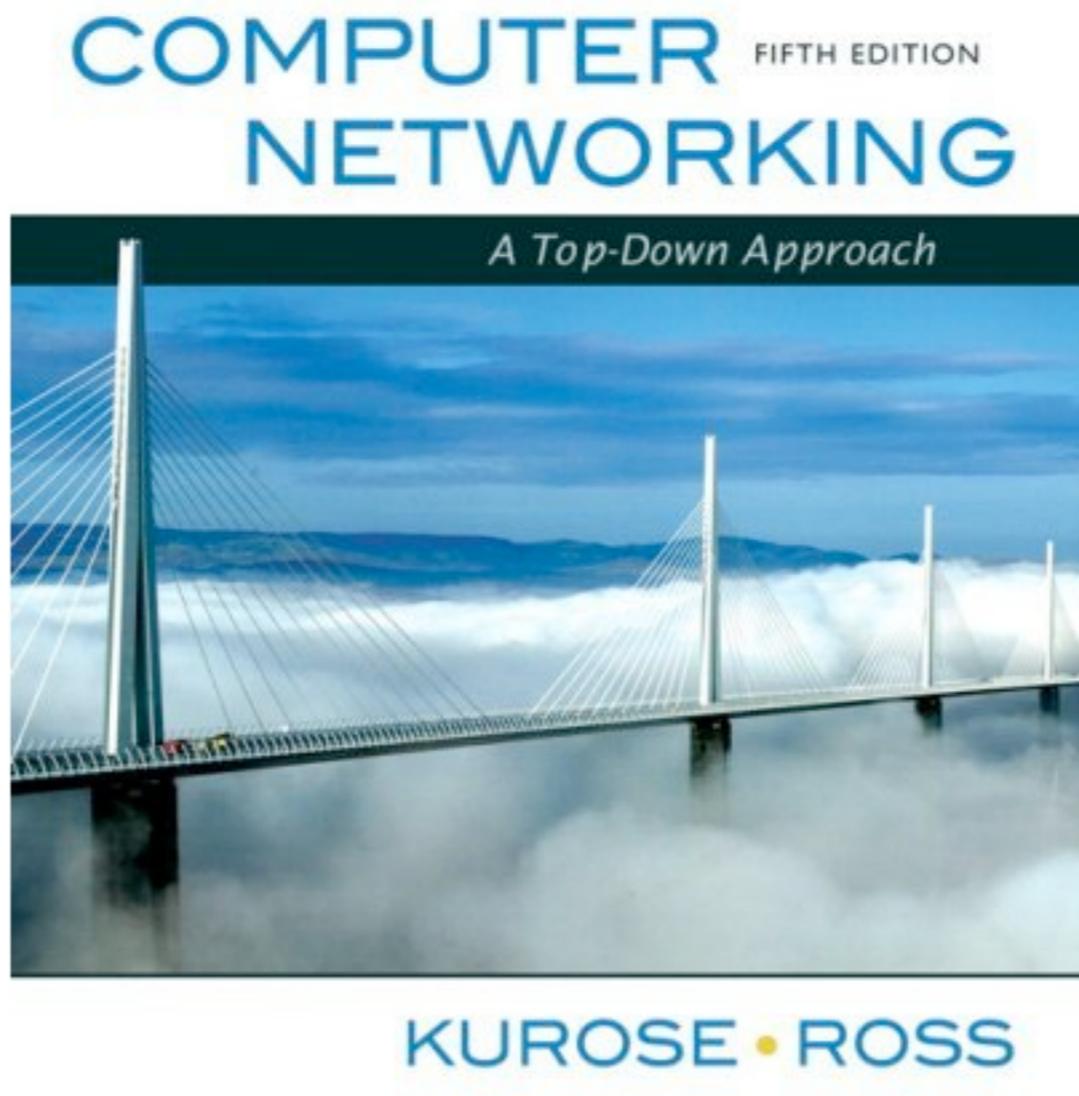
Literatur (I)

- Das Buch Nr. 1 zur Vorlesung

- Computer Networks, Andrew S. Tanenbaum (Prentice Hall)
- auf Deutsch:
Computernetzwerke
(Taschenbuch)

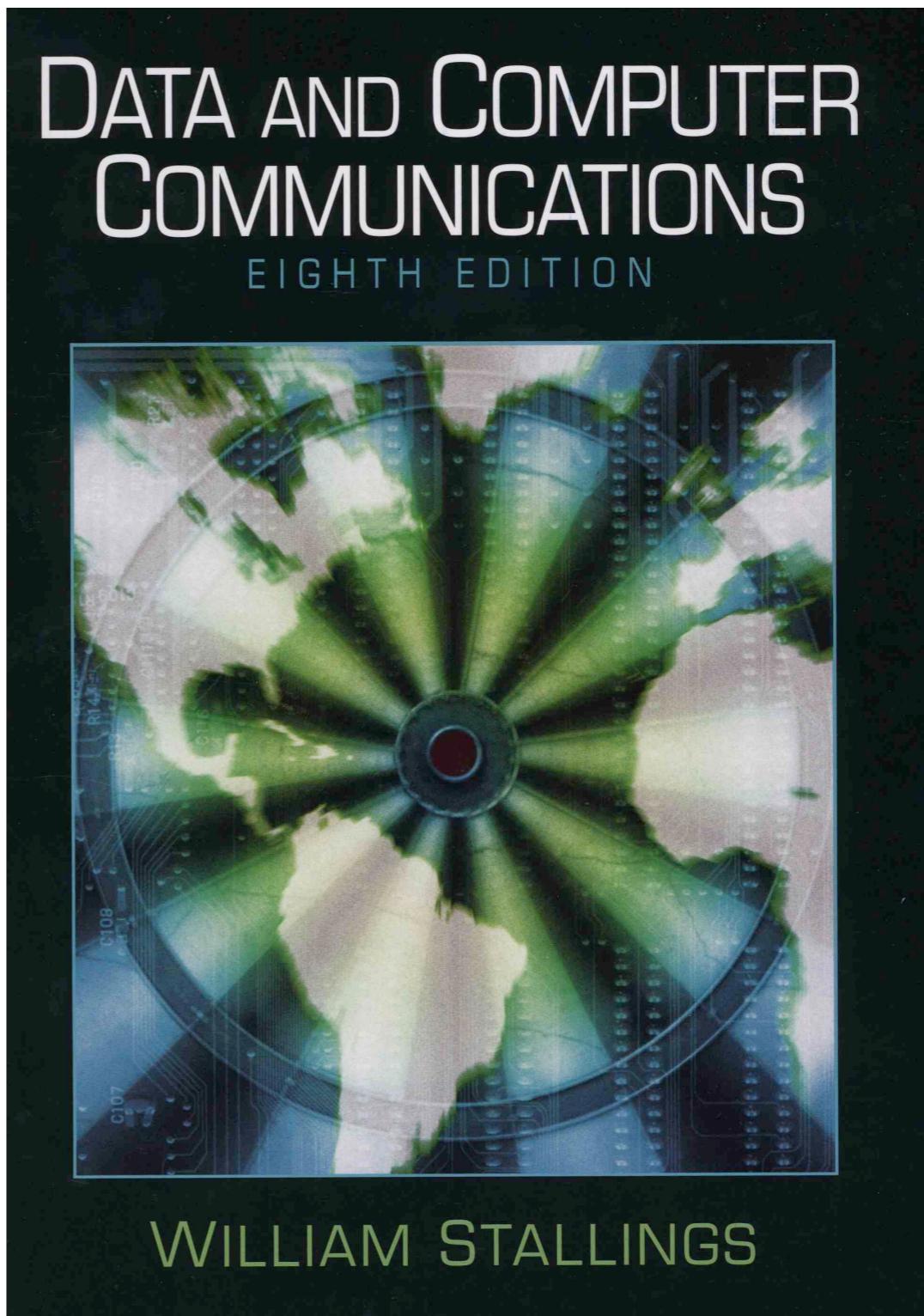


- Das Buch Nr. 2 zur Vorlesung:
 - Computer Networking - A Top-Down Approach Featuring the Internet, James F. Kurose, Keith W. Ross, Prentice Hall

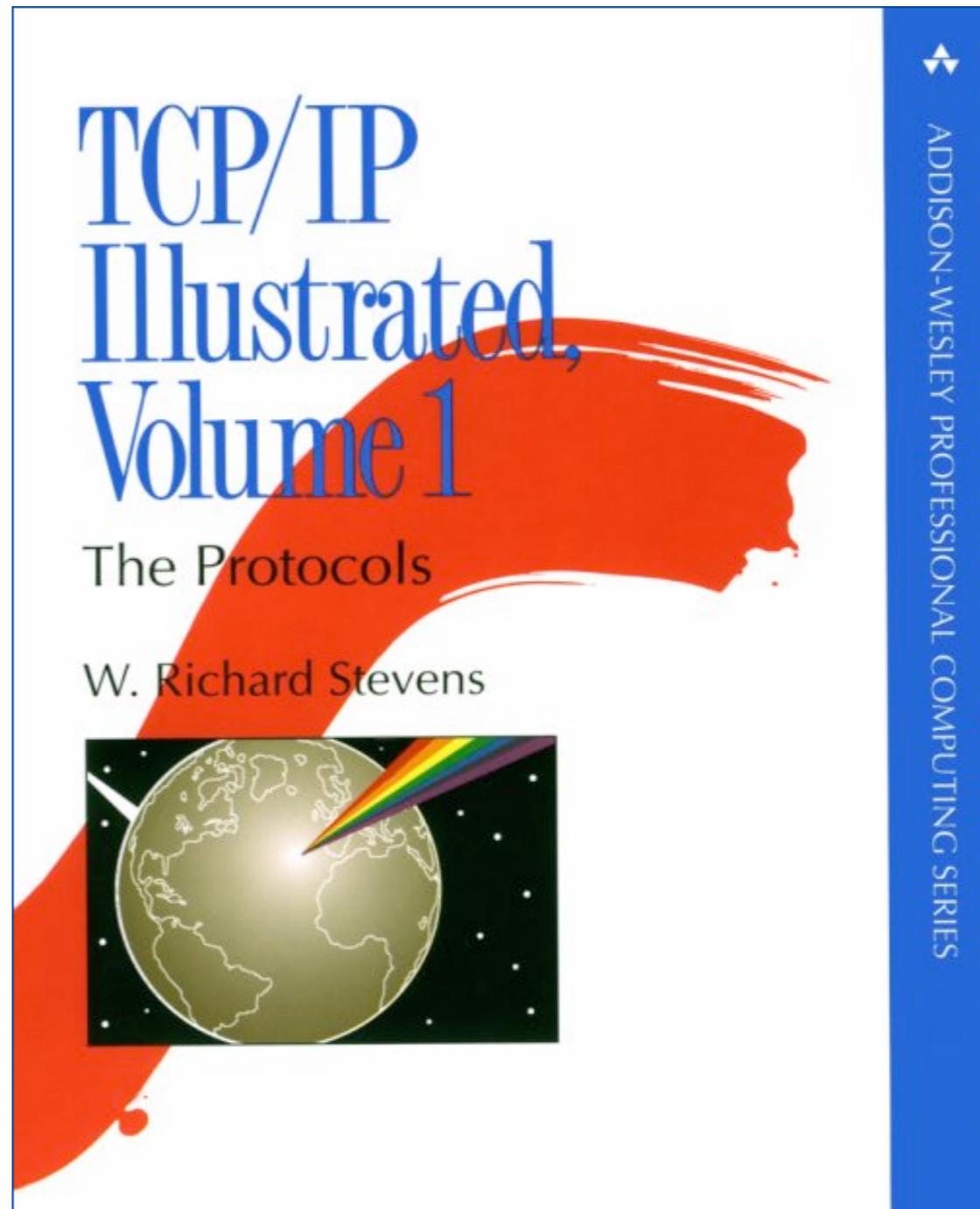


- Buch Nr. 3:

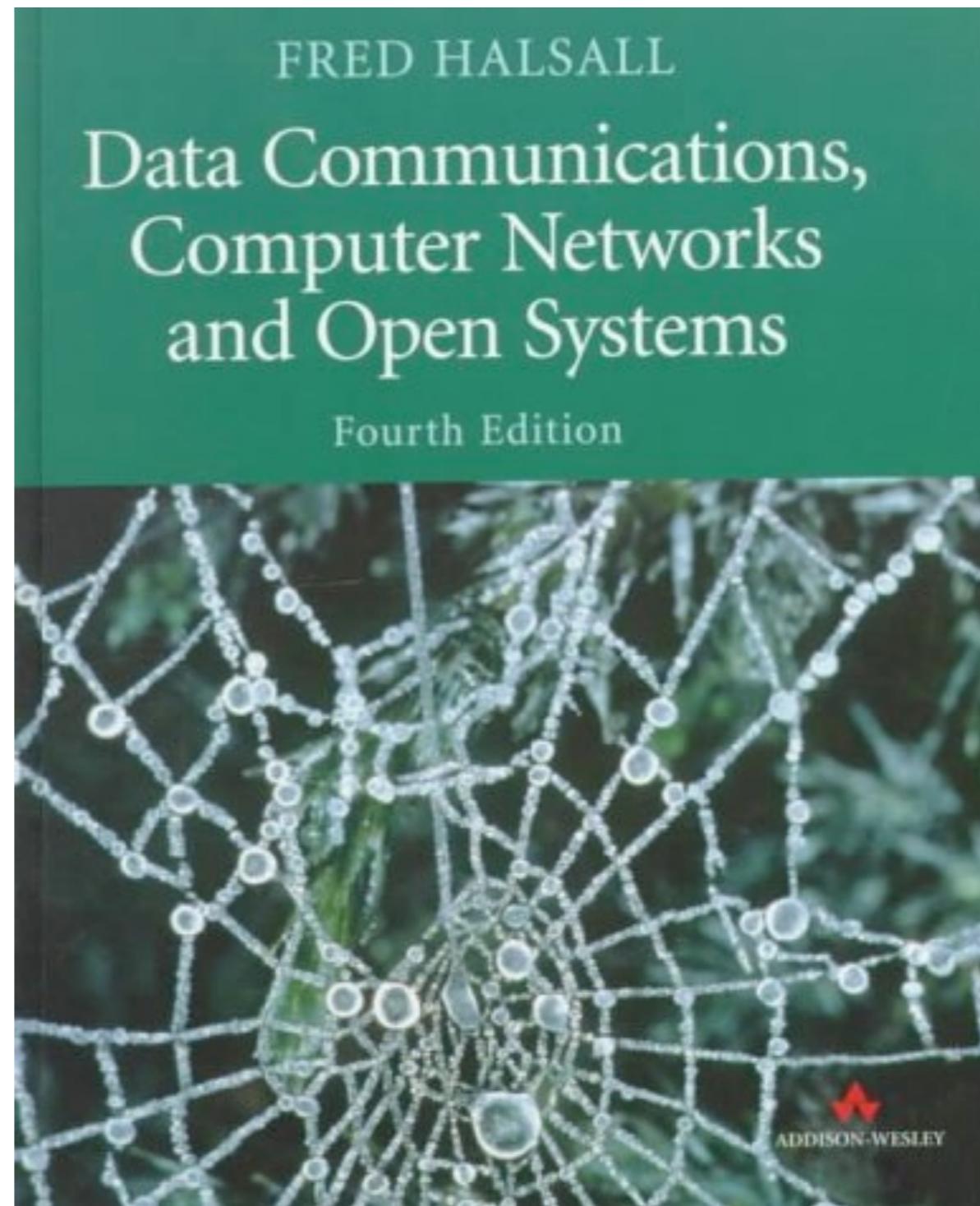
- Data and computer Communications
- William Stallings
- Pearson, Prentice-Hall, 2007



- Zur Vertiefung:
 - TCP/IP Illustrated,
Volume - The Protocols,
W. Richard Stevens,
Addison-Wesley



- Fred Halsal, Data Communications, Computer Networks and Open Systems, Addison-Wesley, 1995



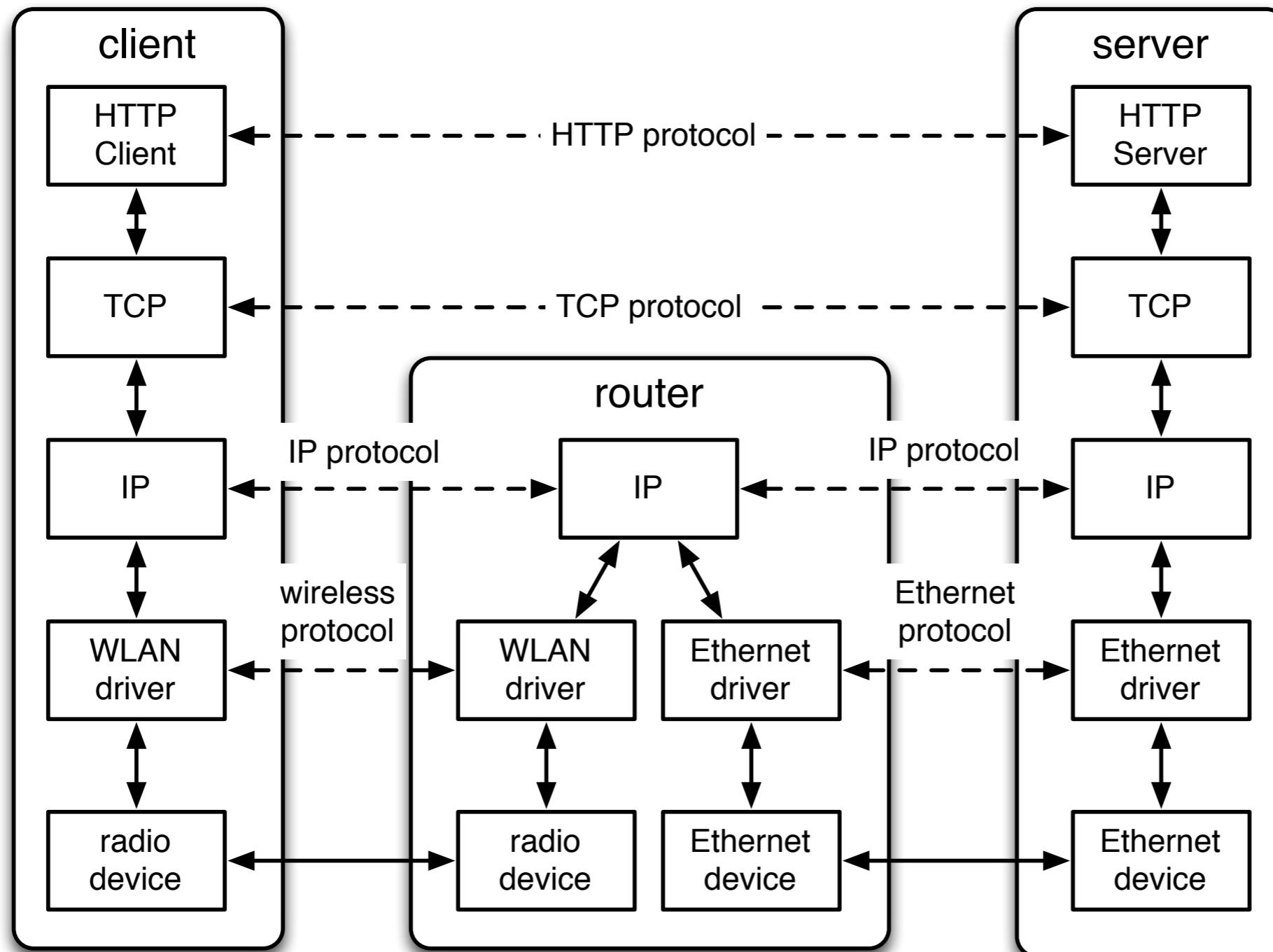
Die Schichtung des Internets

Anwendung	Application	HTTP, SMTP (E-Mail), ...
Transport	Transport	TCP (Transmission Control Protocol) UDP (User Datagram Protocol)
Vermittlung	Network	IP (Internet Protocol) + ICMP (Internet Control Message Protocol) + IGMP (Internet Group Management Protocol)
Verbindung	Host-to-Network	LAN (z.B. Ethernet, WLAN 802.11, etc.)

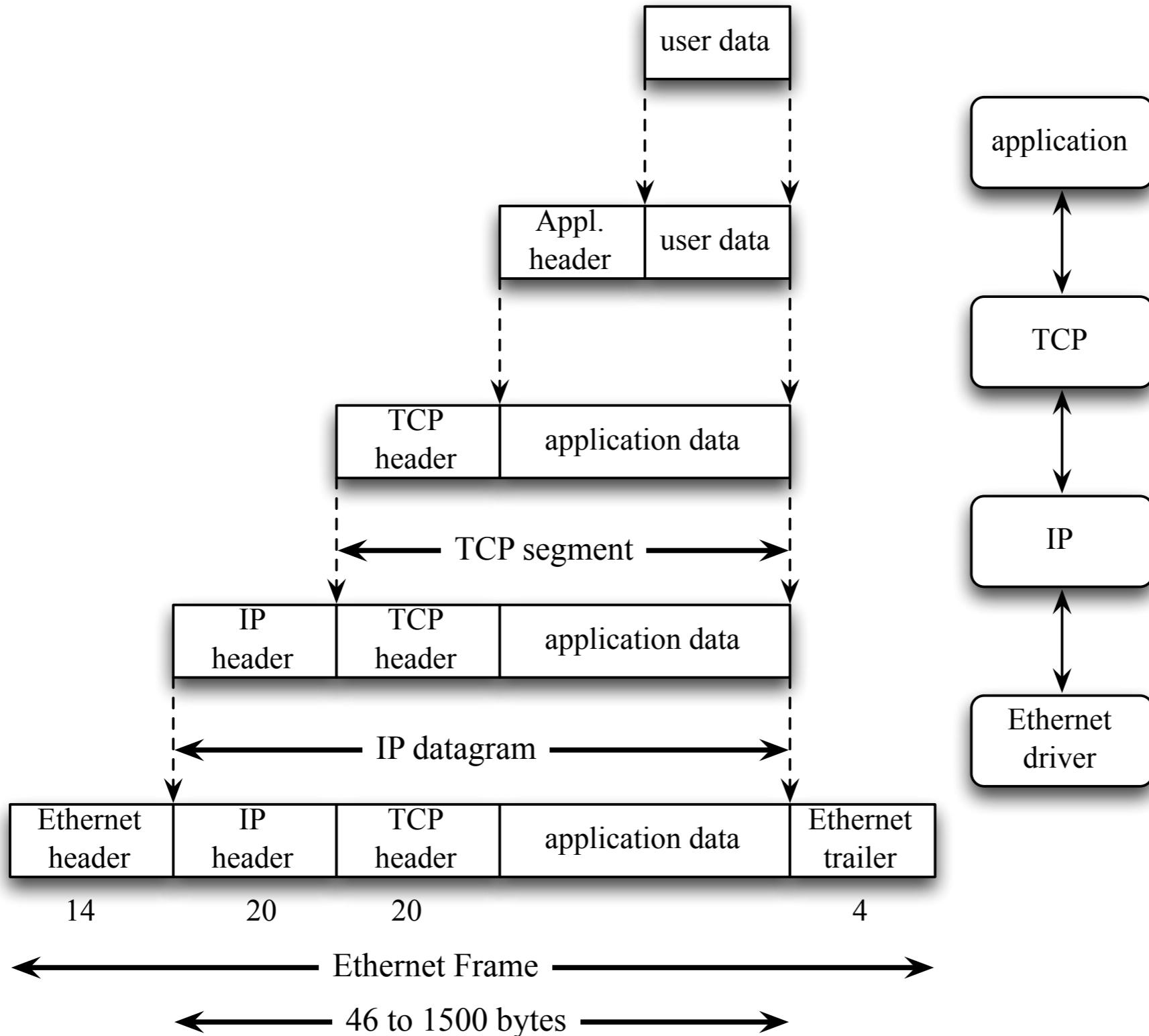
Internet-Schichtenmodell

- 1. Host-to-Network
 - nicht spezifiziert, hängt vom LAN ab, z.B. Ethernet, WLAN 802.11b, PPP, DSL
- 2. Vermittlungsschicht (IP - Internet Protokoll)
 - Spezielles Paketformat und Protokoll
 - Paketweiterleitung
 - Routenermittlung
- 3. Transportschicht
 - TCP (Transport Control Protocol)
 - zuverlässiger bidirektonaler Byte-Strom-Übertragungsdienst
 - Fragmentierung, Flusskontrolle, Multiplexing
 - UDP (User Datagram Protocol)
 - Paketübergabe an IP
 - unzuverlässig, keine Flusskontrolle
- 4. Anwendungsschicht
 - zahlreiche Dienste wie SMTP, HTTP, NNTP, FTP, ...

Beispiel zum Zusammenspiel der Schichten

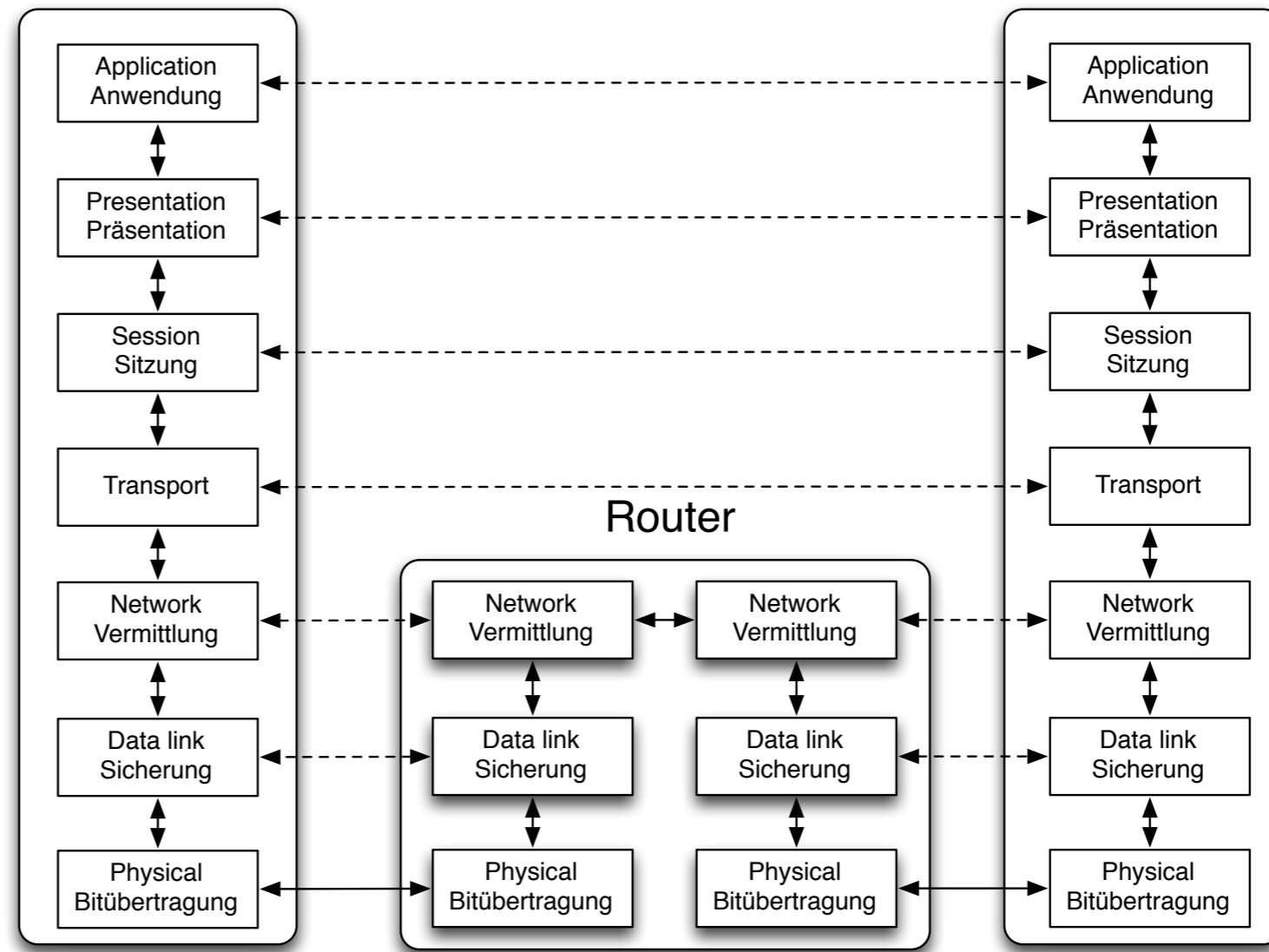


Datenkapselung



Das ISO/OSI Referenzmodell

- 7. Anwendung (Application)
 - Datenübertragung, E-Mail, Terminal, Remote login
- 6. Darstellung (Presentation)
 - Systemabhängige Darstellung der Daten (EBCDIC/ASCII)
- 5. Sitzung (Session)
 - Aufbau, Ende, Wiederaufsetzpunkte
- 4. Transport (Transport)
 - Segmentierung, Stauvermeidung
- 3. Vermittlung (Network)
 - Routing
- 2. Sicherung (Data Link)
 - Prüfsummen, Flusskontrolle
- 1. Bitübertragung (Physical)
 - Mechanische, elektrische Hilfsmittel



- Aküfi
 - ISO: International Standards Organisation
 - OSI: Open Systems Interconnections
- **1. Bitübertragung (Physical)**
 - Übertragung der reinen Bits
 - Technologie (elektronisch/Licht)
 - Physikalische Details (Wellenlänge, Modulation)

2. Sicherung (Data Link Layer)

- Bereinigung von Übertragungsfehler
- Daten werden in Frames unterteilt mit Kontrollinformation
 - (z.B. Checksum)
- Bestätigungsframes werden zurückgesendet
- Löschen von Duplikaten
- Ausgleich schneller Sender - langsamer Empfänger
(Flusssteuerung)
- Lösung von Problemen beim Broadcasting
 - Zugriff auf gemeinsames Medium = Mediumzugriff
(medium access control = MAC)

3. Vermittlungsschicht

- Packetweiterleitung (packet forwarding)
- Routenermittlung/Wegewahl der Pakete (route detection)
- Kontrolle von Flaschenhälzen (bottleneck) in der Wegewahl
- Abrechnung der Pakete (Abrechnungssystem)

4. Transportschicht

- Unterteilung der Daten aus der Sitzungsschicht in kleinere Einheiten (Pakete)
- In der Regel Erstellung **einer** Transportverbindung für jede anfallende Verbindung
- Möglicherweise auch **mehrere** Transportverbindungen zur Durchsatzoptimierung
- Art der Verbindung
 - fehlerfrei, Punkt-zu-punkt (z.B. TCP)
 - fehlerbehaftet, Unidirektional (z.B. UDP)
 - Multicasting (einer an viele)
 - Broadcasting (einer an alle)
- Multiplexing: Zu welcher Verbindung gehört dieses Paket
- Flusskontrolle: Wieviele Pakete können/sollen versendet werden (ohne das Netzwerk zu überfordern)

5. Sitzungsschicht

- Festlegung der Sitzungsart, z.B.
 - Dateitransfer, Einloggen in ein entferntes System
- Dialogkontrolle
 - Falls Kommunikation immer nur abwechselnd in einer Richtung geht, regelt die Richtung die Sitzungsschicht
- Token Management
 - Falls Operationen nicht zur gleichen Zeit auf beiden Seiten der Verbindungen möglich sind, verhindert dies die Sitzungsschicht
- Synchronisation
 - Checkpoints zur Wiederaufnahme abgebrochener Operationen (z.B. Filetransfer)

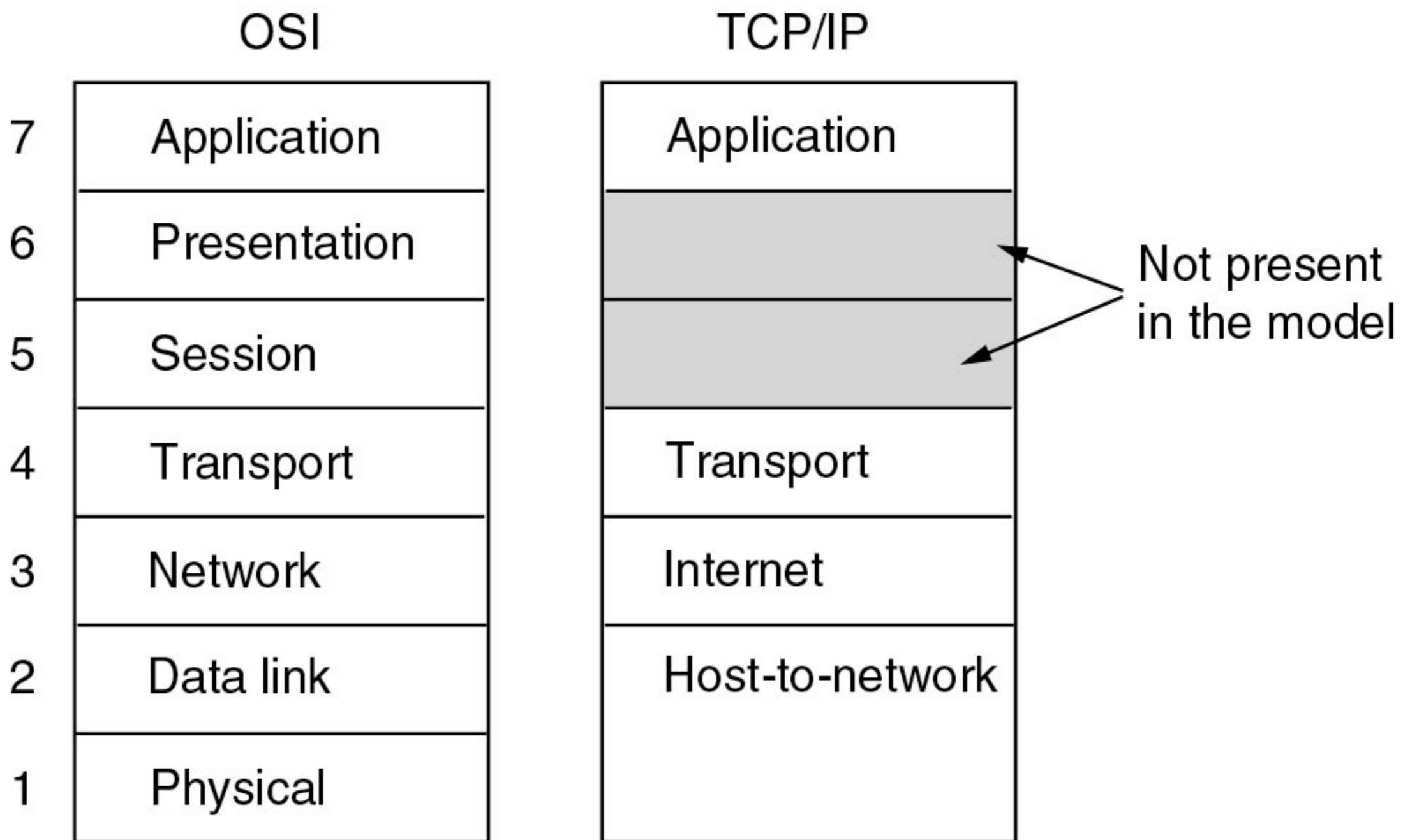
6. Präsentationsschicht

- Anpassung von Kodierungen,
- z.B. Zeichensätze, Namen, Addressfelder, Formulare, etc.

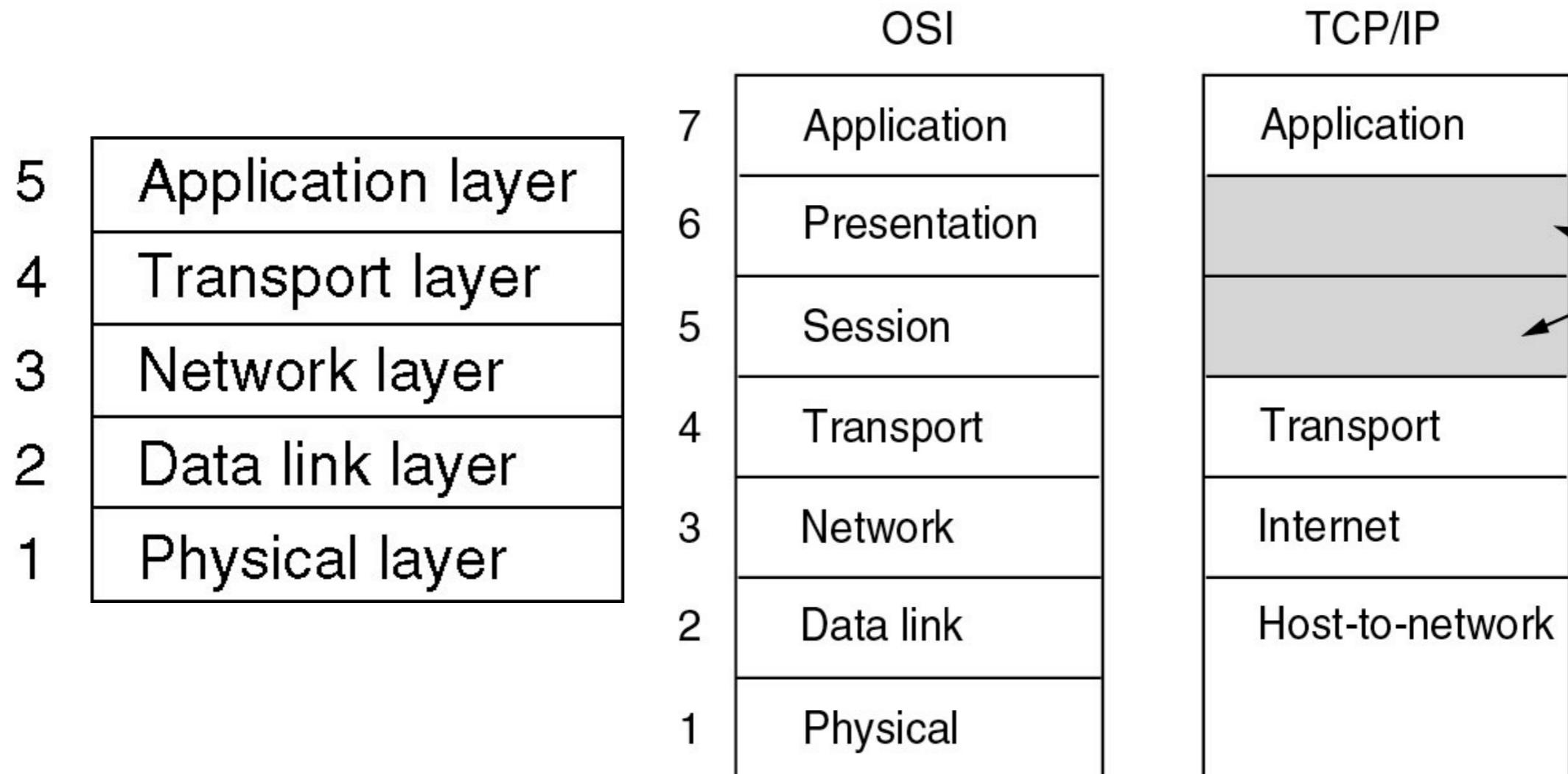
7. Anwendungsschicht

- Große Vielfalt aller möglichen Funktionen, z.B.
 - Virtuelle Terminals, Filetransfer, E-mail, Online-Video, Twitter, Radio-Streams, Internet-Telefonie, Online-Games ...

OSI versus TCP/IP



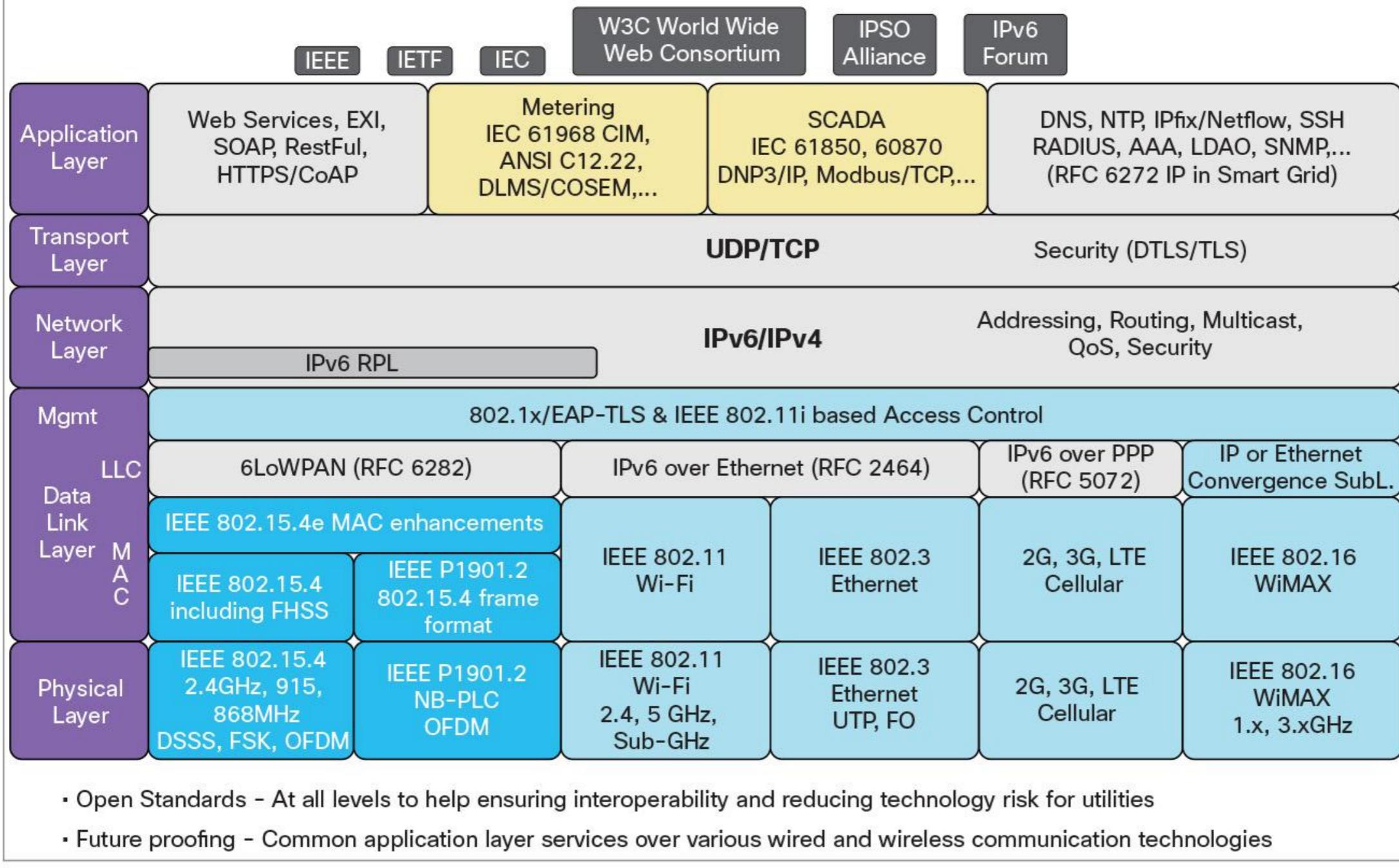
Hybrides Modell



(Aus Tanenbaum)

Beispiel: Smart Grid mit IPv6

Open Standards Reference Model



Source: Cisco

http://www.cisco.com/c/dam/en_us/solutions/industries/docs/energy/ip_arch_sg_wp.pdf

Systeme II

1. Organisation, Literatur, Internet, TCP/IP-Schichtenmodell, ISO/OSI-Schichten

Christian Schindelhauer

Technische Fakultät

Rechnernetze und Telematik

Albert-Ludwigs-Universität Freiburg

Systeme II

2. Die physikalische Schicht

Christian Schindelhauer

Technische Fakultät

Rechnernetze und Telematik

Albert-Ludwigs-Universität Freiburg

Version 26.04.2017

■ ISO-Definition

- Die Bitübertragungsschicht definiert
 - mechanische
 - elektrische
 - funktionale und
 - prozedurale
- Eigenschaften um eine physikalische Verbindung
 - aufzubauen,
 - aufrecht zu erhalten und
 - zu beenden.

Signale, Daten und Information

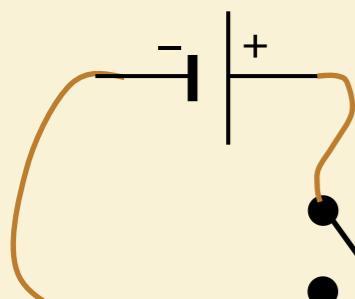
- **Information**
 - Menschliche Interpretation,
 - z.B. schönes Wetter
- **Daten**
 - Formale Präsentation,
 - z.B. 8 Grad Celsius, Niederschlagsmenge 0cm, Wolkenbedeckung 40%
- **Signal**
 - Repräsentation von Daten durch physikalische Variablen,
 - z.B. Stromfluss durch Thermosensor, Videosignale aus Kamera
 - Beispiele für Signale:
 - Strom, Spannung
 - In der digitalen Welt repräsentieren Signale Bits

- Leitungsgebundene Übertragungsmedien
 - Kupferdraht – Twisted Pair
 - Kupferdraht – Koaxialkabel
 - Glasfaser
- Drahtlose Übertragung
 - Funkübertragung
 - Mikrowellenübertragung
 - Infrarot
 - Lichtwellen

Die einfachste Bitübertragung

- Bit 1: Strom an
- Bit 0: Strom aus

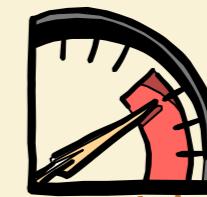
Schicht 1:
Bit zu Spannung



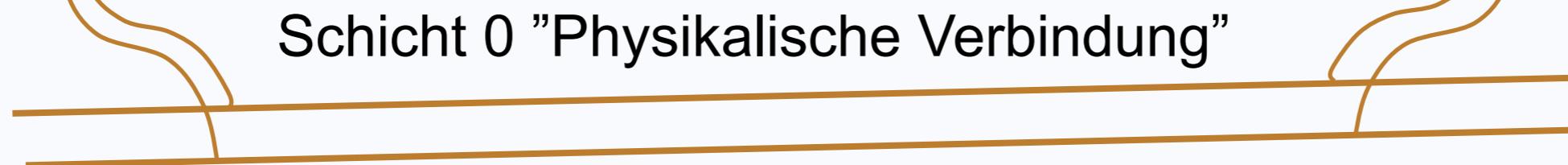
Bit=1: Schalter zu
Bit=0: Schalter auf

Schicht 1:
Spannung zu Bit

Spannung: Bit 1
Keine Spannung: Bit 0



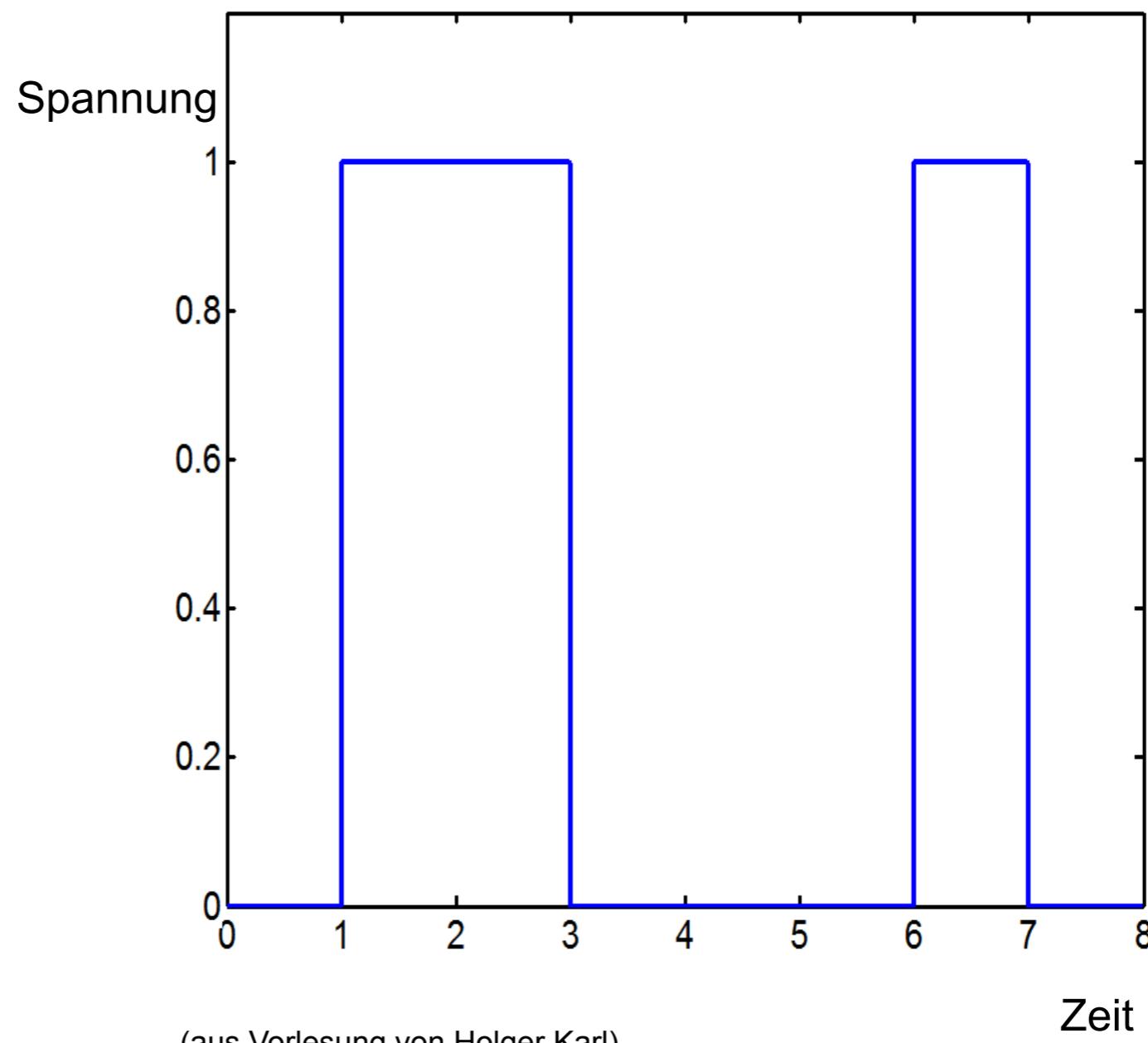
Schicht 0 "Physikalische Verbindung"



(aus Vorlesung von Holger Karl)

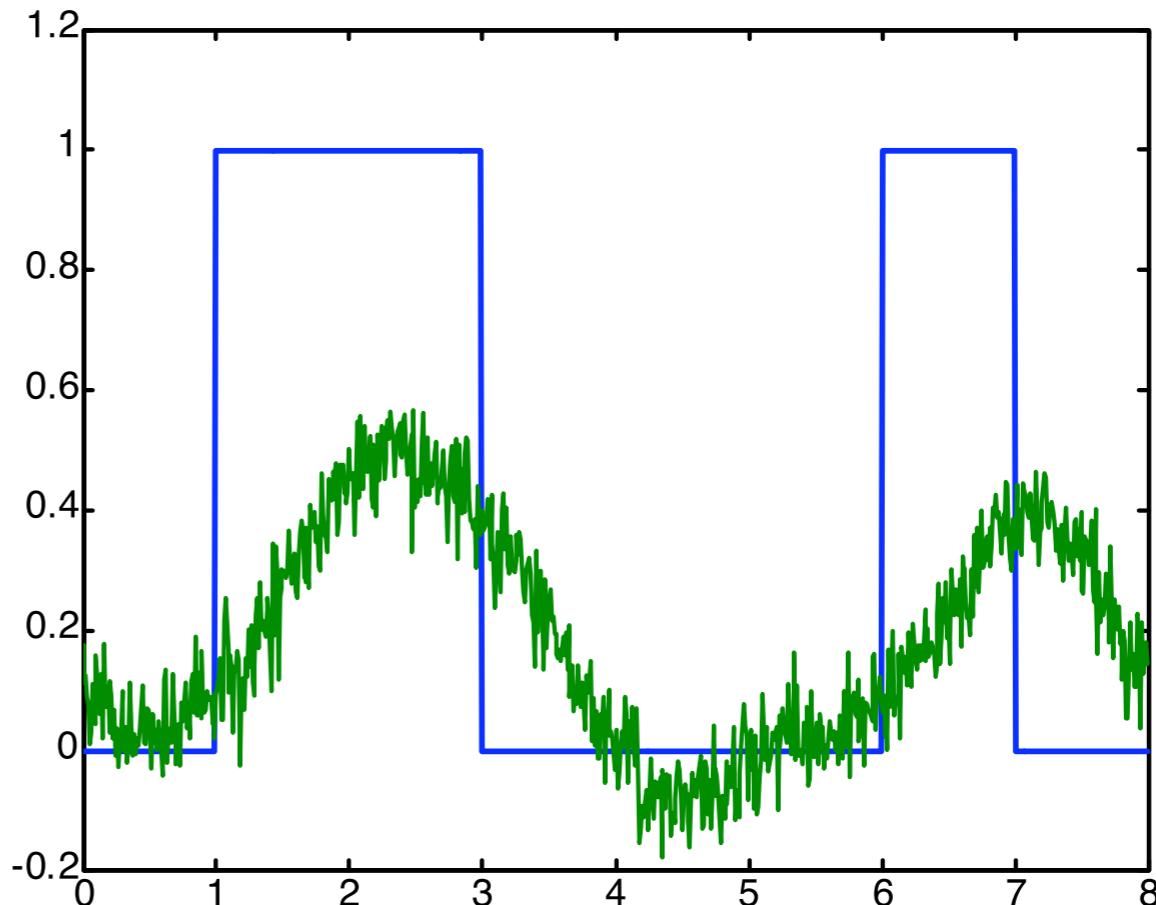
Übertragung eines Buchstabens: “b”

- Zeichen “b” benötigt mehrere Bits
 - z.B. ASCII code of “b” als Binärzahl
01100010
- Spannungsverlauf:



Was kommt an?

- Übertrieben schlechter Empfang
- Was passiert hier?



5 Gründe für den schlechten Empfang

-
1. Allgemeine Dämpfung
 2. Frequenzverlust
 3. Frequenzabhängige Dämpfung
 4. Störung und Verzerrung
 5. Rauschen

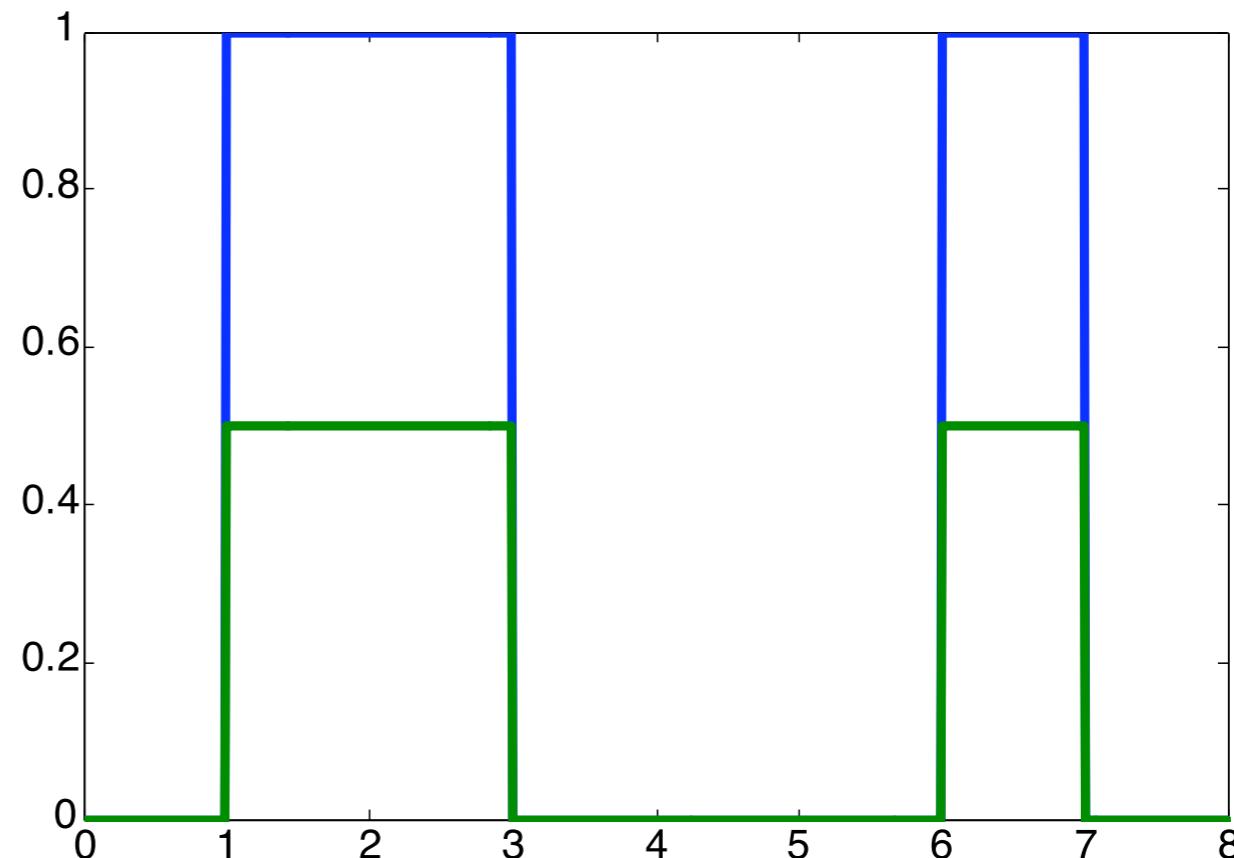
1. Signale werden gedämpft

- Dämpfung α (attenuation)
 - Verhältnis von Sendeenergie P_1 zu Empfangsenergie P_0
 - Bei starker Dämpfung erreicht wenig Energie dem Empfänger
- Dämpfung hängt ab von
 - der Art des Mediums
 - Abstand zwischen Sender und Empfänger
 - ... anderen Faktoren
- Angegeben in deziBel

$$\log_{10} \frac{P_1}{P_0} \quad (\text{in Bel})$$

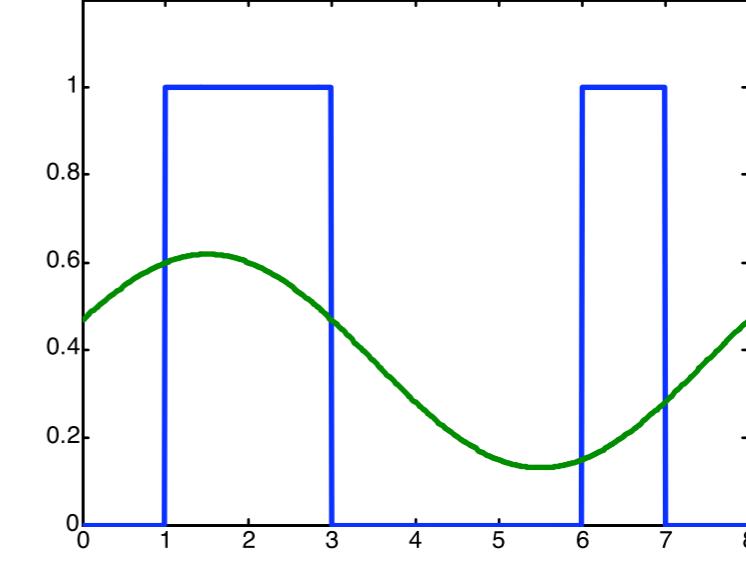
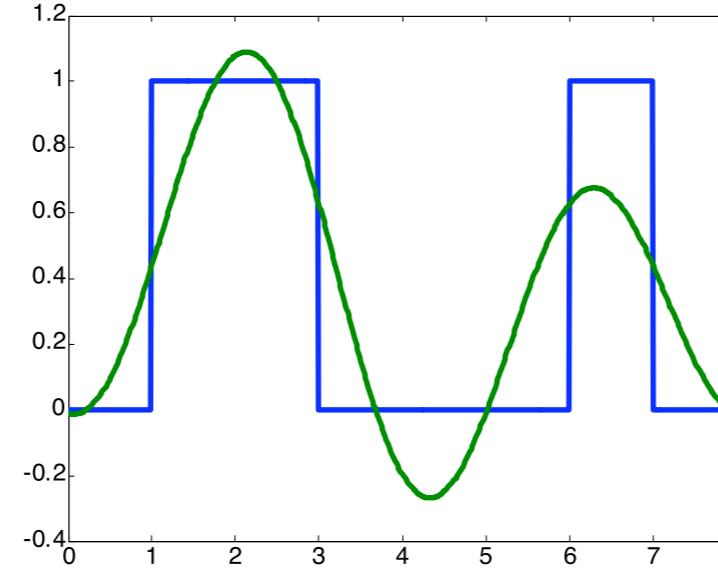
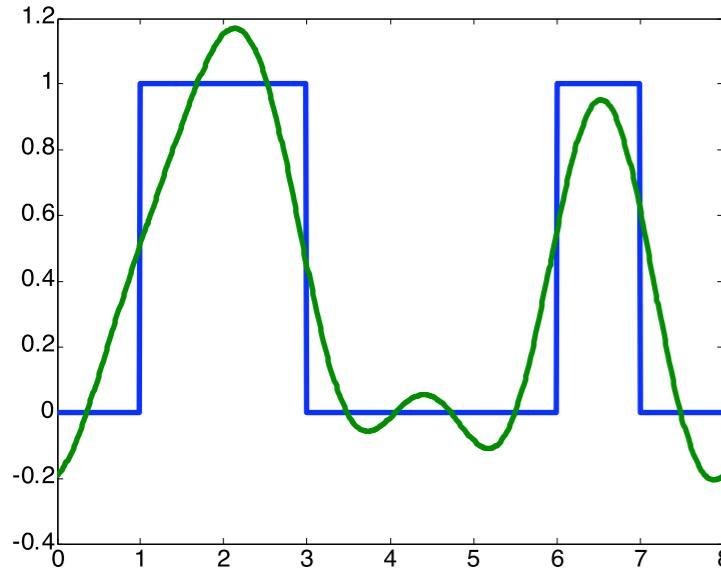
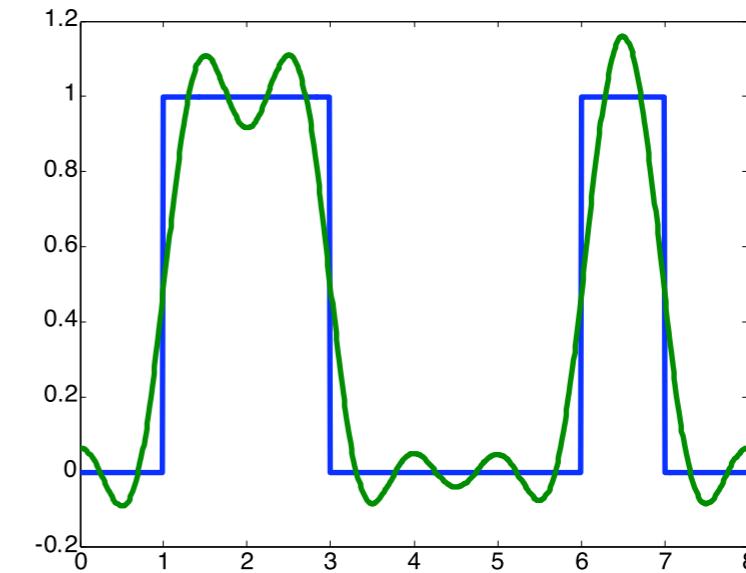
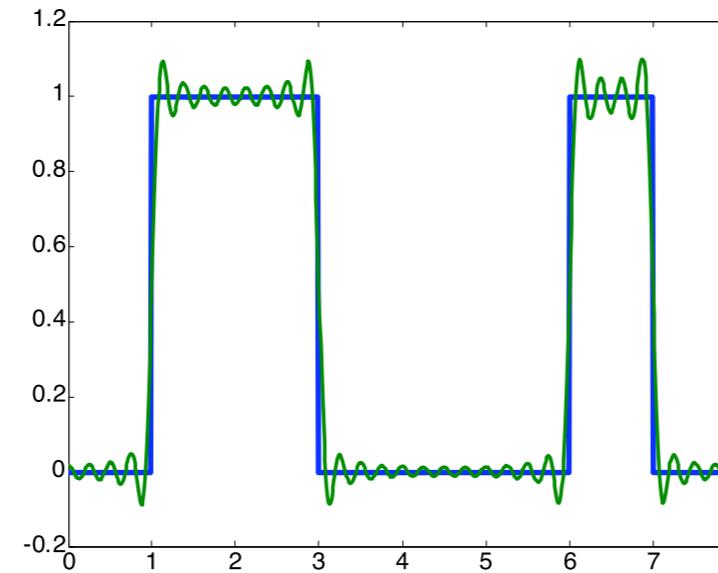
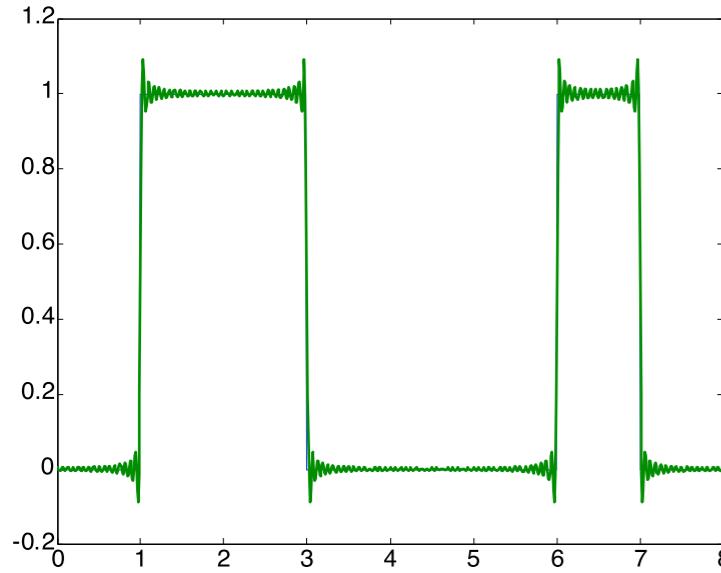
$$= 10 \log_{10} \frac{P_1}{P_0} \quad (\text{in deziBel [dB]})$$

$$\alpha = \frac{P_1}{P_0}$$



2. Nicht alle Frequenzen passieren das Medium

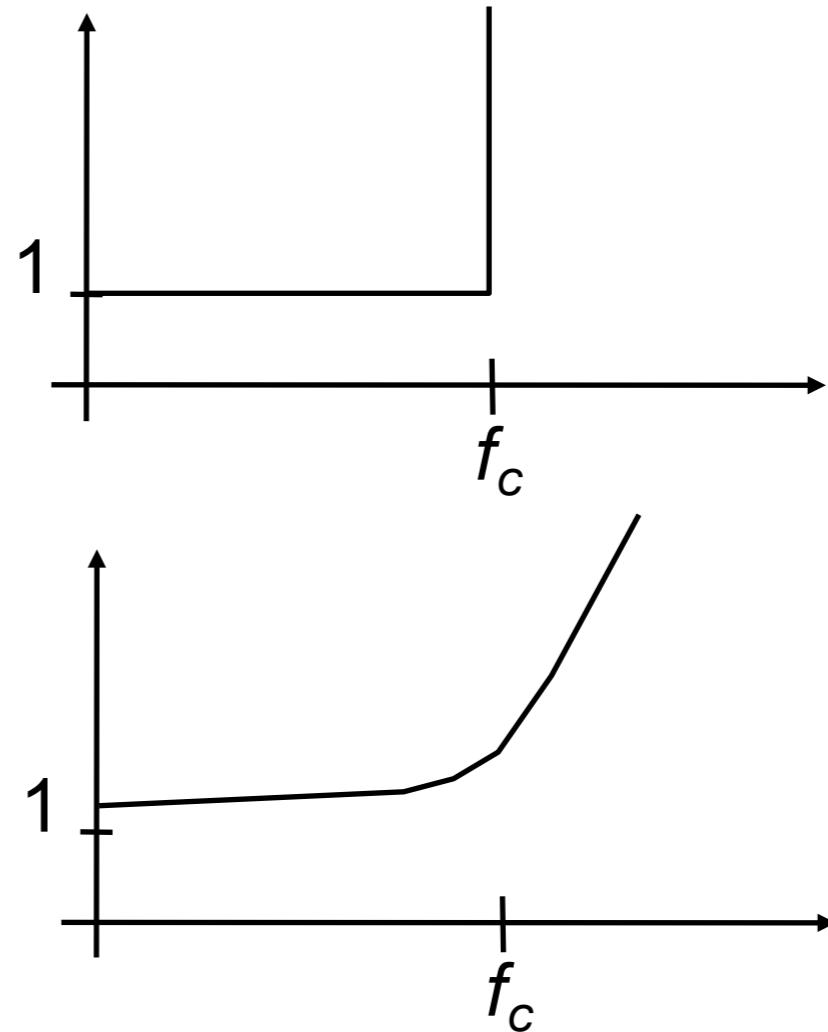
- Das Signal beim Verlust der hohen Frequenzen



(aus Vorlesung von Holger Karl)

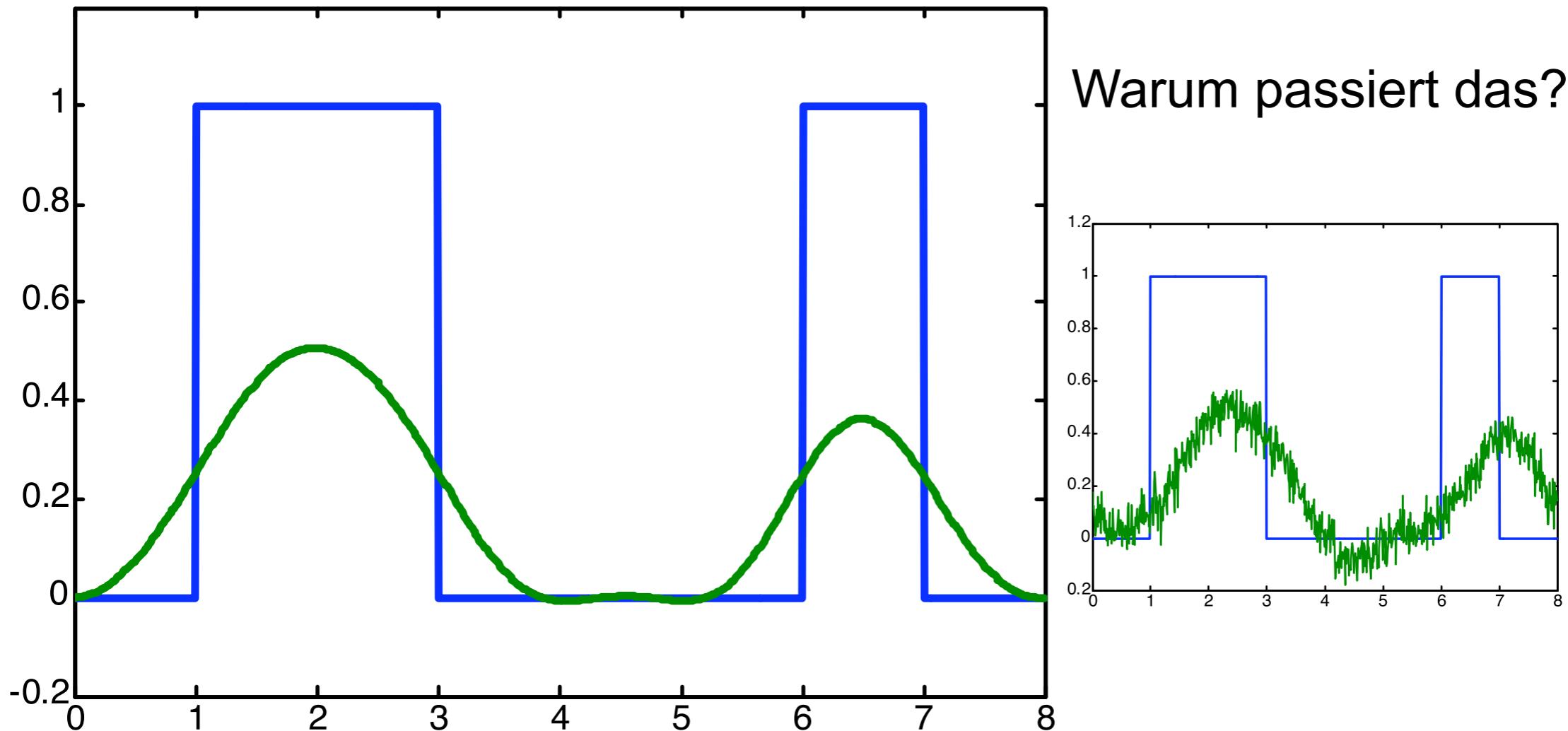
3. Frequenzabhängige Dämpfung

- Vorherige Seite: Cutoff
 - Zuerst ist die Dämpfung 1
 - und dann Unendlich
- Realistischer:
 - Dämpfung steigt kontinuierlich von 1 zu höheren Frequenzen
- Beides:
 - Bandweiten-begrenzter Kanal



Beispiel mit realistischerer Dämpfung

- Beispiel: Dämpfung ist 2; 2,5, 3,333... , 5, 10, ∞ für den ersten, zweiten, ... Fourier-koeffizienten



(aus Vorlesung von Holger Karl)

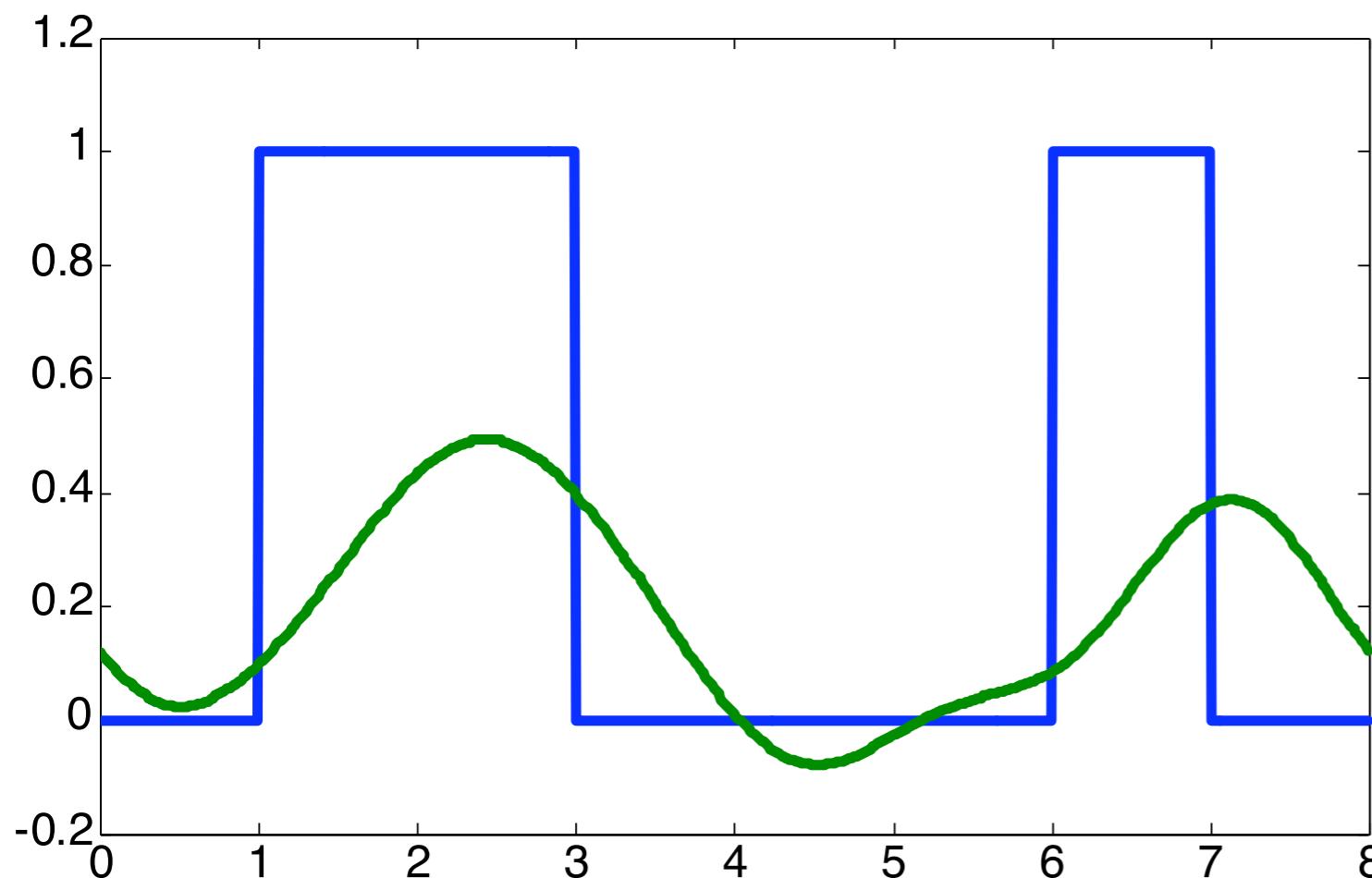
4. Das Medium stört und verzerrt

- In jedem Medium (außer dem Vakuum) haben verschiedene Frequenzen verschiedene Ausbreitungsgeschwindigkeit
 - Resultiert in Phasenverschiebung
 - Die zugrunde liegende Sinuskurve ist bestimmt durch Amplitude a , Frequenz f , and Phase ϕ

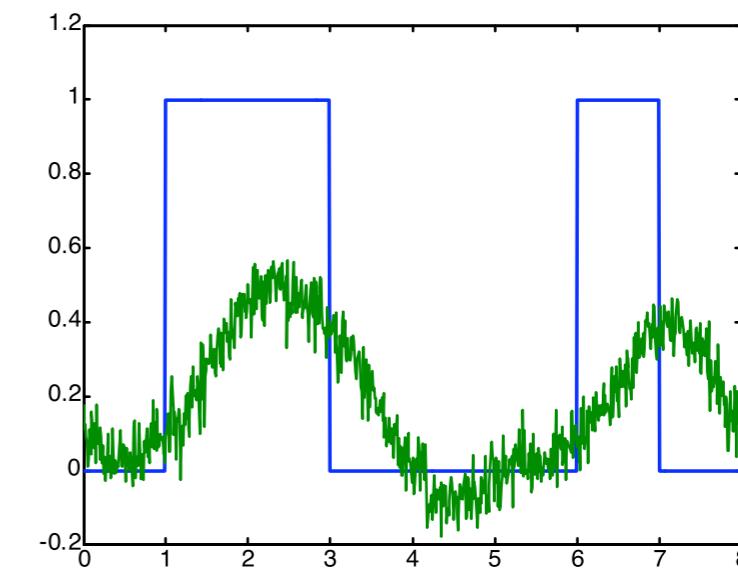
$$a \sin(2\pi ft + \phi)$$

- Die Größe dieser Phasenverschiebung hängt von der Frequenz ab
 - Dieser Effekt heißt Verzerrung (distortion)

Frequenzabhängige Dämpfung und Verzerrung



Warum passiert das:

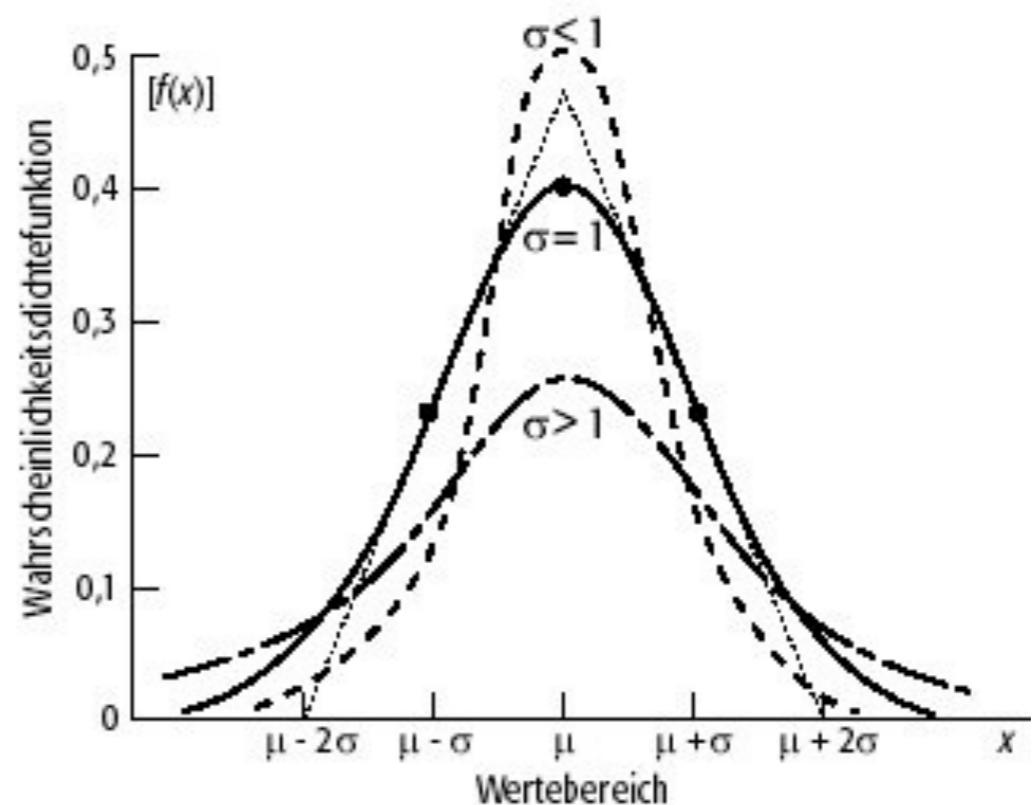


(aus Vorlesung von Holger Karl)

5. Echte Medien rauschen

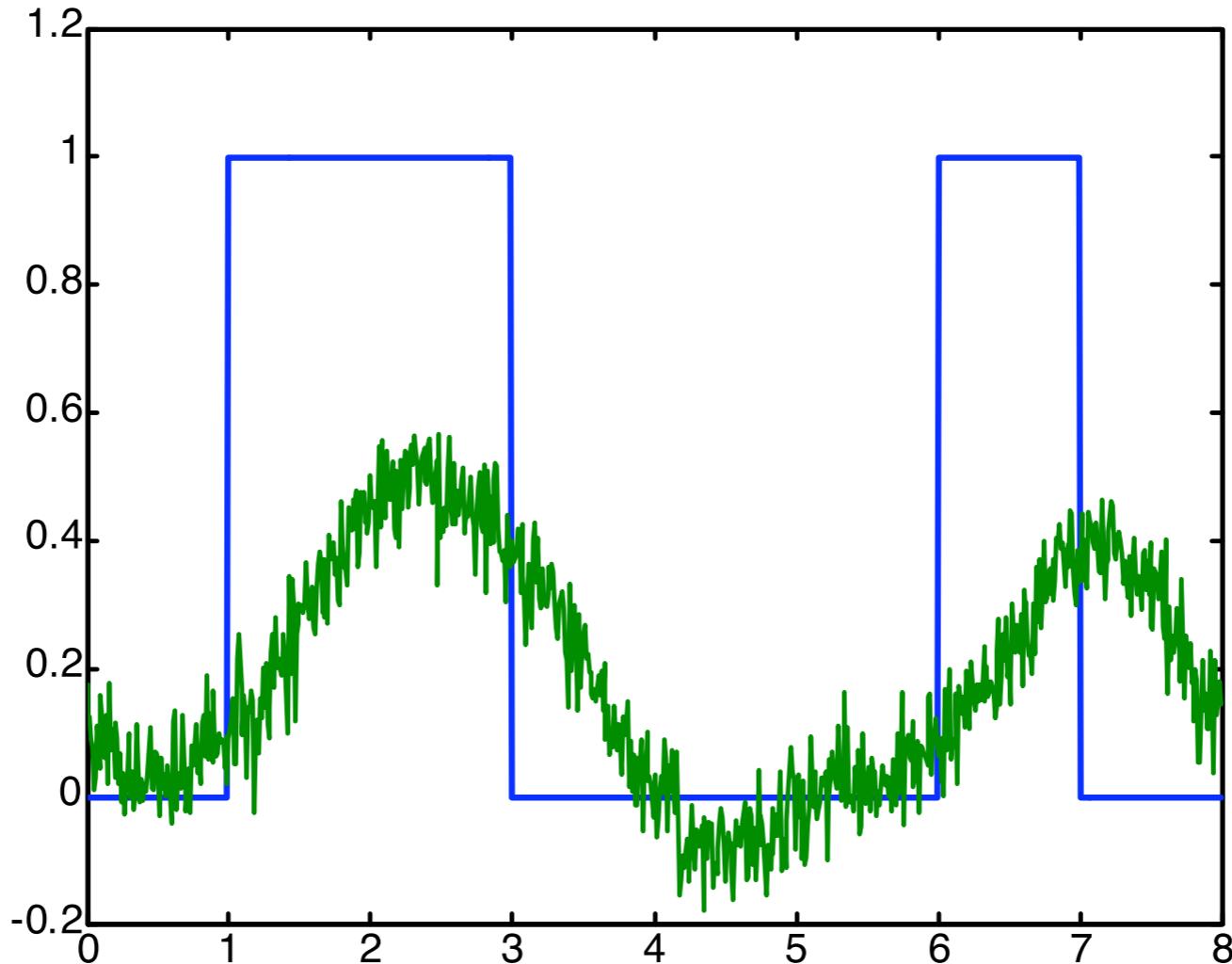
- Jedes Medium und jeder Sender und Empfänger produzieren Rauschen
 - Verursacht durch Wärme, Störungen anderer Geräte, Signale, Wellen, etc.
- Wird beschrieben durch zufällige Fluktuationen des (störungsfreien) Signals
 - Typische Modellierung: Gauß'sche Normalverteilung

$$f(x) = \frac{1}{\sigma \cdot \sqrt{2\pi}} \cdot e^{-\frac{1}{2} \left(\frac{x - \mu}{\sigma} \right)^2}$$



Zusammenfassung

- Dies alles kann das Eingangssignal erklären.



(aus Vorlesung von Holger Karl)

Basisband und Breitband

■ Basisband (baseband)

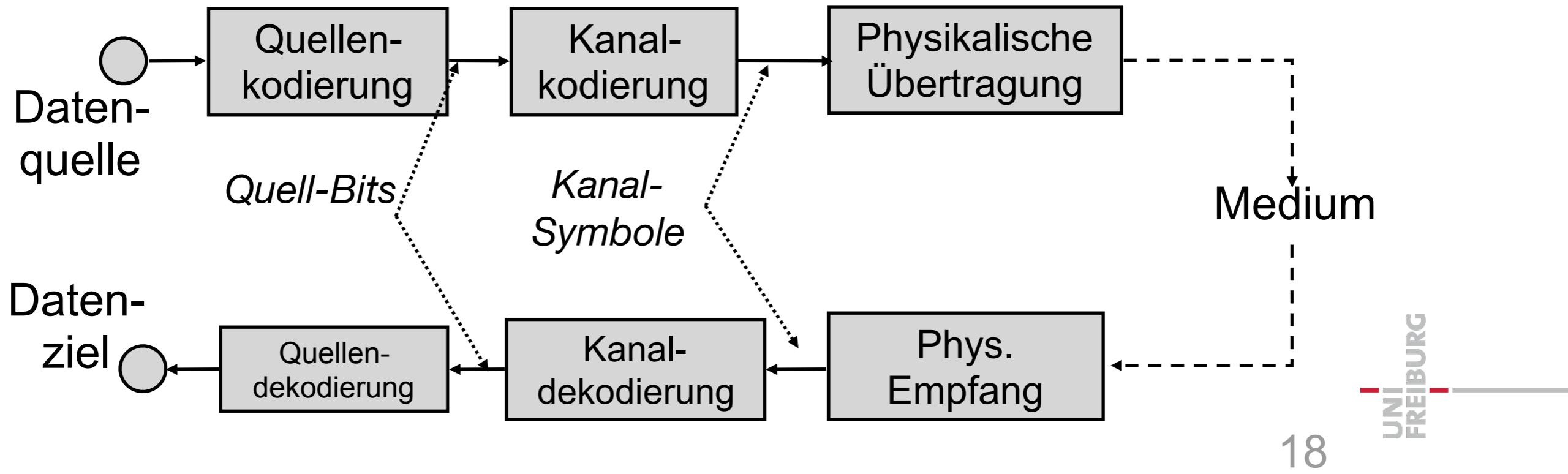
- Das digitale Signal wird direkt in Strom- oder Spannungsveränderungen umgesetzt
- Das Signal wird mit allen Frequenzen übertragen
 - z.B. Durch NRZ (Spannung hoch = 1, Spannung niedrig = 0)
- Problem: Übertragungseinschränkungen

■ Breitband (broadband)

- Die Daten werden durch einen weiten Frequenzbereich übertragen
- Weiter Bereich an Möglichkeiten:
 - Die Daten können auf eine Trägerwelle aufgesetzt werden (Amplitudenmodulation)
 - Die Trägerwelle kann verändert (moduliert) werden (Frequenz/ Phasenmodulation)
 - Verschiedene Trägerwellen können gleichzeitig verwendet werden

Struktur einer digitalen Basisband-Übertragung

- Quellkodierung
 - Entfernen redundanter oder irrelevanter Information
 - Z.B. mit verlustbehafteter Komprimierung (MP3, MPEG 4)
 - oder mit verlustloser Komprimierung (Huffman-Code)
- Kanalkodierung
 - Abbildung der Quellbits auf Kanal-Symbole
 - Möglicherweise Hinzufügen von Redundanz angepasst auf die Kanaleigenschaften
- Physikalische Übertragung
 - Umwandlung in physikalische Ereignisse



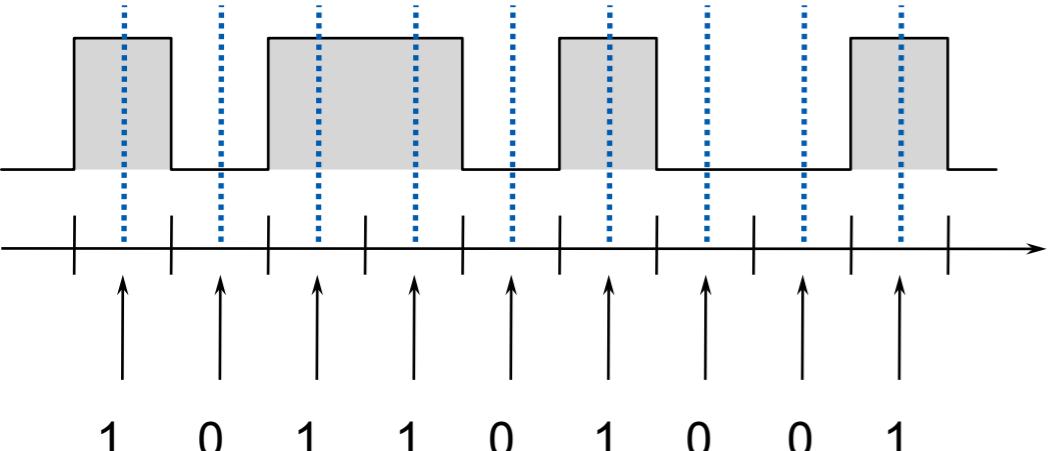
Selbsttaktende Kodierungen

- Wann muss man die Signale messen
 - Typischerweise in der Mitte eines Symbols
 - Wann startet das Symbol?
 - Die Länge des Symbols ist üblicherweise vorher festgelegt.
- Der Empfänger muss auf der Bit-ebene mit dem Sender synchronisiert sein
 - z.B. durch *Frame Synchronization*

Synchronisation

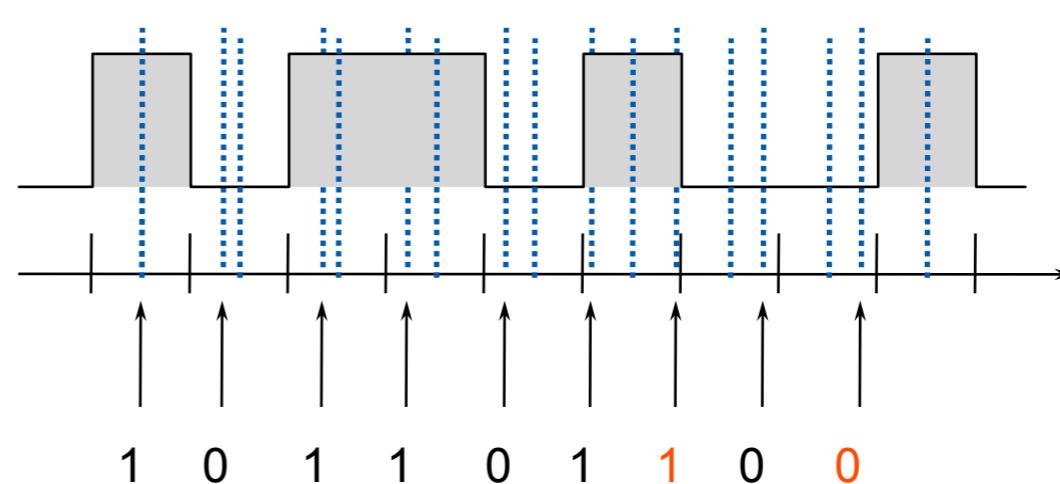
- Was passiert wenn man einfach Uhren benutzt
- Problem
 - Die Uhren driften auseinander
 - Keine zwei (bezahlbare Uhren) bleiben perfekt synchron
- Fehler by Synchronisationsverlust (NRZ):

Sender:



Empfänger mit driftender Uhr

Kanal



Lösung der Synchronisation

- Ohne Kontrolle keine Synchronisation
- Lösung: explizites Uhrensignal
 - Benötigt parallele Übertragung über Extra-Kanal
 - Muss mit den Daten synchronisiert sein
 - Nur für kurze Übertragungen sinnvoll
- Synchronisation an kritischen Zeitpunkten
 - z.B. Start eines Symbols oder eines Blocks
 - Sonst läuft die Uhr völlig frei
 - Vertraut der kurzzeitig funktionierenden Synchronität der Uhren
- Uhrensignal aus der Zeichenkodierung

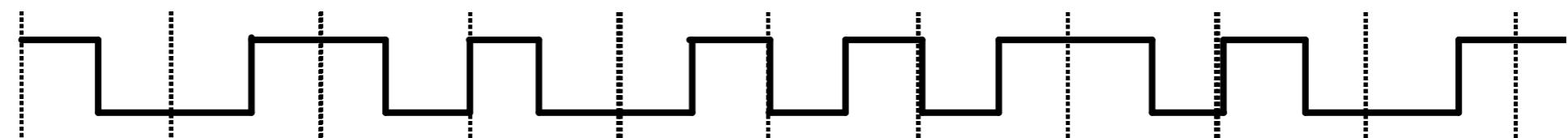
Selbsttaktende Codes

- z.B. Manchester Code (Biphase Level)
 - 1 = Wechsel von hoch zu niedrig in der Intervallmitte
 - 0 = Umgekehrter Wechsel

Daten:

1 0 1 1 0 0 0 1 1 0

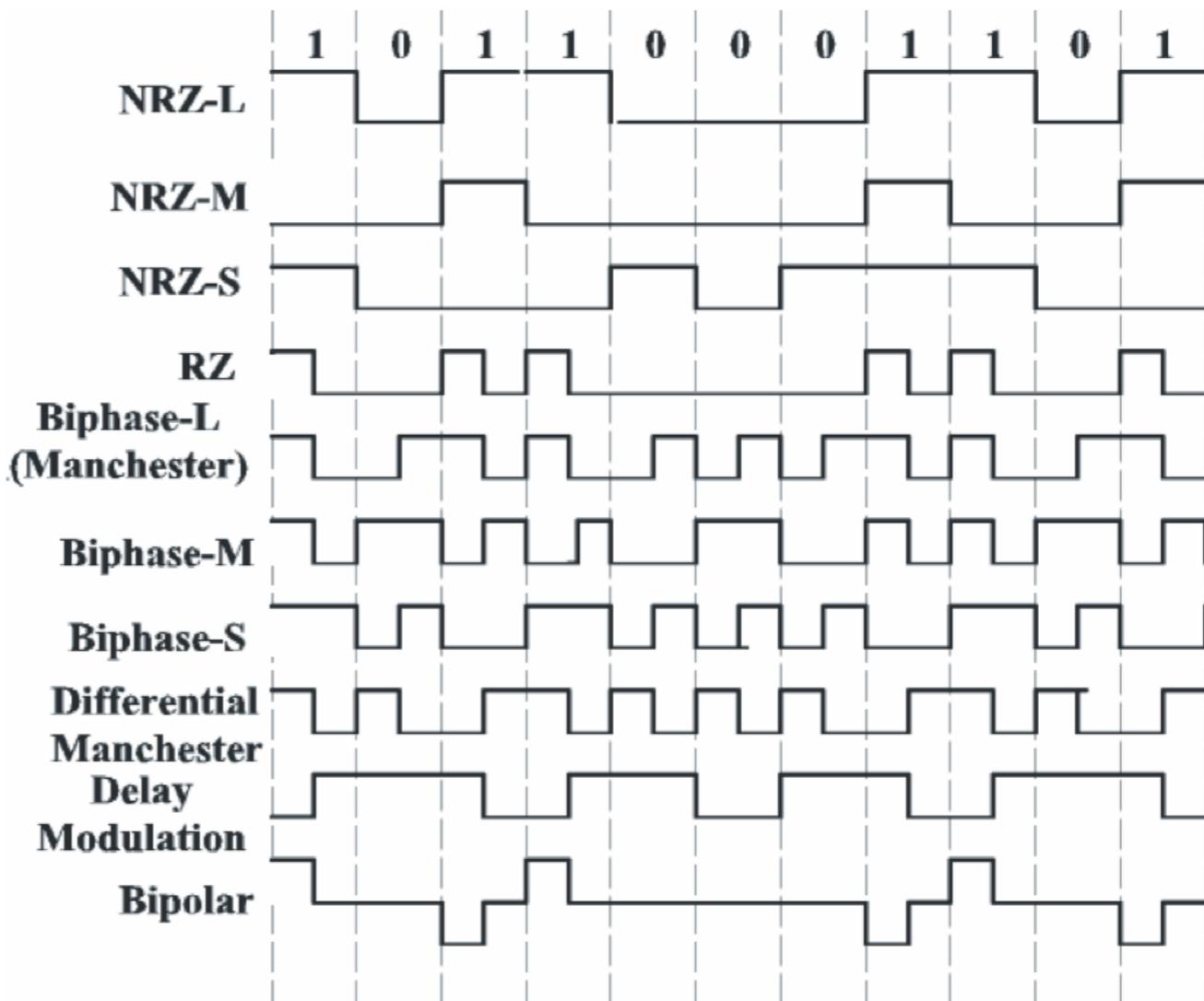
Manchester



- Das Signal beinhaltet die notwendige Information zur Synchronisation

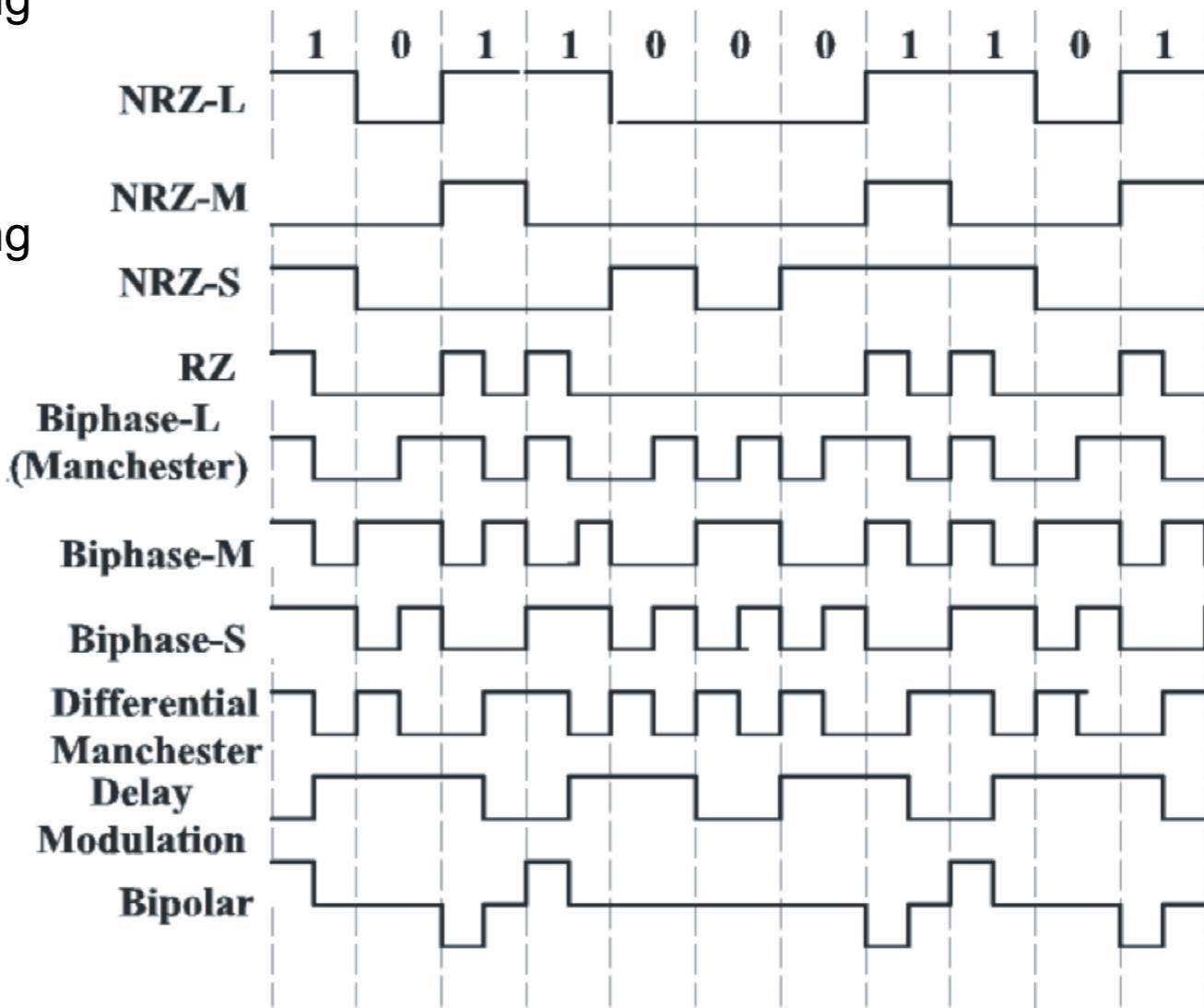
Digitale Kodierungen (I)

- Non-Return to Zero-Level (NRZ-L)
 - 1 = hohe Spannung, 0 = niedrig
- Non-Return to Zero-Mark (NRZ-M)
 - 1 = Wechsel am Anfang des Intervalls
 - 0 = Kein Wechsel
- Non-Return to Zero-Space (NRZ-S)
 - 0 = Wechsel am Intervallanfang
 - 1 = Kein Wechsel
- Return to Zero (RZ)
 - 1 = Rechteckpuls am Intervallanfang
 - 0 = Kein Impuls
- Manchester Code (Biphase Level)
 - 1 = Wechsel von hoch zu niedrig in der Intervallmitte
 - 0 = Umgekehrter Wechsel



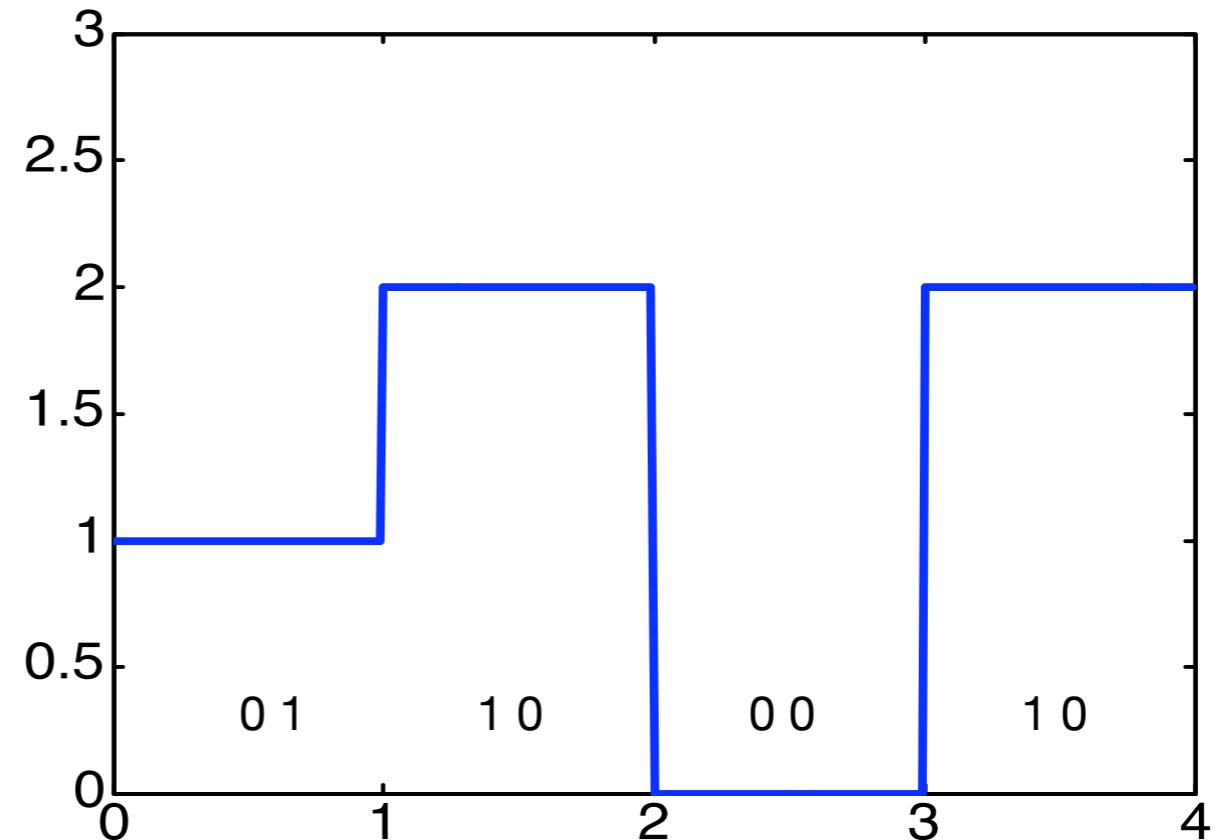
Digitale Kodierungen (II)

- Biphase-Mark
 - Immer: Übergang am Intervallanfang
 - 1 = zweiter Übergang in der Mitte
 - 0 = kein zweiter Übergang
- Biphase-Space
 - Immer: Übergang am Intervallanfang
 - 1/0 umgekehrt wie Biphase-Mark
- Differential Manchester-Code
 - Immer: Übergang in Intervallmitte
 - 1 = Kein Übergang am Intervallanfang
 - 0 = Zusätzlicher Übergang am Intervallanfang
- Delay Modulation (Miller)
 - Übergang am Ende, falls 0 folgt
 - 1 = Übergang in der Mitte des Intervalls
 - 0 = Kein Übergang falls 1 folgt
- Bipolar
 - 1 = Rechteckpuls in der ersten Hälfte, Richtung alterniert (wechselt)
 - 0 = Kein Rechteckpuls



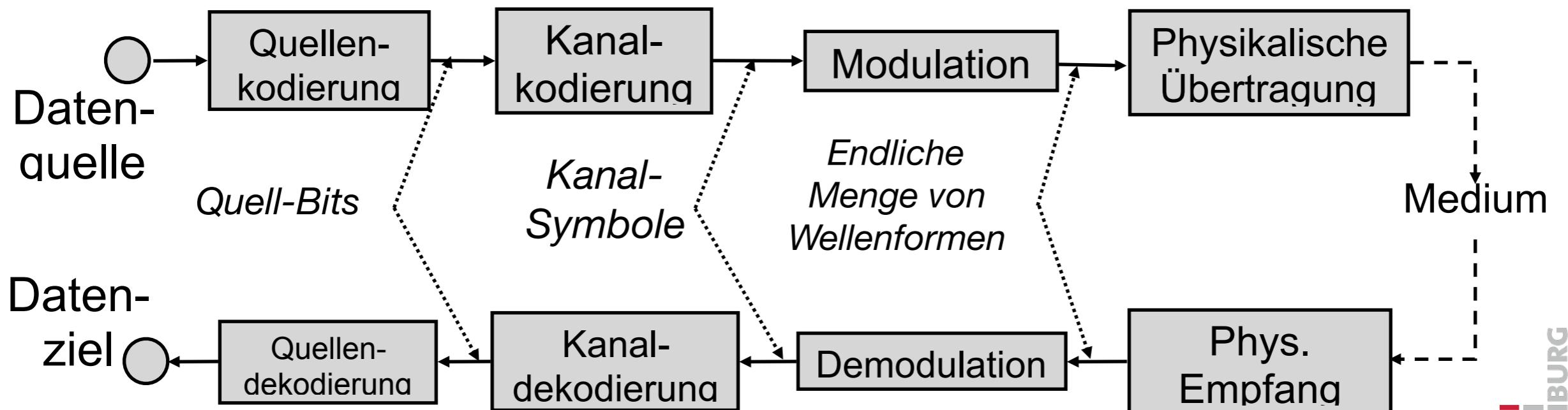
Symbole und Bits

- Für die Datenübertragung können statt Bits auch Symbole verwendet werden
- Z.B. 4 Symbole: A,B,C,D mit
 - A=00, B=01, C=10, D=11
- Symbole
 - Gemessen in Baud
 - Anzahl der Symbole pro Sekunde
- Datenrate
 - Gemessen in Bits pro Sekunde (bit/s)
 - Anzahl der Bits pro Sekunde
- Beispiel
 - 2400 bit/s Modem hat 600 Baud (verwendet 16 Symbole)



Struktur einer digitalen Breitband-Übertragung

- MODulation/DEModulation
 - Übersetzung der Kanalsymbole durch
 - Amplitudenmodulation
 - Phasenmodulation
 - Frequenzmodulation
 - oder einer Kombination davon



Physikalische Grundlagen

- Bewegte elektrisch geladene Teilchen verursachen elektromagnetische Wellen
 - **Frequenz**
 - f : Anzahl der Oszillationen pro Sekunde
 - Maßeinheit: Hertz
 - **Wellenlänge**
 - λ : Distanz (in Metern) zwischen zwei Wellenmaxima
 - Durch Antennen können elektro-magnetische Wellen erzeugt und empfangen werden
 - Die Ausbreitungsgeschwindigkeit von elektro-magnetischen Wellen im Vakuum ist konstant:
 - **Lichtgeschwindigkeit** $c \approx 3 \cdot 10^8$ m/s
- Zusammenhang:

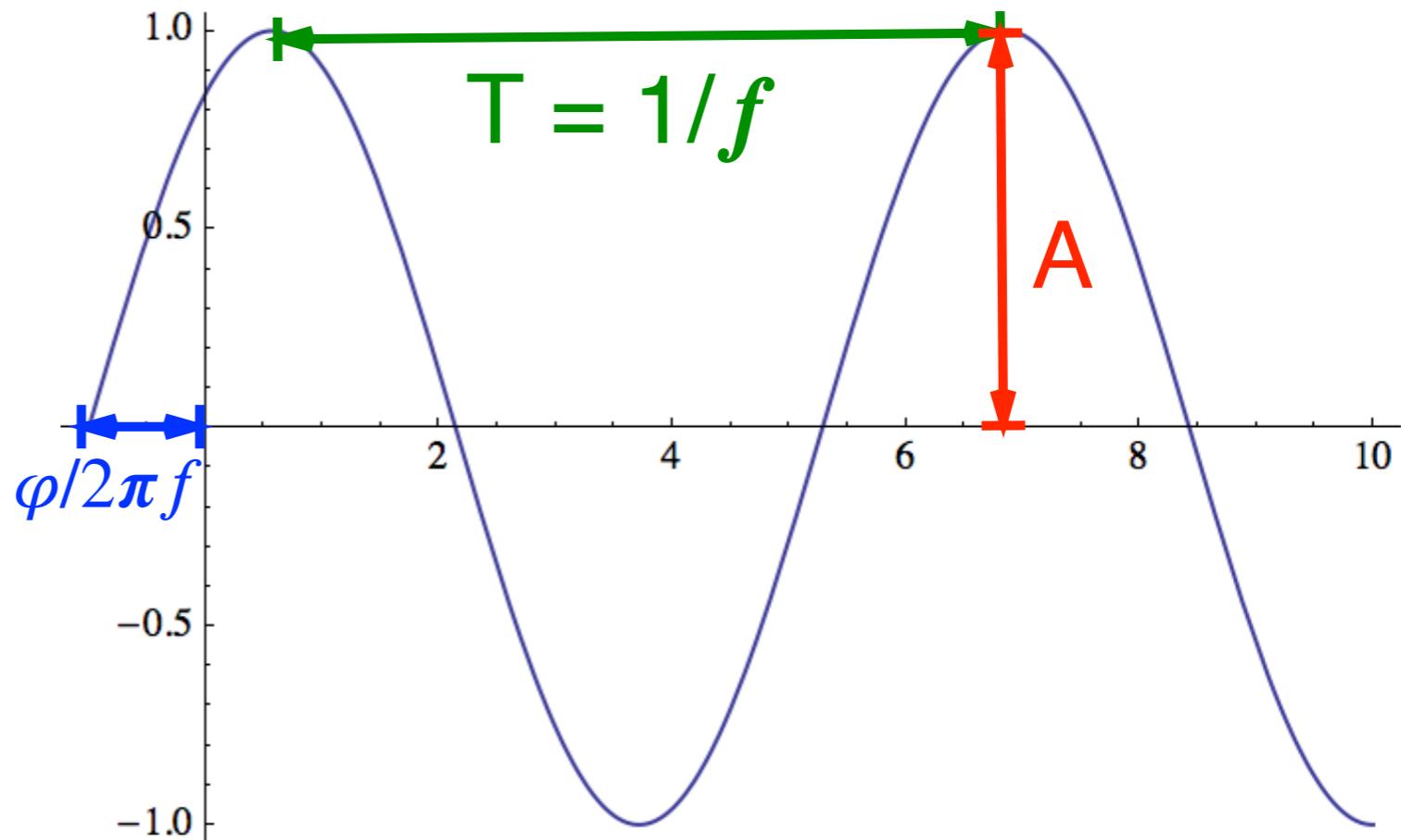
$$\lambda \cdot f = c$$

Amplitudendarstellung

■ Amplitudendarstellung einer Sinusschwingung

$$s(t) = A \sin(2\pi ft + \phi)$$

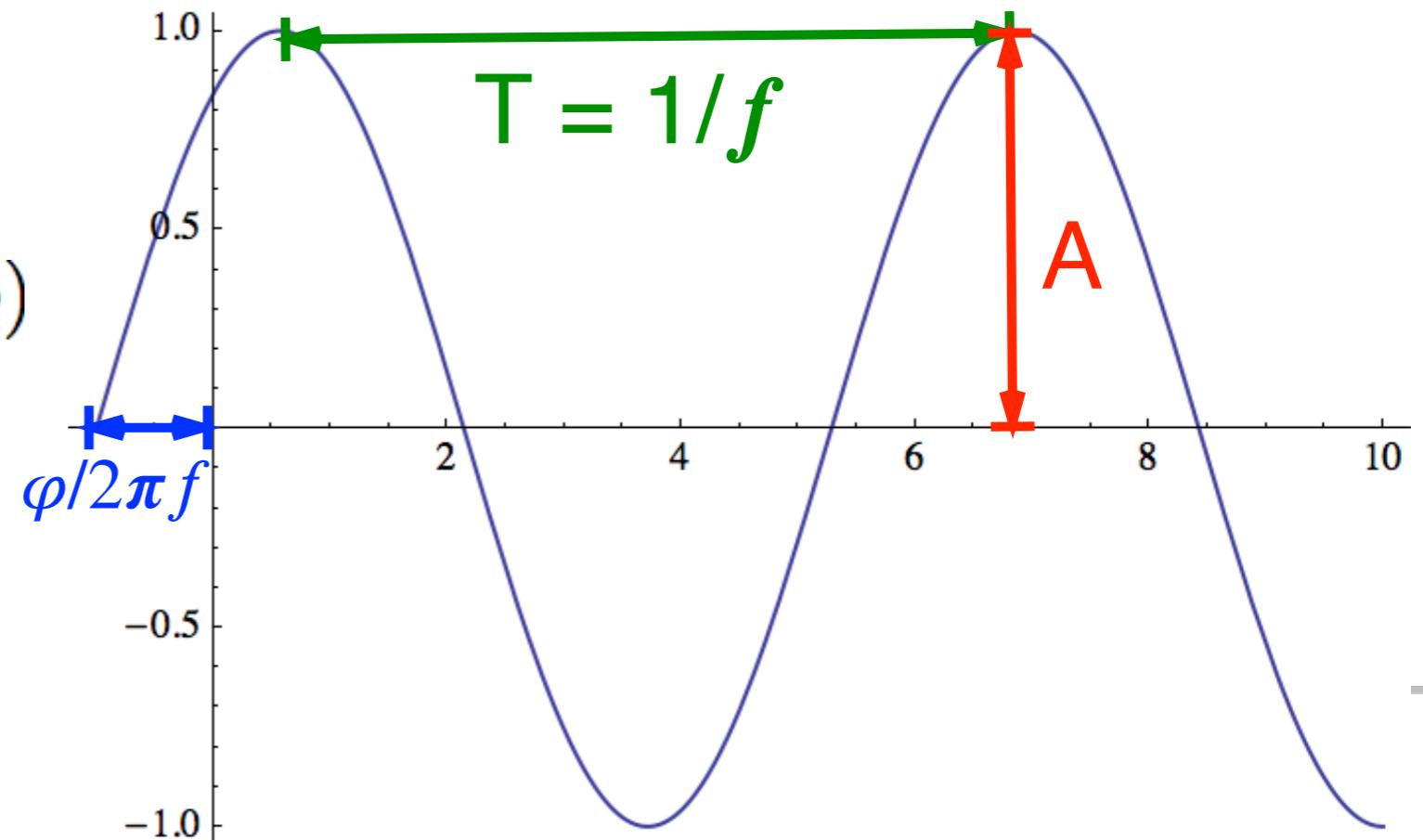
- A: Amplitude
- ϕ : Phasenverschiebung
- f : Frequenz = $1/T$
- T: Periode



Breitband

- Idee:
 - Konzentration auf die idealen Frequenzen des Mediums
 - Benutzung einer Sinuskurve als Trägerwelle der Signale
- Eine Sinuskurve hat keine Information
- Zur Datenübertragung muss die Sinuskurve fortlaufend verändert werden (moduliert)
 - Dadurch Spektralweitung (mehr Frequenzen in der Fourier-Analyse)
- Folgende Parameter können verändert werden:
 - Amplitude A
 - Frequenz $f=1/T$
 - Phase ϕ

$$s(t) = A \sin(2\pi ft + \phi)$$



Amplitudenmodulation

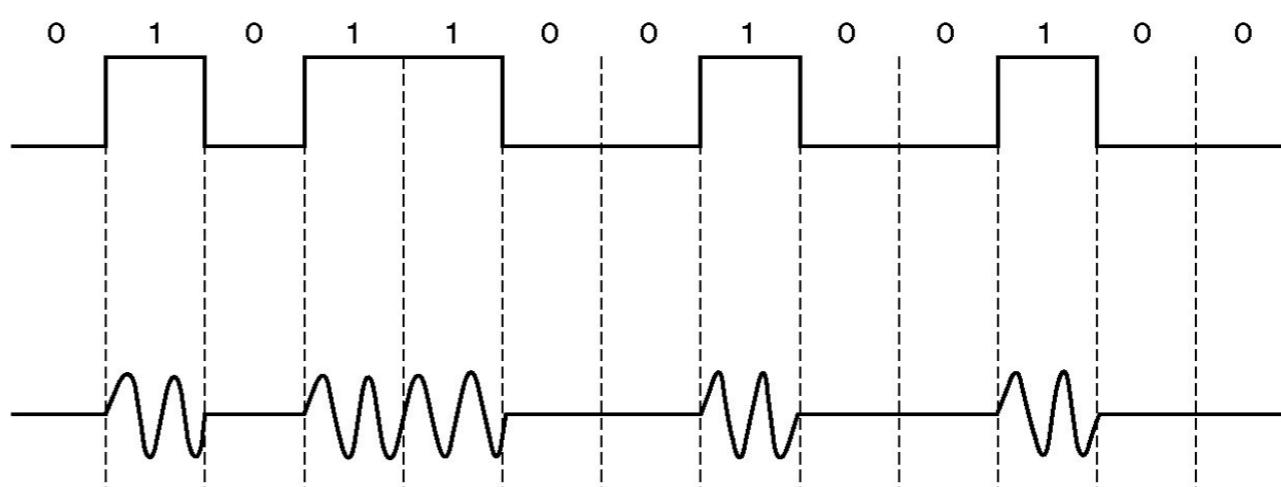
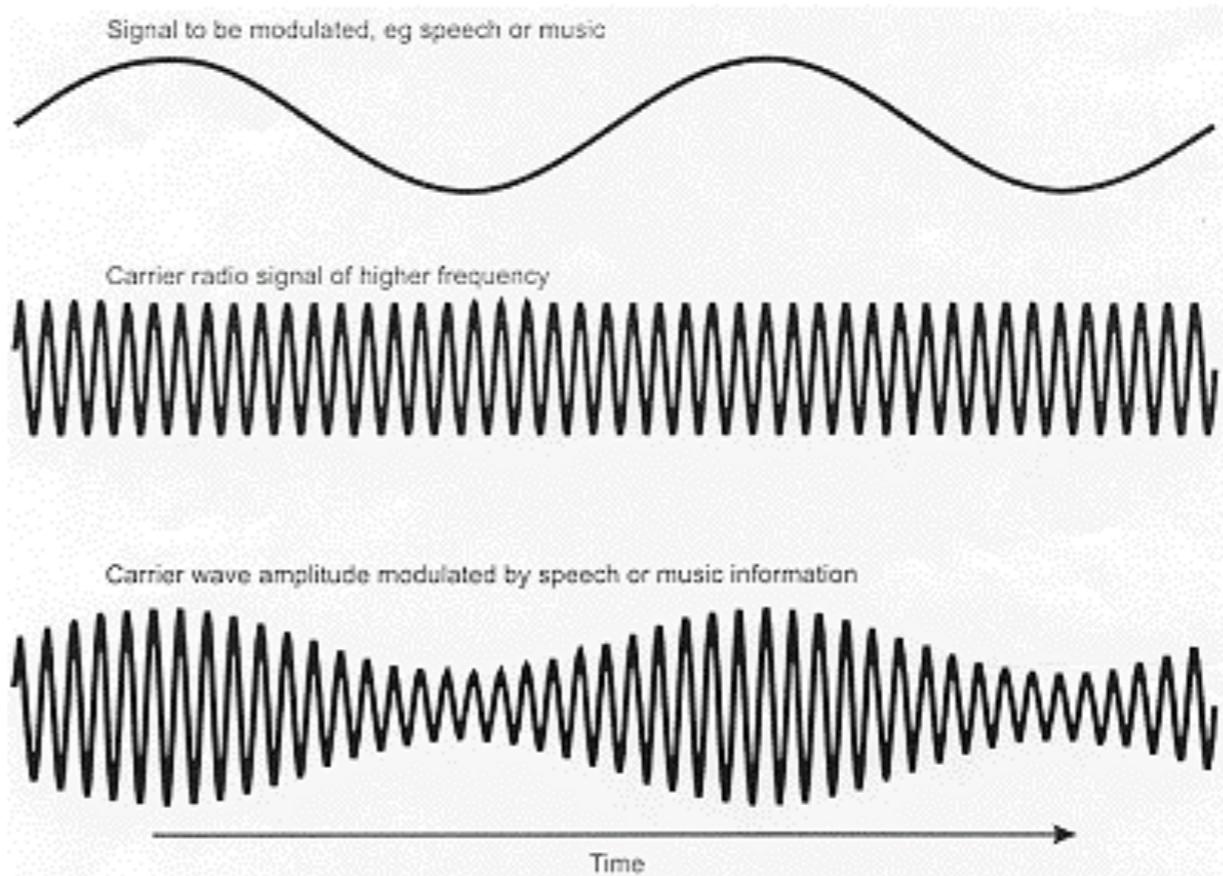
- Das zeitvariable Signal $s(t)$ wird als Amplitude einer Sinuskurve kodiert:

$$f_A(t) = s(t) \sin(2\pi ft + \phi)$$

- Analoges Signal
 - Amplitude Modulation
 - Kontinuierliche Funktion in der Zeit

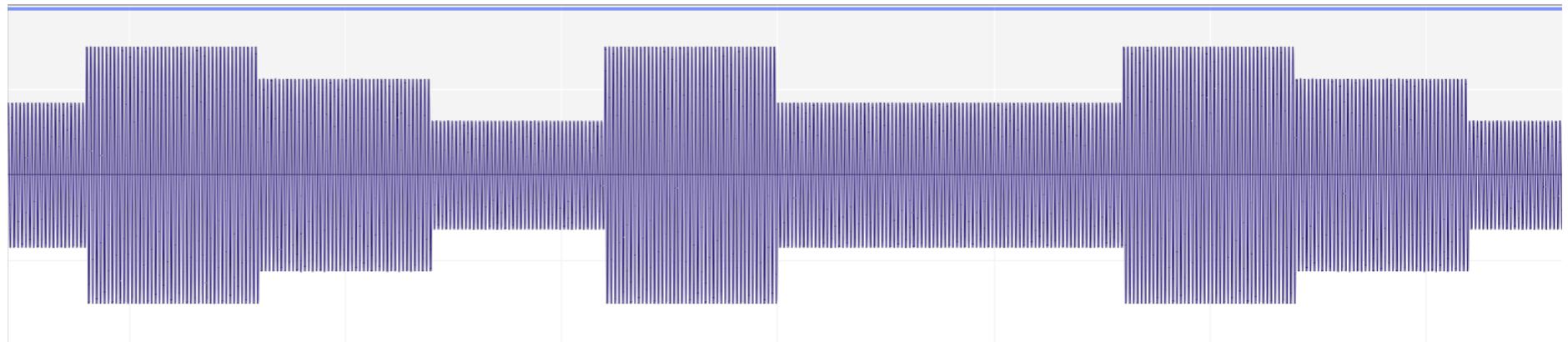
- z.B. zweites längeres Wellensignal (Schallwellen)

- Digitales Signal
 - Amplitude Keying
 - Z.B. durch Symbole gegeben als Symbolstärken
 - Spezialfall: Symbole 0 oder 1
 - on/off keying



Hörbeispiel

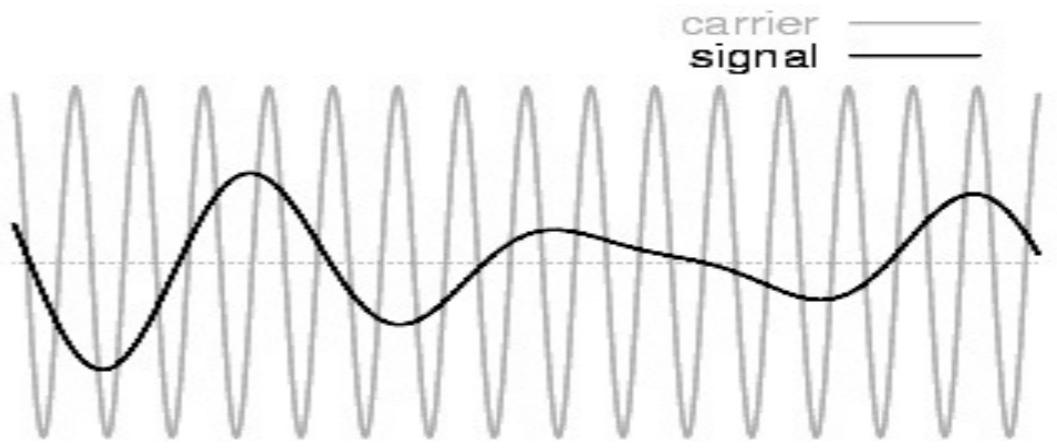
- Amplituden-modulierte Sinuskurve



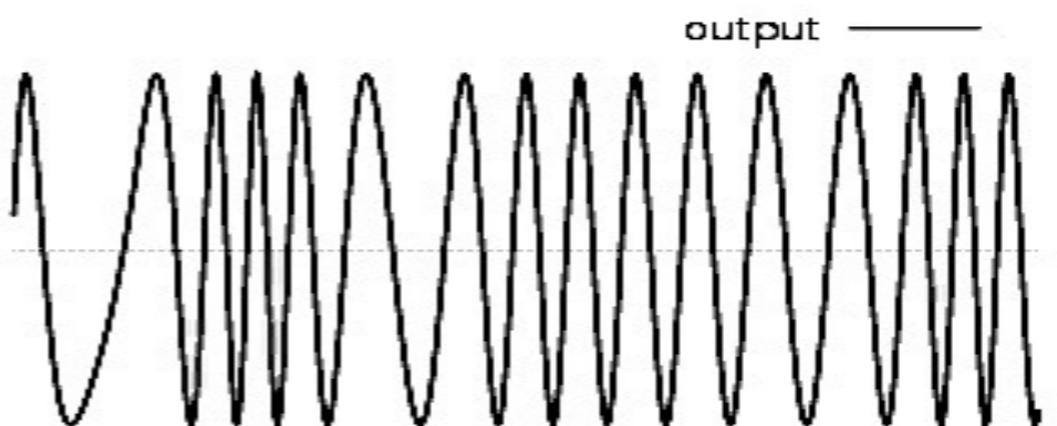
Frequenzmodulation

- Das zeitvariable Signal $s(t)$ wird in der Frequenz der Sinuskurve kodiert:

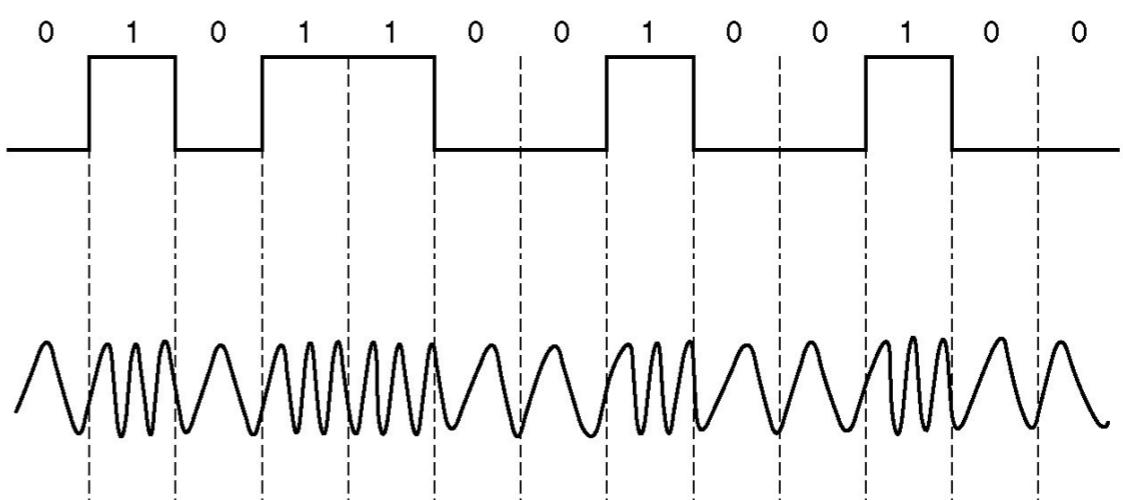
$$f_F(t) = a \sin(2\pi s(t)t + \phi)$$



- Analoges Signal
 - Frequency Modulation (FM)
 - Kontinuierliche Funktion in der Zeit

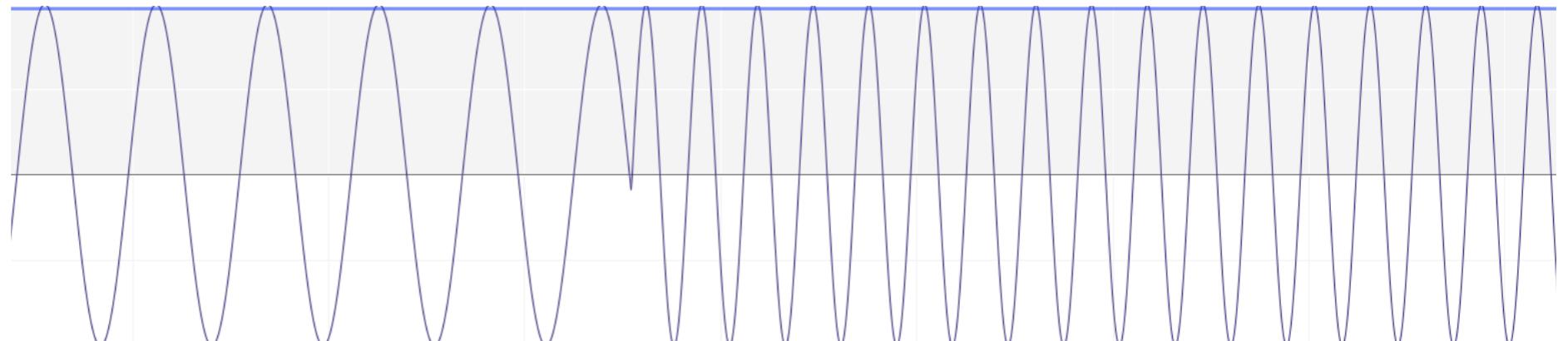
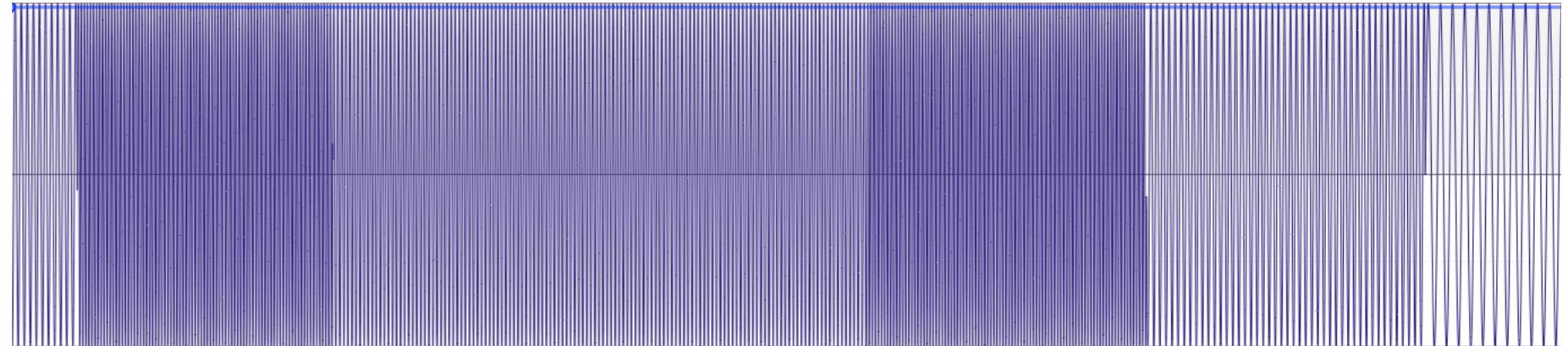


- Digitales Signal
 - Frequency Shift Keying (FSK)
 - Z.B. durch Symbole gegeben als Frequenzen



Hörbeispiel

- frequenz-
modulierte
Sinuskurve

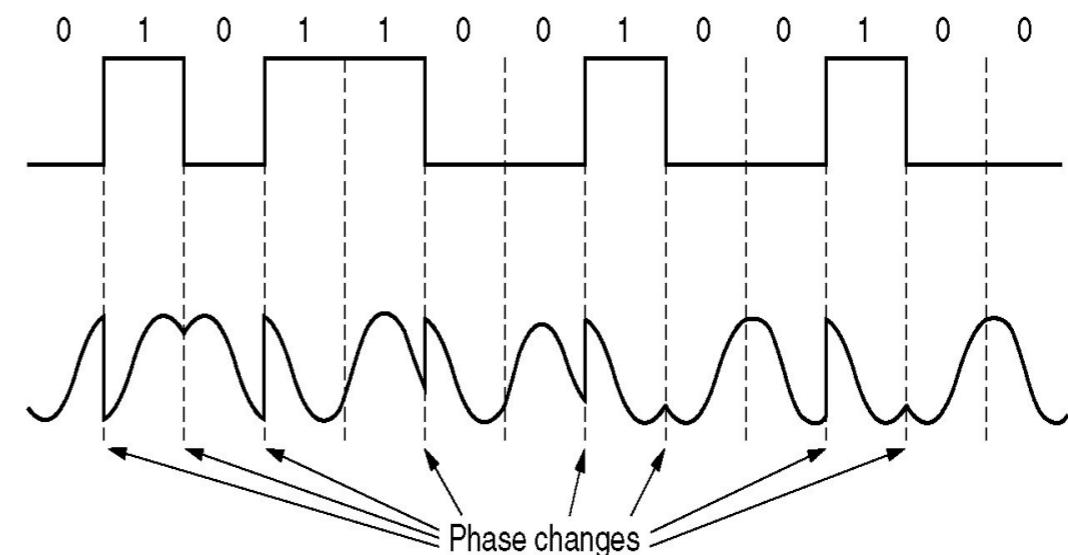
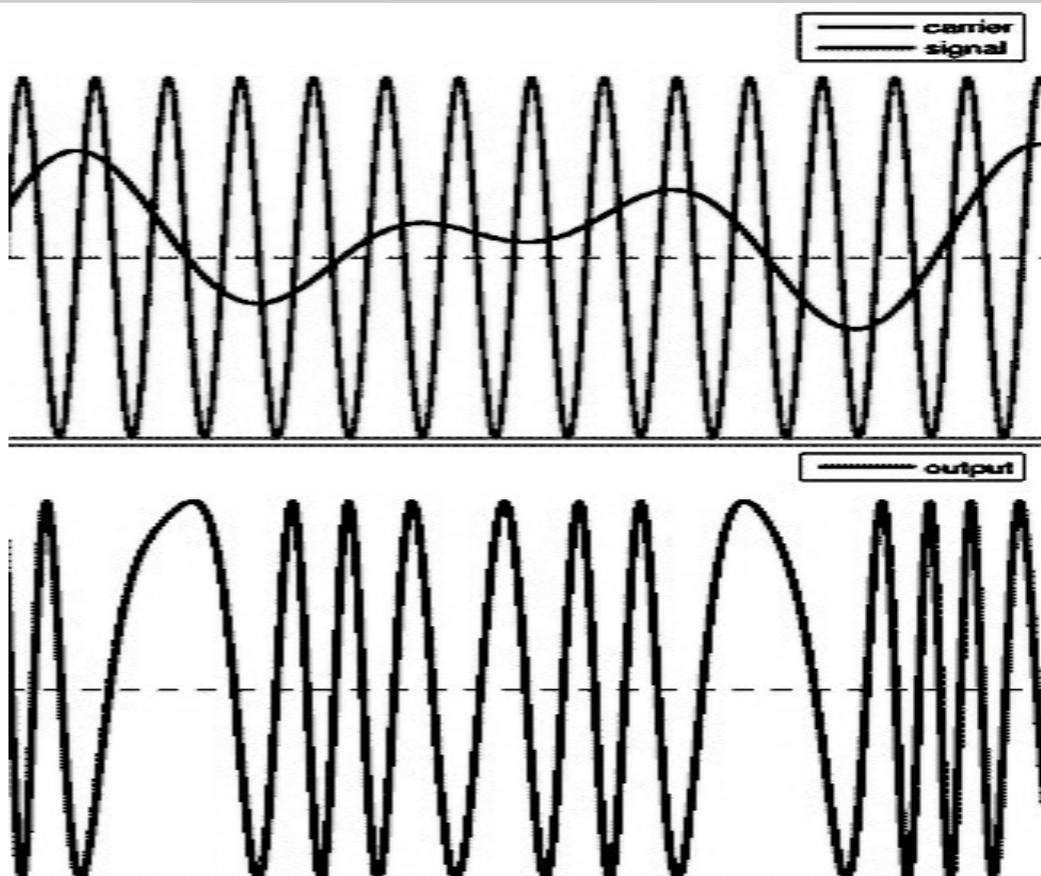


Phasenmodulation

- Das zeitvariable Signal $s(t)$ wird in der Phase der Sinuskurve kodiert:

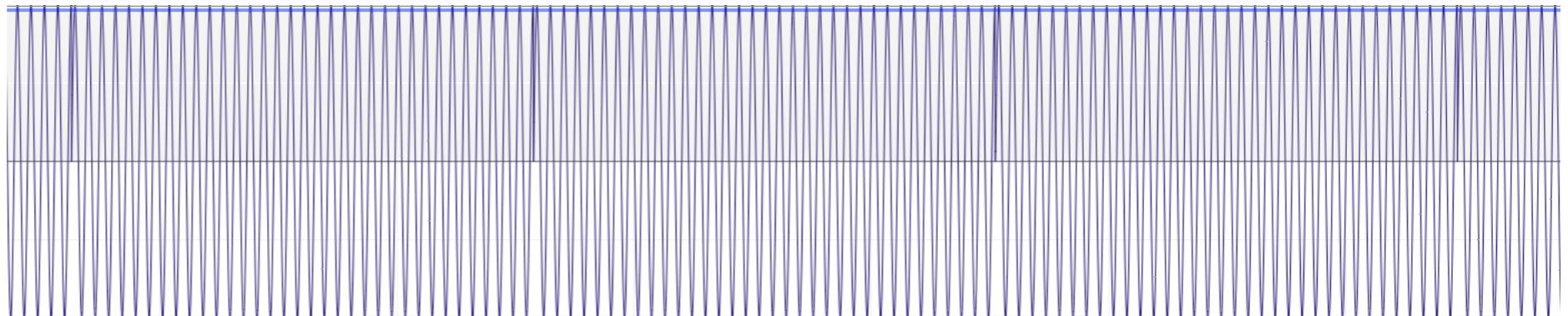
$$f_P(t) = a \sin(2\pi ft + s(t))$$

- Analoges Signal
 - Phase Modulation (PM)
 - Sehr ungünstige Eigenschaften
 - Wird nicht eingesetzt
- Digitales Signal
 - Phase-Shift Keying (PSK)
 - Z.B. durch Symbole gegeben als Phasen



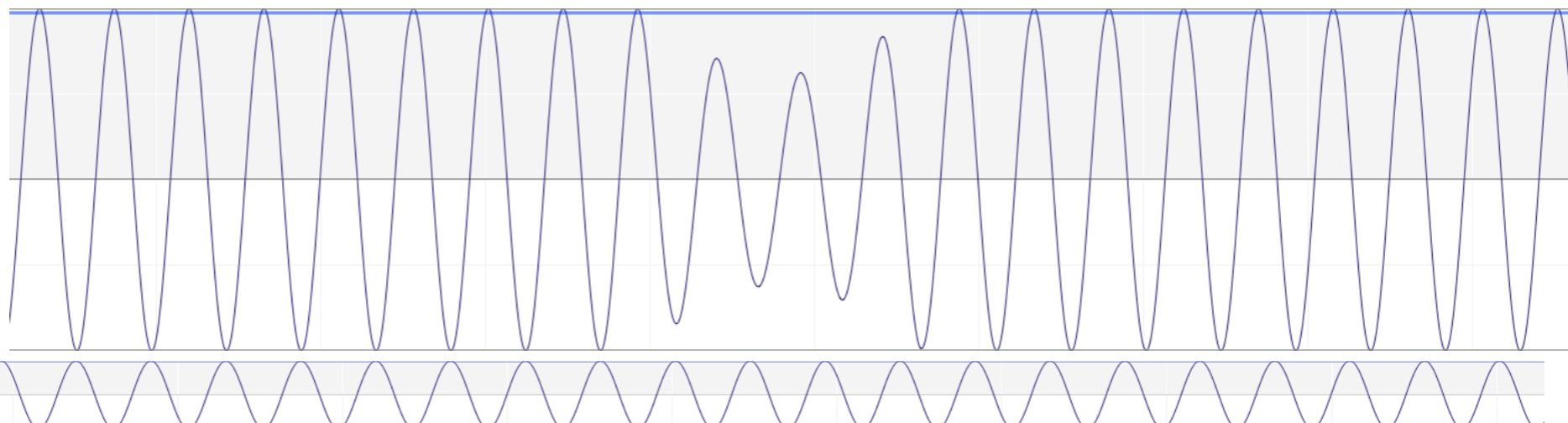
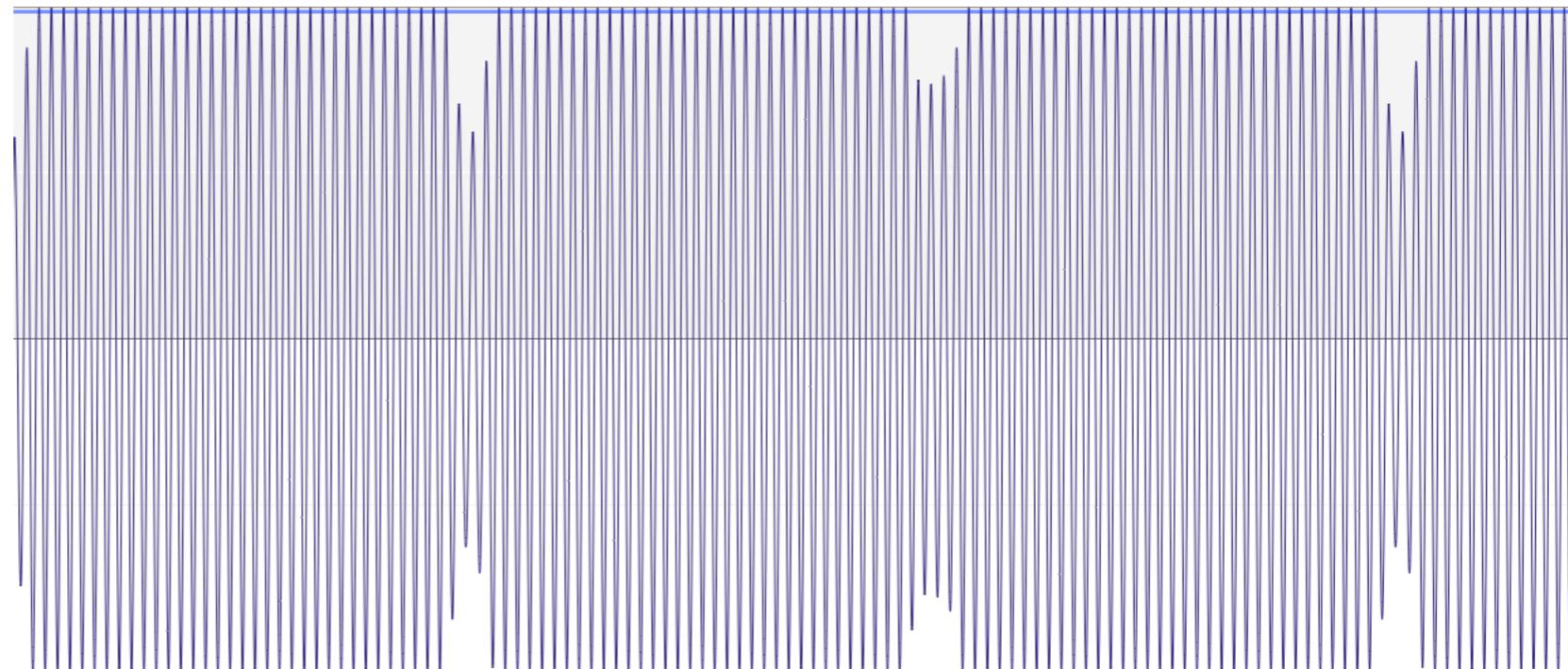
Hörbeispiel

- phasen-
modulierte
Sinuskurve



Hörbeispiel

- phasen-modulierte Sinuskurve
 - mit glatten Übergang

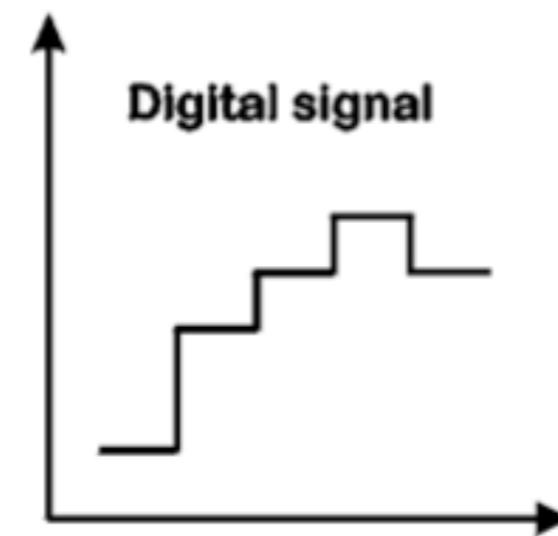
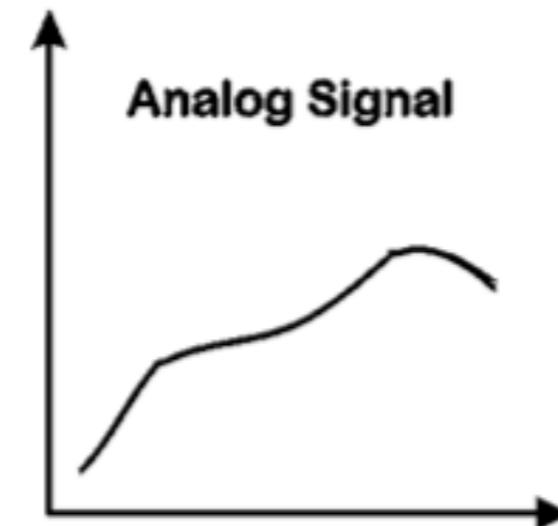


zum Vergleich

Digitale und analoge Signale im Vergleich

- Für einen Sender gibt es zwei Optionen

- Digitale Übertragung
 - Endliche Menge von diskreten Signalen
 - Z.B. endliche Menge von Spannungsgrößen/Stromstärken
 - Analoge Übertragung
 - Unendliche (kontinuierliche) Menge von Signalen
 - Z.B. Signal entspricht Strom oder Spannung im Draht



- Vorteil der digitalen Signale:

- Es gibt die Möglichkeit Empfangsungenauigkeiten zu reparieren und das ursprüngliche Signal zu rekonstruieren
 - Auftretende Fehler in der analogen Übertragung können sich weiter verstärken

Fouriertransformation

- Fouriertransformation einer periodischen Funktion:
 - Zerlegung in verschiedene
 - Sinus/Cosinus-Funktionen
- Dirichletsche Bedingungen einer periodischen Funktion f :
 - $f(x) = f(x+2\pi)$
 - $f(x)$ ist in $(-\pi, \pi)$ in endlich vielen Intervallen stetig und monoton
 - Falls f nicht stetig in x_0 , dann ist $f(x_0) = (f(x_0-0) + f(x_0+0))/2$
- Satz von Dirichlet:
 - $f(x)$ genüge in $(-\pi, \pi)$ den Dirichletschen Bedingungen. Dann existieren Fourerkoeffizienten $a_0, a_1, a_2, \dots, b_1, b_2, \dots$ so dass gilt:

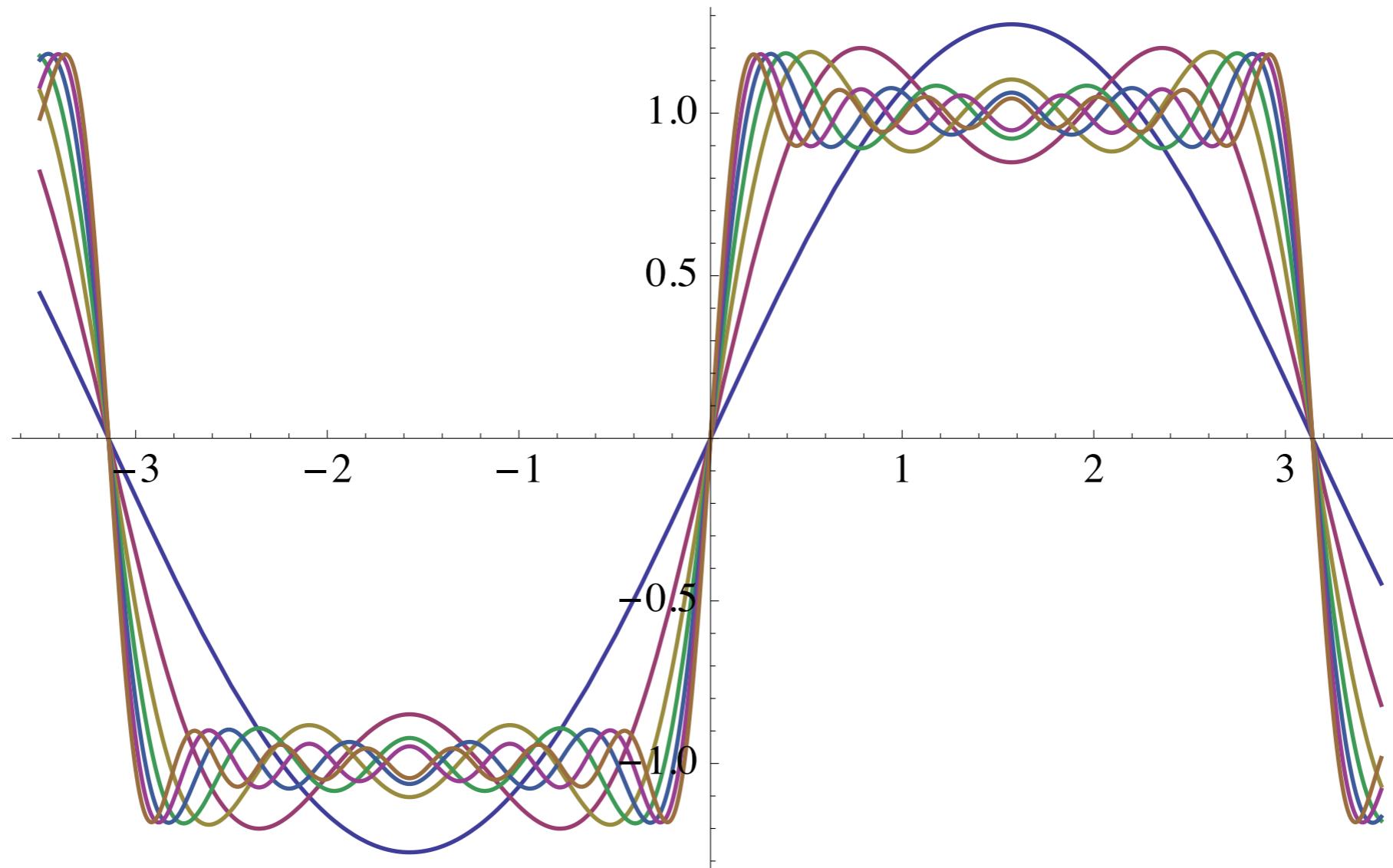
$$\lim_{n \rightarrow \infty} \frac{a_0}{2} + \sum_{k=1}^n a_k \cos kx + b_k \sin kx = f(x)$$

Fouriertransformation

■ Fouriertransformation einer periodischen Funktion:

- Zerlegung in verschiedene
- Sinus/Cosinus-Funktionen

$$\lim_{n \rightarrow \infty} \frac{a_0}{2} + \sum_{k=1}^n a_k \cos kx + b_k \sin kx = f(x)$$



Berechnung der Fourierkoeffizienten

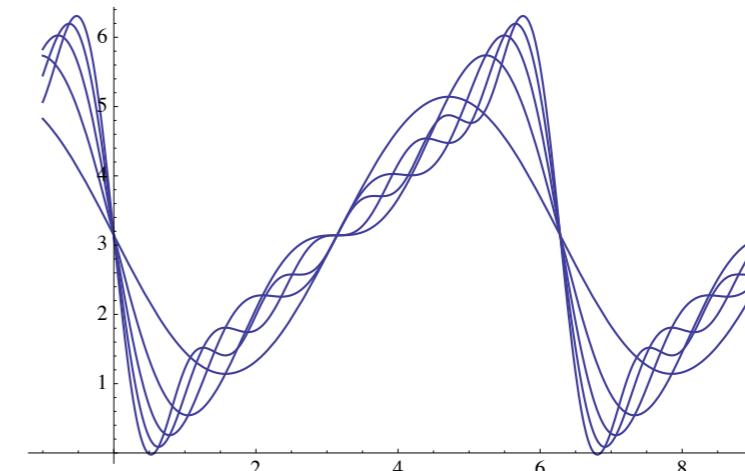
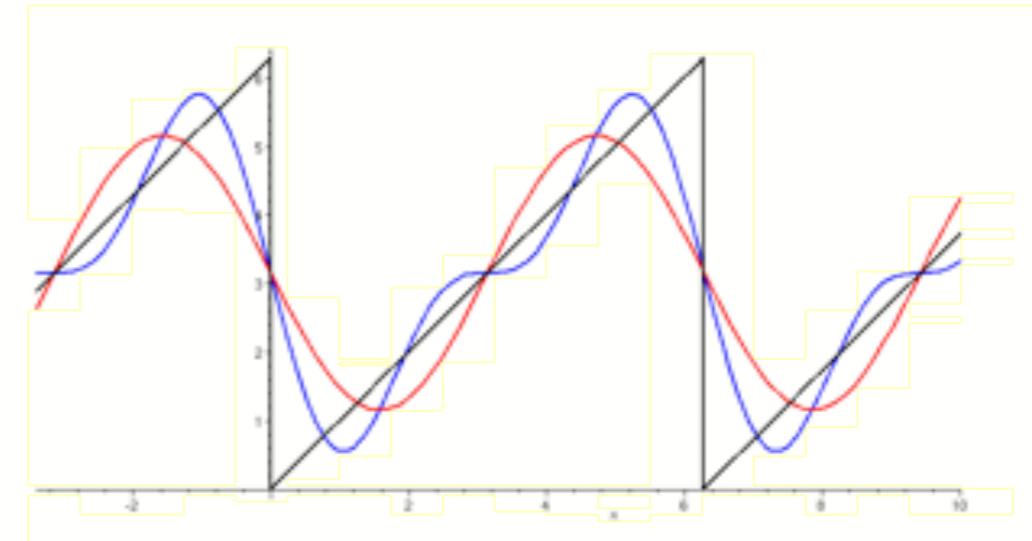
- Die Fourierkoeffizienten a_i, b_i können wie folgt berechnet werden:

- Für $k = 0, 1, 2, \dots$

$$a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos kx \, dx$$

- Für $k = 1, 2, 3, \dots$

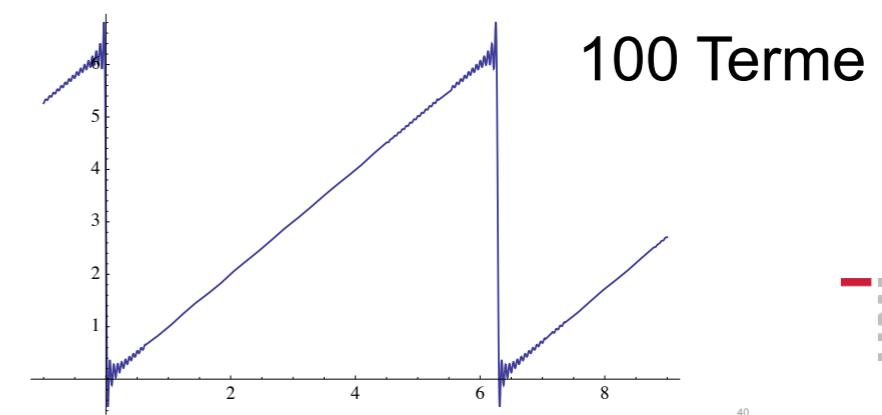
$$b_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin kx \, dx$$



- Beispiel: Sägezahnkurve

$$f(x) = x, \text{ für } 0 < x < 2\pi$$

$$f(x) = \pi - 2 \left(\frac{\sin x}{1} + \frac{\sin 2x}{2} + \frac{\sin 3x}{3} + \dots \right)$$



Fourier-Analyse für allgemeine Periode

- Der Satz von Fourier für Periode $T=1/f$:
 - Die Koeffizienten c , a_n , b_n ergeben sich dann wie folgt

$$g(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos(2\pi k f t) + b_k \sin(2\pi k f t)$$

$$a_k = \frac{2}{T} \int_0^T g(t) \cos(2\pi n f t) dt$$

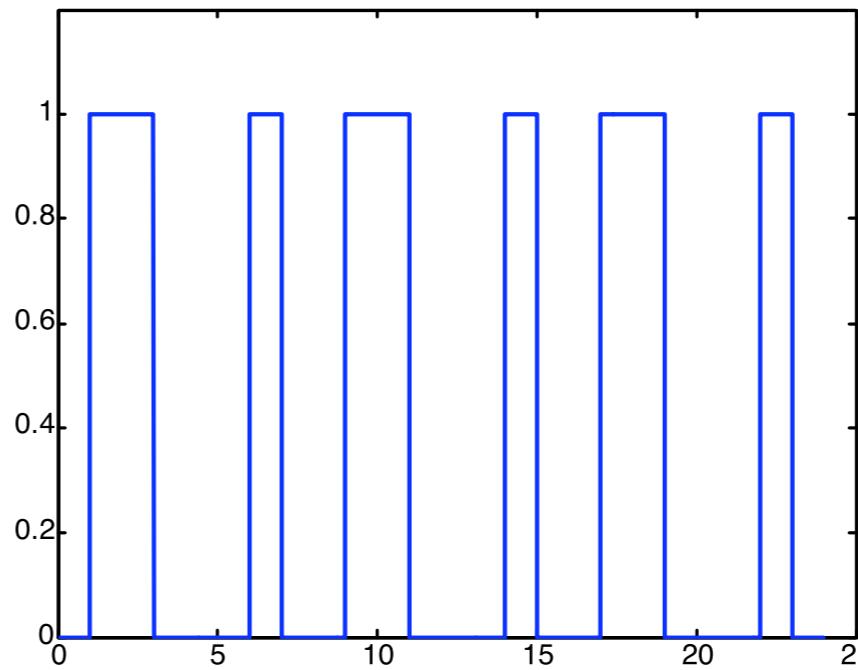
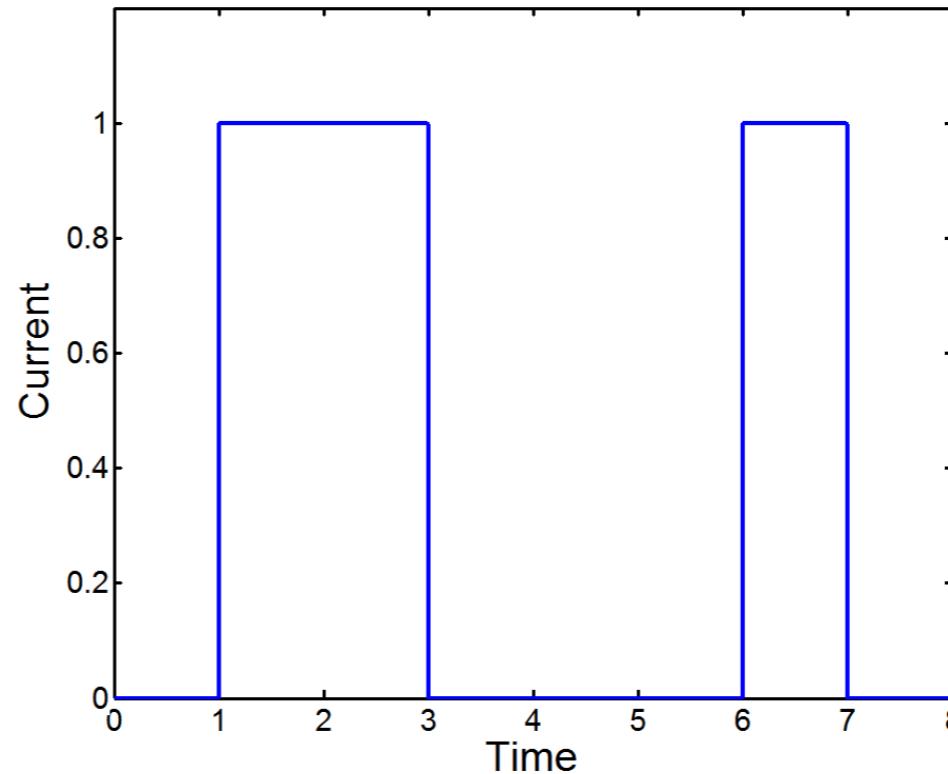
$$b_k = \frac{2}{T} \int_0^T g(t) \sin(2\pi n f t) dt$$

- Die Quadratsumme der k -ten Terme ist proportional zu der Energie, die in dieser Frequenz verbraucht wird:

$$(a_k)^2 + (b_k)^2$$

Anwendung der Fourier-Analyse

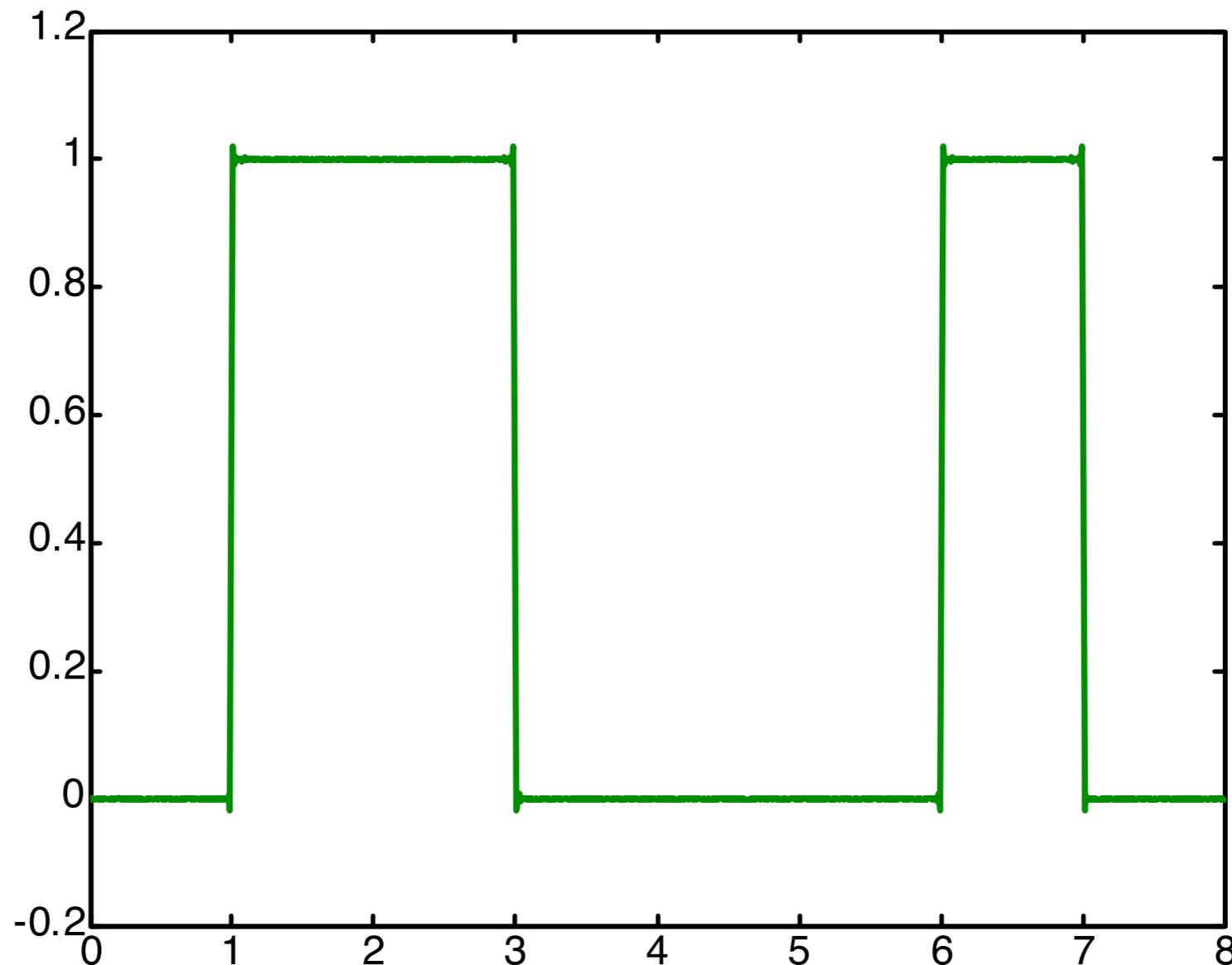
- Problem:
 - Signal ist nicht periodisch
- Lösung:
 - Wiederholung des Signals mit Periode 8



(aus Vorlesung von Holger Karl)

Anwendung der Fourier-Analyse

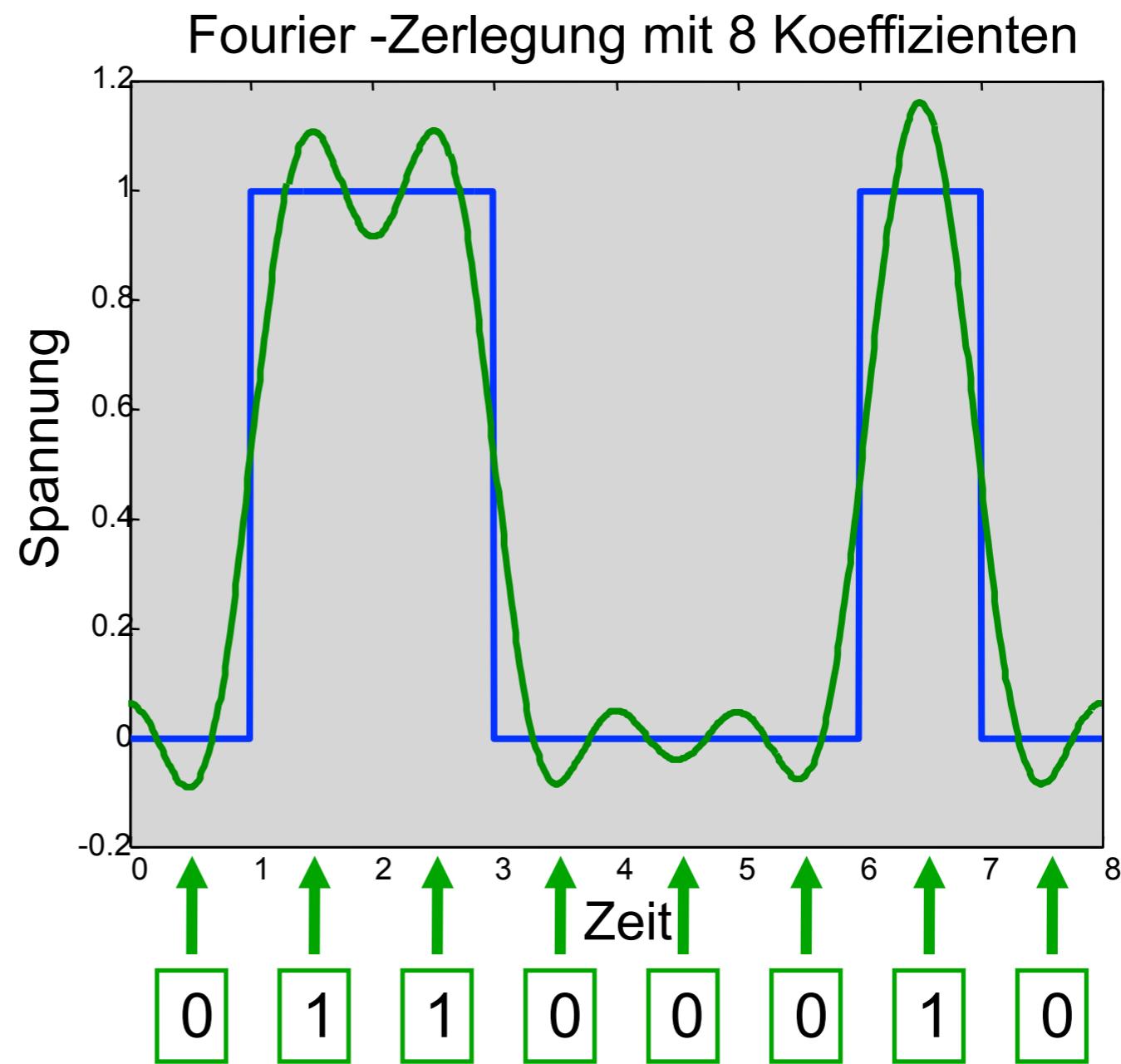
- Fourier-Analyse mit 512 Termen:



(aus Vorlesung von Holger Karl)

Wie oft muss man messen?

- Wie viele Messwerte sind notwendig, um eine Fouriertransformation bis zur k.-ten Komponenten genau zu bestimmen?
- Nyquist-Shannon-Abtasttheorem
 - Um ein kontinuierliches bandbegrenztes Signal mit einer Maximalfrequenz f_{\max} zu rekonstruieren, braucht man mindestens eine Abtastfrequenz von $2 f_{\max}$.



Nyquists Theorem

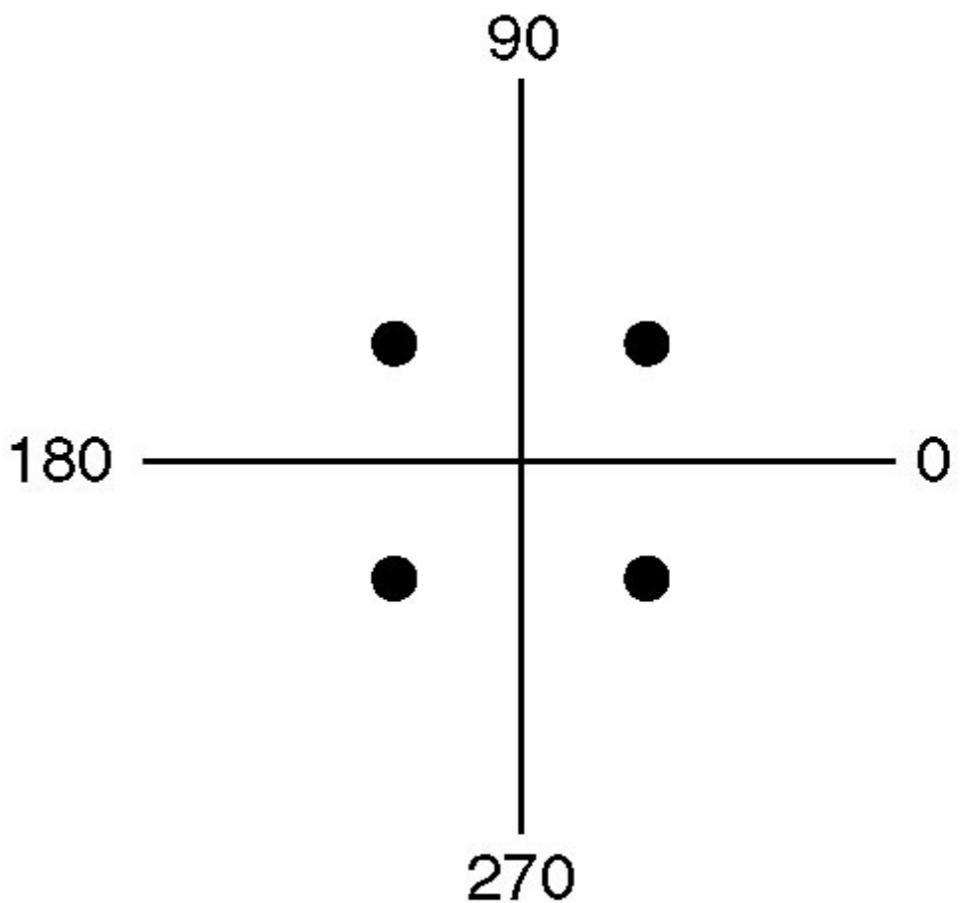
- Definition
 - Die Bandweite H ist die Maximalfrequenz in der Fourier-Zerlegung
- Angenommen:
 - Die maximale Frequenz des empfangenen Signals ist $f=H$ in der Fouriertransformation
 - (Komplette Absorption [unendliche Dämpfung] aller höheren Frequenzen)
 - Die Anzahl der verschiedenen verwendeten Symbole ist V
 - Es treten keinerlei anderen Störungen, Verzerrungen oder Dämpfungen auf
- Theorem von Nyquist
 - Die maximal mögliche Symbolrate ist höchstens $2 H$ baud.
 - Die maximal mögliche Datenrate ist höchstens $2 H \log_2 V$ bit/s.

Helfen mehr Symbole?

- Nyquists Theorem besagt, dass rein theoretisch die Datenrate mit der Anzahl der verwendeten Symbole vergrößert werden könnten
- Diskussion:
 - Nyquists Theorem liefert nur eine theoretische obere Schranke und kein Verfahren zur Übertragung
 - In der Praxis gibt es Schranken in der Messgenauigkeit
 - Nyquists Theorem berücksichtigt nicht das Problem des Rauschens

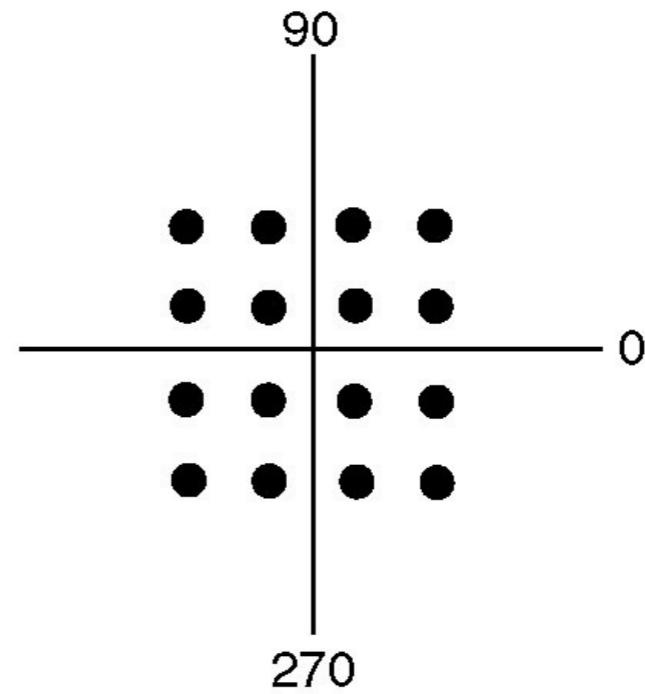
PSK mit verschiedenen Symbolen

- Phasenverschiebungen können vom Empfänger sehr gut erkannt werden
- Kodierung verschiedener Symbole sehr einfach
 - Man verwendet Phasenverschiebung z.B. $\pi/4$, $3/4\pi$, $5/4\pi$, $7/4\pi$
 - selten: Phasenverschiebung 0 (wegen Synchronisation)
 - Bei vier Symbolen ist die Datenrate doppelt so groß wie die Symbolrate
- Diese Methode heißt Quadrature Phase Shift Keying (QPSK)



Amplituden- und Phasenmodulation

- Amplituden- und Phasenmodulation können erfolgreich kombiniert werden
- Beispiel: 16-QAM (Quadrature Amplitude Modulation)
 - Man verwendet 16 verschiedene Kombinationen von Phasen und Amplituden für jedes Symbol
 - Jedes Symbol kodiert vier Bits ($2^4 = 16$)
 - Die Datenrate ist also viermal so groß wie die Symbolrate



Wiederholung: Komplexe Zahlen

- **i: imaginäre Zahl mit**
 - $i^2 = -1$
- **Komplexe Zahl ist lineare Kombination aus Realteil a und Imaginärteil b**
 - $z = a + bi$
- **Rechenregeln:**
 - $(a+bi)+(c+di) = (a+c) + (b+d)i$
 - $(a+bi)(c+di) = (ac - bd) + (ad + bc)i$
 - $1/(a+bi) = (a-bi)/(a^2+b^2)$
- **Komplex konjugierte Zahl**
 - $(a+bi)^* = (a - bi)$
 - $(a+bi)^*(a+bi) = a^2+b^2$

- Wichtige Gleichung:
 - $e^{i\pi} = -1$
 - $e^{i\varphi} = \cos \varphi + i \sin \varphi$
- Exponentiation einer komplexen Zahl
 - $e^{a+bi} = e^a e^{bi} = e^a (\cos b + i \sin b)$
 - Realteil von $e^{i\varphi}$: $\operatorname{Re}(e^{i\varphi}) = \cos \varphi$
 - Imaginärteil von $e^{i\varphi}$: $\operatorname{Im}(e^{i\varphi}) = \sin \varphi$

Äquivalente Darstellungen der FFT

- Realzahlendarstellung
 - Sinus und Cosinus-Funktionen der einzelnen Frequenzen

$$g(x) = \sum_{k=0}^{N-1} a_k \cos \frac{2\pi k t}{T} + b_k \sin \frac{2\pi k t}{T}$$

- Berechnung der Inversen durch Integralprodukt mit Cosinus/Sinus

$$a_k = \frac{2}{T} \int_0^T g(t) \cos(2\pi n f t) dt$$

$$b_k = \frac{2}{T} \int_0^T g(t) \sin(2\pi n f t) dt$$

- Komplexe Darstellung
 - Realteil der Exponentialfunktion der verschiedenen Frequenzen

$$f(x) = \sum_{k=0}^{N-1} z_k e^{i 2\pi k t / T}$$

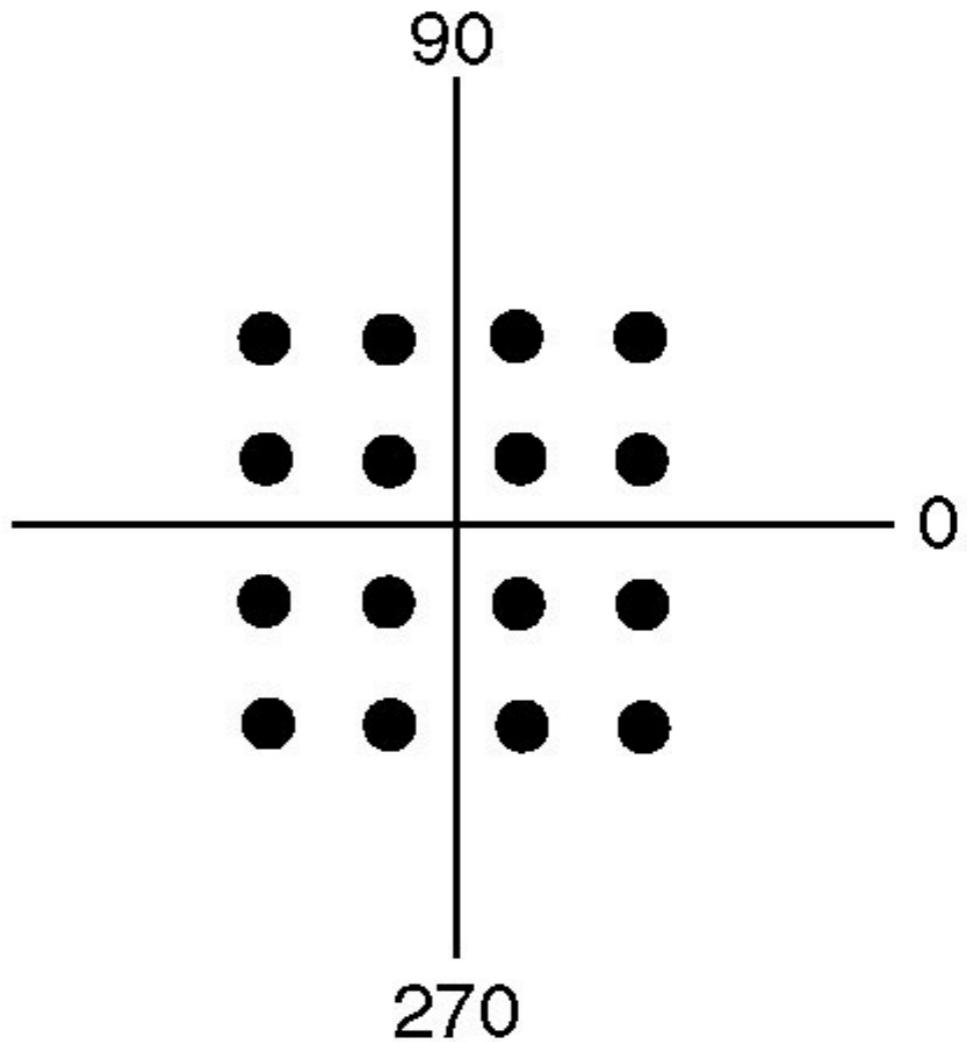
- Berechnung der Inversen durch Integral des Produkts mit der komplex konjugierten Trägerwelle

$$z_k = \frac{1}{T} \int_0^T \left(e^{i 2\pi k t / T} \right)^* f(x) dt$$

Vorteil der komplexen Darstellung

- Jedes Symbol des QAM kann direkt als komplexe Zahl dargestellt werden

$$f(x) = \sum_{k=0}^{N-1} z_k e^{i2\pi kt/T}$$

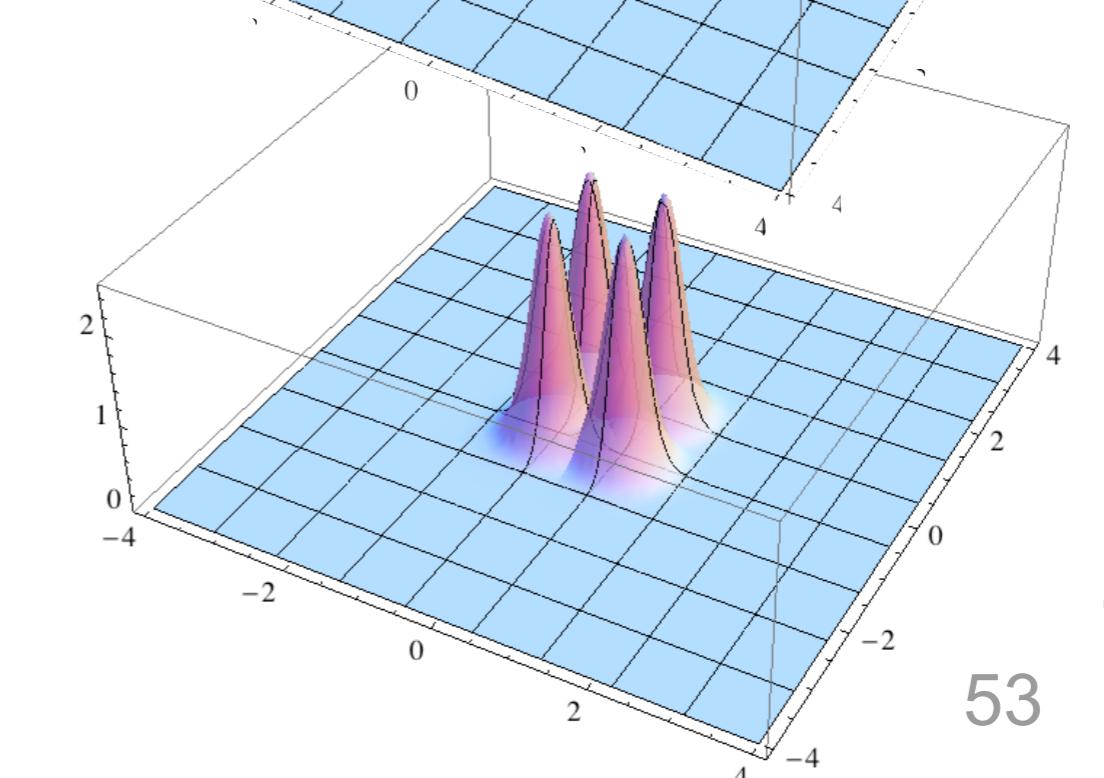
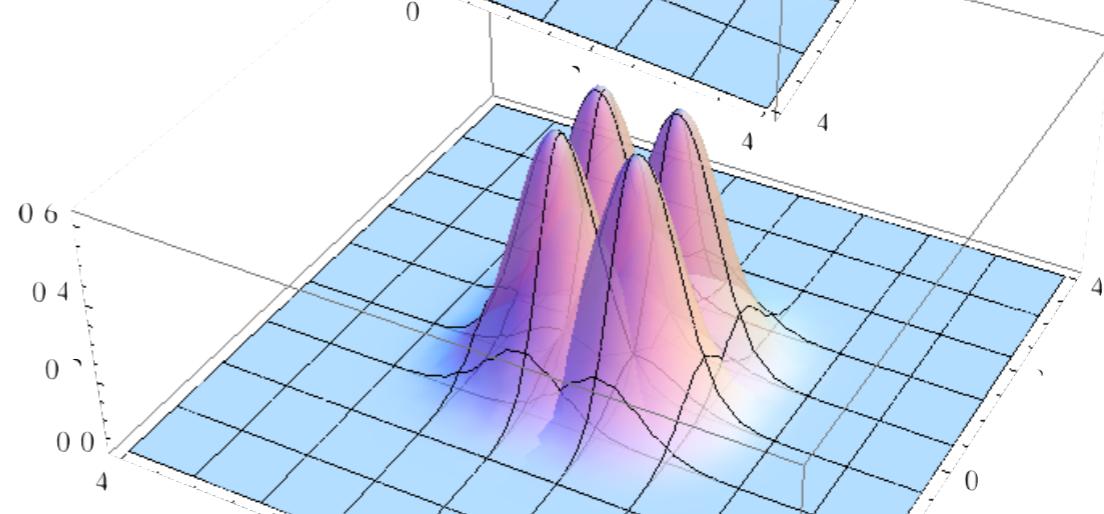
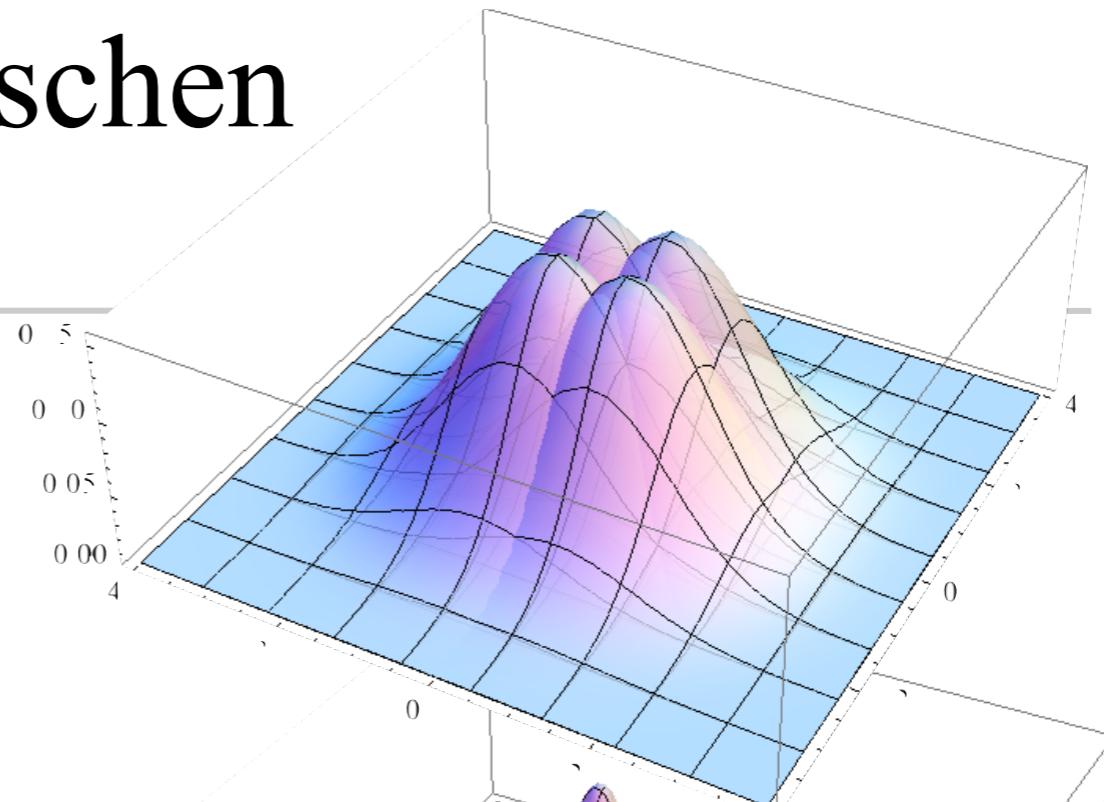


QAM und Rauschen

- Rauschen wird mit der Normalverteilung beschrieben

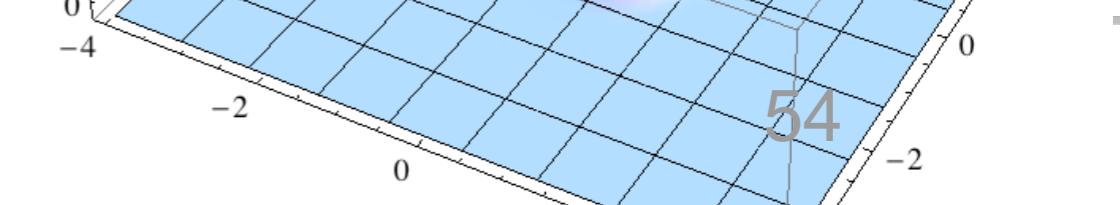
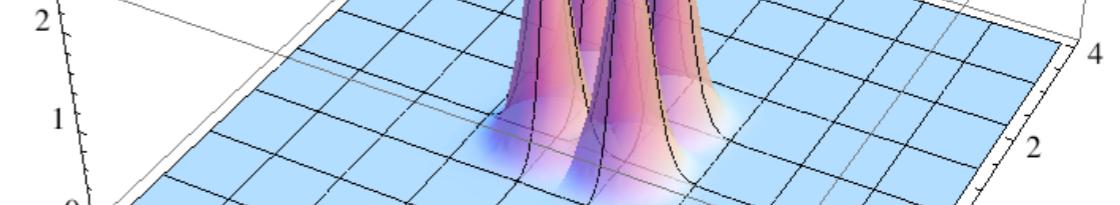
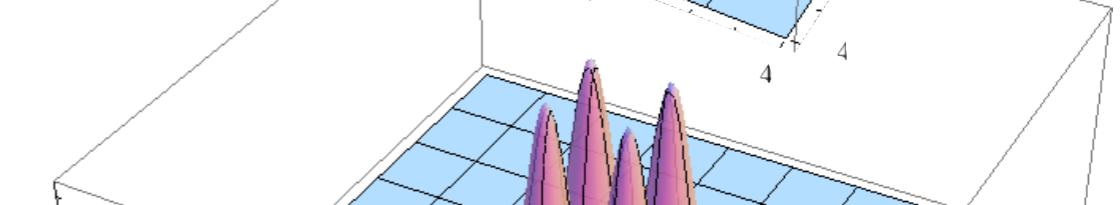
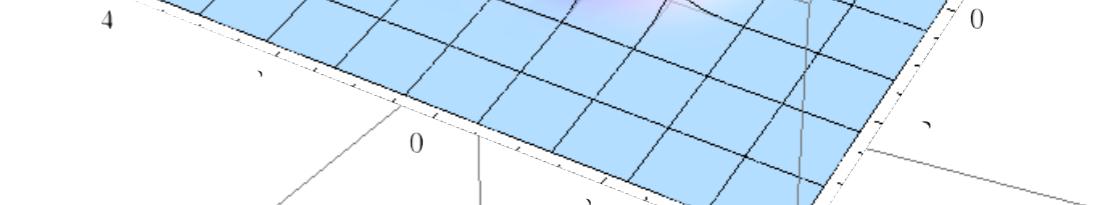
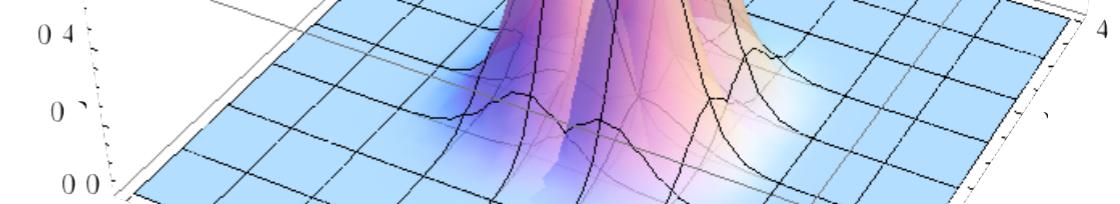
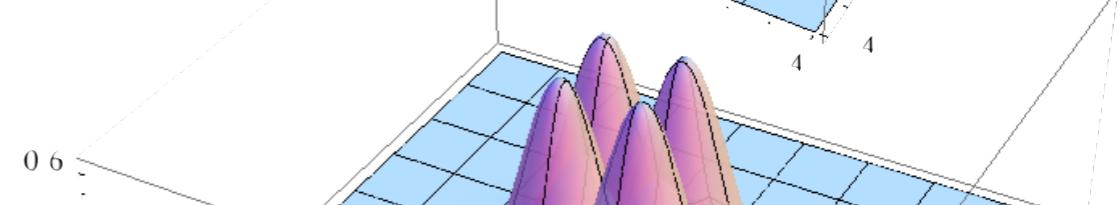
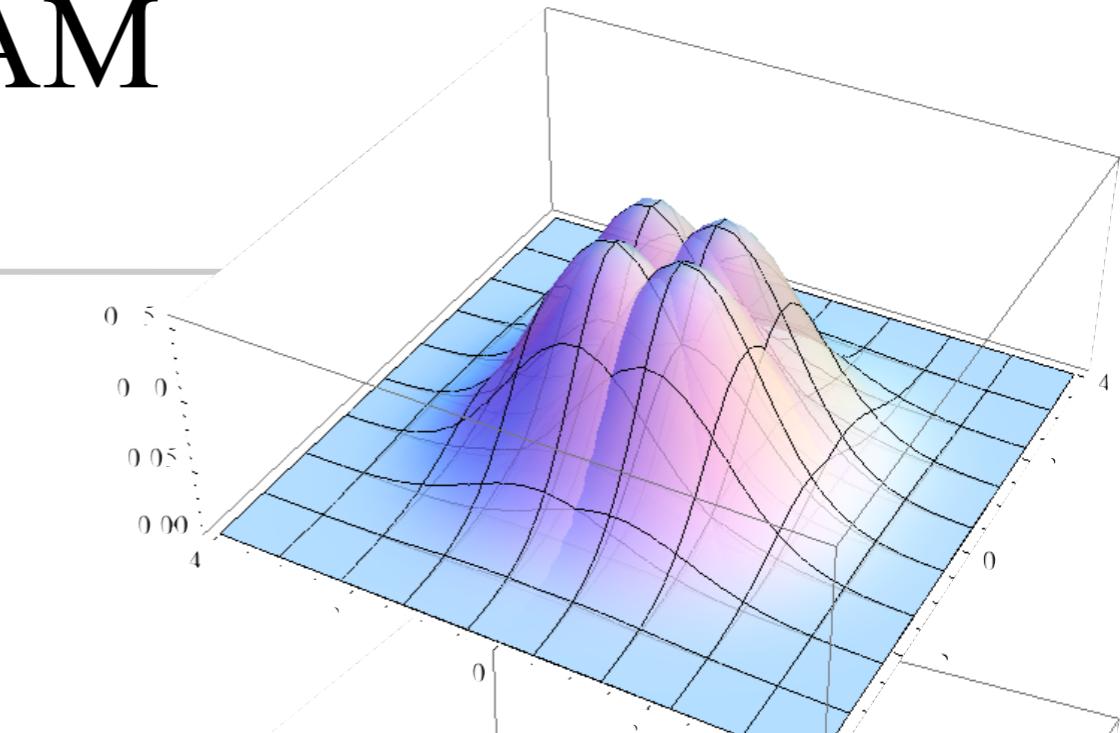
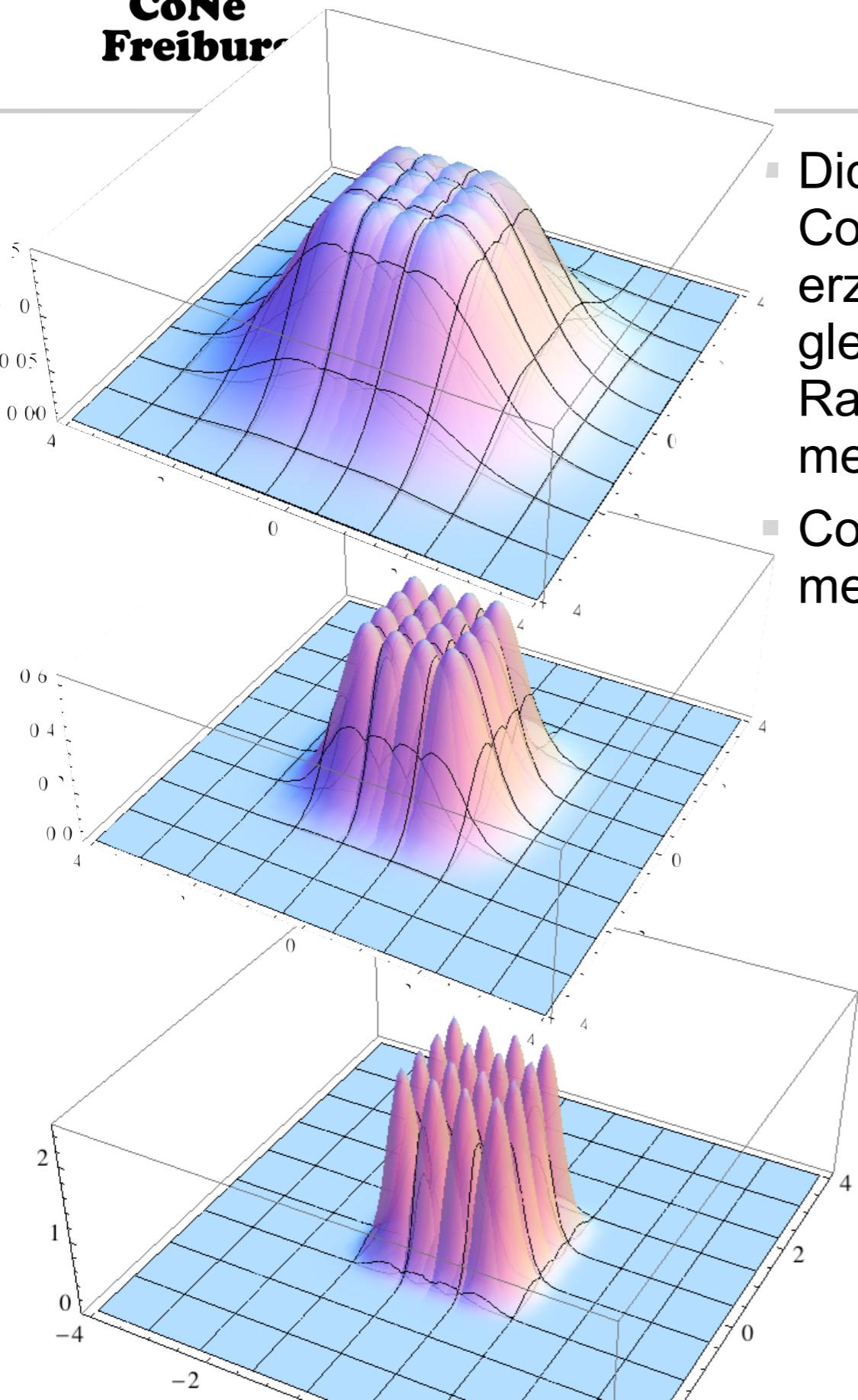
$$f(x) = \frac{1}{\sigma \cdot \sqrt{2\pi}} \cdot e^{-\frac{1}{2} \left(\frac{x - \mu}{\sigma} \right)^2}$$

- Bitfehler entstehen, wenn das dekodierte Signal zu stark abweicht
- Das Signal/Rauschverhältnis korreliert mit der Standardabweichung σ



QAM versus 16QAM

- Dichtere Codes erzeugen bei gleichem Rauschen mehr Fehler
- Codieren aber mehr Bits



Die Bitfehlerhäufigkeit und das Signalrauschverhältnis

- Je höher das Signal-Rausch-Verhältnis, desto geringer ist der auftretende Fehler
- Bitfehlerhäufigkeit (bit error rate - BER)
 - Bezeichnet den Anteil fehlerhaft empfangener Bits
- Abhängig von
 - Signalstärke,
 - Rauschen,
 - Übertragungsgeschwindigkeit
 - Verwendetem Verfahren
- Abhängigkeit der Bitfehlerhäufigkeit (BER) vom Signal-Rausch-Verhältnis
 - Beispiel:
4 QAM, 16 QAM, 64 QAM, 256 QAM

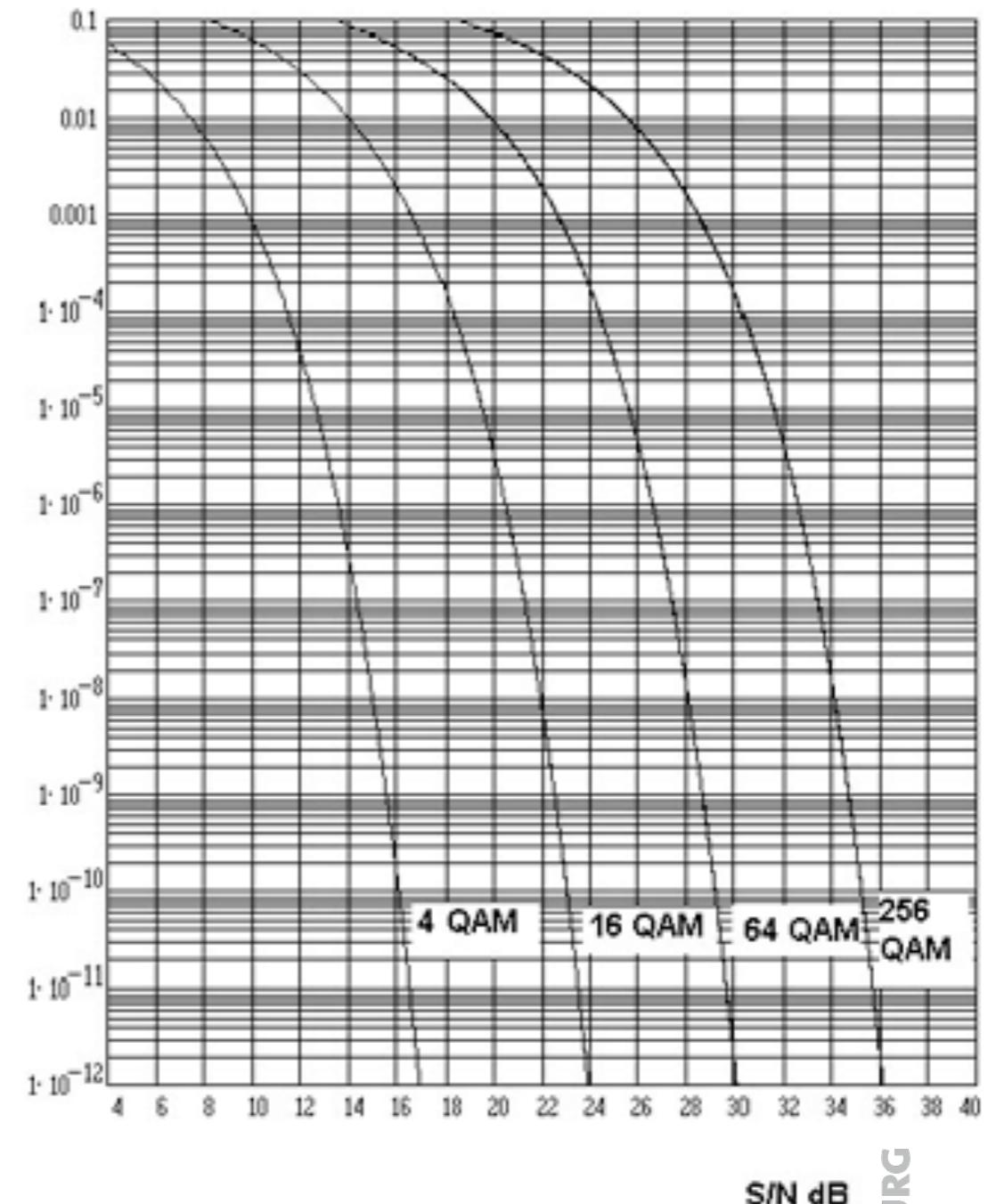


Abb. aus http://www.blondertongue.com/QAM-Transmodulator/Digital_Signal_Analysis.php

OFDM

- OFDM (Orthogonal Frequency Division Multiplex)
 - Signale werden in parallele Signalströme aufgeteilt
 - Parallele Signale werden auf Trägerwellen verschiedener Frequenzen Phasen/Amplituden moduliert
 - z.B. 16-QAM
 - Die Trägersignale werden zusammengefasst und gleichzeitig gesendet
- Sonderform der Frequenz-Multiplex-Verfahren
- Die Trägerwellen verwenden orthogonale Frequenzen:
 - Frequenzen $f, 2f, 3f, 4f, 5f, \dots$

Der Satz von Shannon

- Tatsächlich ist der Einfluss des Rauschens fundamental
 - Betrachte das Verhältnis zwischen Sendestärke S zur Stärke des Rauschens N
 - Je weniger Rauschen desto besser können Signale erkannt werden
- Theorem von Shannon
 - Die maximale mögliche Datenrate ist $H \log_2 (1+S/N)$ bit/s
 - bei Bandweite H
 - Signalstärke S
- Achtung
 - Dies ist eine theoretische obere Schranke
 - Existierende Kodierungen erreichen diesen Wert nicht

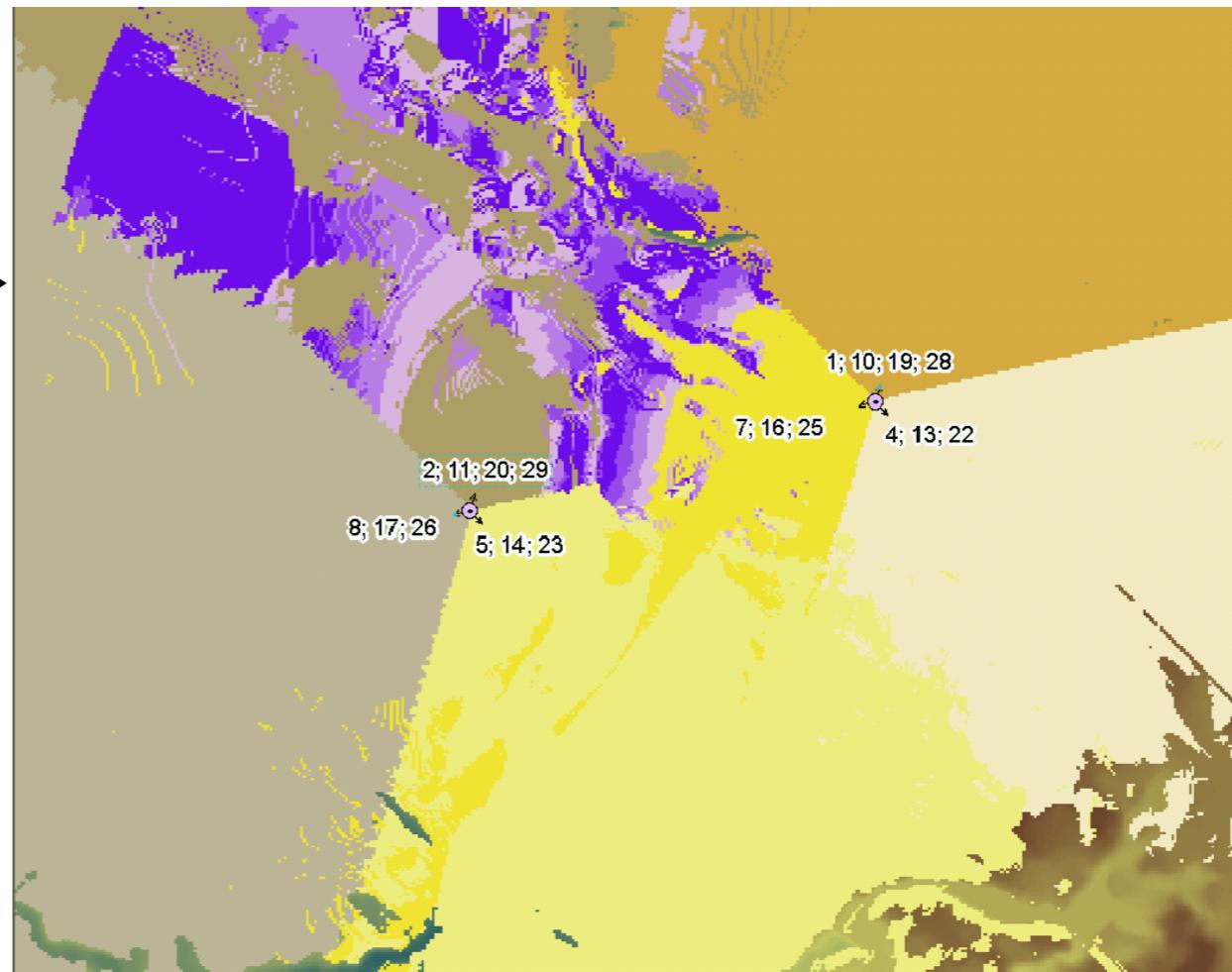
Mehrfachnutzung des Mediums

- Raummultiplexverfahren
 - Parallele und exklusive Nutzung von Übertragungskanäle
 - z.B. Extraleitungen/Zellen/Richtantenne
- Frequenzmultiplexverfahren
 - Mehrere zu übertragende Signale in einem Frequenzbereich gebündelt;
 - Bei Funkübertragung werden unterschiedlichen Sendern unterschiedliche Frequenzen zugewiesen.
- Zeitmultiplexverfahren
 - Zeitversetztes Senden mehrerer Signale
- Wellenlängenmultiplexverfahren
 - Optisches Frequenzmultiplexverfahren für die Übertragung in Glasfaserkabel
- Codemultiplexverfahren
 - Nur in Funktechnik: Kodierung des Signals in orthogonale Codes, die nun gleichzeitig auf einer Frequenz gesendet werden können
 - Dekodierung auch bei Überlagerung möglich

Raum

- Raumaufteilung (Space-Multiplexing)

- Ausnutzung des Abstandsverlusts zum parallelen Betriebs verschiedener Funkzellen → zellulare Netze
- Verwendung gerichteter Antennen zur gerichteten Kommunikations
 - GSM-Antennen mit Richtcharakteristik
 - Richtfunk mit Parabolantenne
 - Laserkommunikation
 - Infrarotkommunikation



- Zeitaufteilung (Time-Multiplexing)
 - Zeitliche Aufteilung des Sende-/Empfangskanals
 - Verschiedene Teilnehmer erhalten exklusive Zeiträume (Slots) auf dem Medium
 - Genaue Synchronisation notwendig
 - Koordination notwendig, oder starre Einteilung
- Wird in der Medium-Zugriffsschicht koordiniert

Frequenz

■ Frequenzmultiplex

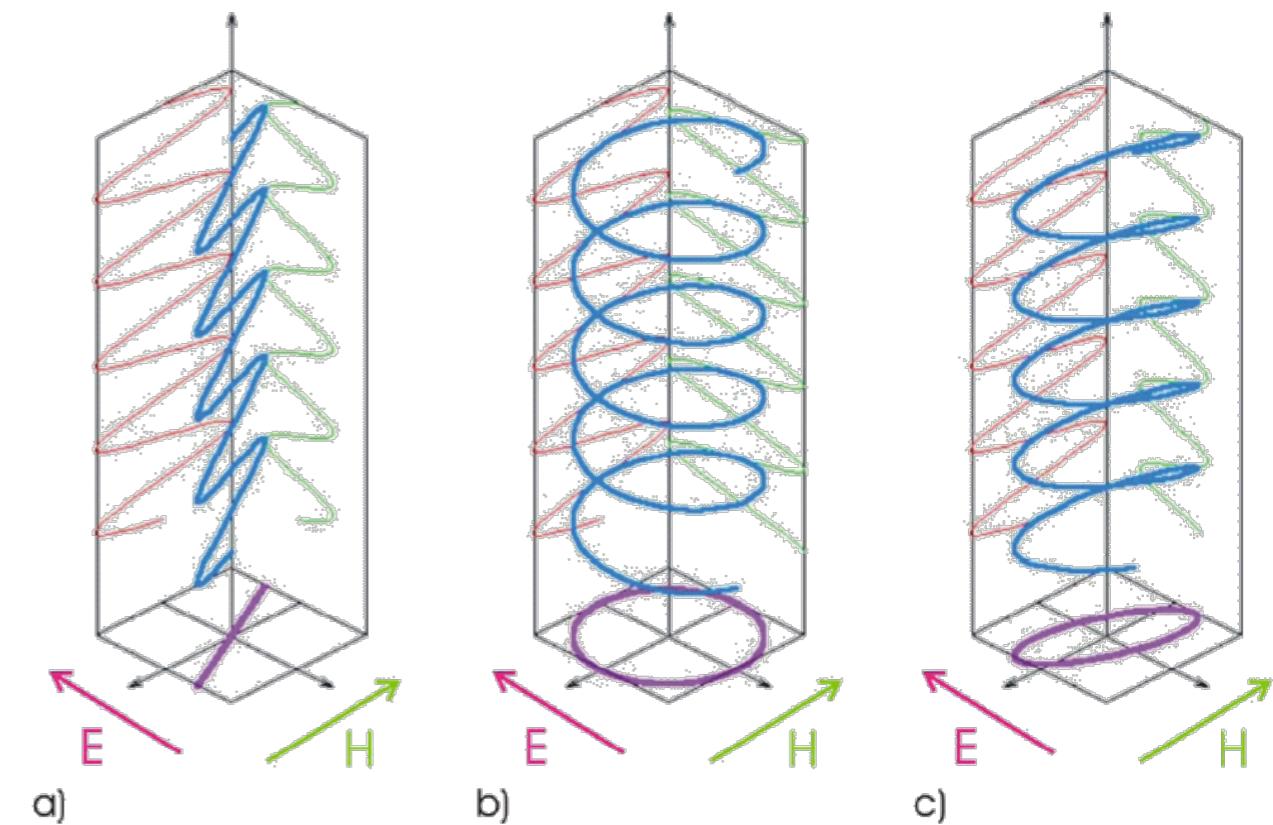
- Aufteilung der Bandbreite in Frequenzabschnitte
- Spreizen der Kanäle und Hopping
 - Direct Sequence Spread Spectrum (DSSS)
 - Xor eines Signals mit einer Folge Pseudozufallszahlen beim Sender und Empfänger (Verwandt mit Codemultiplex)
 - Fremde Signale erscheinen als Hintergrundrauschen
 - Frequency Hopping Spread Spectrum (FHSS)
 - Frequenzwechsel durch Pseudozufallszahlen
 - Zwei Versionen
 - Schneller Wechsel (fast hopping): Mehrere Frequenzen pro Nutzdatenbit
 - Langsamer Wechsel (slow hopping): Mehrere Nutzdatenbits pro Frequenz

Code

- CDMA (Code Division Multiple Access)
 - z.B. GSM (Global System for Mobile Communication)
 - oder UMTS (Universal Mobile Telecommunications System)
- Beispiel:
 - Sender A:
 - 0 ist $(-1, -1)$
 - 1 ist $(+1, +1)$
 - Sender B:
 - 0 ist $(-1, +1)$
 - 1 ist $(+1, -1)$
 - A sendet 0, B sendet 0:
 - Ergebnis: $(-2, 0)$
 - C empfängt $(-2, 0)$:
 - Dekodierung bzgl. A: $(-2, 0) \cdot (-1, -1) = (-2)(-1) + 0(-1) = 2$
 - A hat also 0 gesendet (da Ergebnis positiv)

Polarization-division multiplexing

- Spezialfall des Wellenlängenmultiplex-Verfahren
- Polarisation
 - Durch die Bewegungsrichtung der elektrischen Ladung ergibt sich eine Polarisation
- Z.B.
 - linear: horizontal, vertikal
 - zirkular
 - elliptisch (allgemeiner Fall)
- Die Verwendung verschiedener Polarisationen kann zur Trennung oder zur Modulation verwendet werden
 - in Kombination mit QPSK = 4-PSK
 - Z.B. 112 Gb/s PM-QPSK in Glasfaser mit Übertragungen bis zu 6000 km mit 200 km Distanz zwischen den Verstärkern



<http://optikwiki.harzoptics.de/doku.php?id=polarisation>

- Leitungsgebundene Übertragungsmedien
 - Kupferdraht – Twisted Pair
 - Kupferdraht – Koaxialkabel
 - Glasfaser
- Drahtlose Übertragung
 - Funkübertragung
 - Mikrowellenübertragung
 - Infrarot
 - Lichtwellen

Twisted Pair



(a)

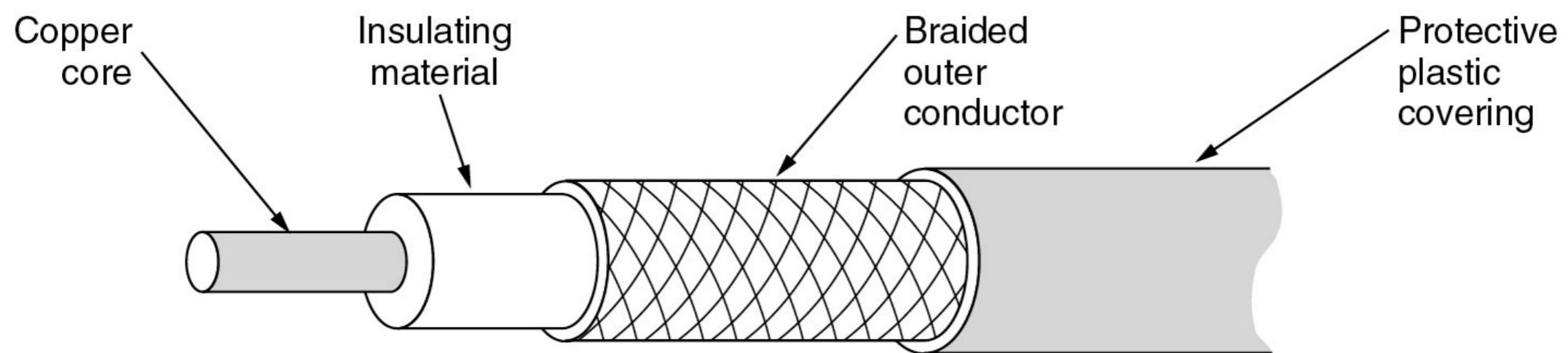


(b)

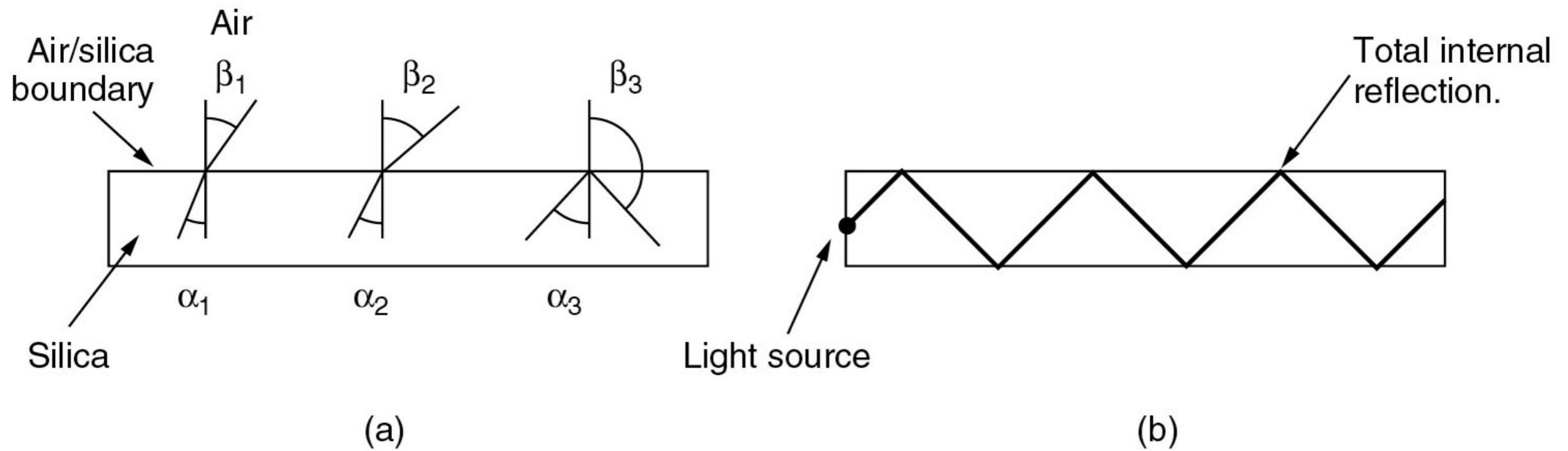
(a) Category 3 UTP.

(b) Category 5 UTP.

Koaxialkabel



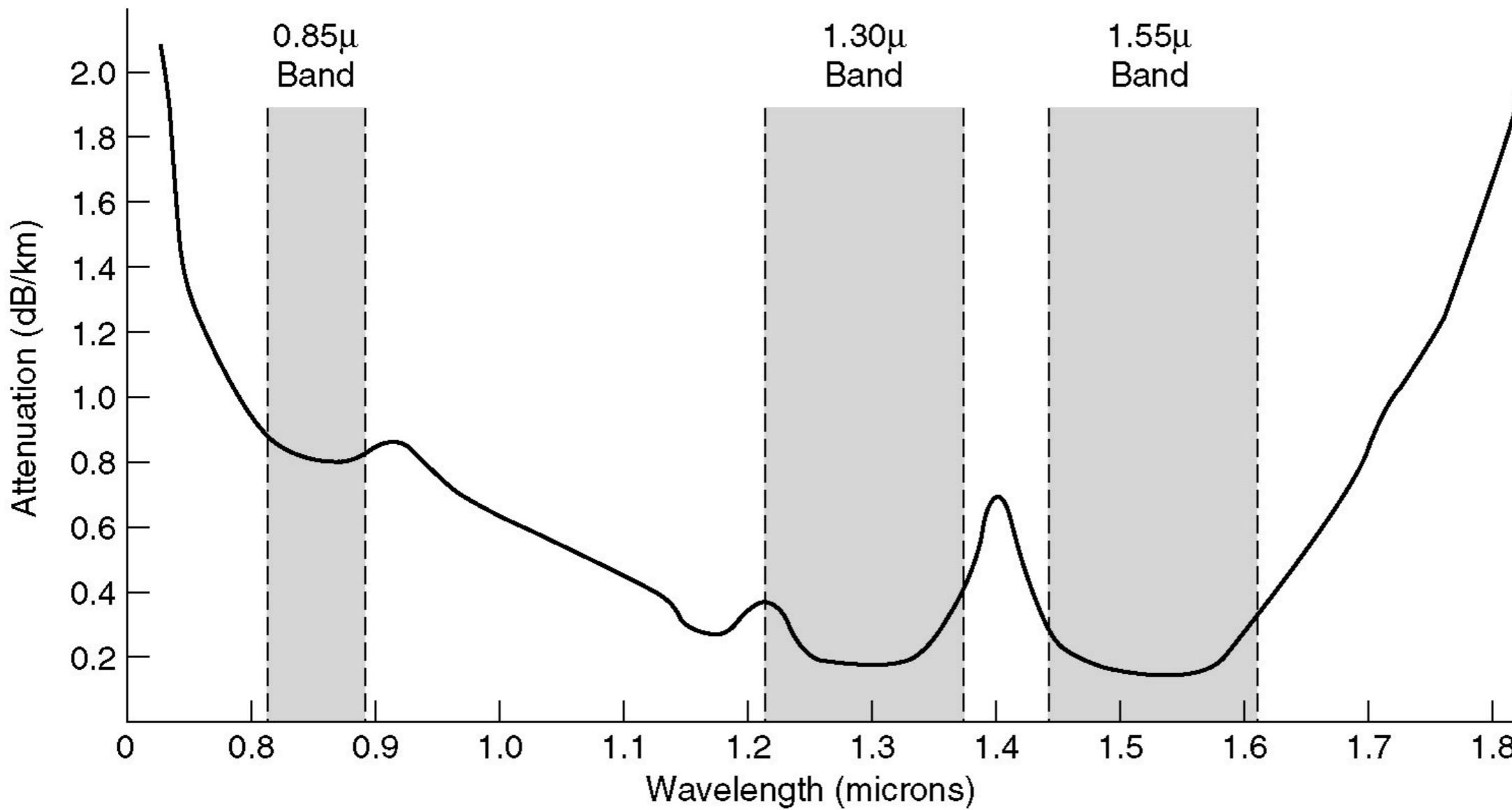
Glasfaser



Gesetz von Snellius:
$$\frac{\sin \alpha}{\sin \beta} = \frac{c_{\text{Glas}}}{c_{\text{Luft}}}$$

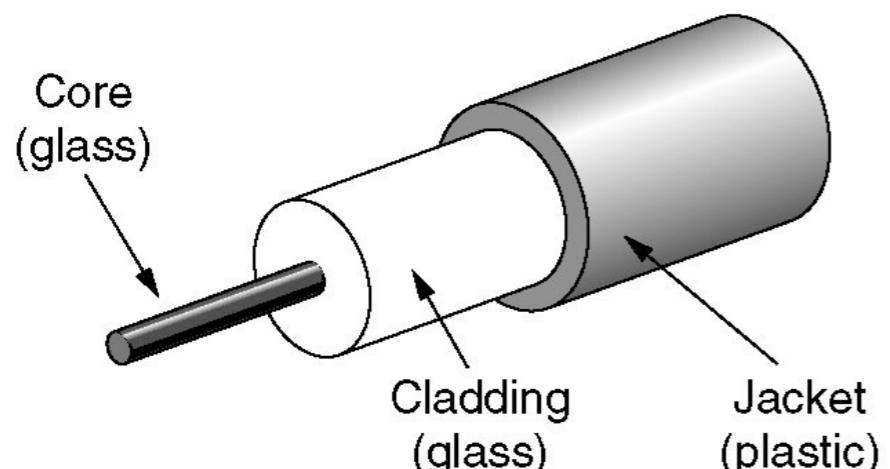
- (a) Beugung und Reflektion an der Luft/Silizium-Grenze bei unterschiedlichen Winkeln
- (b) Licht gefangen durch die Reflektion

- Dämpfung von Infrarotlicht in Glasfaser

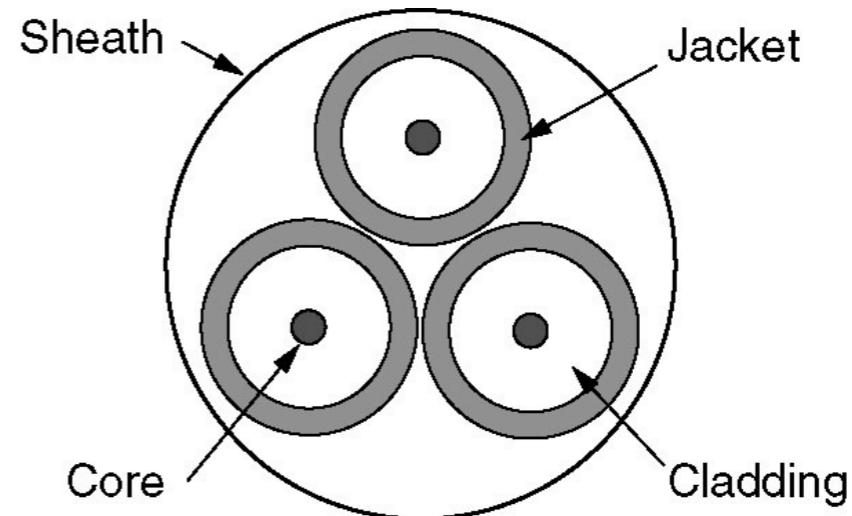


Glasfaser

- (a) Seitenansicht einer einfachen Faser
- (b) Schnittansicht eines Dreier-Glasfaserbündels



(a)



(b)

Fallbeispiel: Ethernet

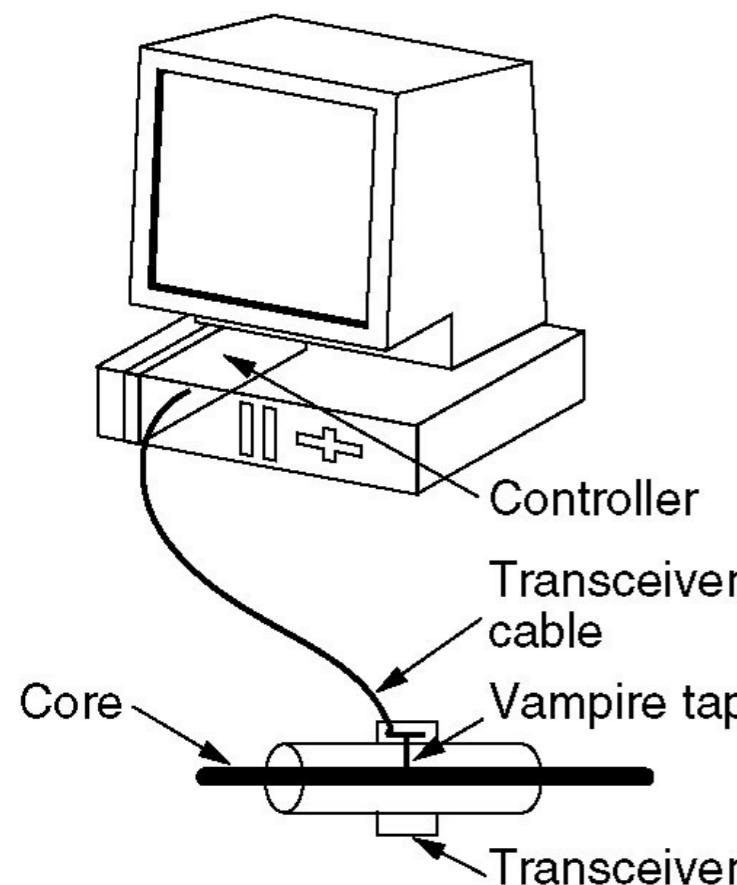
- Beispiel aus der Praxis mit Mediumzugriff:
Ethernet
 - IEEE Standard 802.3
- Punkte im Standard
 - Verkabelung
 - Bitübertragungsschicht
 - Sicherungsschicht mit Mediumzugriff

Bitübertragungsschicht Ethernet

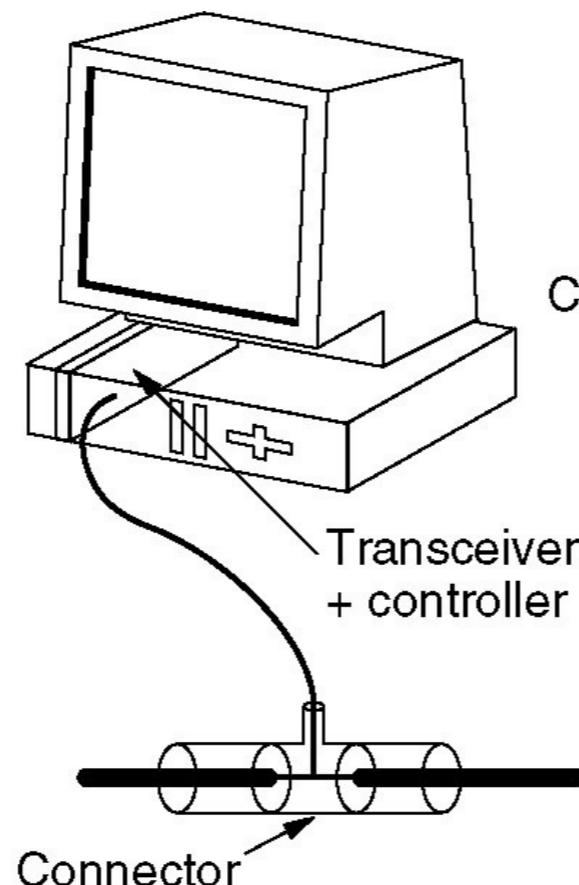
- Mediumabhängig
- Typisch: Manchester encoding
 - mit +/- 0.85 V
- Code-Verletzung zeigt Frame-Grenzen auf

Ethernet cabling

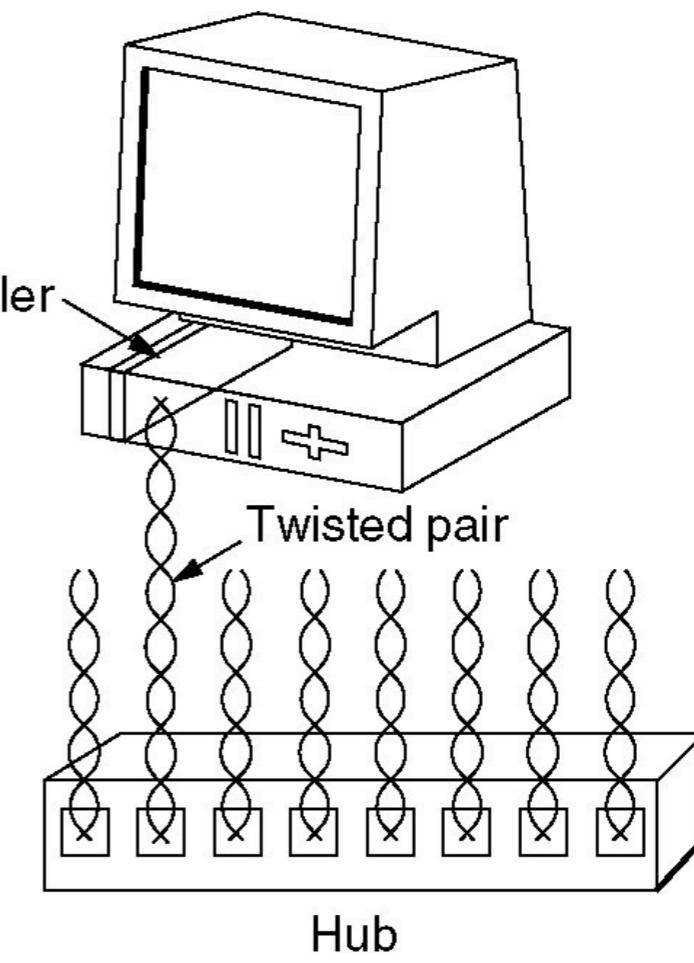
Name	Cable	Max. seg.	Nodes/seg.	Advantages
10Base5	Thick coax	500 m	100	Original cable; now obsolete
10Base2	Thin coax	185 m	30	No hub needed
10Base-T	Twisted pair	100 m	1024	Cheapest system
10Base-F	Fiber optics	2000 m	1024	Best between buildings



10Base5



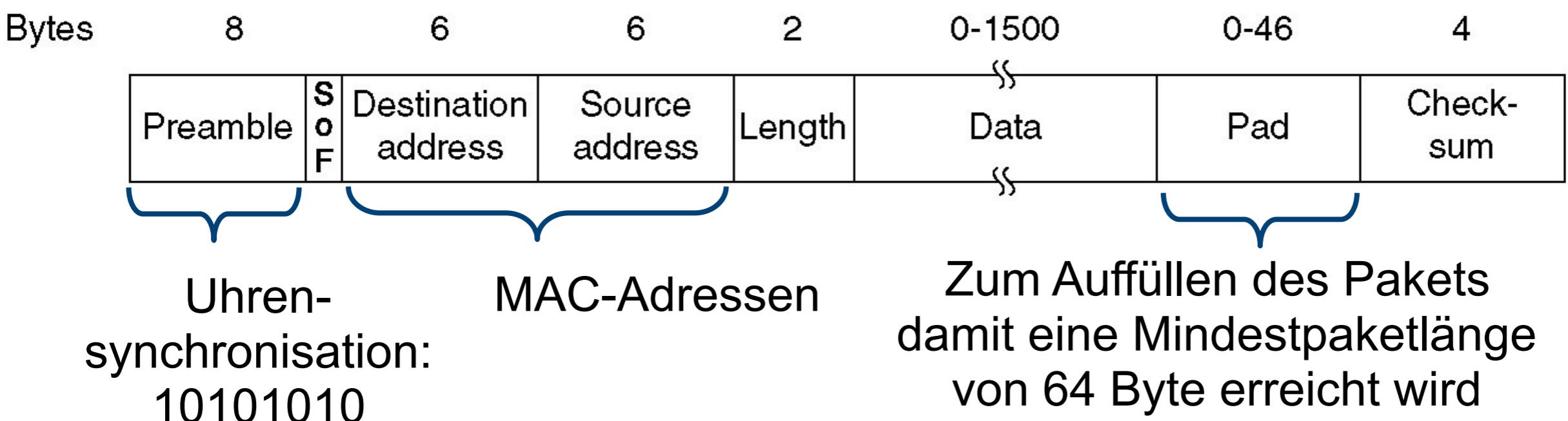
10Base2



10BaseT

Ethernet MAC-Schicht

- Im wesentlichen: CSMA/CD mit binary exponential backoff
- Frame-Format



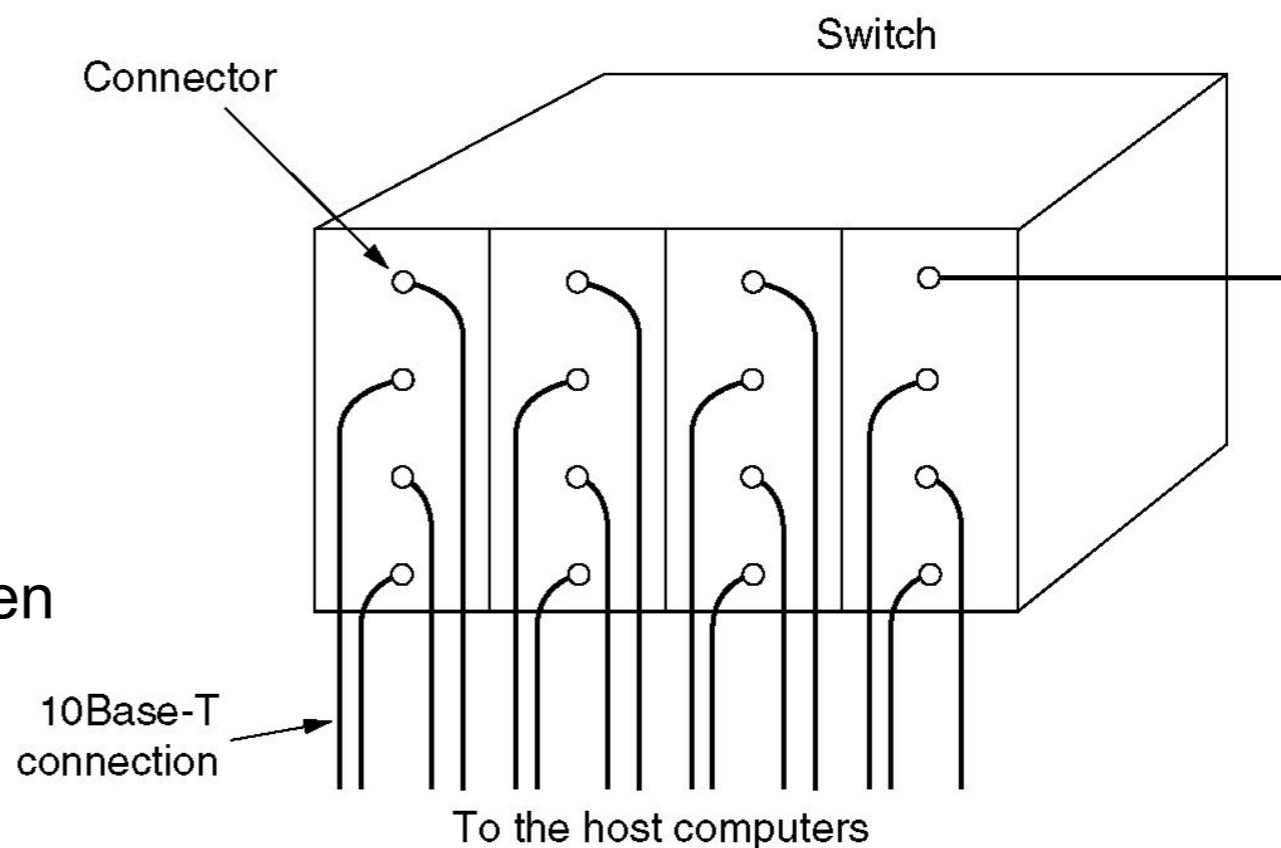
Switch versus Hub

■ Hub

- verknüpft Ethernet-Leitungen nabenförmig
- jede Verbindung hört alles
- Durch CSMA/CD wird die Übertragungsrate reduziert

■ Switch

- unterteilt die eingehenden Verbindungen in kleinere Kollisionsteilmengen
- die Prüfsumme eines eingehenden Pakets wird überprüft
- Kollisionen werden nicht weiter gegeben
- interpretiert die Zieladresse und leitet das Paket nur in diese Richtung weiter



Fast Ethernet

- Ursprünglich erreichte Ethernet 10 MBit/s
- 1992: Fast Ethernet
 - Ziele: Rückwärtskompatibilität
 - Resultat: 802.3u
- Fast Ethernet
 - Frame-Format ist gleichgeblieben
 - Bit-Zeit wurde von 100 ns auf 10 ns reduziert
 - Dadurch verkürzt sich die maximale Kabellänge (und die minimale Paket-Größe steigt).
 - Unvermeidbare Kollisionen CSMA

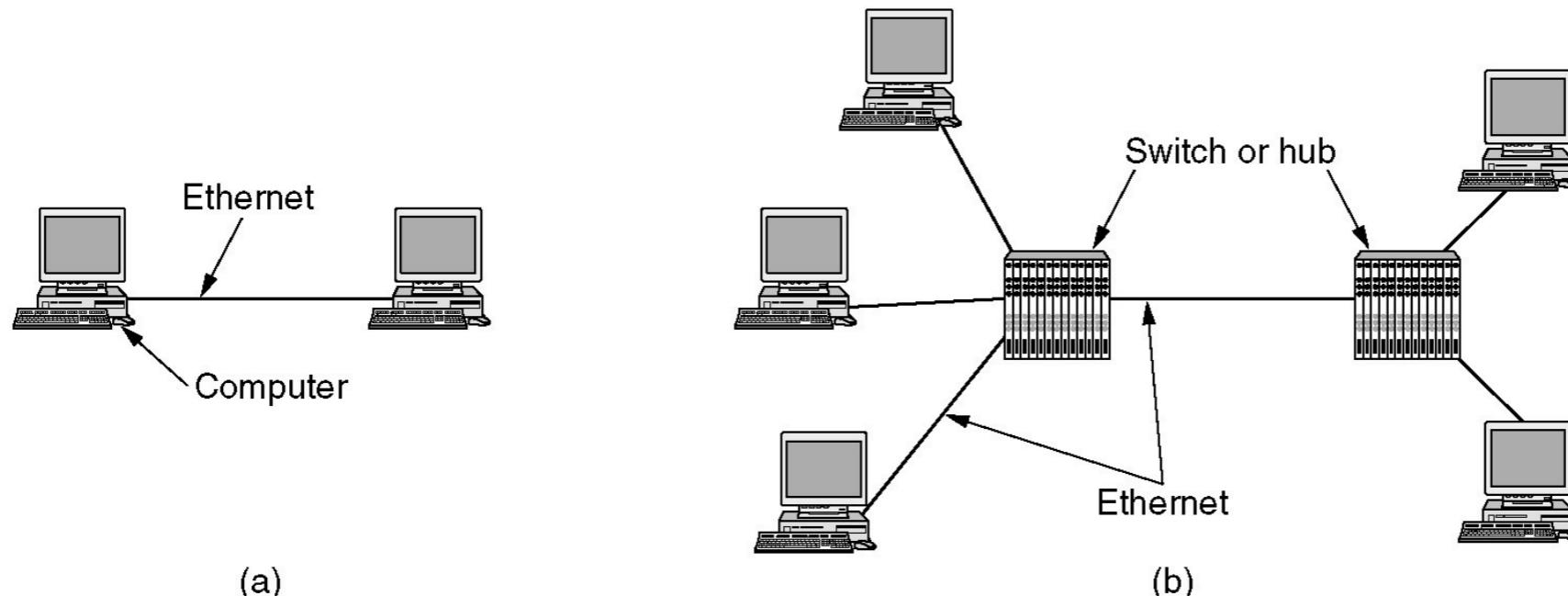
Fast Ethernet – Verkabelung

- Standard Cat-3 twisted pair unterstützt nicht 200 MBaud über 100 m
 - Lösung: Verwendung von 2 Kabelpaaren bei reduzierter Baudrate
- Wechseln von Manchester auf 4B/5B-Kodierung auf Cat-5-Kabeln

Name	Cable	Max. segment	Advantages
100Base-T4	Twisted pair	100 m	Uses category 3 UTP
100Base-TX	Twisted pair	100 m	Full duplex at 100 Mbps
100Base-FX	Fiber optics	2000 m	Full duplex at 100 Mbps; long runs

Gigabit Ethernet

- Gigabit-Ethernet: 1995
 - Ziel: Weitgehende Übernahme des Ethernet-Standards
- Ziel wurde erreicht durch Einschränkung auf Punkt-zu-Punkt-Verbindungen
 - In Gigabit-Ethernet sind an jedem Kabel genau zwei Maschinen
 - oder zumindestens ein Switch oder Hub



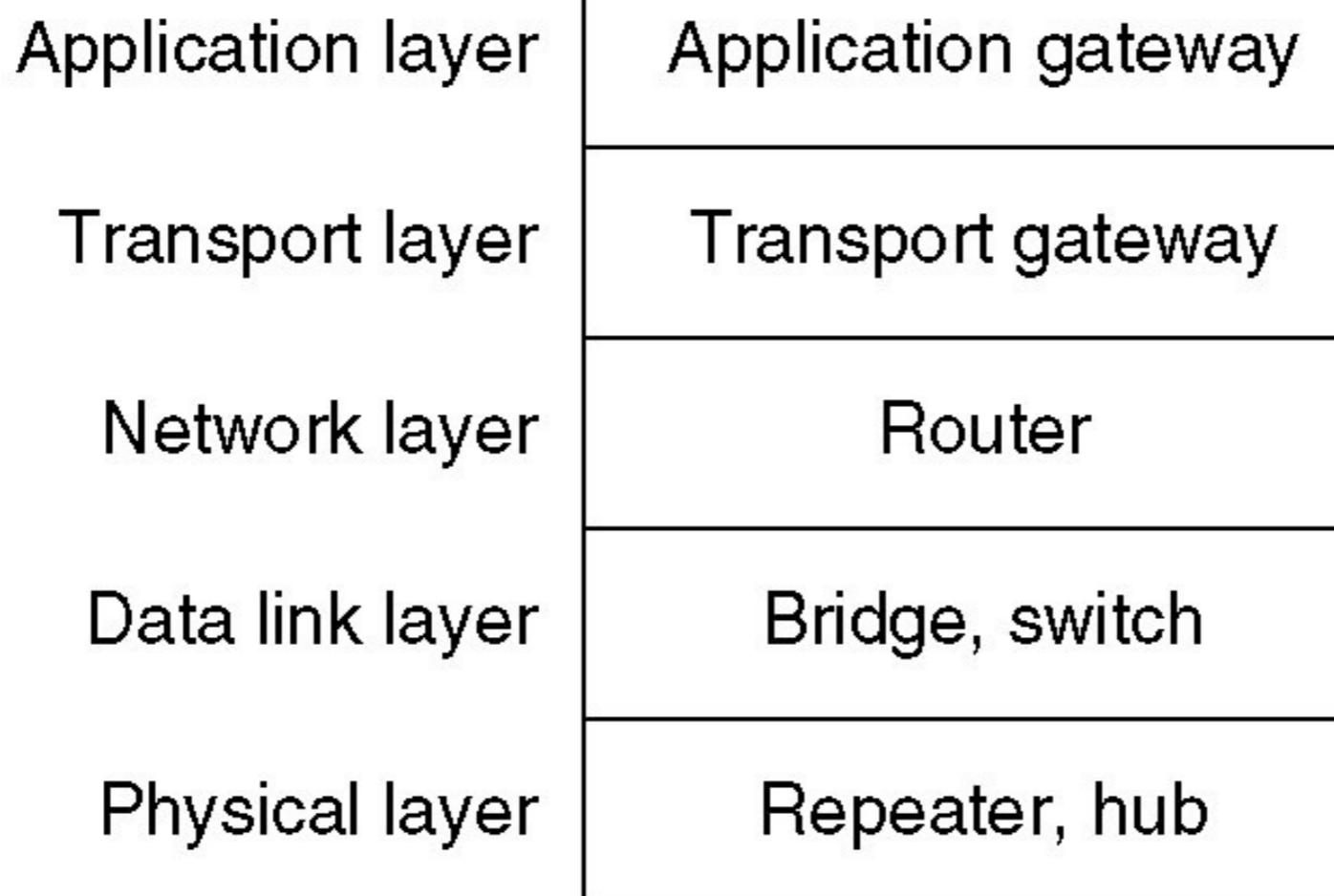
Gigabit Ethernet

- Mit Switch
 - Keine Kollisionen → CSMA/CD unnötig
 - Erlaubt full-duplex für jeden Link
- Mit Hub
 - Kollisionen, nur Halb-Duplex (d.h. abwechselnd Simplex), CSMA/CD
 - Kabellängen auf 25 m reduziert

Gigabit Ethernet – Cabling

Name	Cable	Max. segment	Advantages
1000Base-SX	Fiber optics	550 m	Multimode fiber (50, 62.5 microns)
1000Base-LX	Fiber optics	5000 m	Single (10 μ) or multimode (50, 62.5 μ)
1000Base-CX	2 Pairs of STP	25 m	Shielded twisted pair
1000Base-T	4 Pairs of UTP	100 m	Standard category 5 UTP

Verbinden von LANs



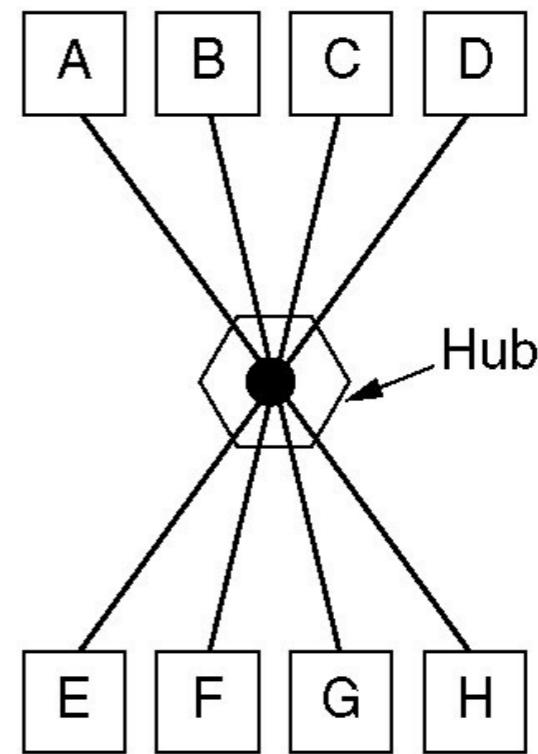
Repeater

- Signalregenerator
 - Empfängt Signal und bereitet es auf
 - Nur das elektrische und optische Singal wird aufbereitet
 - Information bleibt unbeeinflusst
- Bitübertragungsschicht
- Repeater teilen das Netz in physische Segmente
 - logische Topologien bleiben erhalten

Hub

- Verbindet sternförmig Netzsegmente
 - im Prinzip wie ein Repeater
 - Signale werden auf alle angebundenen Leitungen verteilt

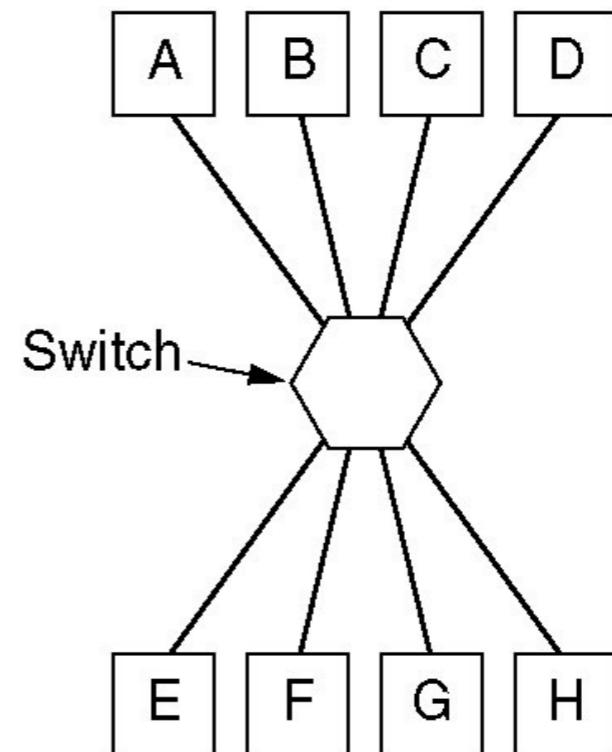
- Bitübertragungsschicht
 - Information und Logik der Daten bleibt unberücksichtigt
 - Insbesondere für Kollisionen



Switch

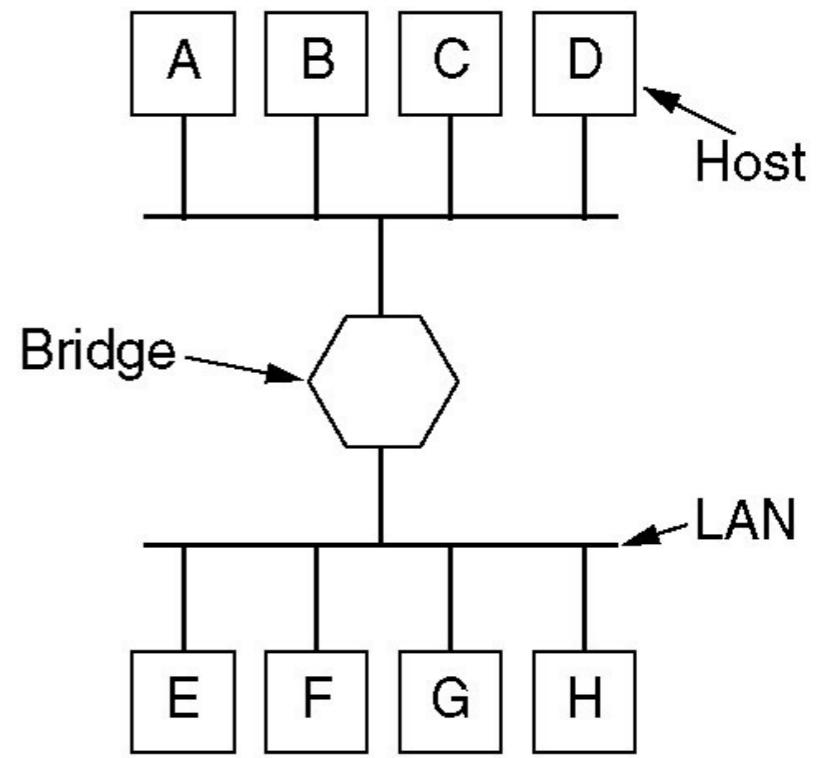
- Verbindet sternförmig Netzsegmente
 - Leitet die Daten nur in die betreffende Verbindung weiter
 - Gibt keine Kollisionen weiter

- Sicherungsschicht
 - Signale werden neu erzeugt
 - Kollisionen abgeschirmt und reduziert
 - Frames aber nicht verwendet
 - Rudimentäre Routingtabelle durch Beobachtung, wo Nachrichten herkommen



Bridge

- Verbindet zwei lokale Netzwerke
 - im Gegensatz zum Switch (dort nur Terminals)
 - trennt Kollisionen
- Sicherungsschicht
 - Weitergabe an die andere Seite, falls die Ziel-Adresse aus dem anderen Netzwerk bekannt ist oder auf beiden Seiten noch nicht gehört wurde
 - Nur korrekte Frames werden weitergereicht
 - Übergang zwischen Bridge und Switch ist fließend



Beispiel: Internet über Telefon

- Analog
 - typisch 3-4 kBit/s
 - maximal bis 56 kBit/s
- ISDN (Integrated Services Digital Network)
 - 128 kBit/s (Nutzdaten)
 - Hin/Rückrichtung jeweils 64 kBit/s
 - Pulse-Code Modulation (Amplitudenmodulation)
- DSL
 - maximal
 - bis 25 Mbit/s Downstream
 - bis 3,5 Mbit/s Upstream
 - typisch (DSL 6000)
 - 6 Mbit/s Downstream
 - 0,5 Mbit/s Upstream

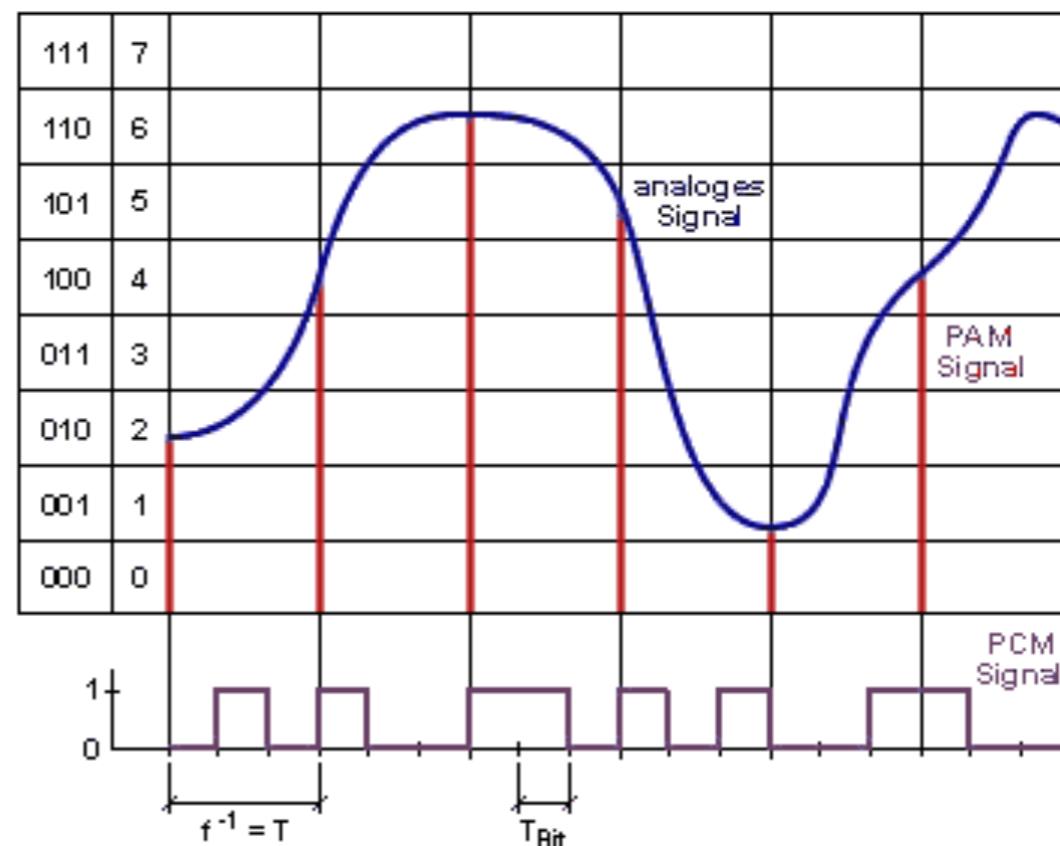
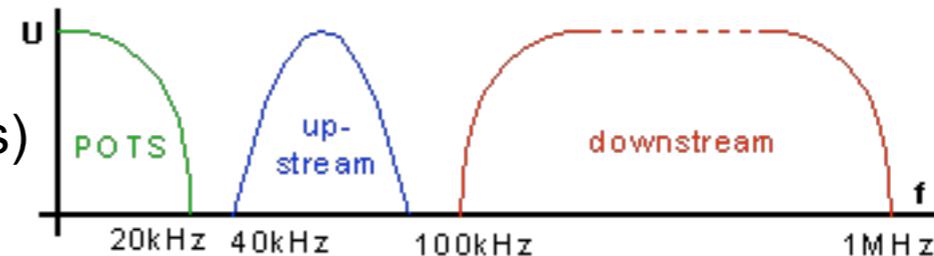


Abb. aus <http://de.wikipedia.org/wiki/Puls-Code-Modulation>

Beispiel DSL

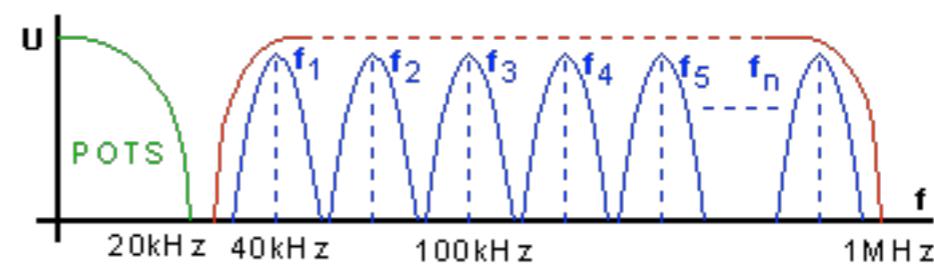
■ Asymmetric Digital Subscriber Line (ADSL)

- momentan der Standard zur Anbindung von Endverbrauchern zu ISP (Internet Service Providers)
- verwendet herkömmliche Kupferkabel



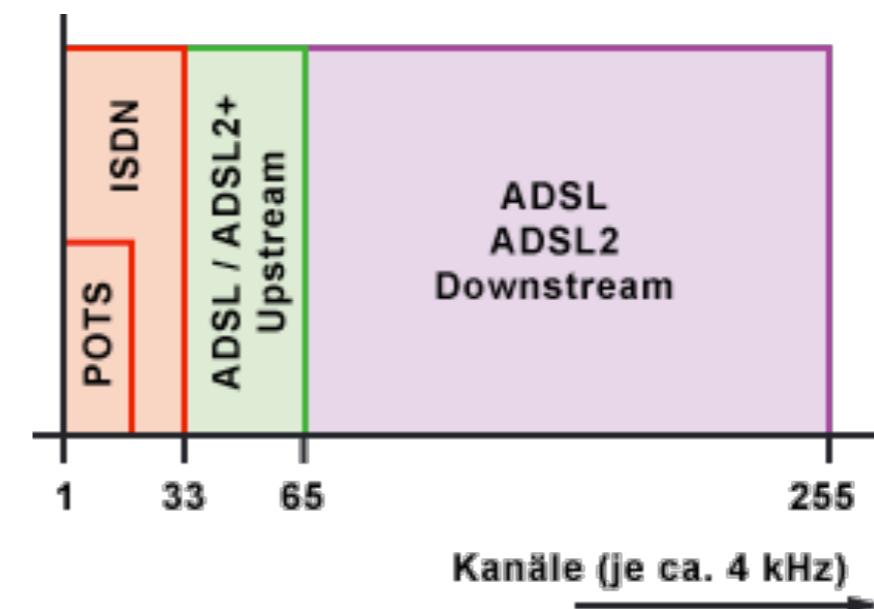
■ Übertragungsverfahren:

- Carrier-less Amplitude/Phase Modulation CAP (wie QAM)
 - Eine Modulation für Upstream/Downstream
- Discrete Multitone Modulation (DMT)
 - 256 Kanäle mit je 4 kHz Bandbreite



■ DMT: 3 Kanälstränge:

- POTS/ISDN (public switched telephone network/ Integrated Services Digital Network)
 - bleibt im Frequenzbereich 1-20 kHz von ADSL unberührt
- Upstream
 - 32 Trägerkanäle für Verbindung zum ISP
- Downstream
 - 190 Trägerkanäle für Verbindung vom ISP



Das elektromagnetische Spektrum

leitungsgebundene Übertragungstechniken

verdrillte Drähte Koaxialkabel

Hohlleiter

optische Glasfaser

10^3 10^5 10^7 10^9 10^{11} 10^{13} 10^{15}

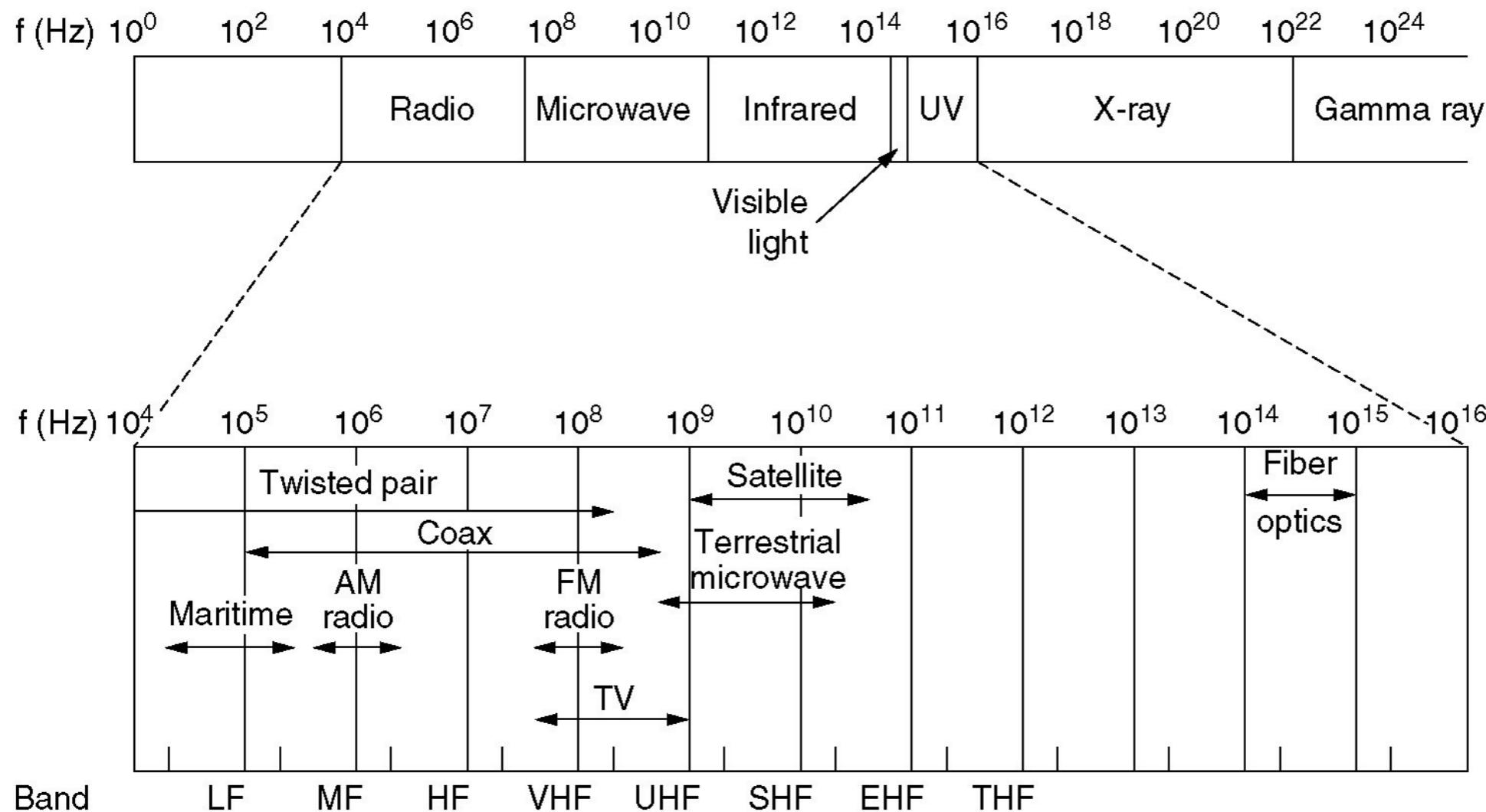
Hz

Langwellen-
Radio Mittelwellen
-Radio Kurzwelle Fernsehen Mikrowellen Infrarot sichtbares
Licht

nicht-leitungsgebundene Übertragungstechniken

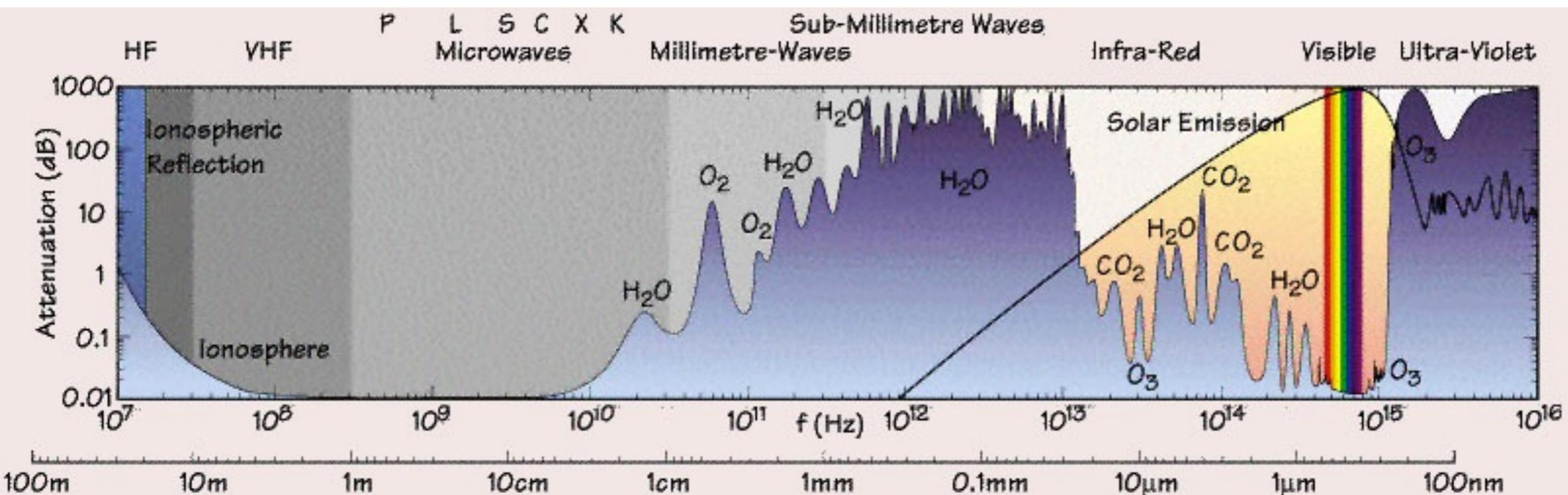
Frequenzbereiche

- LF Low Frequency =
 - LW Langwelle
- MF Medium Frequency =
 - MW Mittelwelle
- HF High Frequency =
 - KW Kurzwelle
- VHF Very High Frequency =
 - UKW Ultrakurzwelle
- UHF Ultra High Frequency
- SHF Super High Frequency
- EHF Extra High Frequency
- UV Ultraviolettes Licht
- X-ray Röntgenstrahlung



Dämpfung in verschiedenen Frequenzbereichen

- Frequenzabhängige Dämpfung elektromagnetischer Wellen in der Atmosphäre



http://www.geographie.uni-muenchen.de/igf/Multimedia/Klimatologie/physik_arbeit.htm

Frequenzbänder für Funknetzwerke

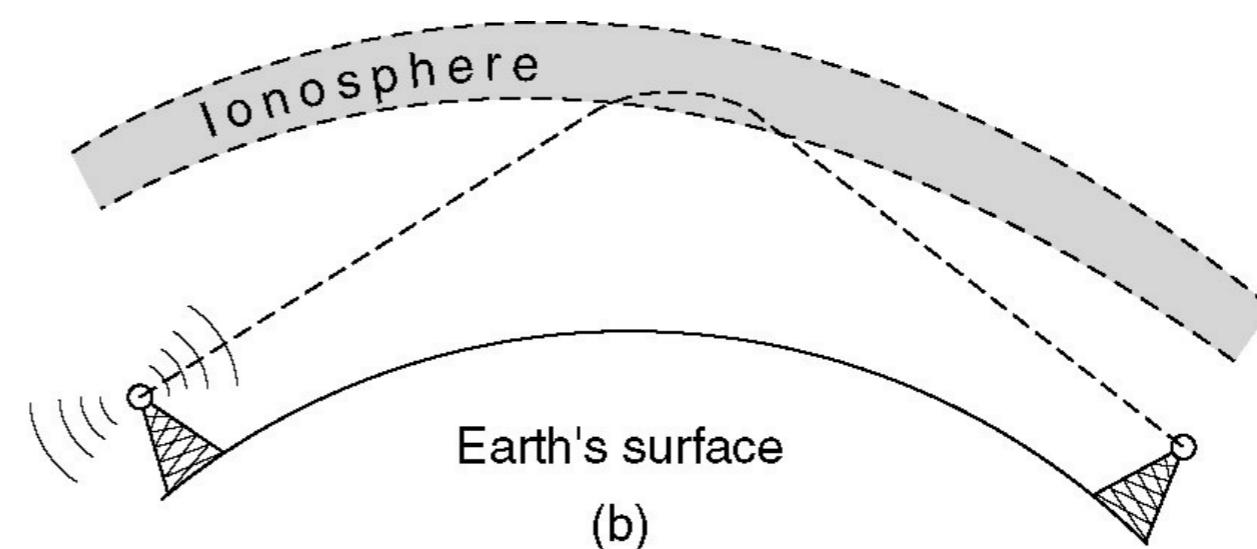
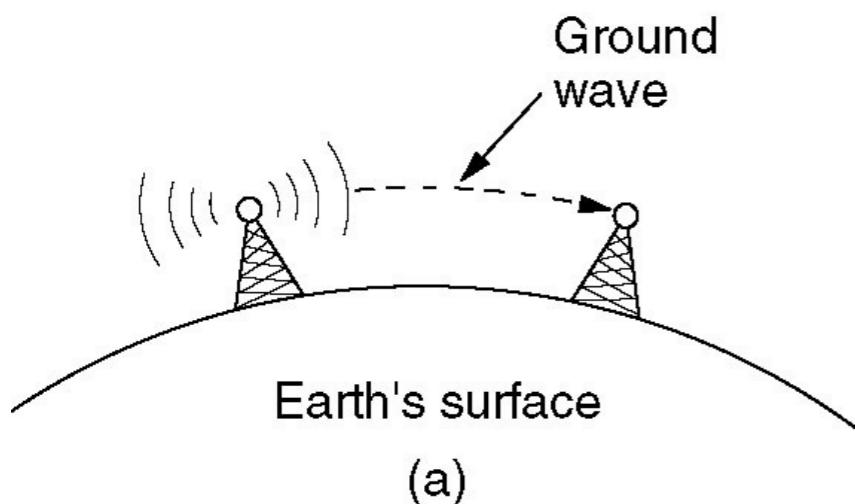
- VHF/UHF für Mobilfunk
 - Antennenlänge
- SHF für Richtfunkstrecken, Satellitenkommunikation
- Drahtloses (Wireless) LAN: UHF bis SHF
 - Geplant: EHF
- Sichtbares Licht
 - Kommunikation durch Laser
- Infrarot
 - Fernsteuerungen
 - Lokales LAN in geschlossenen Räumen

Ausbreitungsverhalten (I)

- Geradlinige Ausbreitung im Vakuum
- Empfangsleistung nimmt mit $1/d^2$ ab
 - Theoretisch, praktisch mit höheren Exponenten bis zu 4 oder 5
- Einschränkung durch
 - Dämpfung in der Luft (insbesondere HV, VHF)
 - Abschattung
 - Reflektion
 - Streuung an kleinen Hindernissen
 - Beugung an scharfen Kanten

Ausbreitungsverhalten (II)

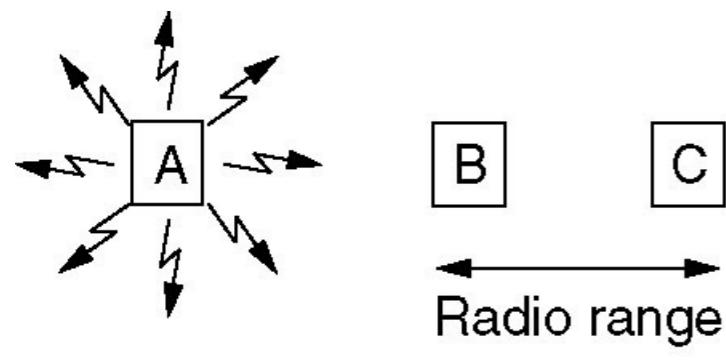
- VLF, LF, MF-Wellen
 - folgen der Erdkrümmung (bis zu 1000 km in VLF)
 - Durchdringen Gebäude
- HF, VHF-Wellen
 - Werden am Boden absorbiert
 - Werden von der Ionosphäre in 100-500 km Höhe reflektiert
- Ab 100 MHz
 - Wellenausbreitung geradlinig
 - Kaum Gebäudedurchdringung
 - Gute Fokussierung
- Ab 8 GHz Absorption durch Regen



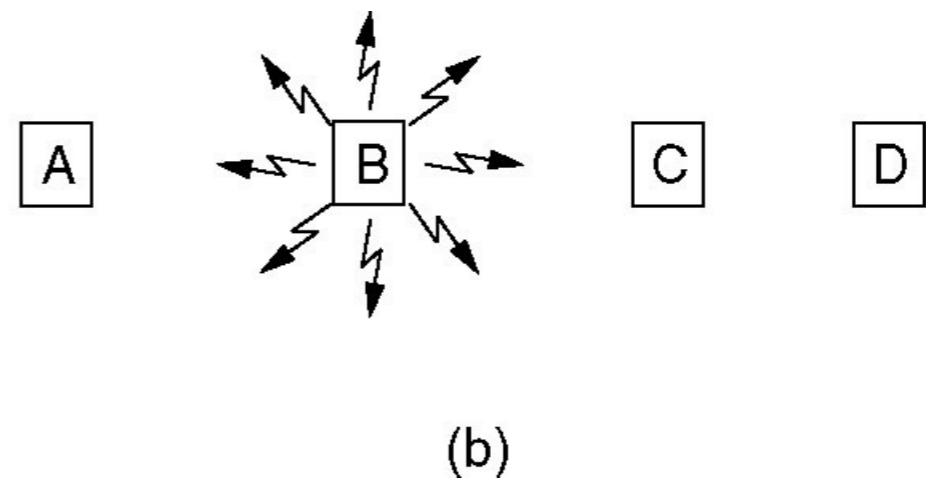
Ausbreitungsverhalten (III)

- Mehrwegeausbreitung (Multiple Path Fading)
 - Signal kommt aufgrund von Reflektion, Streuung und Beugung auf mehreren Wegen beim Empfänger an
 - Zeitliche Streuung führt zu Interferenzen
 - Fehlerhafter Dekodierung
 - Abschwächung
- Probleme durch Mobilität
 - Kurzzeitige Einbrüche (schnelles Fading)
 - Andere Übertragungswege
 - Unterschiedliche Phasenlage
 - Langsame Veränderung der Empfangsleistung (langsam Fading)
 - Durch Verkürzen, Verlängern der Entfernung Sender-Empfänger

Spezielle Probleme in drahtlosen Netzwerken



(a)

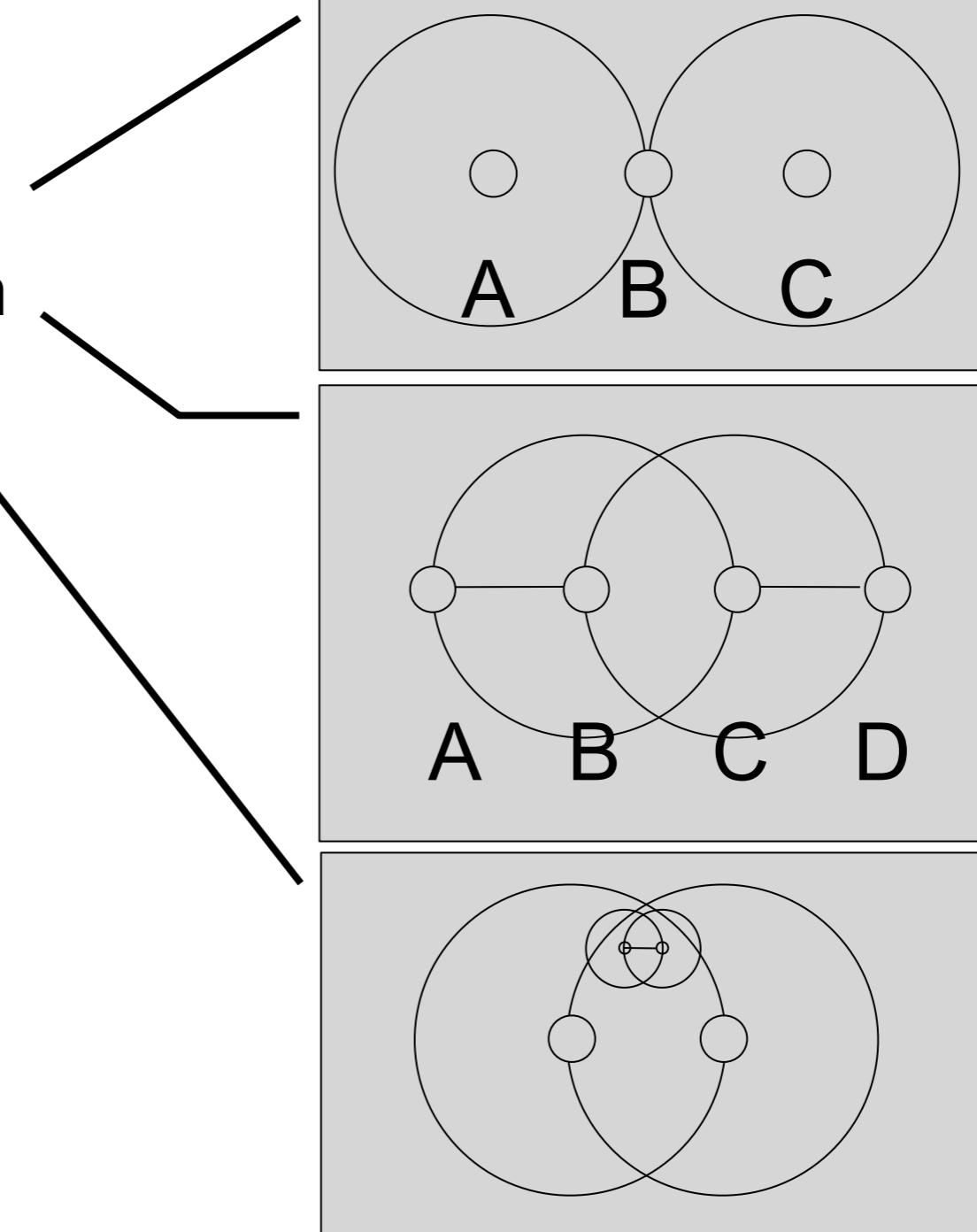


(b)

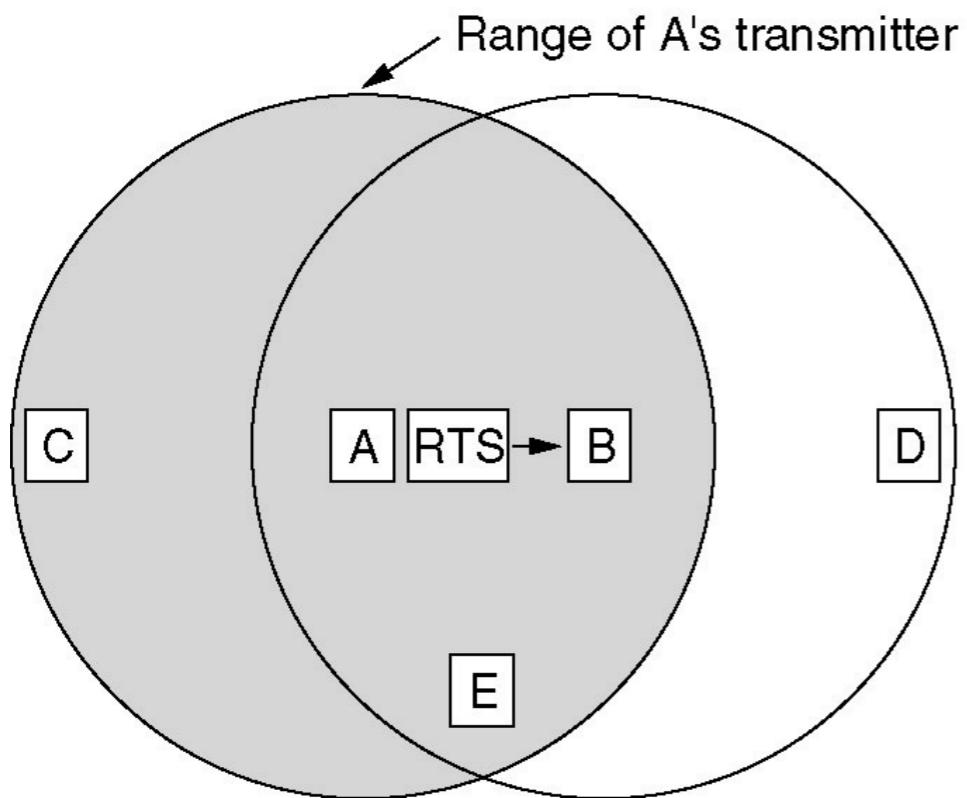
Probleme im W-LAN

- Interferenzen

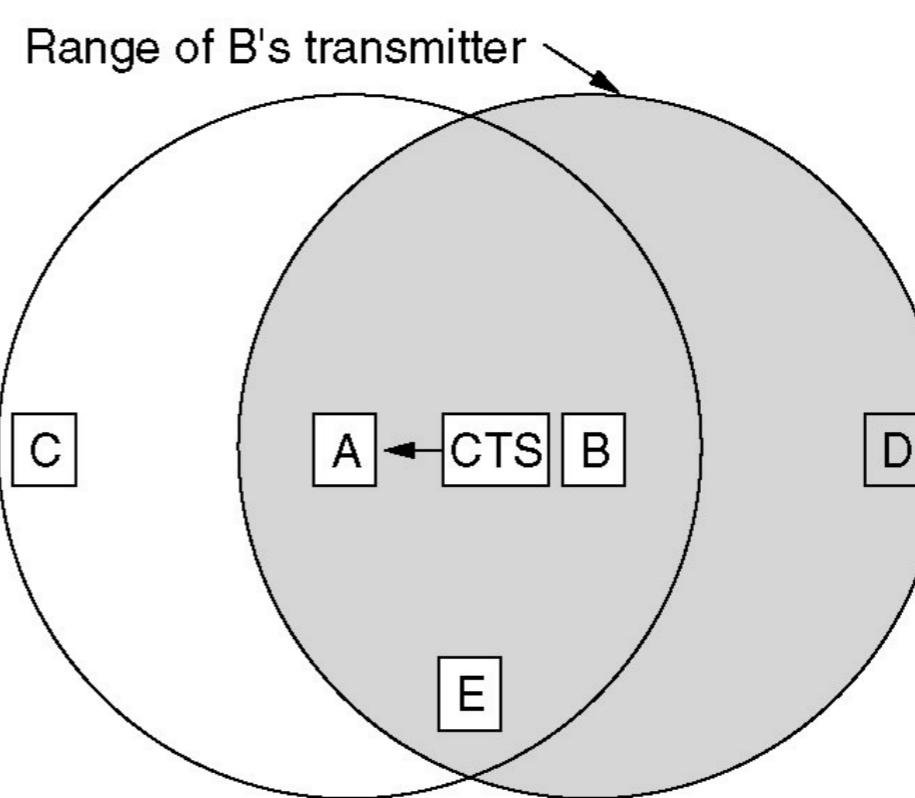
- Hidden Terminal Problem
- Exposed Terminal Problem
- Asymmetrie (var. Reichweite)



Multiple Access with Collision Avoidance



(a)

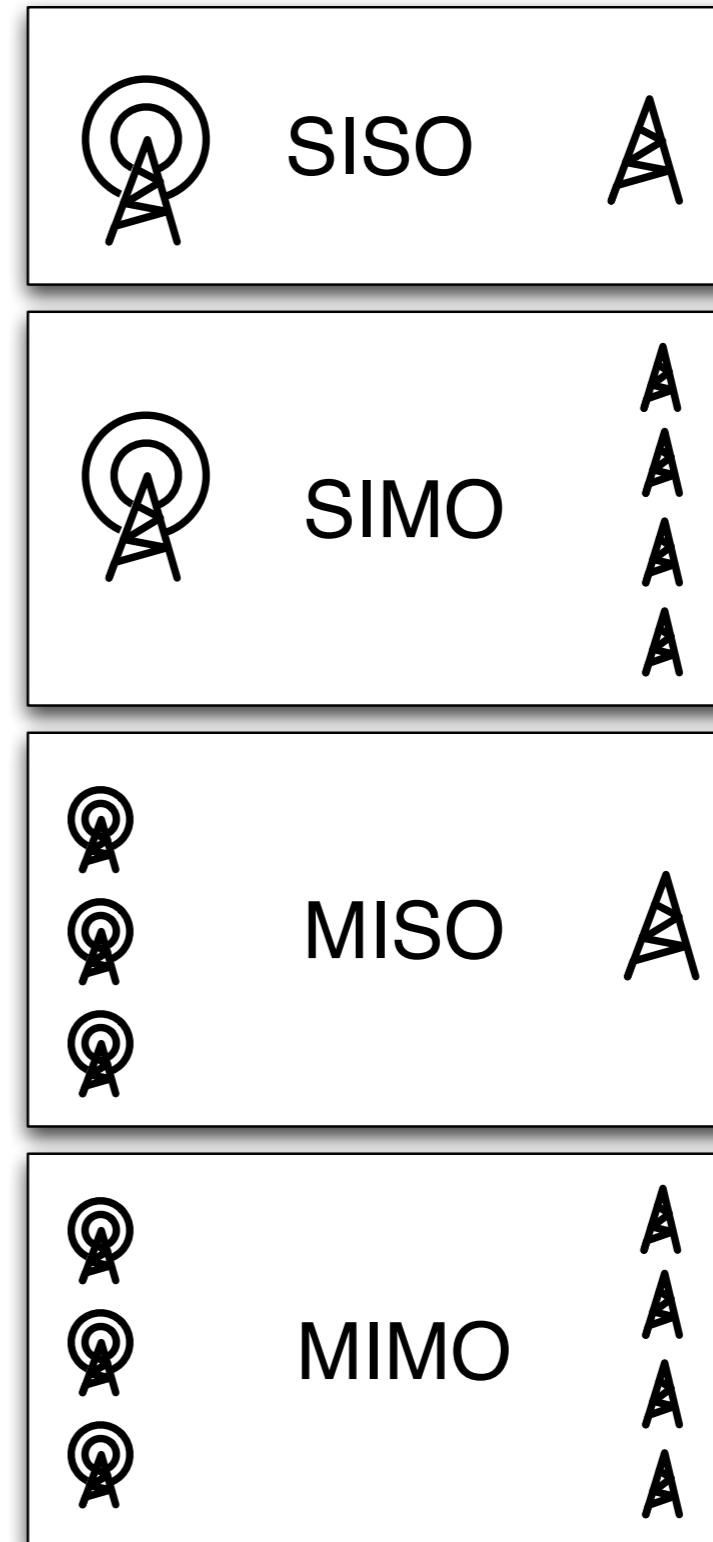


(b)

- (a) A sendet Request to Send (RTS) an B.
- (b) B antwortet mit Clear to Send (CTS) an A.

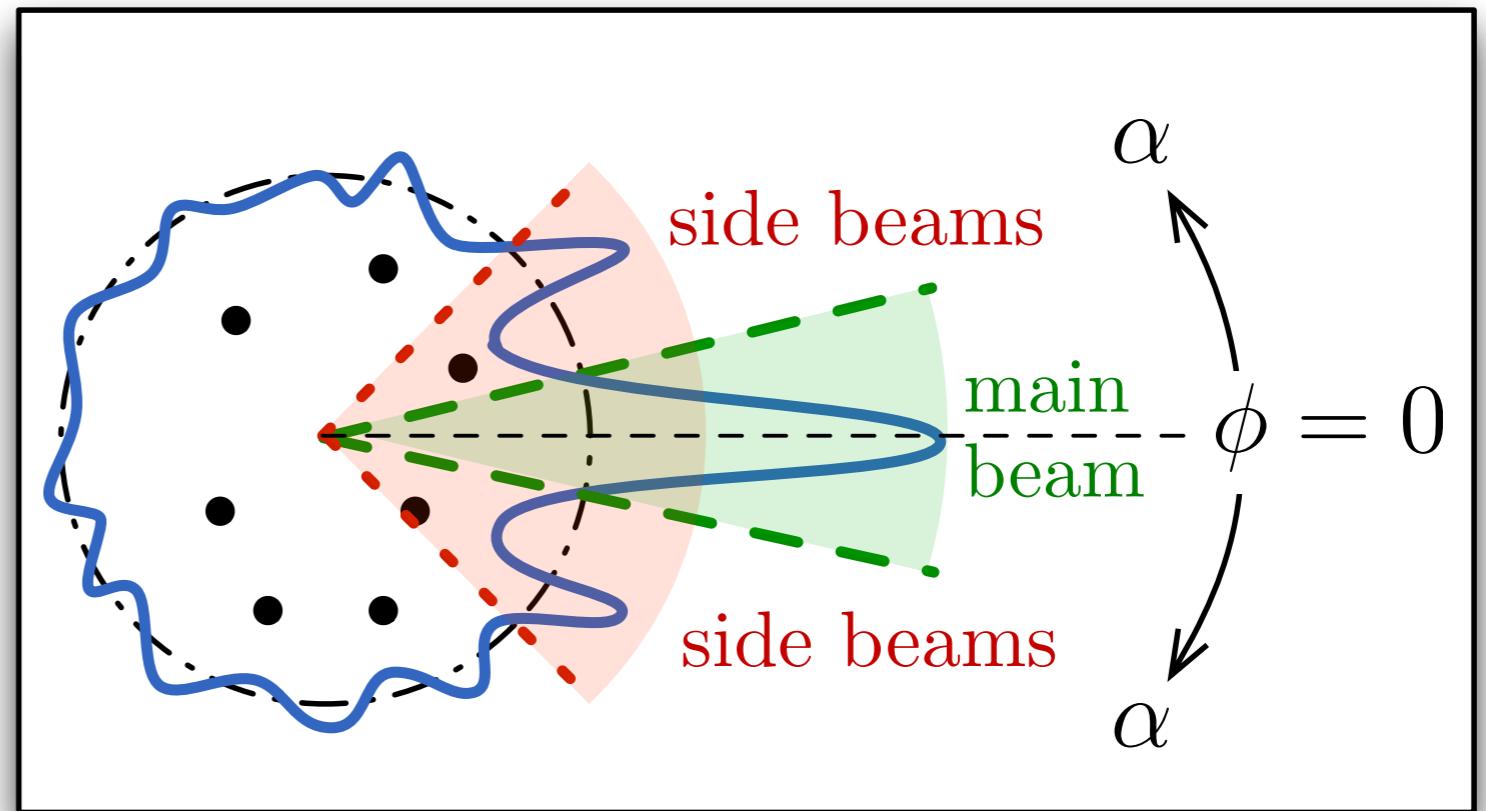
Smart Antennas, MIMO, SIMO, MISO

- Smart antennas
 - MIMO (multiple input/multiple output)
 - SIMO (single input/multiple output)
 - MISO, SISO
 - sind mehrere Antennen, welche koordiniert Signale übertragen und empfangen
- Vorteile
 - Beam forming
 - Power gain
 - Diversity gain
- Anwendungen
 - IEEE-802.11n-WLAN



Beamforming

- Durch geschickte Phasenverschiebung kann ein gerichteter Sendestrahl gesendet werden
 - oder symmetrisch auch empfangen werden

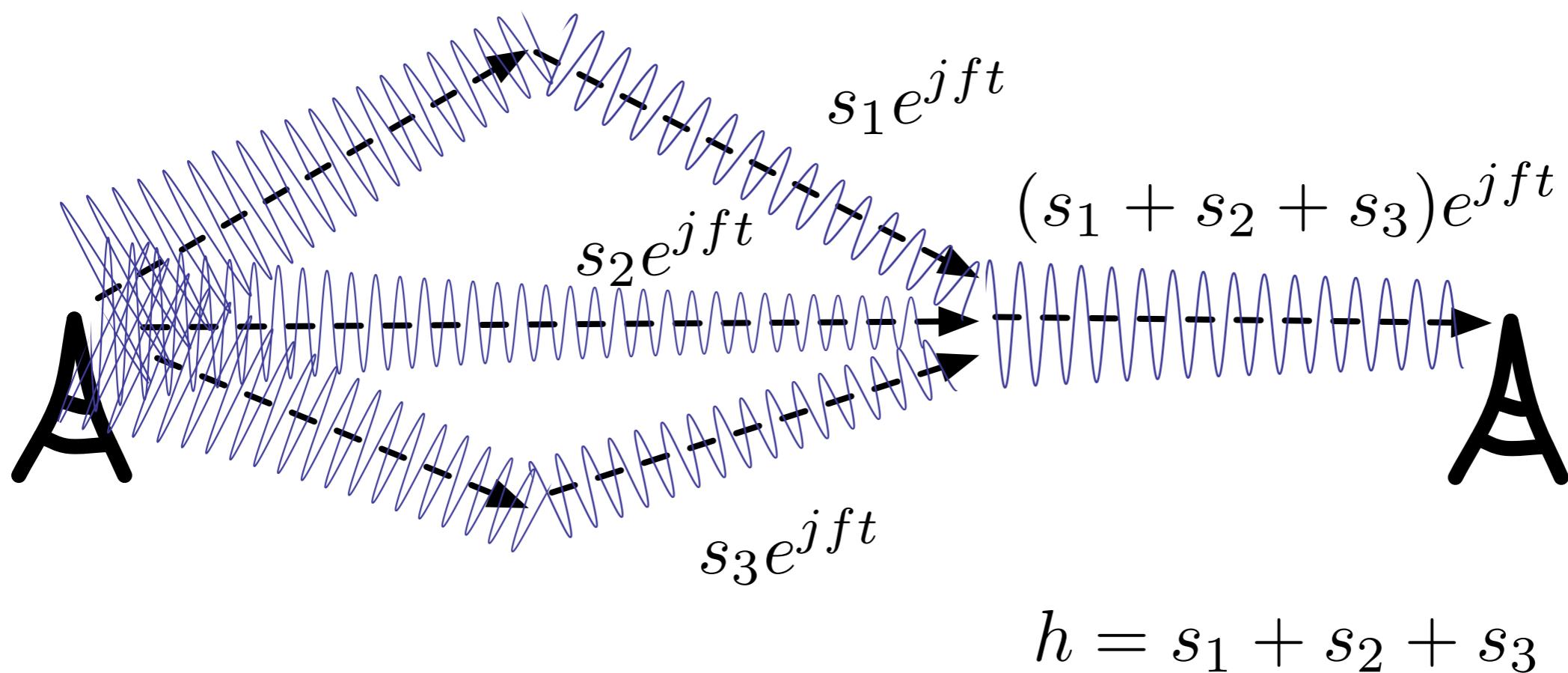


Power Gain

- Wieso können n Sender oder n Empfänger weiterreichen als 1 Sender und Empfänger?
 - mit gleichen Antennen
 - mit gleicher Energie
- Superposition:
 - Die elektrischen Felder überlagern sich (nicht die Energie)
 - Energy = $P \sim E^2 = (\text{el. Feld})^2$
 - El. Feldstärke = $D \sim 1/d$
- 1 Sender
 - Energie: P
 - Energie im Abstand d : P/d^2
- n Sender
 - Energie von n Sendern: P
 - Feldstärke eines von n Sendern: $\sqrt{\frac{P}{n}}$
 - Feldstärke im Abstand d von n Sendern: $\frac{n}{d} \sqrt{\frac{P}{n}} = \frac{\sqrt{Pn}}{d}$
 - Gesamtenergie im Abstand d : $n \cdot \frac{P}{d^2}$
- Der selbe Effekt funktioniert auch beim Empfänger
 - führt zu einem Power Gain von Faktor n für n Sender und n Empfänger

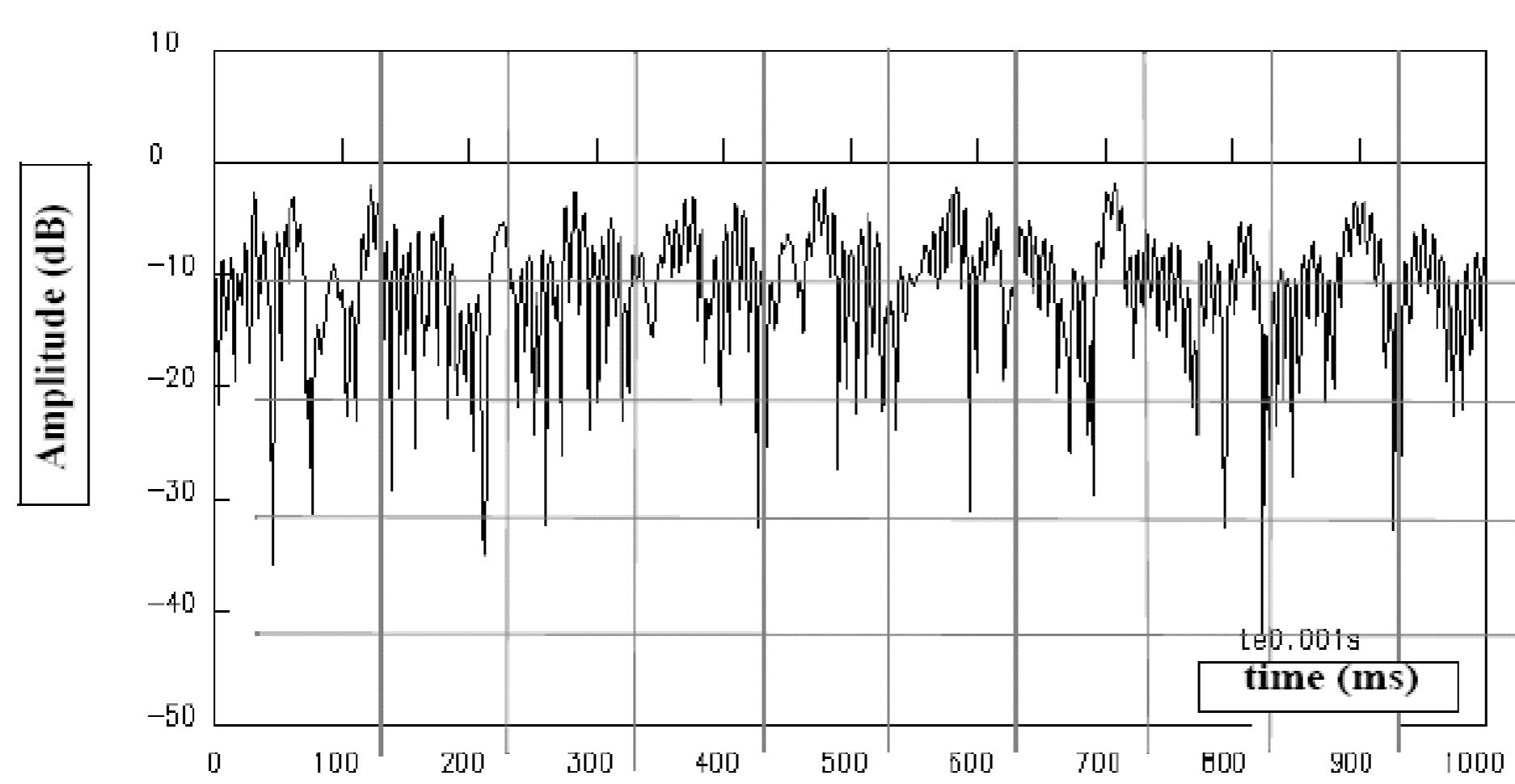
Multipath Channel

- Superposition von Reflektionen



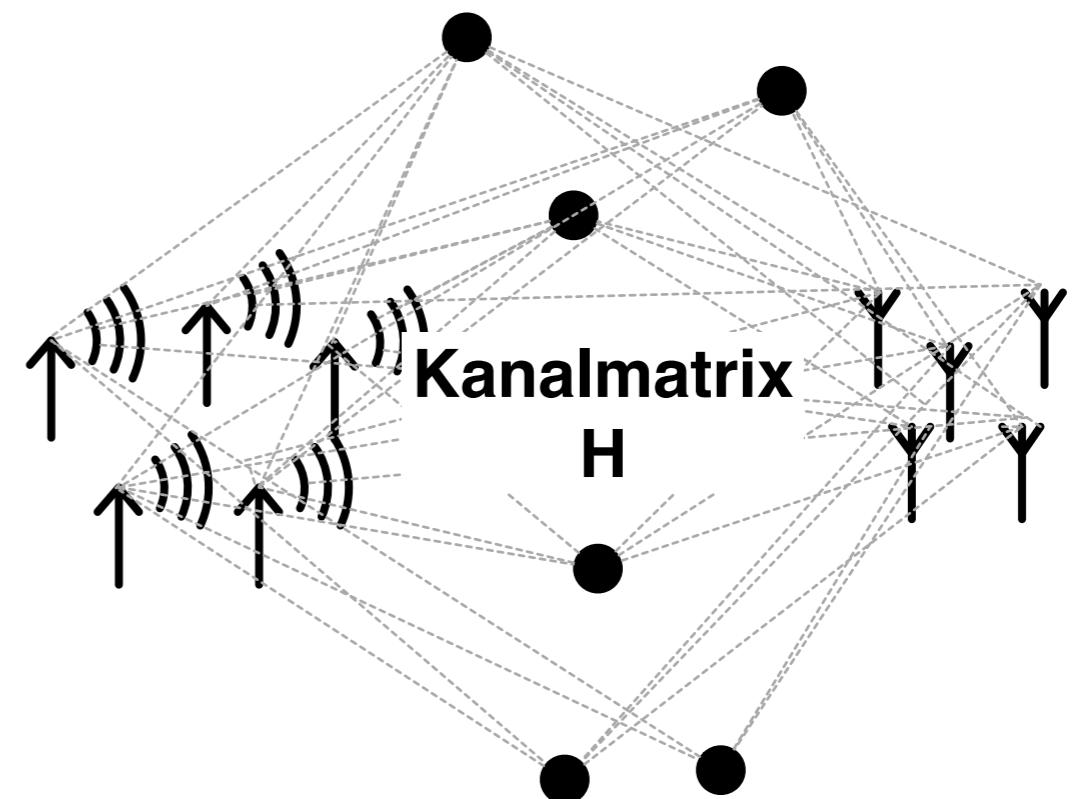
Rayleigh fading

- Superposition führt zu drastischen Einbrüchen



Diversity Gain

- Wenn in der Umgebung viele Reflektoren (scatterers) vorhanden sind,
 - dann ergibt sich für die Beschreibung der Sender-/Empfänger-Beziehung eine Kanalmatrix H
- $H_{i,j} =$
 - resultierende Dämpfung und Phasenverschiebung zwischen Sender i und Empfänger j
- Für geeignete Kanalmatrizen
 - mit „guter“ Singulärwertzerlegung
 - können bis zu $\max\{\#\text{Sender}, \#\text{Empfänger}\}$ parallele Kommunikationskanäle verwendet werden
- Dadurch können mehr Daten übertragen werden, als Shannons Theorem für SISO zulässt



Systeme II

2. Die physikalische Schicht

Christian Schindelhauer

Technische Fakultät

Rechnernetze und Telematik

Albert-Ludwigs-Universität Freiburg

Version 26.04.2017

Systeme II

3. Die Datensicherungsschicht

Christian Schindelhauer

Technische Fakultät

Rechnernetze und Telematik

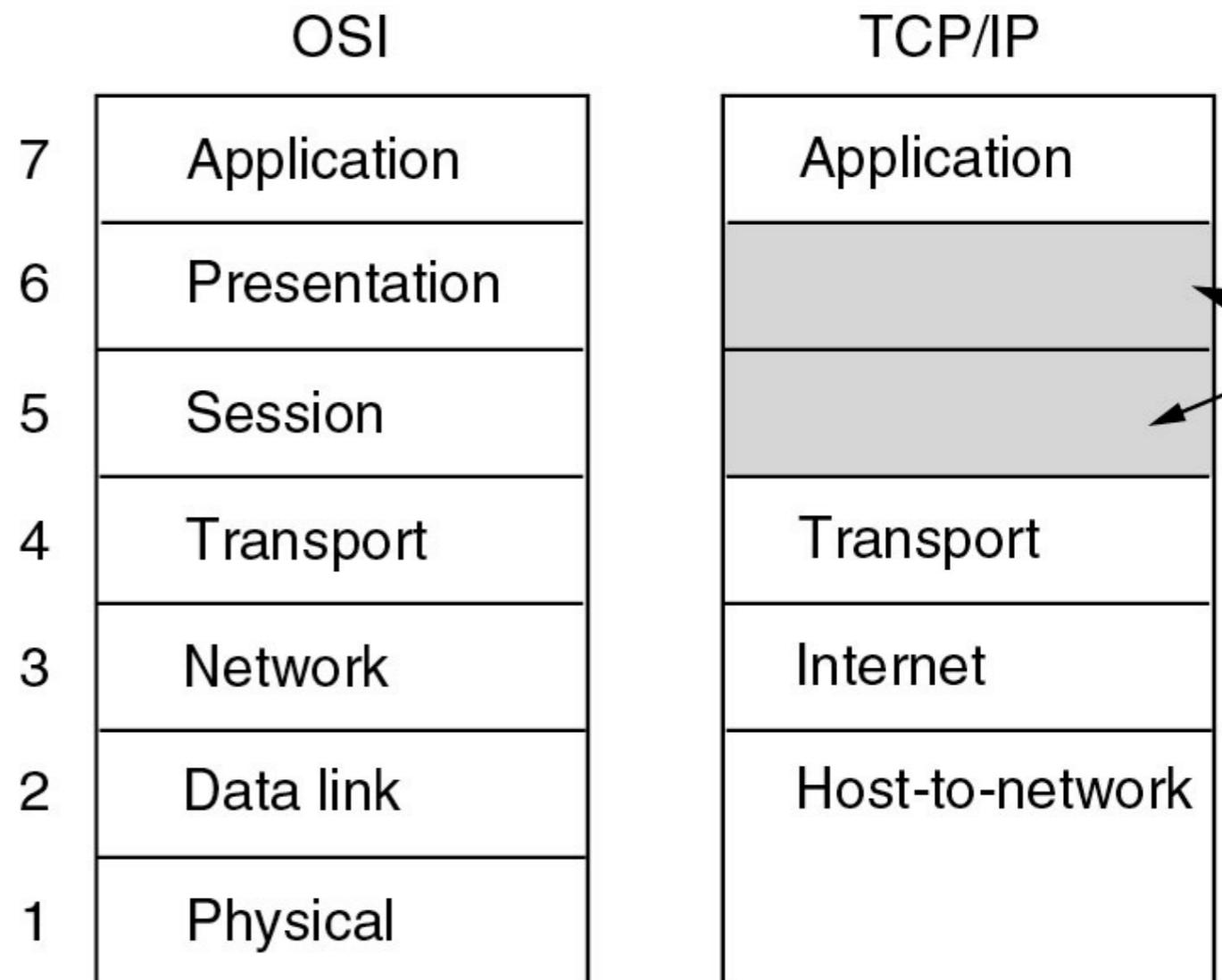
Albert-Ludwigs-Universität Freiburg

Version 31.05.2017

Die Sicherungsschicht

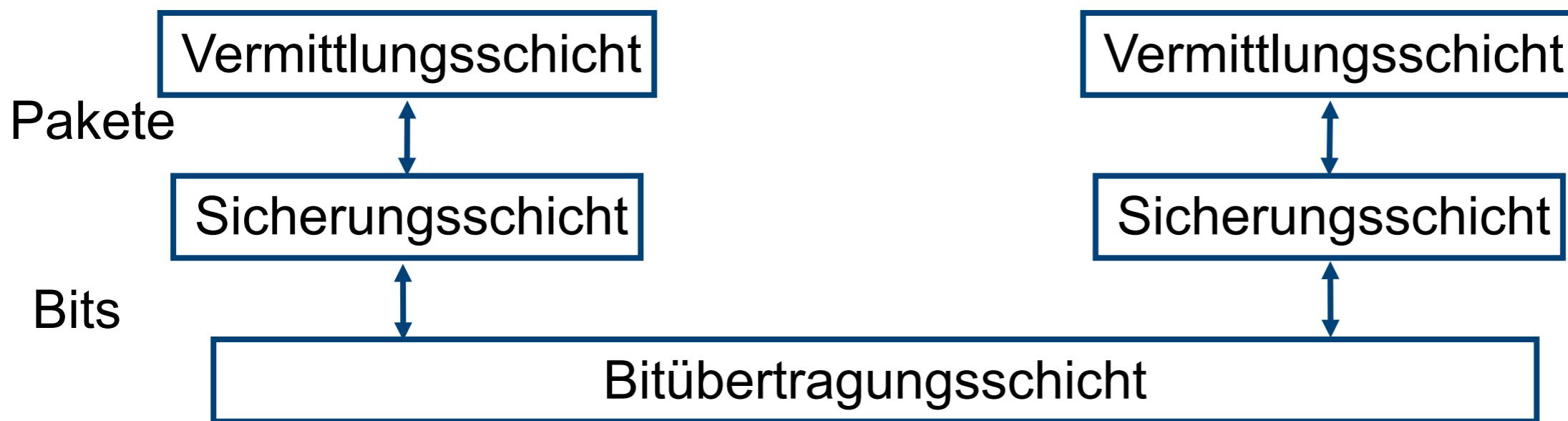
- Aufgaben der Sicherungsschicht (Data Link Layer)

- Dienste für die Vermittlungsschicht
- Frames
- Fehlerkontrolle
- Flusskontrolle



Dienste der Sicherungsschicht

- Situation der Sicherungsschicht
 - Die Bitübertragungsschicht überträgt Bits
 - Aber unstrukturiert und möglicherweise fehlerbehaftet
- Die Vermittlungsschicht erwartet von der Sicherungsschicht
 - Fehlerfreie Übermittlung
 - Übermittlung von strukturierten Daten
 - Datenpakete oder Datenströme
 - Störungsfreien Datenfluss



Mögliche Dienste der Sicherungsschicht

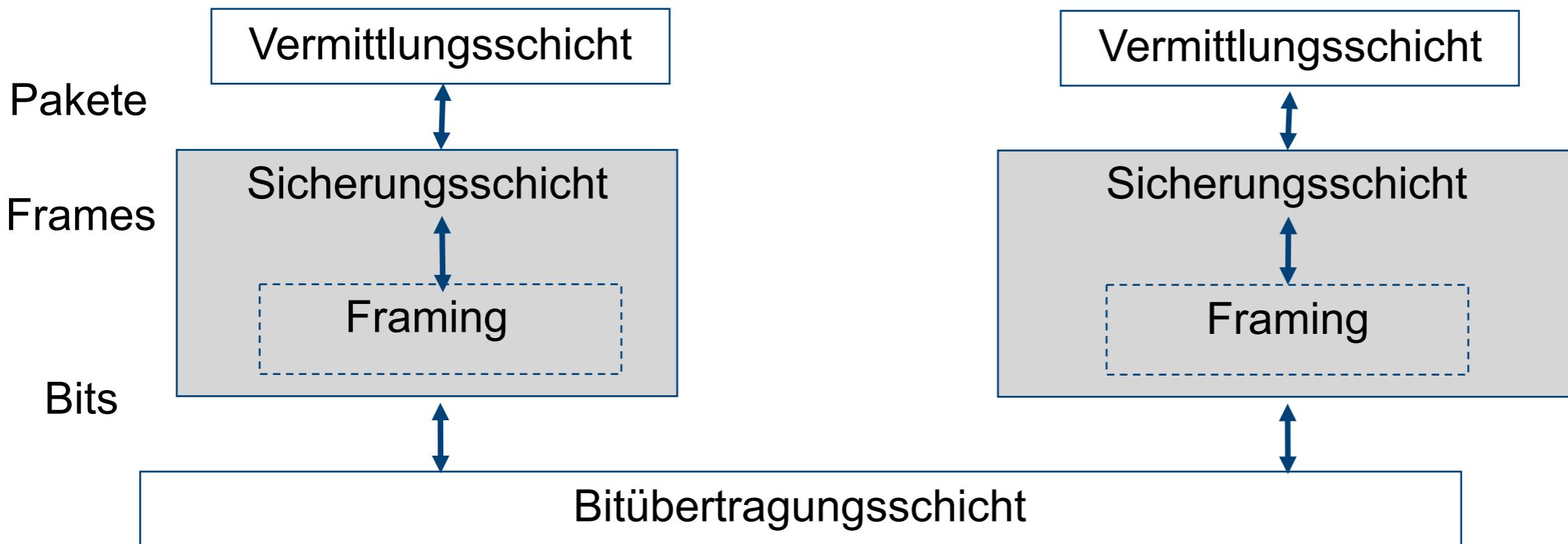
- Verlässlicher Dienst?
 - Das ausgelieferte und das empfangene Paket müssen identisch sein
 - Alle Pakete sollen (irgendwann) ankommen
 - Pakete sollen in der richtigen Reihenfolge ankommen
 - Fehlerkontrolle ist möglicherweise notwendig
- Verbindungsorientiert?
 - Ist die Punkt-zu-Punktverbindung in einem größerem Kontext?
 - Reservierung der Verbindung notwendig?
- Pakete oder Datenströme (Bitströme)?

Unterscheidung: Dienst und Implementation

- Beispiel
 - Verbindungsloser und verlässlicher Dienst wird durch die Vermittlungsschicht gefordert
 - Sicherungsschicht verwendet intern verbindungsorientierten Dienst mit Fehlerkontrolle
- Andere Kombinationen sind möglich

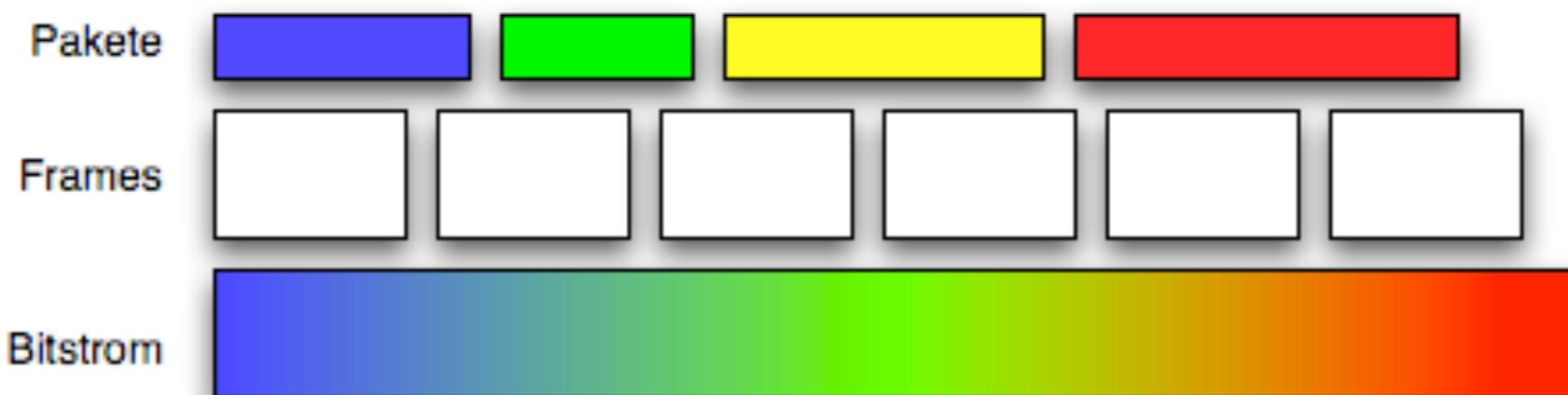
Frames

- Der Bitstrom der Bitübertragungsschicht wird in kleinere “Frames” unterteilt
 - Notwendig zur Fehlerkontrolle
 - Frames sind Pakete der Sicherungsschicht
- Frame-Unterteilung (Fragmentierung) und Defragmentierung sind notwendig
 - Falls die Pakete der Vermittlungsschicht größer sind als die Frames

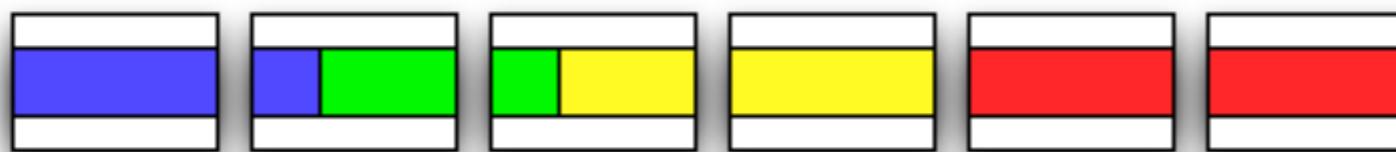


Frames

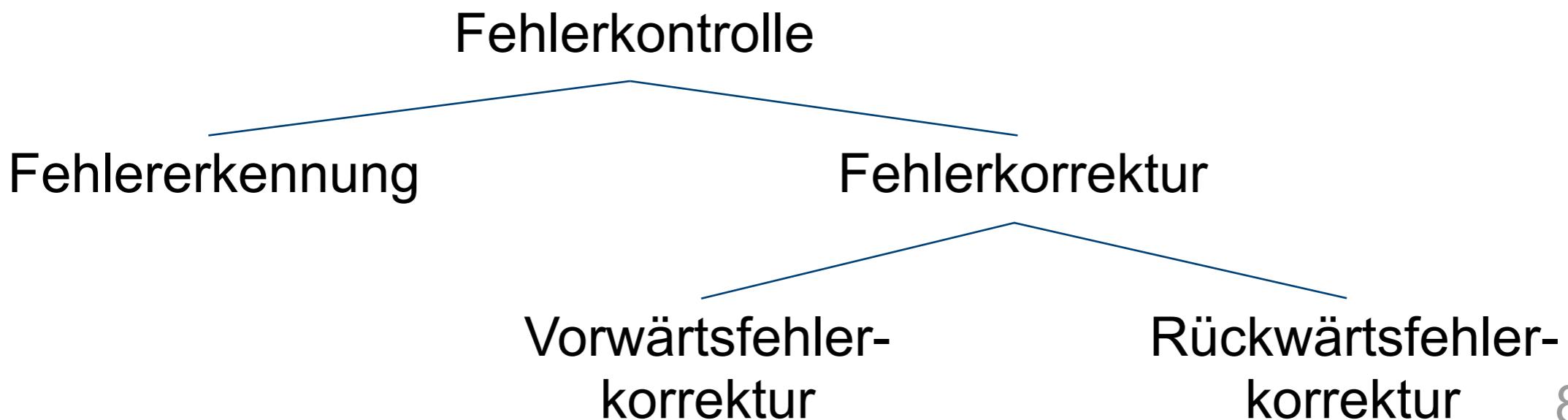
- Die Sicherungsschicht zwischen der Bitübertragungsschicht mit Bitstrom und der Vermittlungsschicht mit Paketen



- Pakete werden in Framegröße fragmentiert



- Zumeist gefordert von der Vermittlungsschicht
 - Mit Hilfe der Frames
- Fehlererkennung
 - Gibt es fehlerhaft übertragene Bits?
- Fehlerkorrektur
 - Behebung von Bitfehlern
 - Vorwärtsfehlerkorrektur (Forward Error Correction)
 - Verwendung von redundanter Kodierung, die es ermöglicht Fehler ohne zusätzliche Übertragungen zu beheben
 - Rückwärtsfehlerkorrektur (Backward Error Correction)
 - Nach Erkennen eines Fehlers, wird durch weitere Kommunikation der Fehler behoben



Verbindungsauftbau

- Nutzen von Verbindungen
 - Kontrolle des Verbindungsstatus
 - Korrektheit des Protokolls
 - Fehlerkontrolle
 - Verschiedene Fehlerkontrollverfahren vertrauen auf gemeinsamen Kontext von Sender und Empfänger
- Aufbau und Terminierung von Verbindungen
 - “Virtuelle Verbindungen”
 - Es werden keine Schalter umgelegt
 - Interpretation des Bitstroms
 - Kontrollinformationen in Frames
 - Besonders wichtig bei drahtlosen Medien
- Das Problem wird im Rahmen der Transportschicht ausführlich diskutiert
 - Vgl. Sitzungsschicht vom OSI-Modell

Flusskontrolle

- Problem: Schneller Sender und langsamer Empfänger
 - Der Sender lässt den Empfangspuffer des Empfängers überlaufen
 - Übertragungsbandweite wird durch sinnlosen Mehrfachversand (nach Fehlerkontrolle) verschwendet
- Anpassung der Frame-Sende-Rate an dem Empfänger notwendig



Frames

- Wo fängt der Frame an und wo hört er auf?
 - Achtung:
 - Die Bitübertragungsschicht kann auch Bits liefern, wenn der Sender tatsächlich nichts sendet
 - Der Empfänger
 - könnte das Rauschen auf dem Medium interpretieren
 - könnte die Folge 00000000.... liefern
 - Daten oder Kontrollinformation?

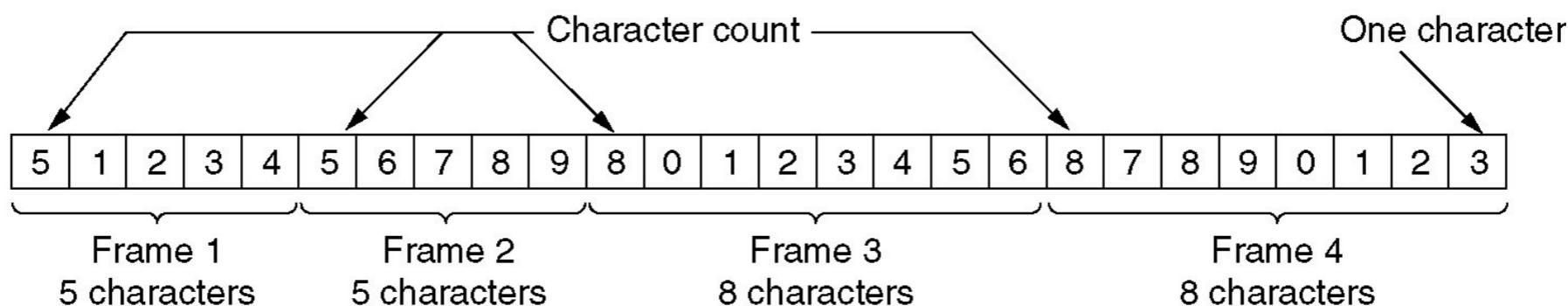
Übertragerer Bitstrom 0110010101110101110010100010101010101010101100010

Frame-Anfang?

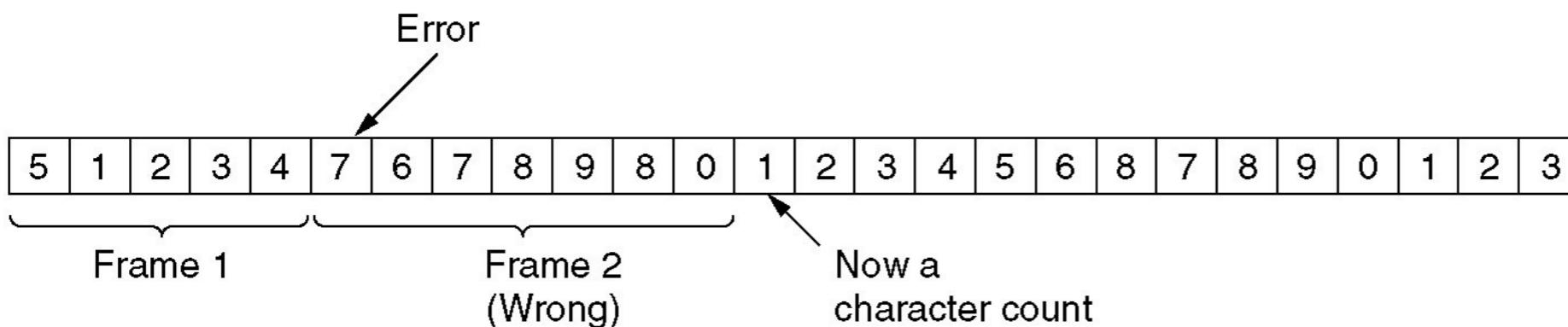
Frame-Ende?

Frame-Grenzen durch Paketlängen?

- Idee: Ankündigung der Bitanzahl im Frame-Header



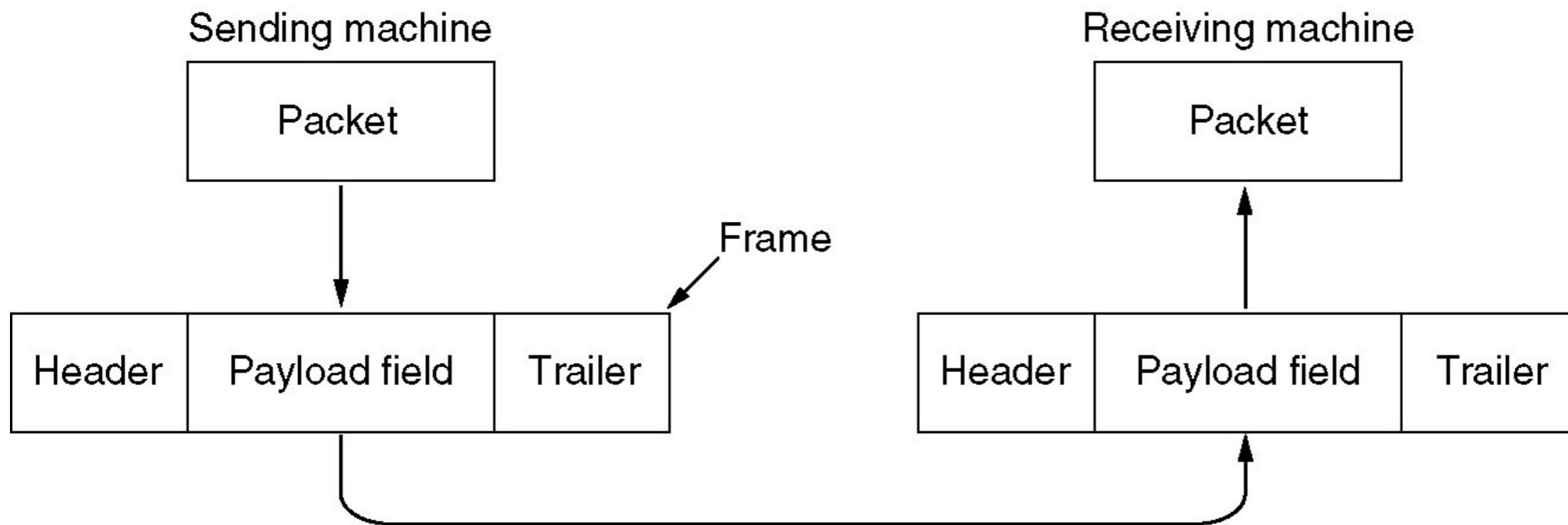
- Problem: Was, wenn die Frame-Länge fehlerhaft übertragen wird?
 - Der Empfänger kommt aus dem Takt und interpretiert neue, sinnlose Frames
 - Variable Frame-Größen mit Längeninformation sind daher kein gutes Konzept



Header und Trailer

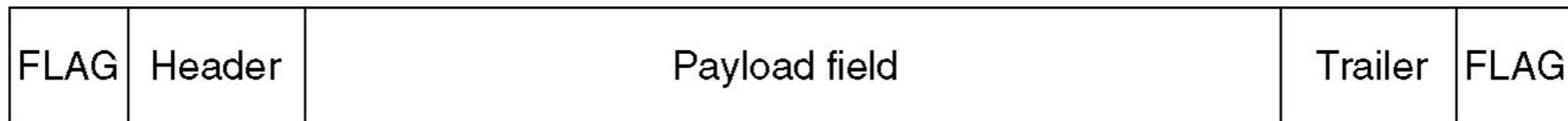
■ Header und Trailer

- Zumeist verwendet man Header am Anfang des Frames, mitunter auch Trailer am Ende des Frames
- signalisieren den Frame-Beginn und das Frame-Ende
- tragen Kontrollinformationen
 - z.B. Sender, Empfänger, Frametypen, Fehlerkontrollinformation



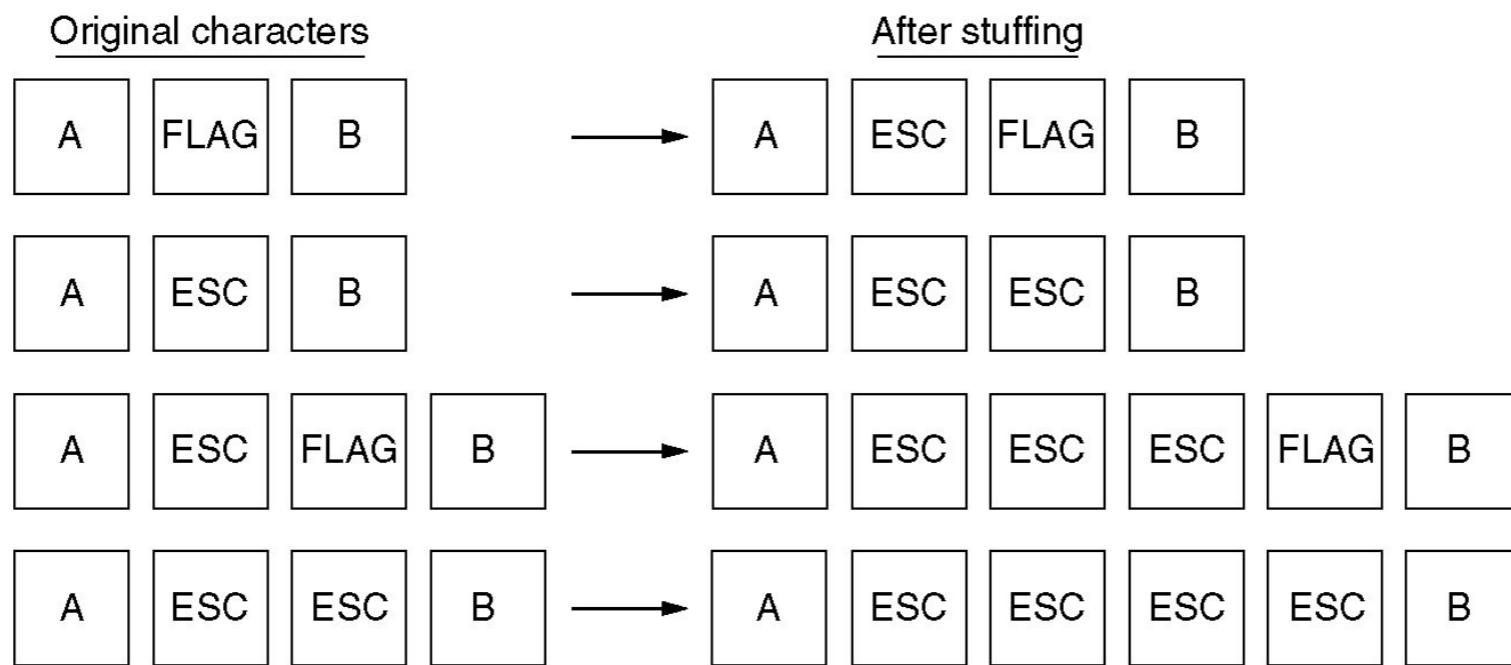
Flag Bytes und Bytestopfen

- Besondere “Flag Bytes” markieren Anfang und Ende eines Frames



- Falls diese Marker in den Nutzdaten vorkommen

- Als Nutzdatenbyte mit Sonderzeichen (Escape) markieren
 - Bytestopfen (byte stuffing)
- Falls Sonderzeichen und “Flag-Byte” erscheinen, dito,
 - etc., etc.



Frames durch Bit-Sequenzen/Bitstopfen

- Bytestopfen verwendet das Byte als elementare Einheit
 - Das Verfahren funktioniert aber auch auf Bitebene
 - Flag Bits und Bitstopfen (bit stuffing)
 - Statt flag byte wird eine Bit-Folge verwendet
 - z.B.: 0111110
 - Bitstopfen
 - Wenn der Sender eine Folge von fünf 1er senden möchte, wird automatisch eine 0 in den Bitstrom eingefügt
 - Außer bei den Flag Bits
 - Der Empfänger entfernt eine 0 nach fünf 1ern

Originale Nutzdate

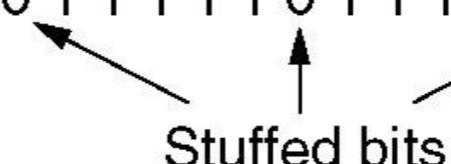
(a) 011011111111111110010

Nach dem Bitstopfen

(b) 01101111011111011111010010

Nach der “Entstopfung”

(c) 011011111111111111110010



Frames durch Code-Verletzung

- Möglicher Spielraum bei Bitübertragungsschicht bei der Kodierung von Bits auf Signale
 - Nicht alle möglichen Kombination werden zur Kodierung verwendet
 - Zum Beispiel: Manchester-Kodierung hat nur tief/hoch und hoch/tief–Übergang
- Durch “Verletzung” der Kodierungsregeln kann man Start und Ende des Rahmens signalisieren
 - Beispiel: Manchester – Hinzunahme von hoch/hoch oder tief/tief
 - Selbsttaktung von Manchester gefährdet?
- Einfache und robuste Methode
 - z.B. verwendet in Ethernet
 - Kosten? Effiziente Verwendung der Bandbreite?

Fehlerkontrolle

■ Aufgaben

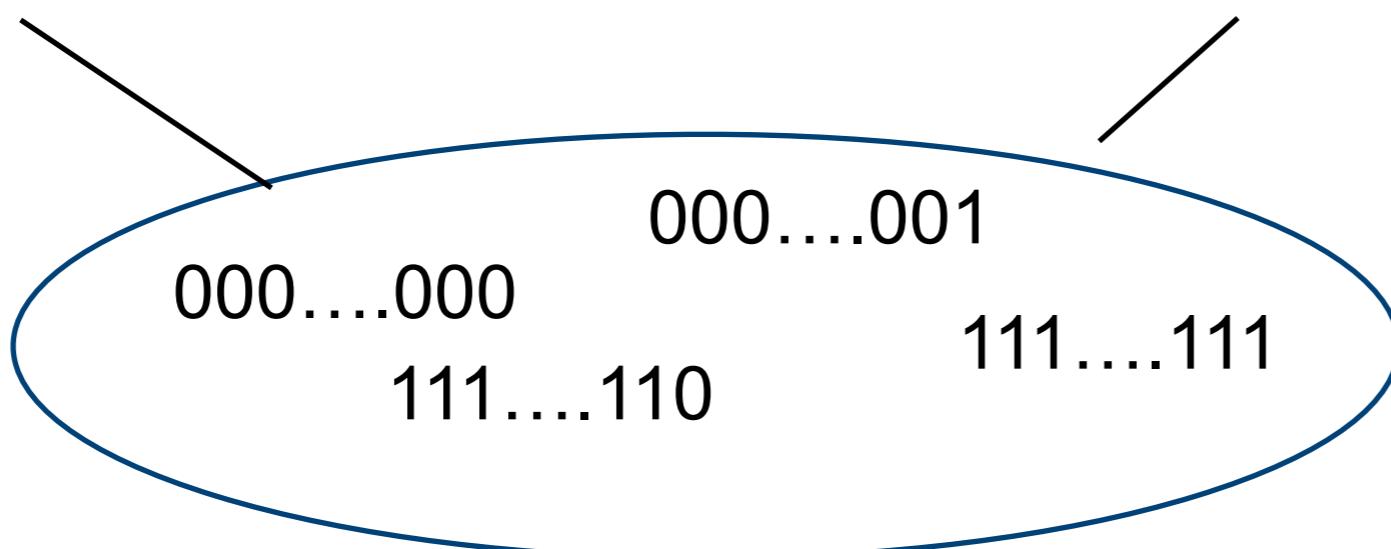
- Erkennung von Fehlern (fehlerhafte Bits) in einem Frame
 - Korrektur von Fehlern in einem Frame
- ## ■ Jede Kombination dieser Aufgaben kommt vor
- Erkennung ohne Korrektur
 - Löschen eines Frames ohne weiter Benachrichtigung (drop a frame)
 - Höhere Schichten müssen sich um das Problem kümmern
 - Korrektur ohne Erkennung
 - Es werden bestmöglich Bitfehler beseitigt, möglicherweise sind aber noch Fehler vorhanden
 - Sinnvoll, falls Anwendung Fehler tolerieren kann
 - Beispiel: Tonübertragung
 - Prinzipiell gerechtfertigt, weil immer eine positive Restfehlerwahrscheinlichkeit bleibt

Redundanz

- Redundanz ist eine Voraussetzung für Fehlerkontrolle
- Ohne Redundanz
 - Ein Frame der Länge m kann 2^m mögliche Daten repräsentieren
 - Jede davon ist erlaubt
- Ein fehlerhaftes Bit ergibt einen neuen Dateninhalt

Menge legaler Frames

Menge möglicher Frames

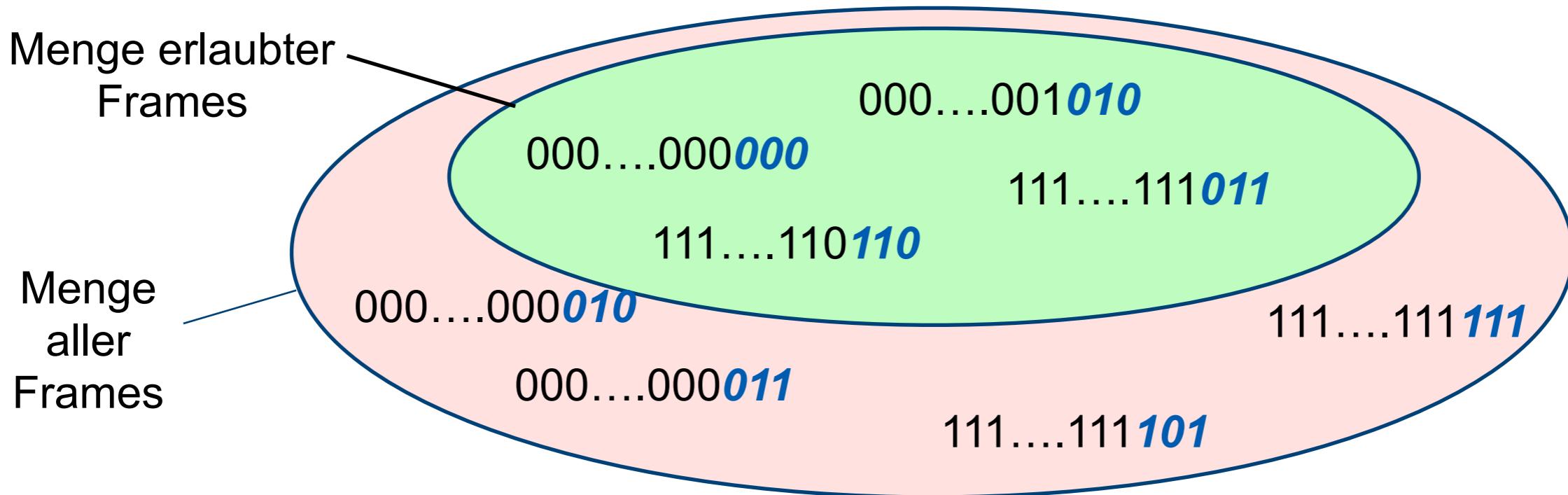


Redundanz

■ Kernidee:

- Einige der möglichen Nachrichten sind verboten
- Um dann 2^m legale Frames darzustellen
 - werden mehr als 2^m mögliche Frames benötigt
 - Also werden mehr als m Bits in einem Frame benötigt
- Der Frame hat also Länge $n > m$
- $r = m - n$ sind die redundanten Bits
 - z.B. Im Header oder Trailer

■ Nur die Einschränkung auf erlaubte und verbotene (legal/illegal) Frames ermöglicht die Fehlerkontrolle



Einfachste Redundanz: Das Paritätsbit

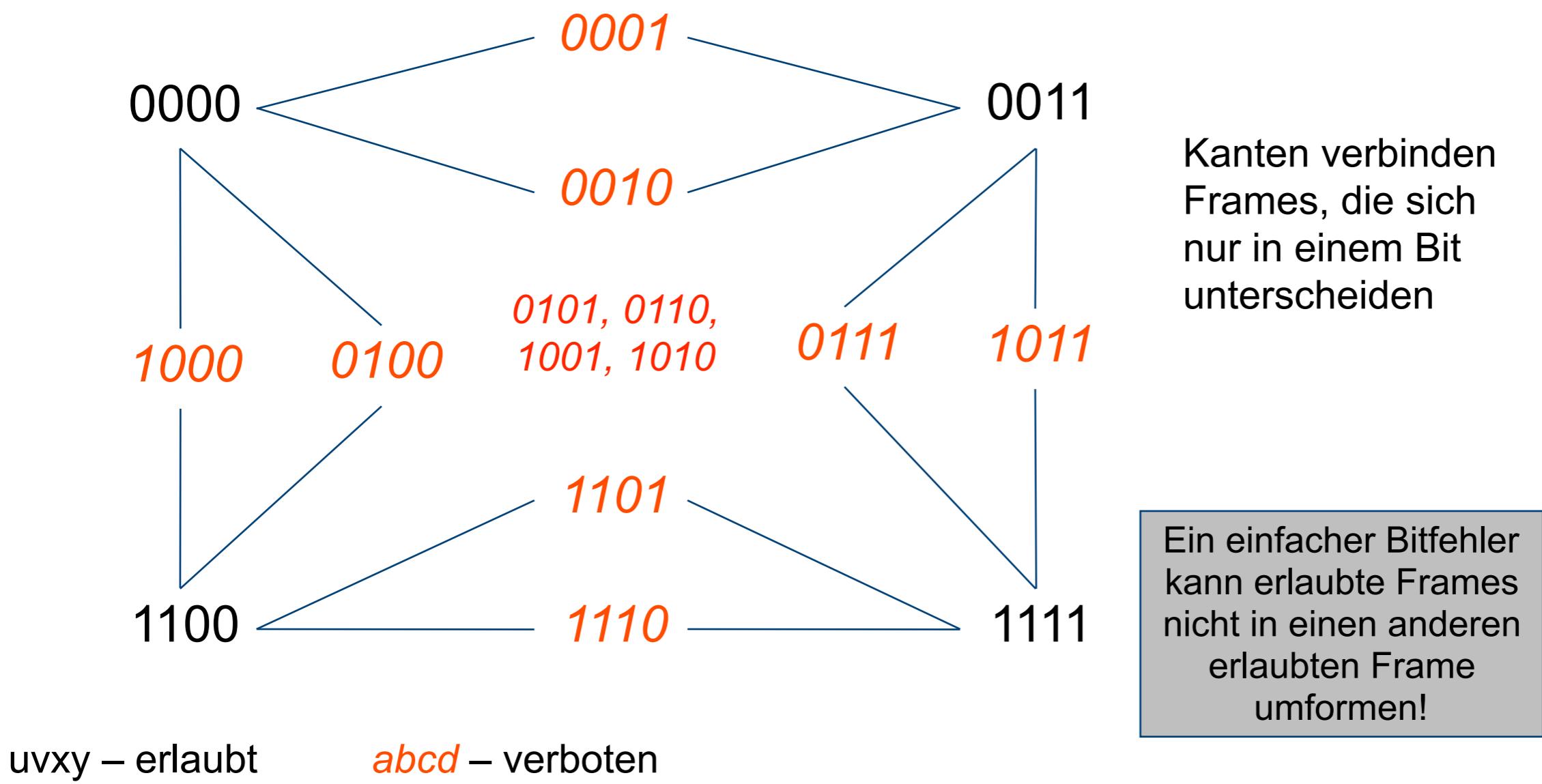
- Eine einfache Regel um ein redundantes Bit zu erzeugen (d.h. $n=m+1$)
- Parität
 - Odd parity
 - Eine Eins wird hinzugefügt, so dass die Anzahl der 1er in der Nachricht ungerade wird (ansonsten eine Null)
 - Even parity
 - Eine Eins wird hinzugefügt, so dass die Anzahl der 1er in der Nachricht gerade wird (ansonsten wird eine Null hinzugefügt)
- Beispiel:
 - Originalnachricht ohne Redundanz: 01101011001
 - Odd parity: 011010110011
 - Even parity: 011010110010

Der Nutzen illegaler Frames

- Der Sender sendet nur erlaubte Frames
- In der Bitübertragungsschicht könnten Bits verfälscht werden
- Hoffnung:
 - Legale Frames werden nur in illegale Nachrichten verfälscht
 - Und niemals ein legaler Frame in einen anderen Legalen
- Notwendige Annahme
 - In der Bitübertragungsschicht werden nur eine bestimmte Anzahl von Bits verändert
 - z.B. k Bits pro Frame
 - Die legalen Nachrichten sind verschieden genug, um diese Frame-Fehlerrate zu erkennen

Veränderung der Frames durch Bitfehler

- Angenommen die folgenden Frames sind erlaubt: 0000, 0011, 1100, 1111



Hamming-Distanz

- Der “Abstand” der erlaubten Nachrichten zueinander war immer zwei Bits
- Definition: Hamming-Distanz
 - Seien $x = x_1, \dots, x_n$ und $y = y_1, \dots, y_n$ Nachrichten
 - Dann sei $d(x,y) =$ die Anzahl der 1er Bits in $x \text{ XOR } y$
- Intuitiver: die Anzahl der Positionen, in denen sich x und y unterscheiden

Hamming-Distanz

- Die Hamming-Distanz ist eine Metrik
 - Symmetrie
 - $d(x,y) = d(y,x)$
 - Dreiecksungleichung:
 - $d(x,y) \leq d(x,z) + d(z,y)$
 - Identität
 - $d(x,x) = 0$ und
 $d(x,y) = 0$ gdw. $x = y$
- Beispiel:
 - $x = 0011010111$
 - $y = 0110100101$
 - $x \text{ XOR } y = 0101110010$
 - $d(x,y) = 5$

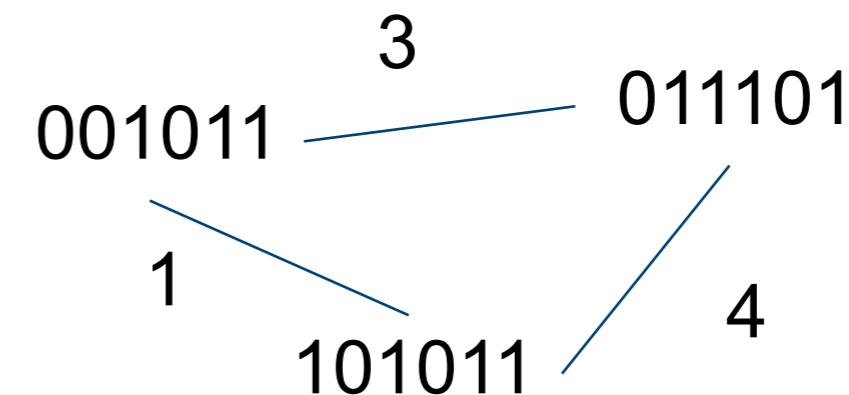
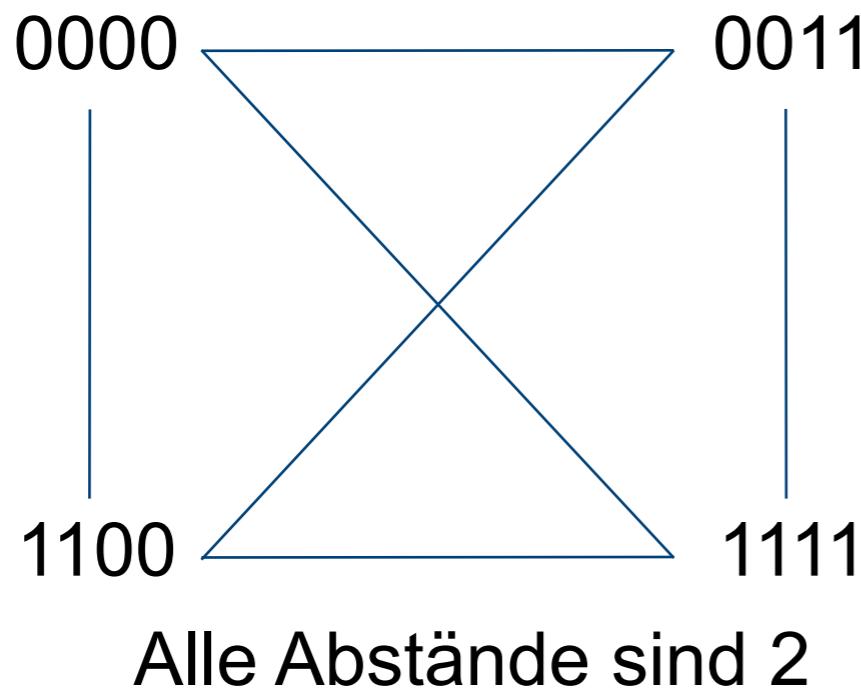
Hamming-Distanz von Nachrichtenmengen

- Die Hamming-Distanz einer Menge von (gleich langen) Bit-Strings S ist:

$$d(S) = \min_{x,y \in S, x \neq y} d(x, y)$$

- d.h. der kleinste Abstand zweier verschiedener Wörter in S

Beispiel:



Ein Abstand ist 1!

- 1. Fall $d(S) = 1$
 - Keine Fehlerkorrektur
 - Legale Frames unterscheiden sich in nur einem Bit
- 2. Fall $d(S) = 2$
 - Dann gibt es nur $x, y \in S$ mit $d(x,y) = 2$
 - Somit ist jedes u mit $d(x,u) = 1$ illegal,
 - wie auch jedes u mit $d(y,u) = 1$



- 1-Bit-Fehler
 - können immer erkannt werden
 - aber nicht korrigiert werden

■ 3. Fall $d(S) = 3$

- Dann gibt es nur $x, y \in S$ mit $d(x,y) = 3$
- Jedes u mit $d(x,u) = 1$ illegal und $d(y,u) > 1$



- Falls u empfangen wird, sind folgende Fälle denkbar:
 - x wurde gesendet und mit 1 Bit-Fehler empfangen
 - y wurde gesendet und mit 2 Bit-Fehlern empfangen
 - Etwas anderes wurde gesendet und mit mindestens 2 Bit-Fehlern empfangen
- Es ist also wahrscheinlicher, dass x gesendet wurde, statt y

Erkennung und Korrektur mit Hamming-Distanzen

- Um d Bit-Fehler zu erkennen ist eine Hamming-Distanz von $d+1$ in der Menge der legalen Frames notwendig
- Um d Bit-Fehler zu korrigieren, ist eine Hamming-Distanz von $2d+1$ in der Menge der legalen Frames notwendig

Codebücher und Kodierungen

- Die Menge der legalen Frames $S \in \{0,1\}^n$ wird **das Code-Buch** oder einfach Kodierung genannt.

- Die Rate R eines Codes S ist definiert als
 - Die Rate charakterisiert die Effizienz des Codes

$$R_S = \frac{\log |S|}{n}$$

- Die Distanz δ des Codes S ist definiert als
 - charakterisiert die Fehlerkorrektur oder Fehlererkennungsmöglichkeiten

$$\delta_S = \frac{d(S)}{n}$$

- Gute Codes haben hohe Raten und hohe Distanz
 - Beides lässt sich nicht zugleich optimieren

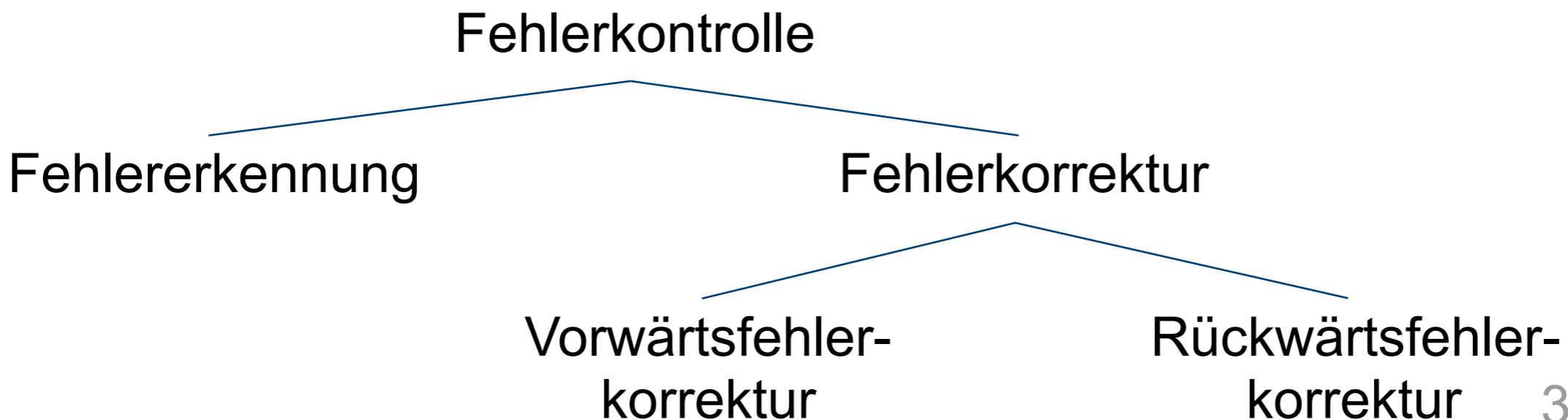
Block-Codes

- Block-Codes kodieren k Bits Originaldaten in n kodierte Bits
 - Zusätzlich werden $n-k$ Symbole hinzugefügt
 - Binäre Block-Codes können höchstens bis zu t Fehler in einem Code-Wort der Länge n mit k Originalbits erkennen, wobei (Gilbert-Varshamov-Schranke):

$$2^{n-k} \geq \sum_{i=0}^t \binom{n}{i}$$

- Das ist eine theoretische obere Schranke
- Beispiele
 - Bose Chaudhuri Hocquenghem (BCH) Codes
 - basierend auf Polynomen über endlichen Körpern (Galois-Körpern)
 - Reed Solomon Codes
 - Spezialfall nichtbinärer BCH-Codes

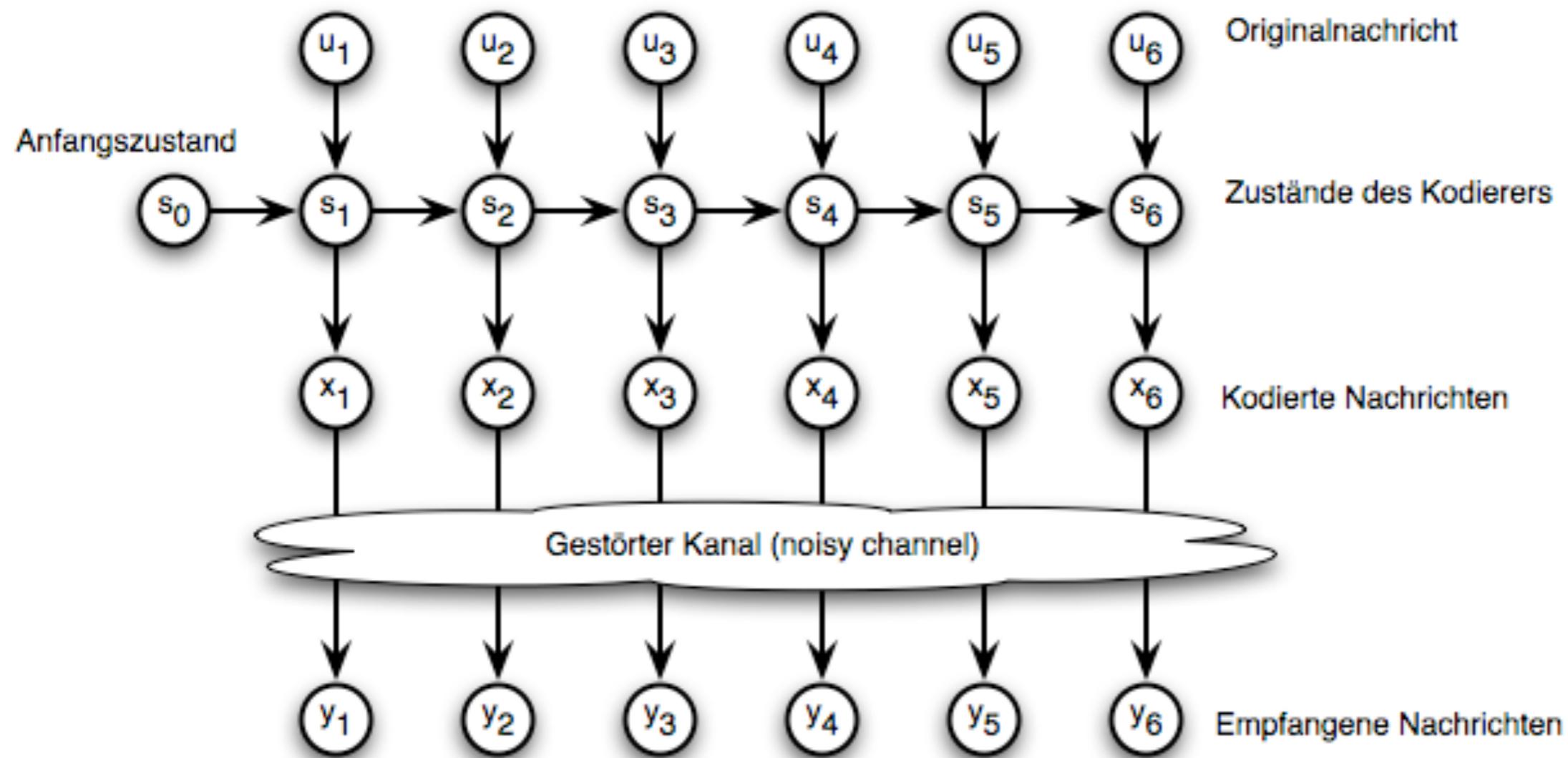
- Zumeist gefordert von der Vermittlungsschicht
 - Mit Hilfe der Frames
- Fehlererkennung
 - Gibt es fehlerhaft übertragene Bits?
- Fehlerkorrektur
 - Behebung von Bitfehlern
 - Vorwärtsfehlerkorrektur (Forward Error Correction)
 - Verwendung von redundanten Kodierung, die es ermöglicht Fehler ohne zusätzliche Übertragungen zu beheben
 - Rückwärtsfehlerkorrektur (Backward Error Correction)
 - Nach Erkennen eines Fehlers, wird durch weitere Kommunikation der Fehler behoben



Faltungs-Codes

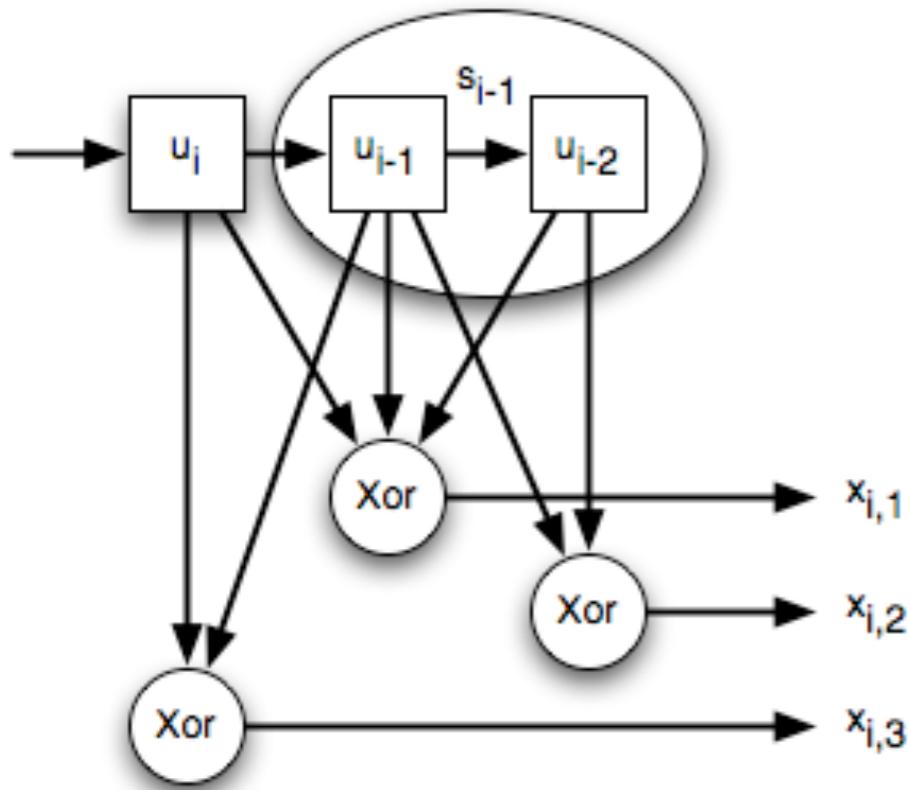
Faltungs-Codes (Convolutional Codes)

- Daten und Fehlerredundanz werden vermischt.
- k Bits werden auf n Bits abgebildet
- Die Ausgabe hängt von den k letzten Bits und dem internen Zustand ab.

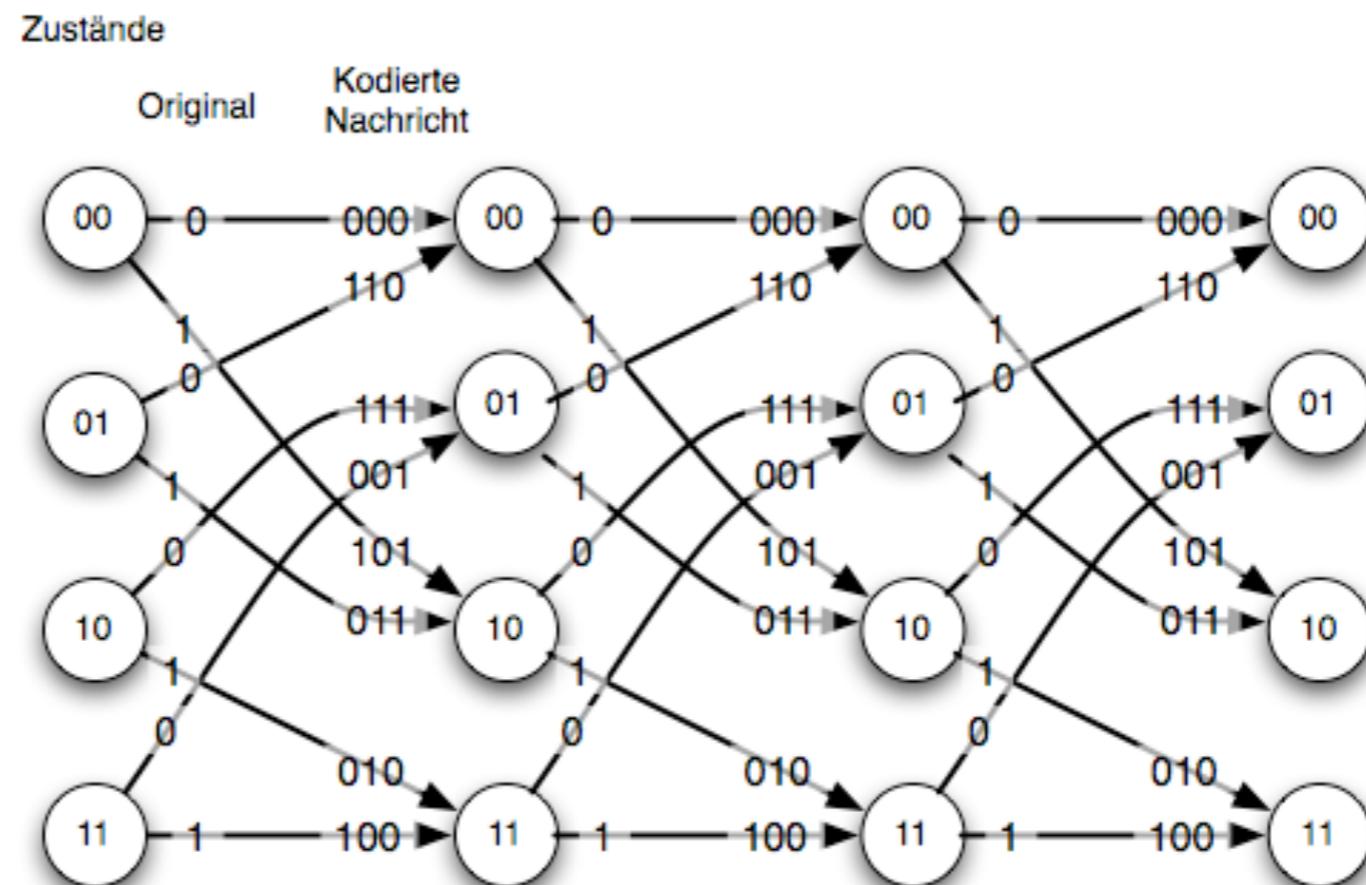


Beispiel

Faltungs-Kodierer



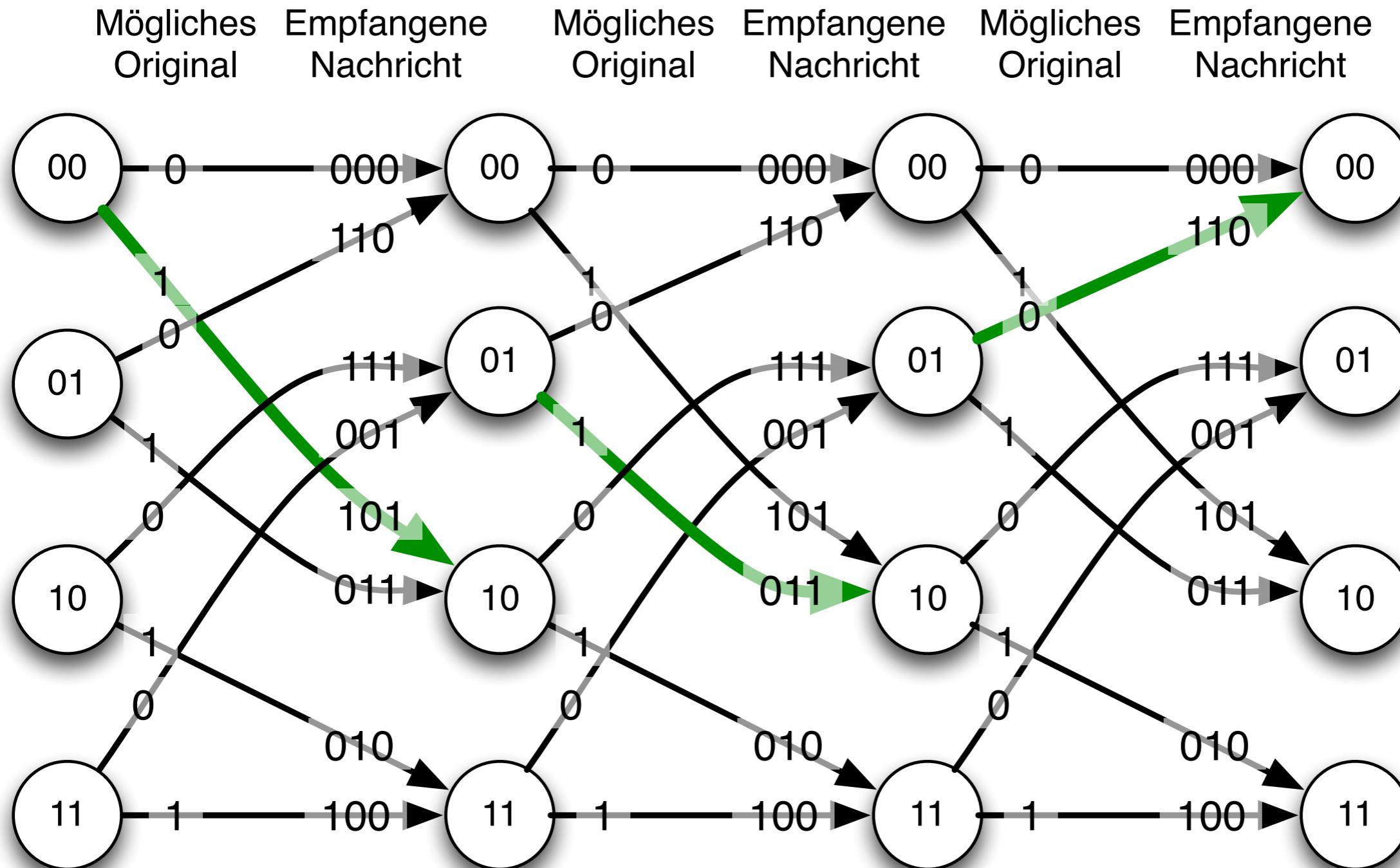
Trellis-Diagramm



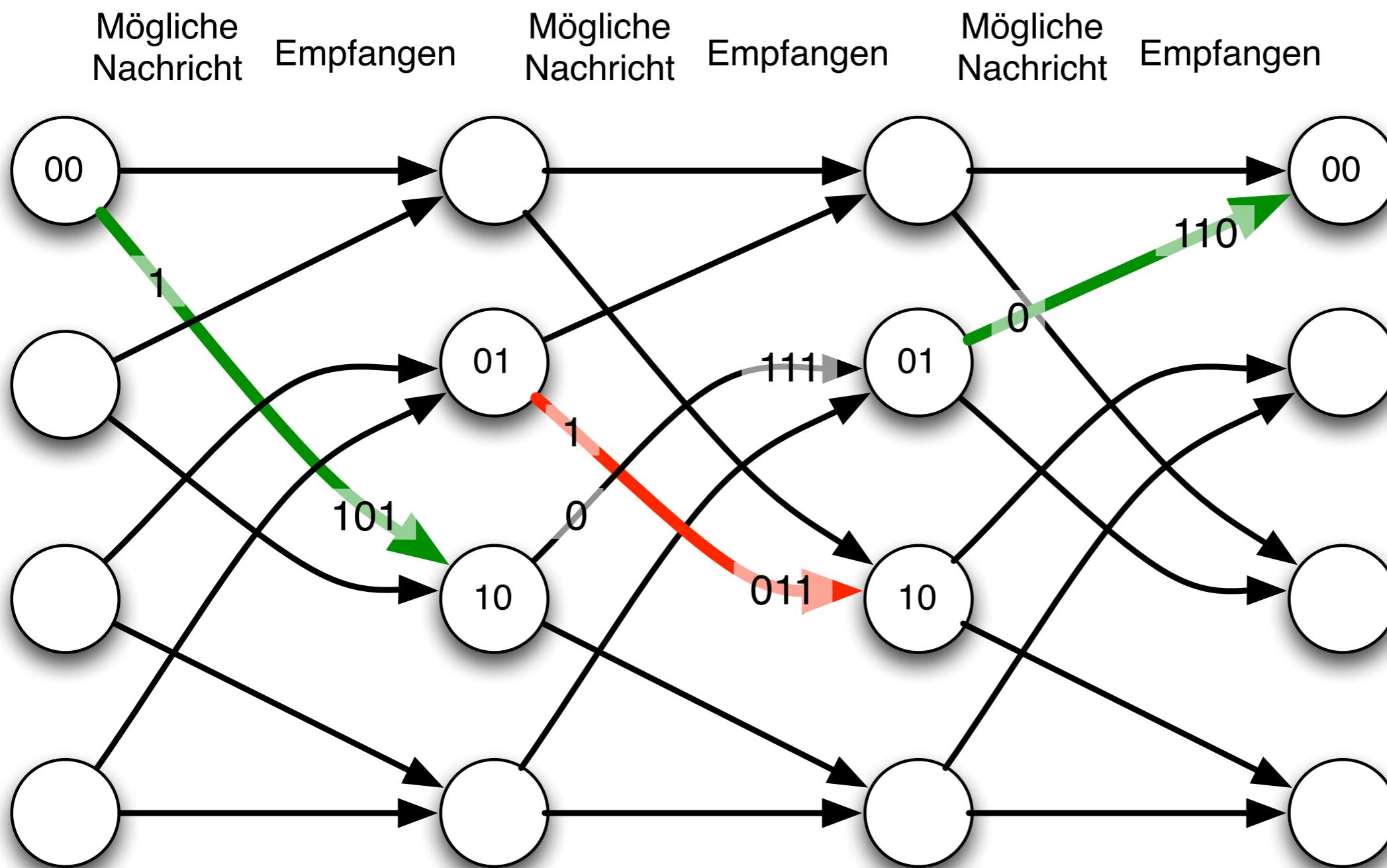
- Dynamische Programmierung
- Zwei notwendige Voraussetzungen für Dekodierung
 - (für den Empfänger) unbekannte Folge von Zuständen
 - beobachtete Folge von empfangenen Bits (möglicherweise mit Fehler)
- Der Algorithmus von Viterbi bestimmt die wahrscheinlichste Folge von Zuständen, welches die empfangenen Bits erklärt
 - Hardware-Implementation möglich

Dekodierung (I)

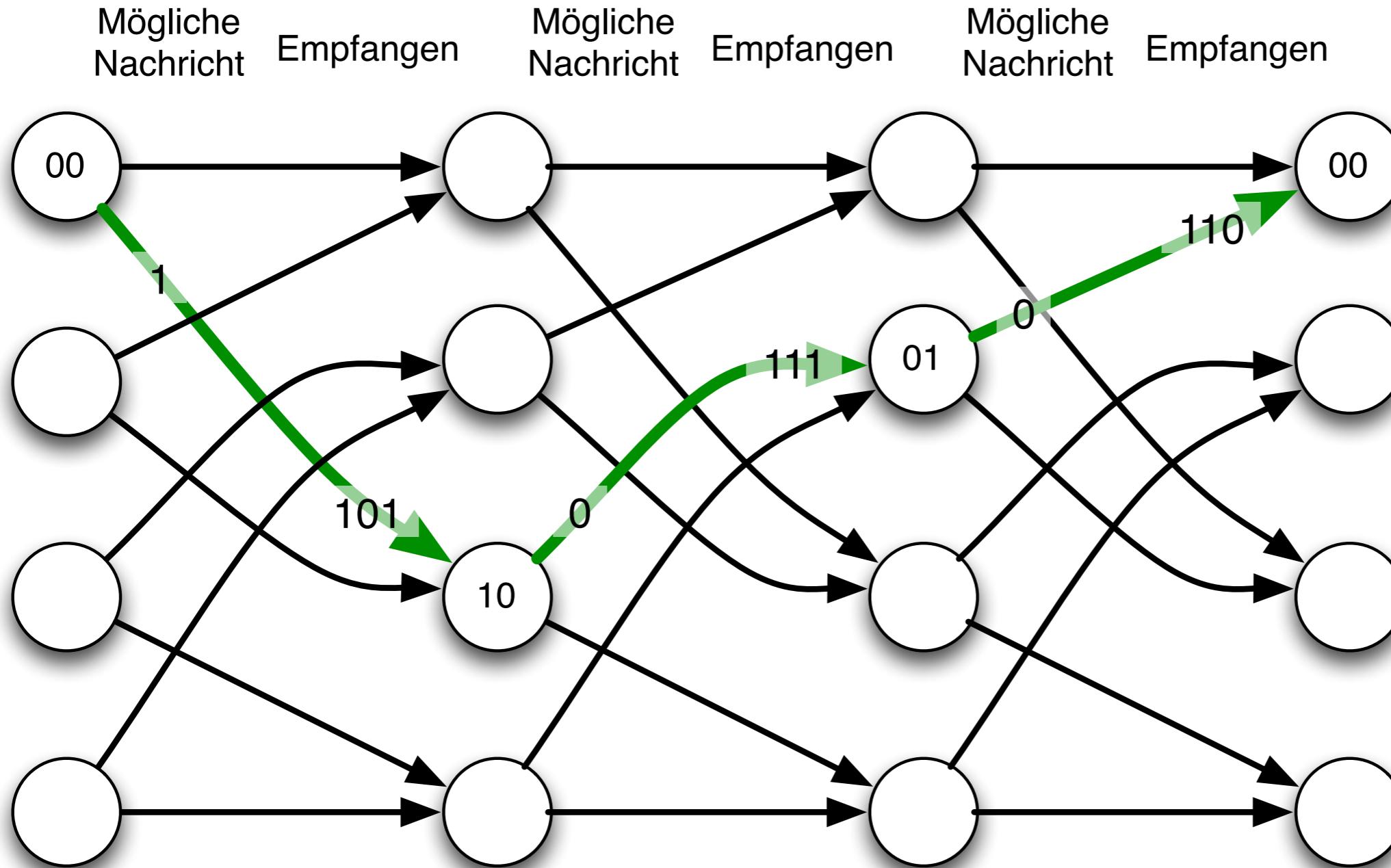
Zustände



Dekodierung (II)

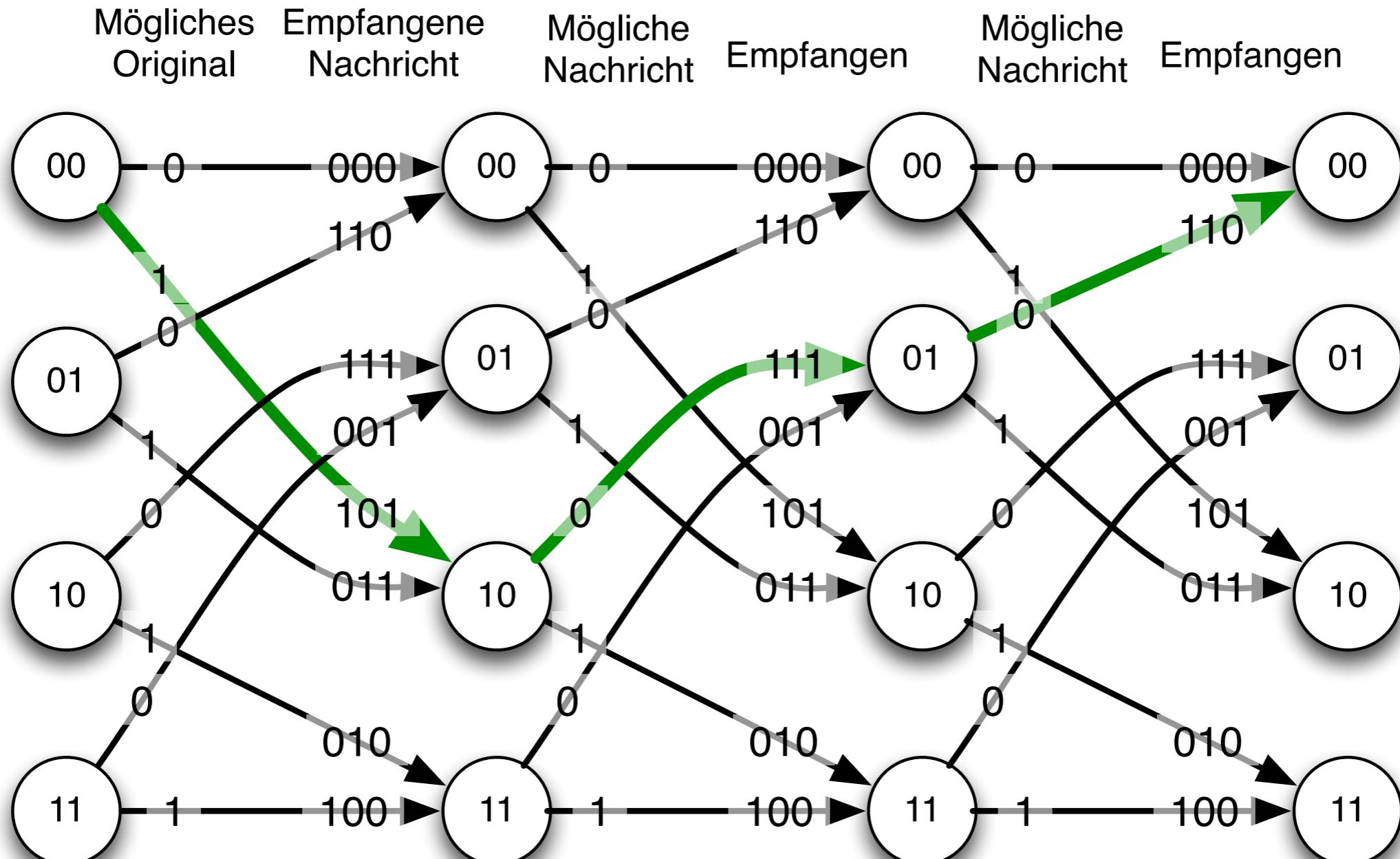


Dekodierung (III)



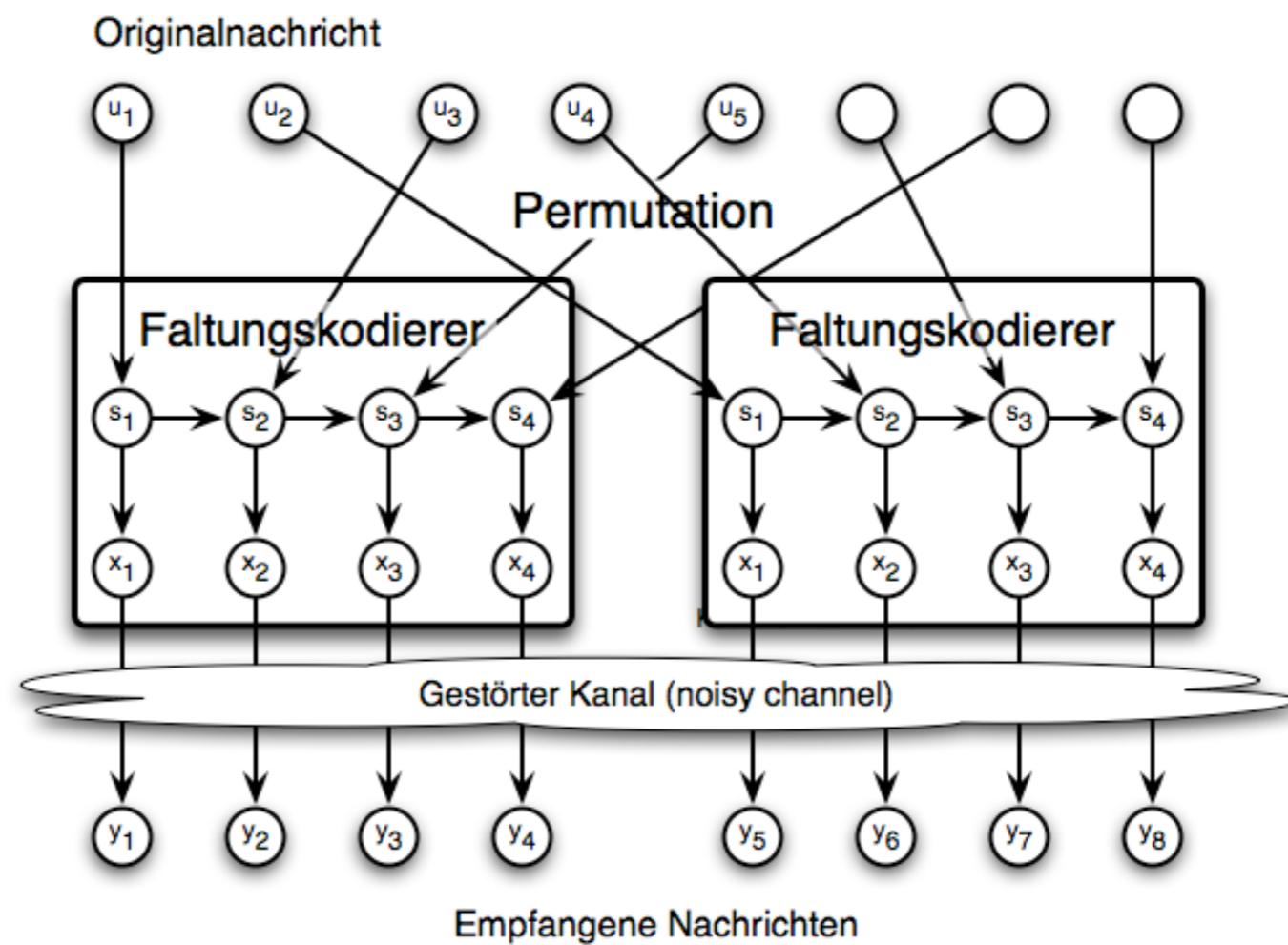
Dekodierung (IV)

Zustände



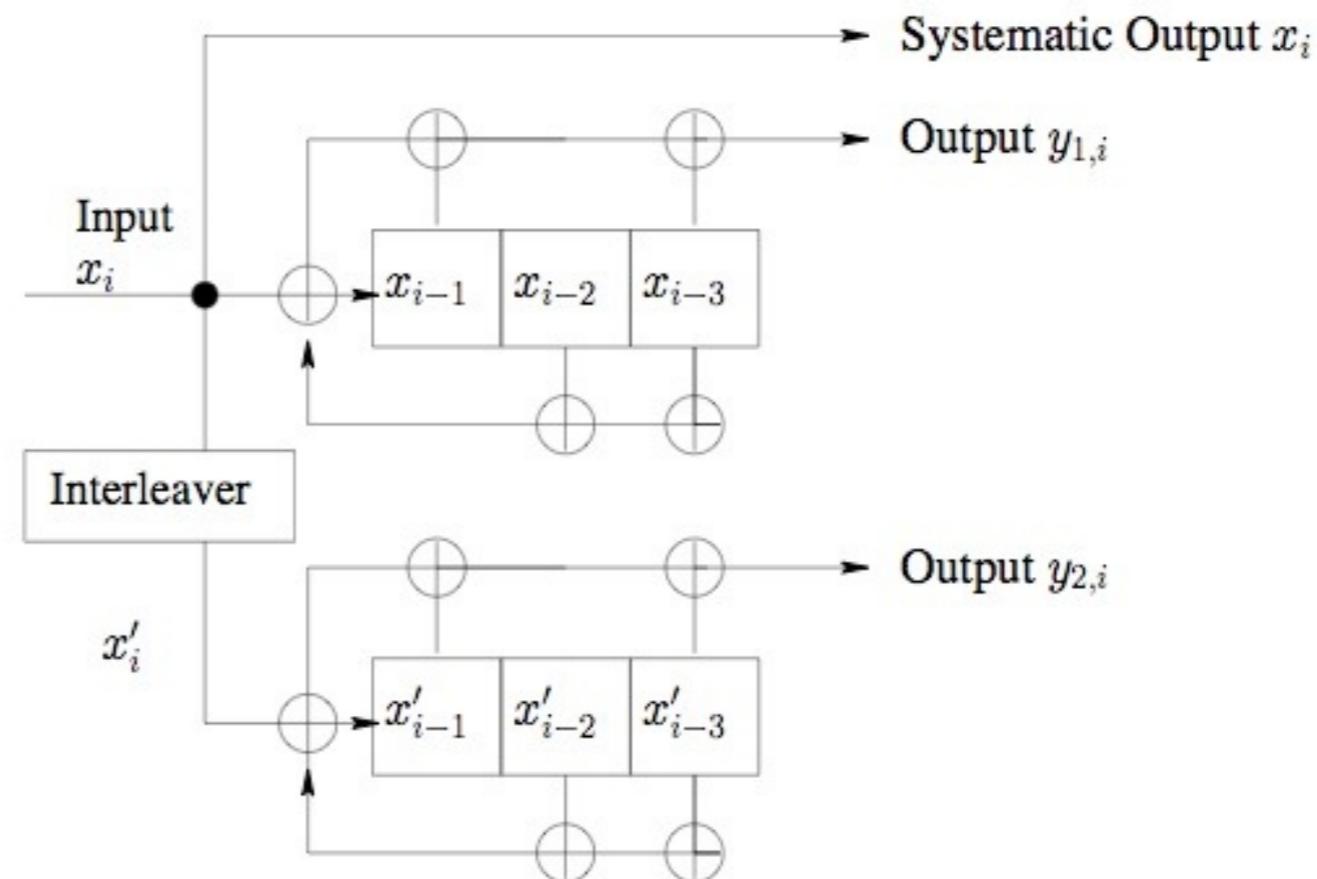
Turbo-Codes

- Turbo-Codes sind wesentlich effizienter als Faltungs-Codes
 - bestehen aus zwei Faltungs-Codes welche abwechselnd mit der Eingabe versorgt werden.
 - Die Eingabe wird durch eine Permutation (Interleaver) im zweiten Faltungs-Code umsortiert



Turbo-Codes

- Beispiel:
 - UMTS Turbo-Kodierer
- Dekodierung von Turbo-Codes ist effizienter möglich als bei Faltungscodes
- Kompensation von Bursts



Interleavers

- Fehler treten oftmals gehäuft auf (Bursts)
 - z.B.: Daten: 0 1 2 3 4 5 6 7 8 9 A B C D E F
 - mit Fehler: 0 1 2 3 ? ? ? ? ? 9 A B C D E F
- Dann scheitern klassische Kodierer ohne Interleavers
 - Nach Fehlerkorrektur (zwei Zeichen in Folge reparierbar):
0 1 2 3 4 5 ? 7 8 9 A B C D E F
- Interleaver:
 - Permutation der Eingabekodierung:
0 1 2 3
4 5 6 7
8 9 A B
C D E F
 - z.B. Row-column Interleaver:
0 4 8 C 1 5 9 D 2 6 A E 3 7 B F
 - mit Fehler: 0 4 8 C ? ? ? ? ? 6 A E 3 7 B F
 - Rückpermutiert: 0 ? ? 3 4 ? 6 7 8 ? A B C D ? F
 - nach FEC: 0 1 2 3 4 5 6 7 8 9 A B C D E F

Fehlererkennung: CRC

- Effiziente Fehlererkennung: Cyclic Redundancy Check (CRC)
- Praktisch häufig verwendeter Code
 - Hoher Fehlererkennungsrate
 - Effizient in Hardware umsetzbar
- Beruht auf Polynomarithmetik im Restklassenring \mathbb{Z}_2
 - Zeichenketten sind Polynome
 - Bits sind Koeffizienten des Polynoms

Rechnen in Z_2

- Rechnen modulo 2:
- Regeln:
 - Addition modulo 2 = Xor = Subtraktion modulo 2
 - Multiplikation modulo 2 = And

A	B	$A + B$
0	0	0
0	1	1
1	0	1
1	1	0

A	B	$A - B$
0	0	0
0	1	1
1	0	1
1	1	0

A	B	$A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

- Beispiel: $0 + (1 \cdot 0) + 1 + (1 \cdot 1) =$

Polynomarithmetik modulo 2

- Betrachte Polynome über den Restklassenring \mathbb{Z}_2
 - $p(x) = a_n x^n + \dots + a_1 x^1 + a_0$
 - Koeffizienten a_i und Variable x sind aus $\{0,1\}$
 - Berechnung erfolgt modulo 2
- Addition, Subtraktion, Multiplikation, Division von Polynomen wie gehabt

Zeichenketten und Polynomarithmetik

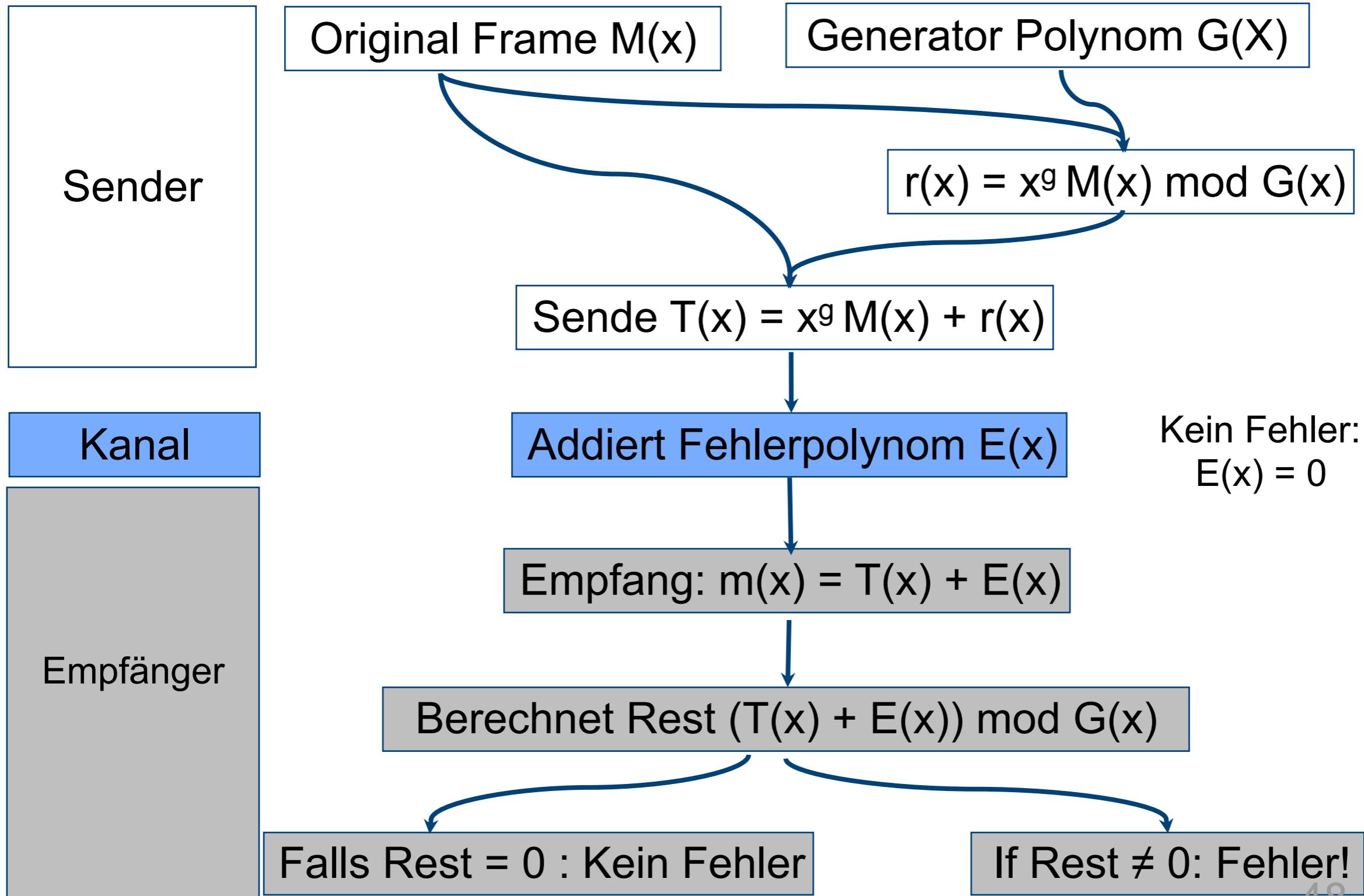
- Idee:
 - Betrachte Bitstring der Länge n als Variablen eines Polynoms
- Bit string: $b_n b_{n-1} \dots b_1 b_0$
Polynom: $b_n x^n + \dots + b_1 x^1 + b_0$
 - Bitstring mit $(n+1)$ Bits entspricht Polynom des Grads n
- Beispiel
 - $A \text{ xor } B = A(x) + B(x)$
 - Wenn man A um k Stellen nach links verschiebt, entspricht das
 - $B(x) = A(x) x^k$
- Mit diesem Isomorphismus kann man Bitstrings dividieren

Polynome zur Erzeugung von Redundanz: CRC

- Definiere ein Generatorpolynom $G(x)$ von Grad g
 - Dem Empfänger und Sender bekannt
 - Wir erzeugen g redundante Bits
- Gegeben:
 - Frame (Nachricht) M , als Polynom $M(x)$
- Sender
 - Berechne den Rest der Division $r(x) = x^g M(x) \bmod G(x)$
 - Übertrage $T(x) = x^g M(x) + r(x)$
 - Beachte: $x^g M(x) + r(x)$ ist ein Vielfaches von $G(x)$
- Empfänger
 - Empfängt $m(x)$
 - Berechnet den Rest: $m(x) \bmod G(x)$

- Keine Fehler:
 - $T(x)$ wird korrekt empfangen
- Bitfehler: $T(x)$ hat veränderte Bits
 - Äquivalent zur Addition eines Fehlerpolynoms $E(x)$
 - Beim Empfänger kommt $T(x) + E(x)$ an
- Empfänger
 - Empfangen: $m(x)$
 - Berechnet Rest $m(x) \bmod G(x)$
 - Kein Fehler: $m(x) = T(x)$,
 - dann ist der Rest 0
 - Bit errors: $m(x) \bmod G(x) = (T(x) + E(x)) \bmod G(x)$
 $= T(x) \bmod G(x) + E(x) \bmod G(x)$ 

CRC – Überblick



Der Generator bestimmt die CRC-Eigenschaften

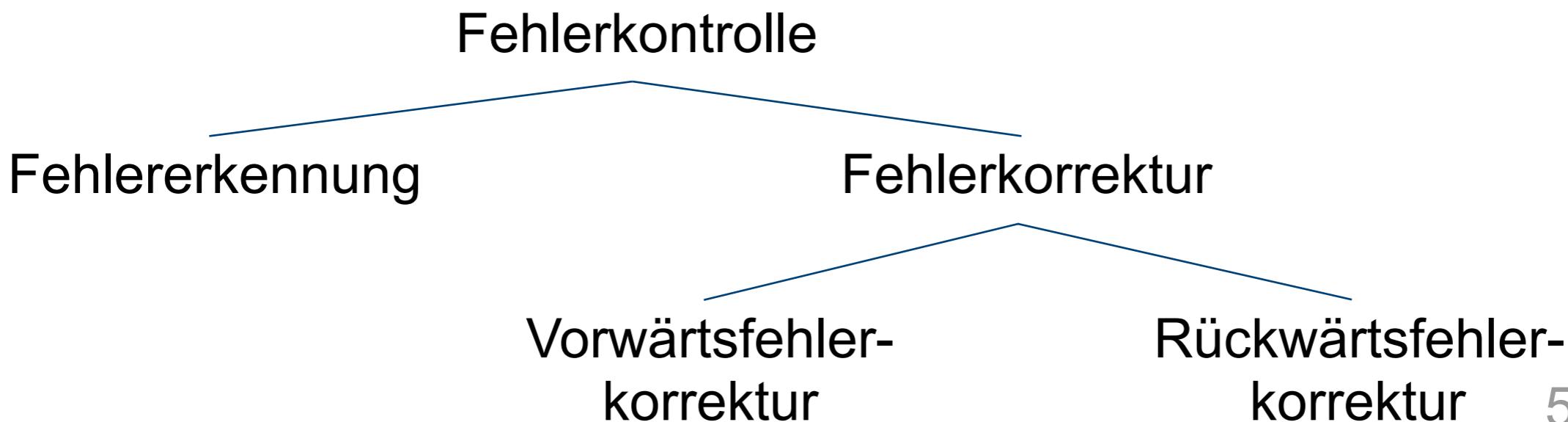
- Bit-Fehler werden nur übersehen, falls $E(x)$ ein Vielfaches von $G(x)$ ist
- Die Wahl von $G(x)$ ist trickreich:
 - Einzel-Bit-Fehler: $E(x) = x^i$ für Fehler an Position i
 - $G(x)$ hat mindestens zwei Summenterme, dann ist $E(x)$ kein Vielfaches von $G(x)$ ist
 - Zwei-Bit-Fehler: $E(x) = x^i + x^j = x^j (x^{i-j} + 1)$ für $i > j$
 - $G(x)$ darf nicht $(x^k + 1)$ teilen für alle k bis zur maximalen Frame-Länge
 - Ungerade Anzahl von Fehlern:
 - $E(x)$ hat nicht $(x+1)$ als Faktor
 - Gute Idee (?): Wähle $(x+1)$ als Faktor von $G(x)$
 - Dann ist $E(x)$ kein Vielfaches von $G(x)$
- Bei guter Wahl von $G(x)$:
 - kann jede Folge von r Fehlern erfolgreich erkannt werden
- Häufig:
 - $G(x)$ wird als irreduzibles Polynom gewählt, das heißt es ist kein Vielfache eines anderen (kleineren) Polynoms

CRC in der Praxis

- Verwendetes irreduzibles Polynom gemäß IEEE 802:
 - $x^{32}+x^{26}+x^{23}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$
- Achtung:
 - Fehler sind immer noch möglich
 - Insbesondere wenn der Bitfehler ein Vielfaches von $G(x)$ ist.
- Implementation:
 - Für jedes Polynom x^i wird $r(x,i) = x^i \bmod G(x)$ berechnet
 - Ergebnis von $B(x) \bmod G(x)$ ergibt sich aus
 - $b_0 r(x,0) + b_1 r(x,1) + b_2 r(x,2) + \dots + b_{k-1} r(x,k-1)$
 - Einfache Xor-Operation

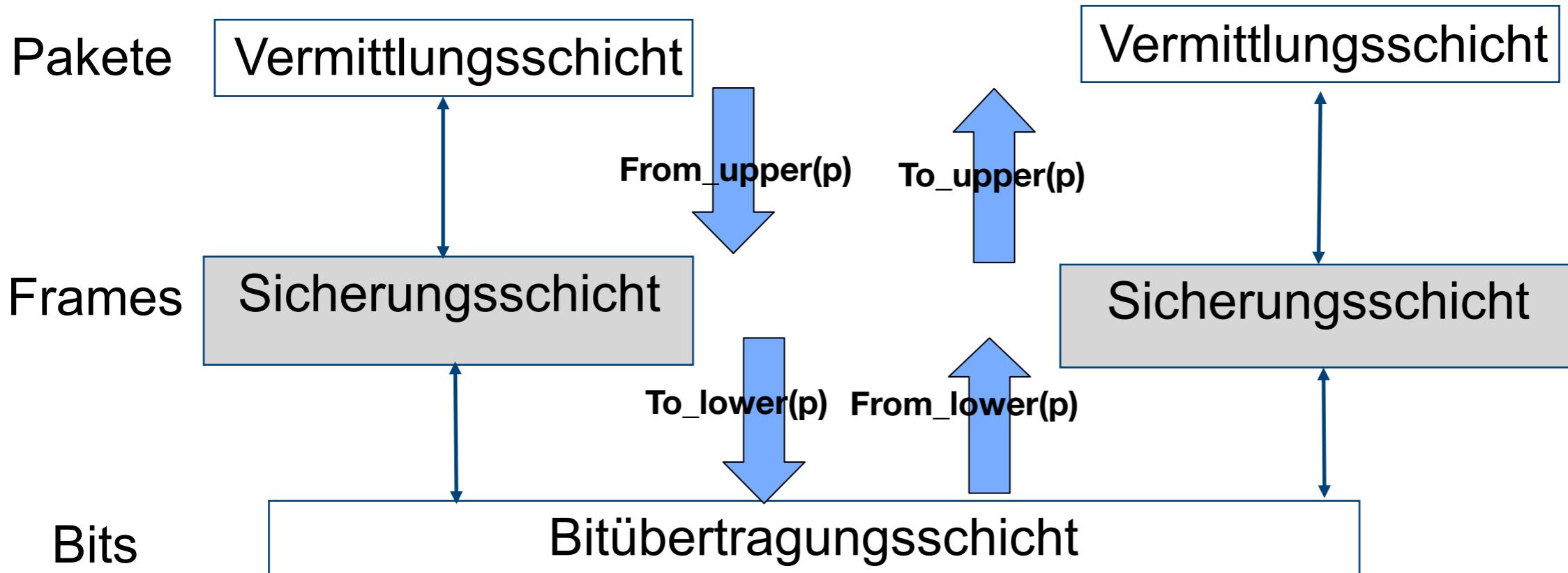
Fehlerkontrolle

- Zumeist gefordert von der Vermittlungsschicht
 - Mit Hilfe der Frames
- Fehlererkennung
 - Gibt es fehlerhaft übertragene Bits?
- Fehlerkorrektur
 - Behebung von Bitfehlern
 - Vorwärtsfehlerkorrektur (Forward Error Correction)
 - Verwendung von redundanten Kodierung, die es ermöglicht Fehler ohne zusätzliche Übertragungen zu beheben
 - Rückwärtsfehlerkorrektur (Backward Error Correction)
 - Nach Erkennen eines Fehlers, wird durch weitere Kommunikation der Fehler behoben



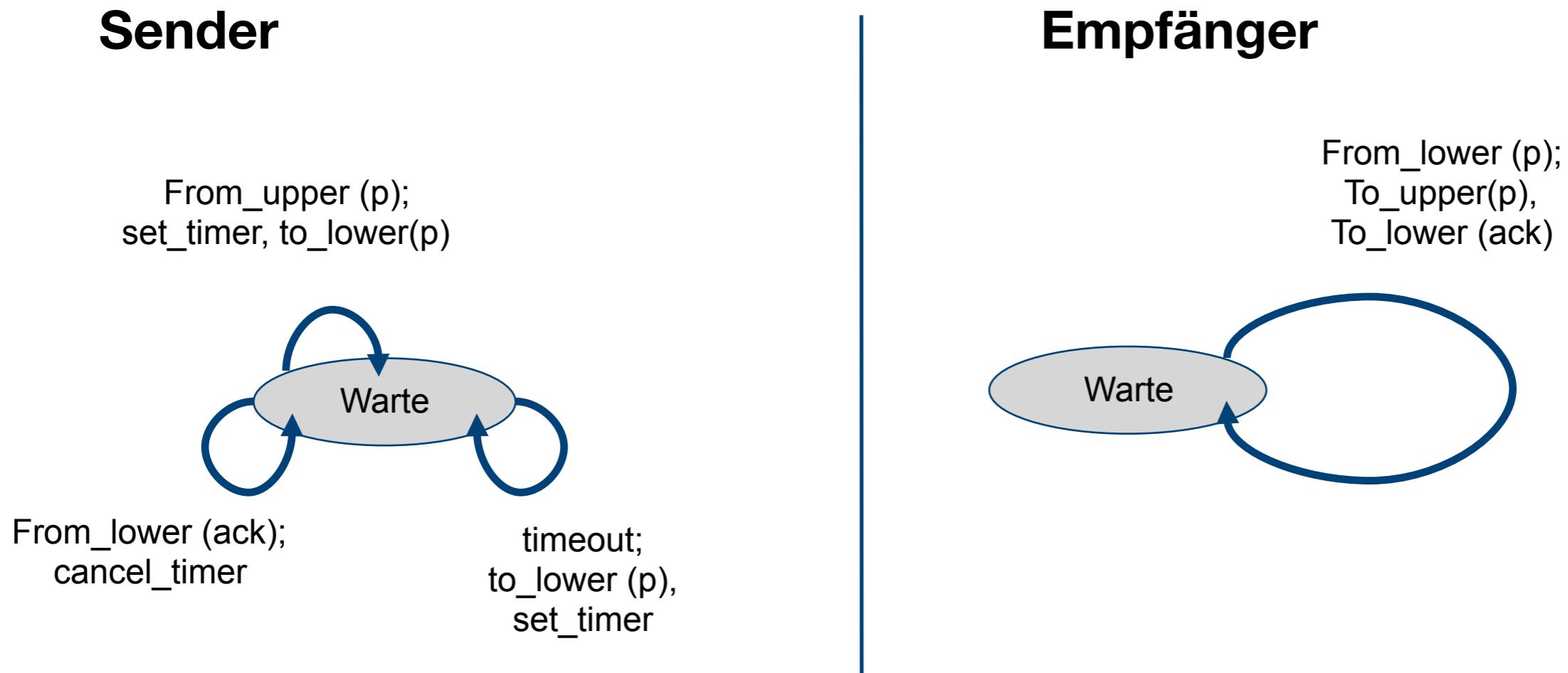
Rückwärtsfehlerkorrektur

- Bei Fehlererkennung muss der Frame nochmal geschickt werden
- Wie ist das Zusammenspiel zwischen Sender und Empfänger?



to_lower, from_lower beinhalten CRC
oder (bei Bedarf) Vorwärtsfehlerkorrektur

- Empfänger bestätigt Pakete dem Sender
 - Der Sender wartet für eine bestimmte Zeit auf die Bestätigung (acknowledgment)
 - Falls die Zeit abgelaufen ist, wird das Paket wieder versendet
- Erster Lösungsansatz



Diskussion

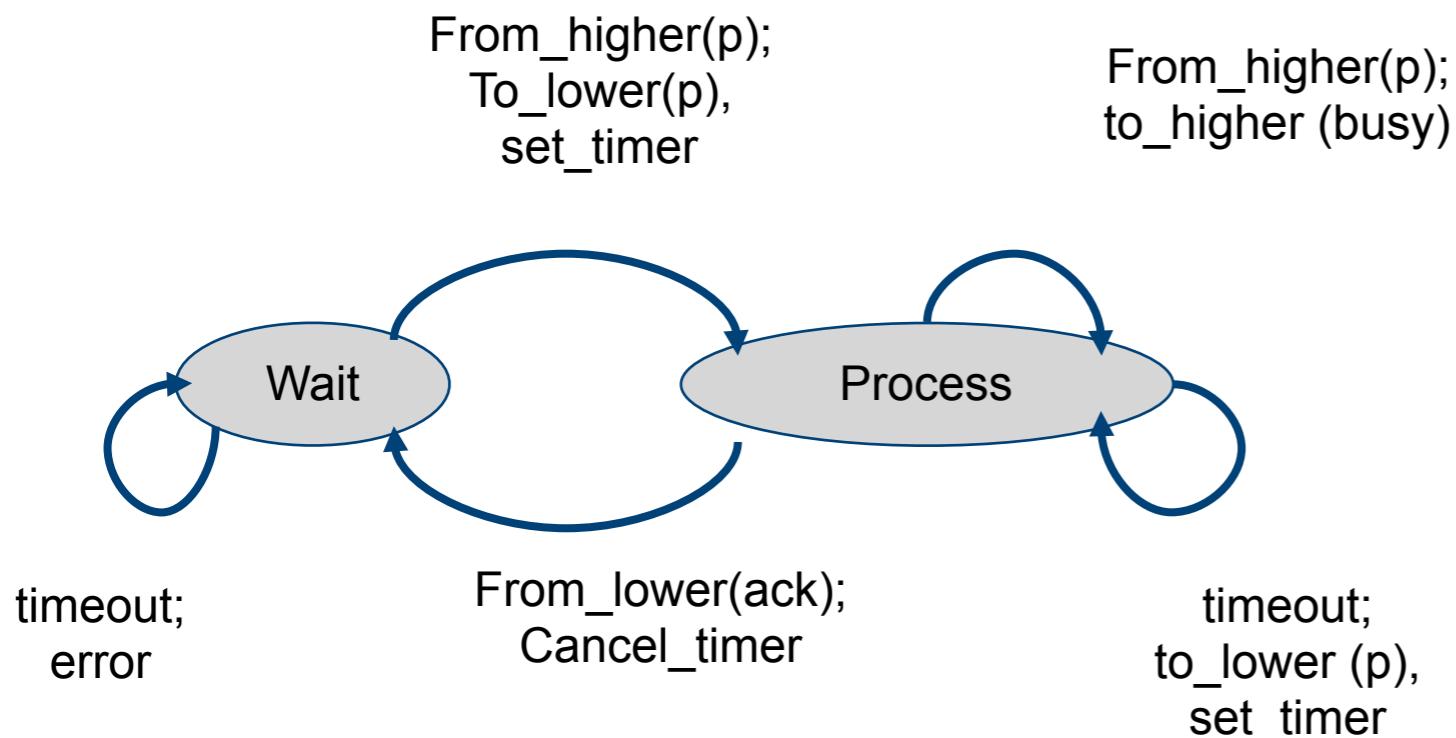
- Probleme
 - Sender ist schneller als Empfänger
 - Was passiert, wenn Bestätigungen verloren gehen?

2. Versuch

- Lösung des ersten Problems

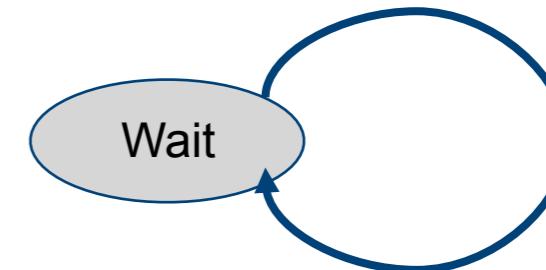
- Ein Paket nach dem anderen

- Sender



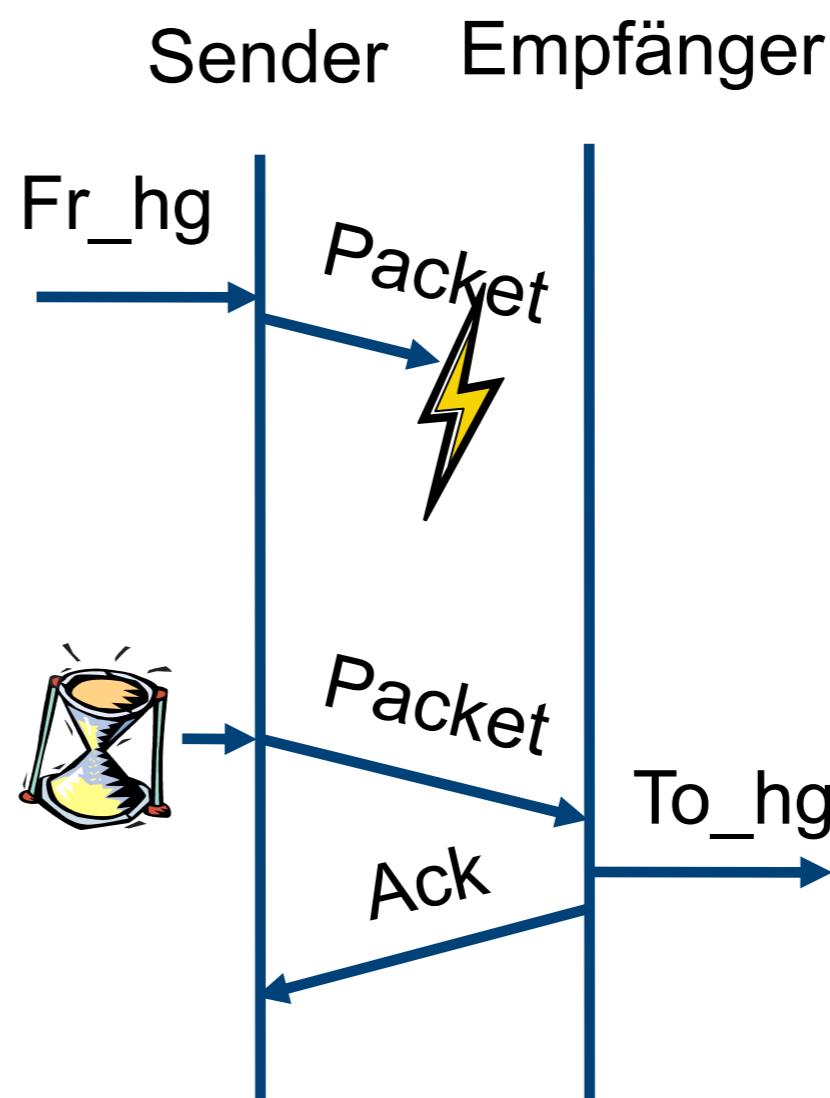
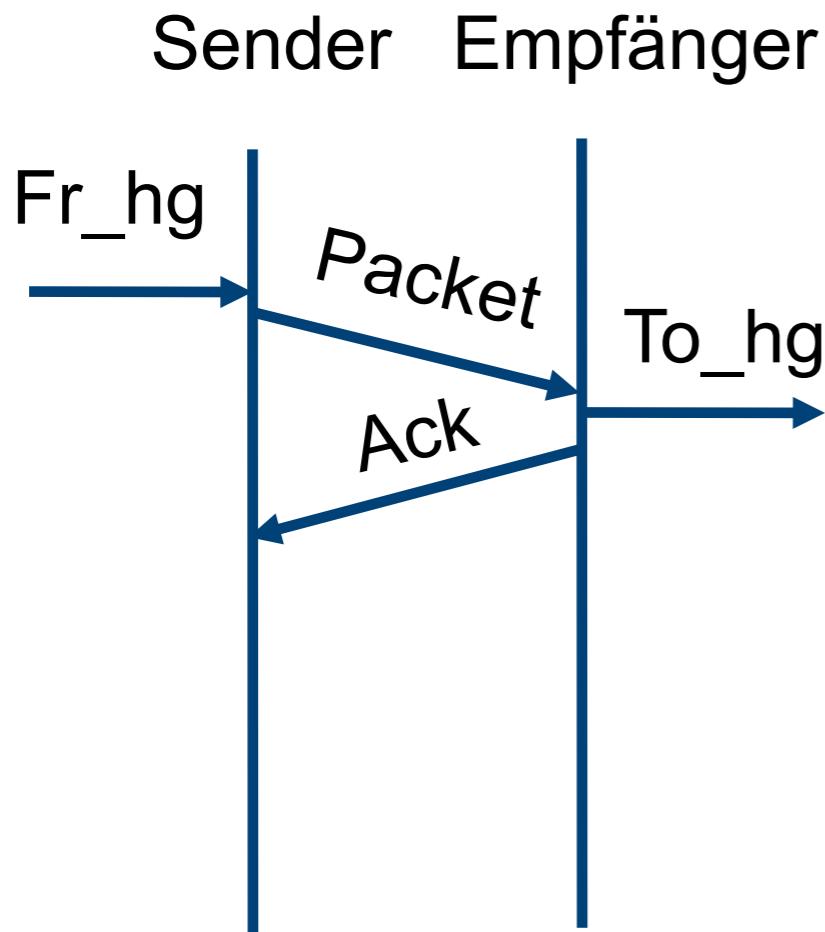
Empfänger

From_lower(p);
To_upper(p),
to_lower(ack)



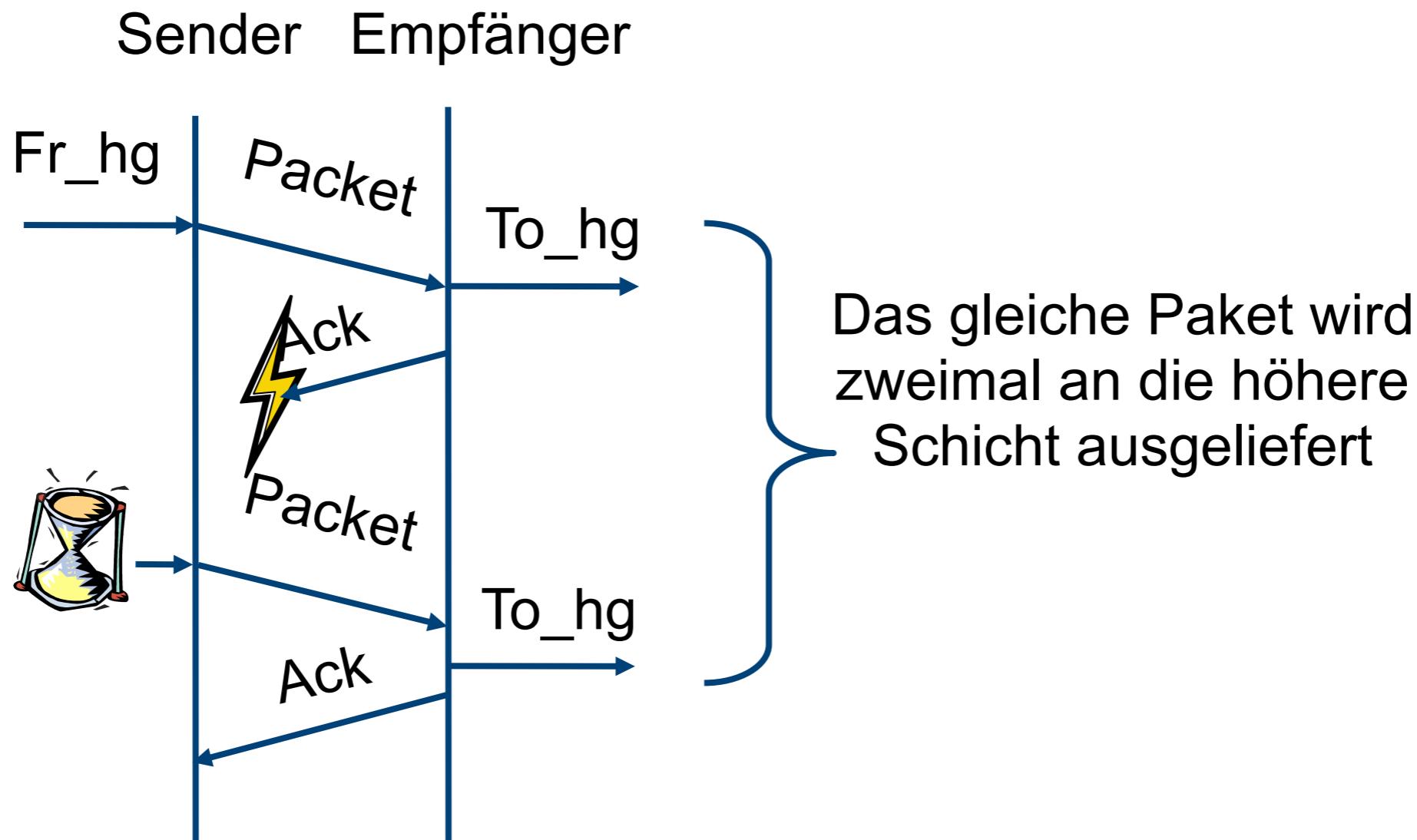
Diskussion

- Protokoll etabliert elementare Flusskontrolle



Diskussion

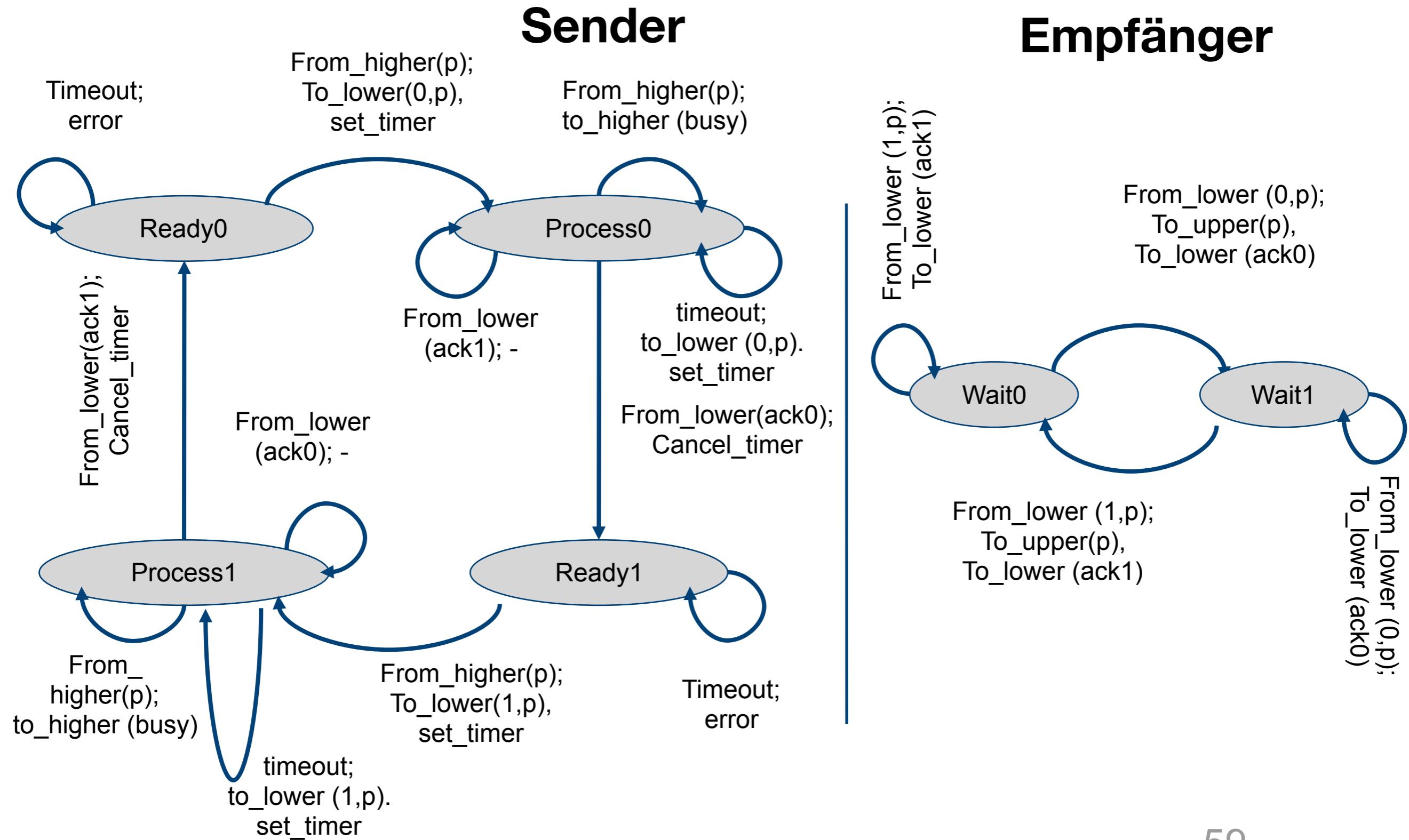
■ 2. Fall: Verlust von Bestätigung



Probleme der 2. Version

- Sender kann nicht zwischen verlorenem Paket und verlorener Bestätigung unterscheiden
 - Paket muss neu versendet werden
- Empfänger kann nicht zwischen Paket und redundanter Kopie eines alten Pakets unterscheiden
 - Zusätzliche Information ist notwendig
- Idee:
 - Einführung einer Sequenznummer in jedes Paket, um den Empfänger Identifikation zu ermöglichen
 - Sequenznummer ist im Header jedes Pakets
 - Hier: nur 0 oder 1
- Notwendig in Paket und Bestätigung
 - In der Bestätigung wird die Sequenznummer des letzten korrekt empfangenen Pakets mitgeteilt
 - (reine Konvention)

3. Versuch: Bestätigung und Sequenznummern

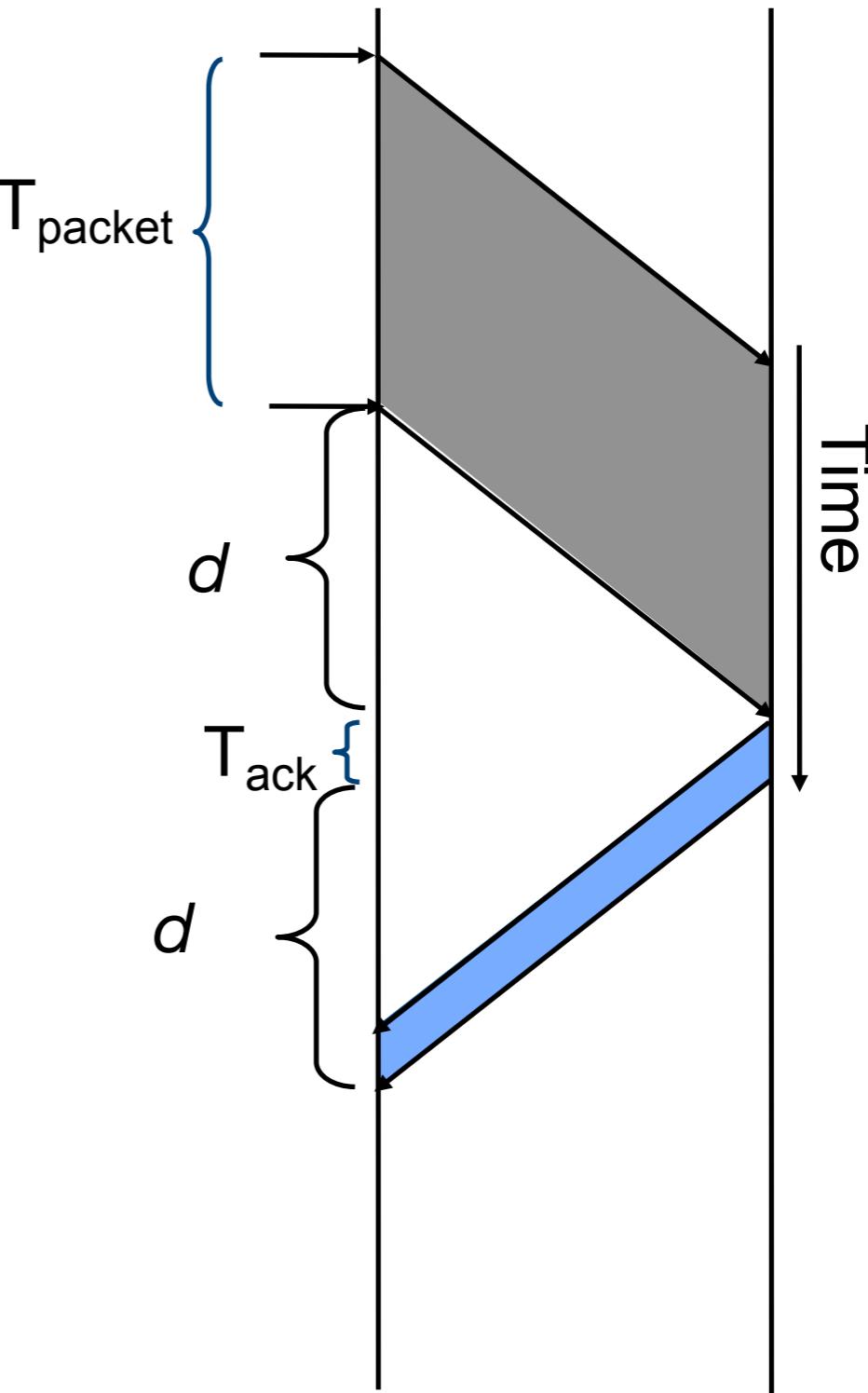


3. Version Alternating Bit Protocol

- Die 3. Version ist eine korrekte Implementation eines verlässlichen Protokolls über einen gestörten Kanal
 - Alternating Bit Protokoll
 - aus der Klasse der Automatic Repeat reQuest (ARQ) Protokolle
 - beinhaltet auch eine einfache Form der Flusskontrolle
- Zwei Aufgaben einer Bestätigung
 - Bestätigung, dass Paket angekommen ist
 - Erlaubnis ein neues Paket zu schicken

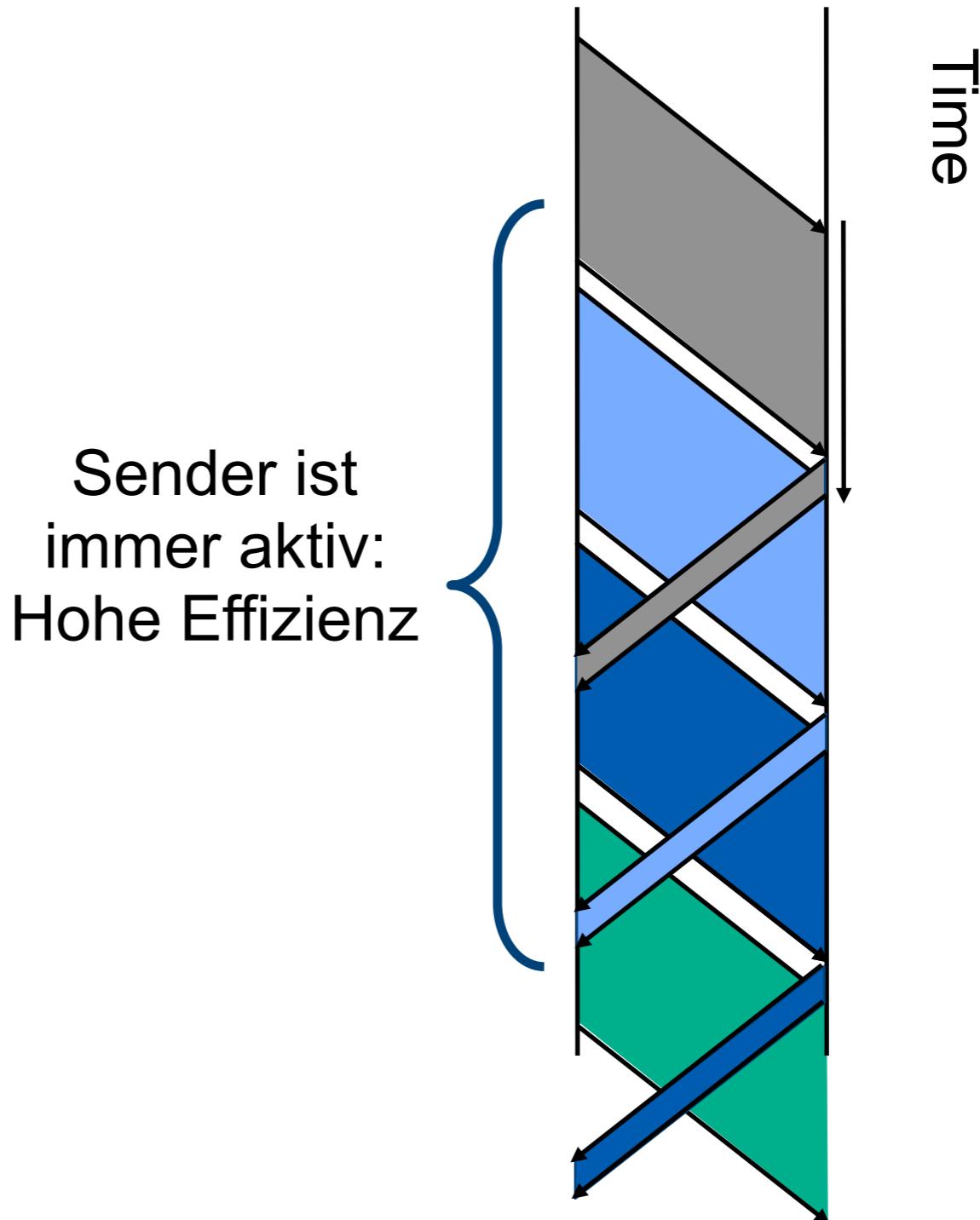
Alternating Bit Protocol – Effizienz

- Effizienz η
 - Definiert als das Verhältnis zwischen
 - der Zeit um zu senden
 - und der Zeit bis neue Information gesendet werden kann
 - (auf fehlerfreien Kanal)
 - $\eta = T_{\text{packet}} / (T_{\text{packet}} + d + T_{\text{ack}} + d)$
- Bei großen Delay ist das Alternating Bit Protocol nicht effizient



Verbesserung der Effizienz

- Durchgehendes Senden von Paketen erhöht Effizienz
 - Mehr “ausstehende” nicht bestätigte Pakete erhöhen die Effizienz
 - “Pipeline” von Paketen
- Nicht mit nur 1-Bit-Sequenznummer möglich



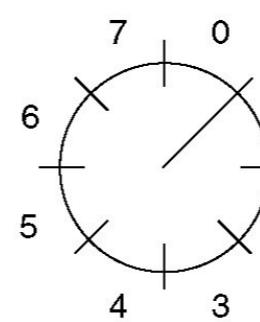
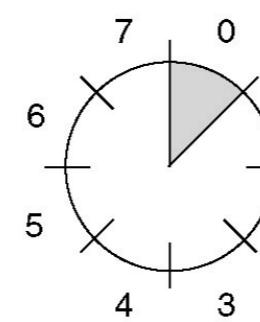
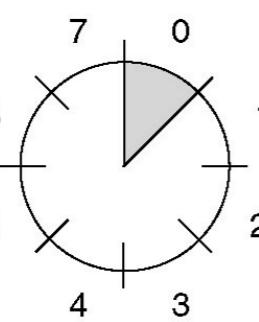
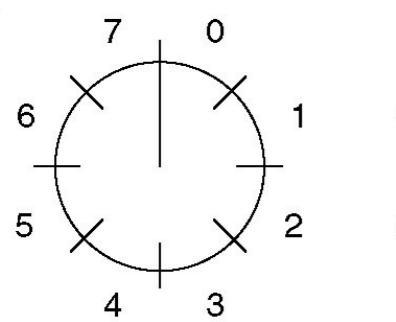
Gleitende Fenster

- Der Raum für Sequenznummern wird vergrößert
 - auf n Bits oder 2^n Sequenznummern
- Nicht alle davon können gleichzeitig verwendet werden
 - auch bei Alternating Bit Protocol nicht möglich
- “Gleitende Fenster” (sliding windows) bei Sender und Empfänger behandeln dieses Problem
 - Sender: Sende-Fenster
 - Folge von Sequenznummer, die zu einer bestimmten Zeit gesendet werden können
 - Empfänger: Empfangsfenster
 - Folge von Sequenznummer, die er zu einer bestimmten Zeit zu akzeptieren bereit ist
 - Größe der Fenster können fest sein oder mit der Zeit verändert werden
 - Fenstergröße entspricht Flusskontrolle

Beispiel

- “Sliding Window”-Beispiel für $n=3$ und fester Fenstergröße = 1
- Der Sender zeigt die momentan unbestätigten Sequenznummern an
 - Falls die maximale Anzahl nicht bestätigter Frames bekannt ist, dann ist das das Sende-Fenster

Sender



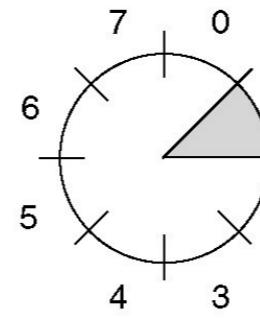
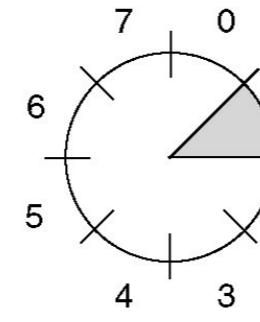
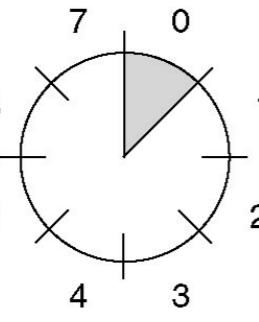
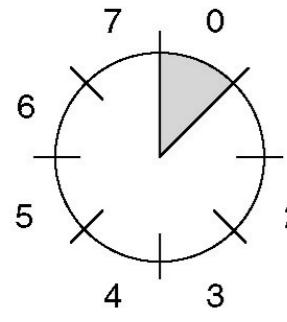
a. Initial: Nichts versendet

b. Nach Senden des 1. Frames mit Seq.Nr. 0

c. Nach dem Empfang des 1. Frame

d. Nach dem Empfang der Bestätigung

Receiver



(a)

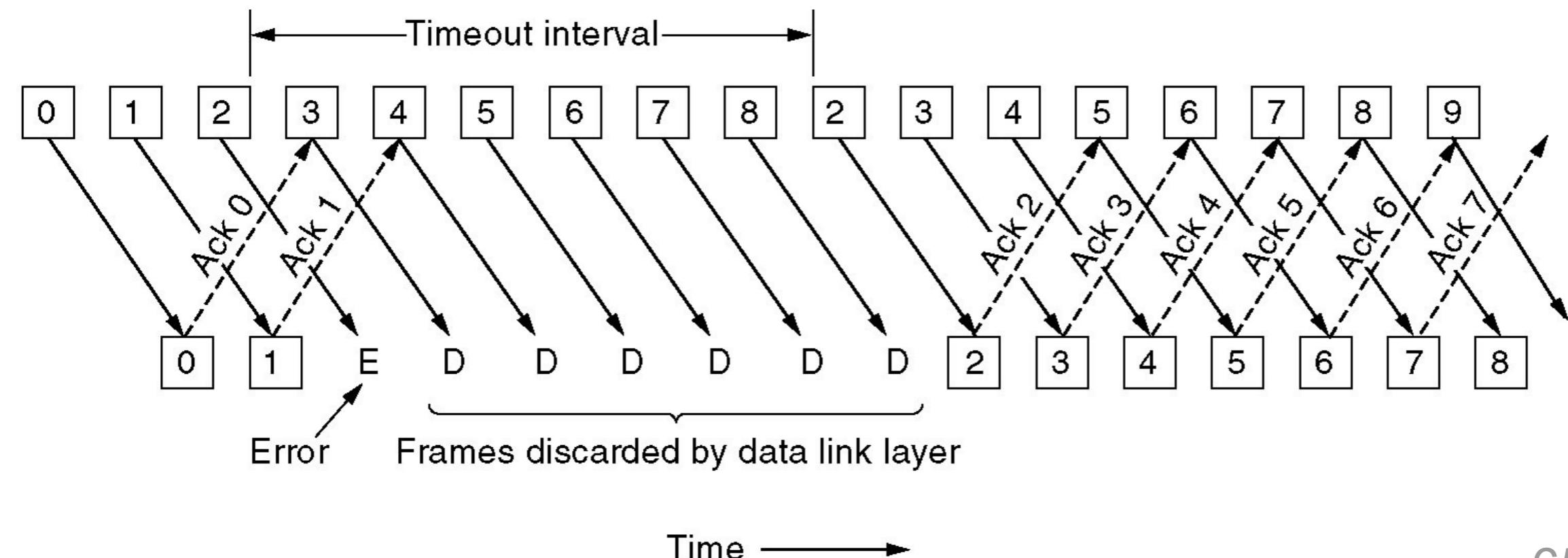
(b)

(c)

(d)

Übertragungsfehler und Empfangsfenster

- Annahme:
 - Sicherungsschicht muss alle Frames korrekt in der richtigen Reihenfolge verschicken
 - Sender “pipelined” Paket zur Erhöhung der Effizienz
- Bei Paketverlust:
 - werden alle folgenden Pakete ebenfalls fallen gelassen



Go-back-N

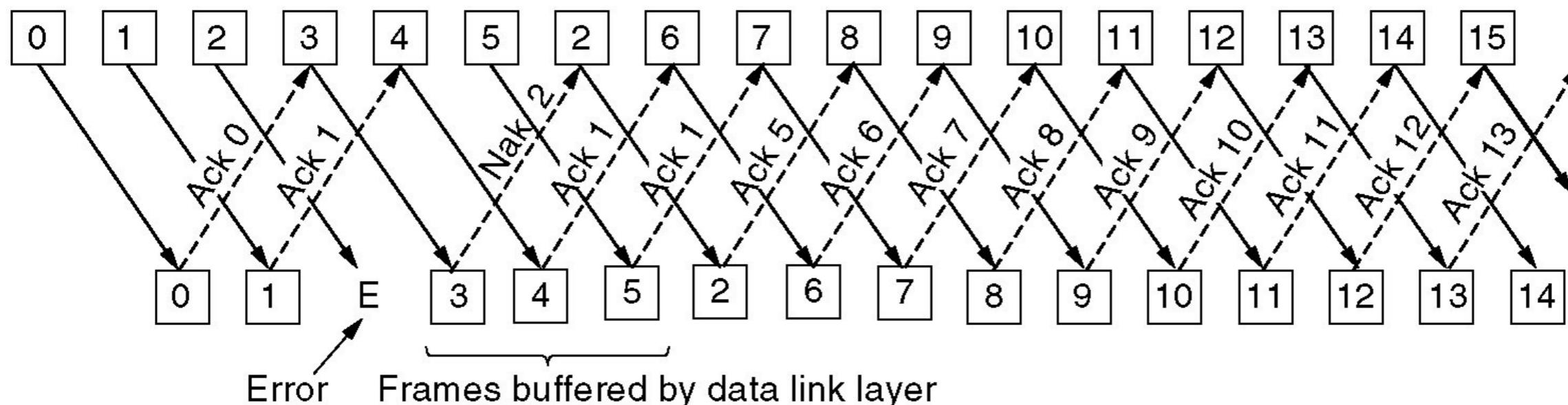
- Mit Empfangsfenster der Größe 1 können die Frames, die einem verloren Frame folgen, nicht durch den Empfänger bearbeitet werden
 - Sie können einfach nicht bestätigt werden, da nur eine Bestätigung für das letzte korrekt empfangene Paket verschickt wird
- Der Sender wird einen “Time-Out” erhalten
 - Alle in der Zwischenzeit versandten Frames müssen wieder geschickt werden
 - “Go-back N” Frames!
- Kritik
 - Unnötige Verschwendung des Mediums
 - Spart aber Overhead beim Empfänger

Selektierte Wiederholung

■ Angenommen

- der Empfänger kann die Pakete puffern, welche in der Zwischenzeit angekommen sind
- d.h. das Empfangsfenster ist größer als 1

■ Beispiel



- Der Empfänger informiert dem Sender fehlende Pakete mit negativer Bestätigung
- Der Sender verschickt die fehlenden Frames selektiv
- Sobald der fehlende Frame ankommt, werden alle (in der korrekten Reihenfolge) der Vermittlungsschicht übergeben

Duplex-Betrieb und Huckepack

- **Simplex**

- Senden von Informationen in einer Richtung

- **Duplex**

- Senden von Informationen in beide Richtungen

- **Bis jetzt:**

- Simplex in der Vermittlungsschicht
 - Duplex in der Sicherungsschicht

- **Duplex in den höheren Schichten**

- Nachrichten und Datenpakete separat in jeder Richtung
 - Oder Rucksack-Technik
 - Die Bestätigung wird im Header eines entgegen kommenden Frames gepackt



Der Mediumzugriff in der Sicherungsschicht

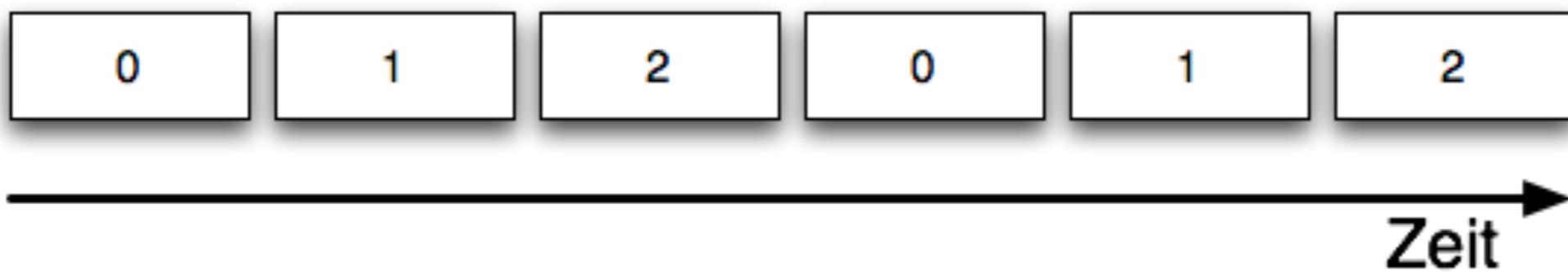
- Die Bitübertragung kann erst stattfinden, wenn das Medium reserviert wurde
 - Funkfrequenz bei drahtloser Verbindung (z.B. W-LAN 802.11, GSM, GPRSM)
 - Zeitraum bei einem Kabel mit mehreren Rechnern (z.B. Ethernet)
- Aufgabe der Sicherungsschicht
 - Koordination zu komplex für die “einfache” Bitübertragungsschicht

Der Mediumzugriff in der Sicherungsschicht

- Statisches Multiplexen
- Dynamische Kanalbelegung
 - Kollisionsbasierte Protokolle
 - Kollisionsfreie Protokolle (contention-free)
 - Protokolle mit beschränkten Wettbewerb (limited contention)

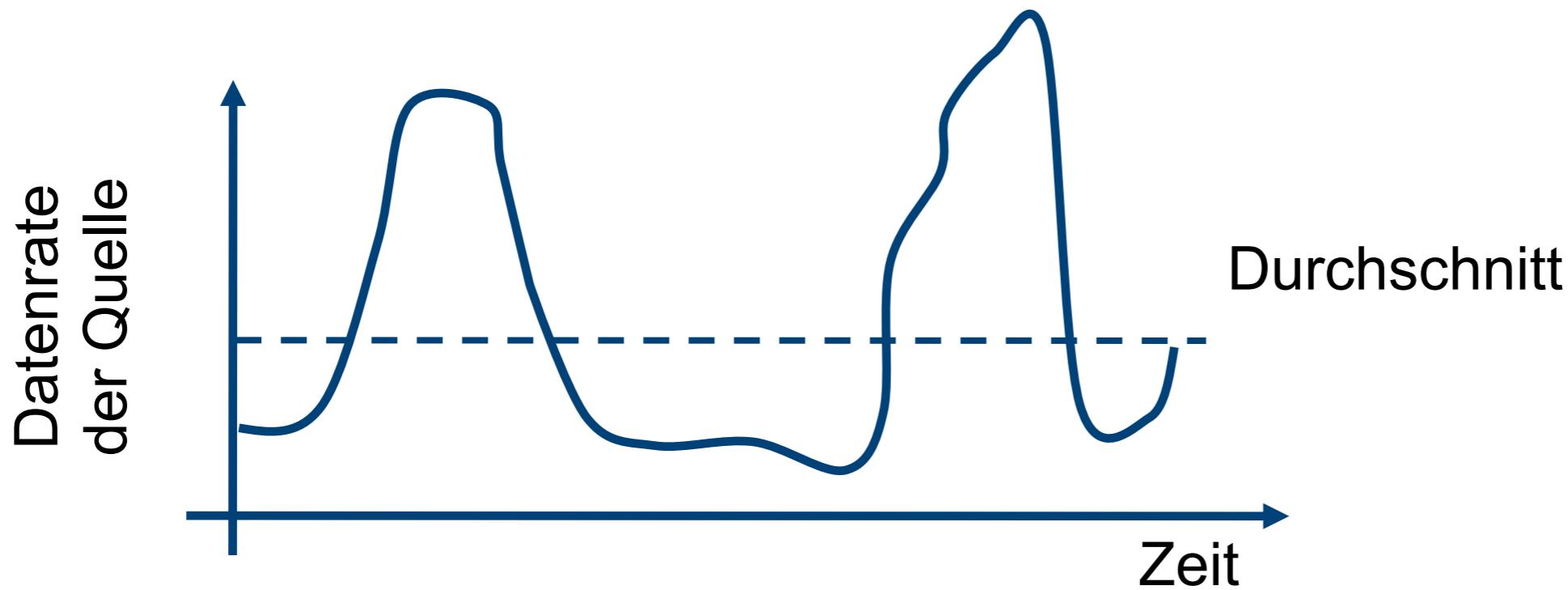
Statisches Multiplexen

- Gegeben sei eine einzelne Leitung (Ressource)
- Mehreren Kommunikations-verbindungen werden feste Zeiträume/Kanäle (slots/channels) zugewiesen
 - Oder: Feste Frequenzbänder werden ihnen zugewiesen
- Feste Datenraten und entsprechenden Anteilen am Kanal
 - Quellen lasten die Leitung aus



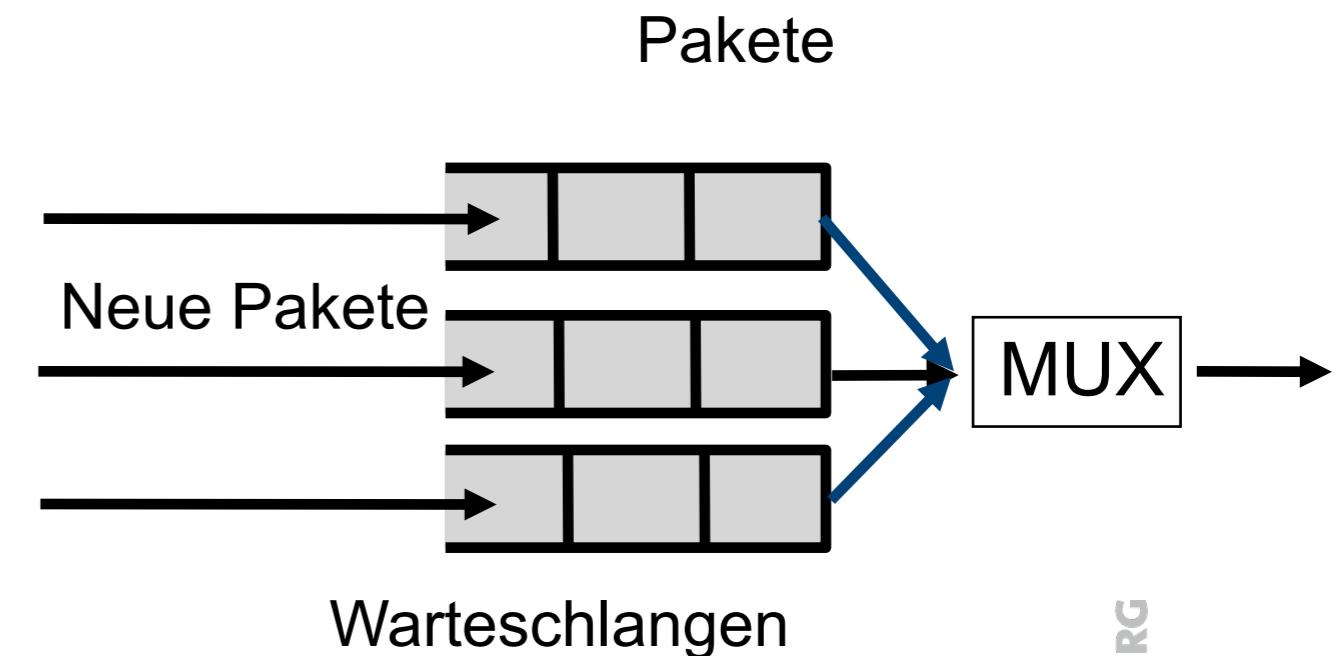
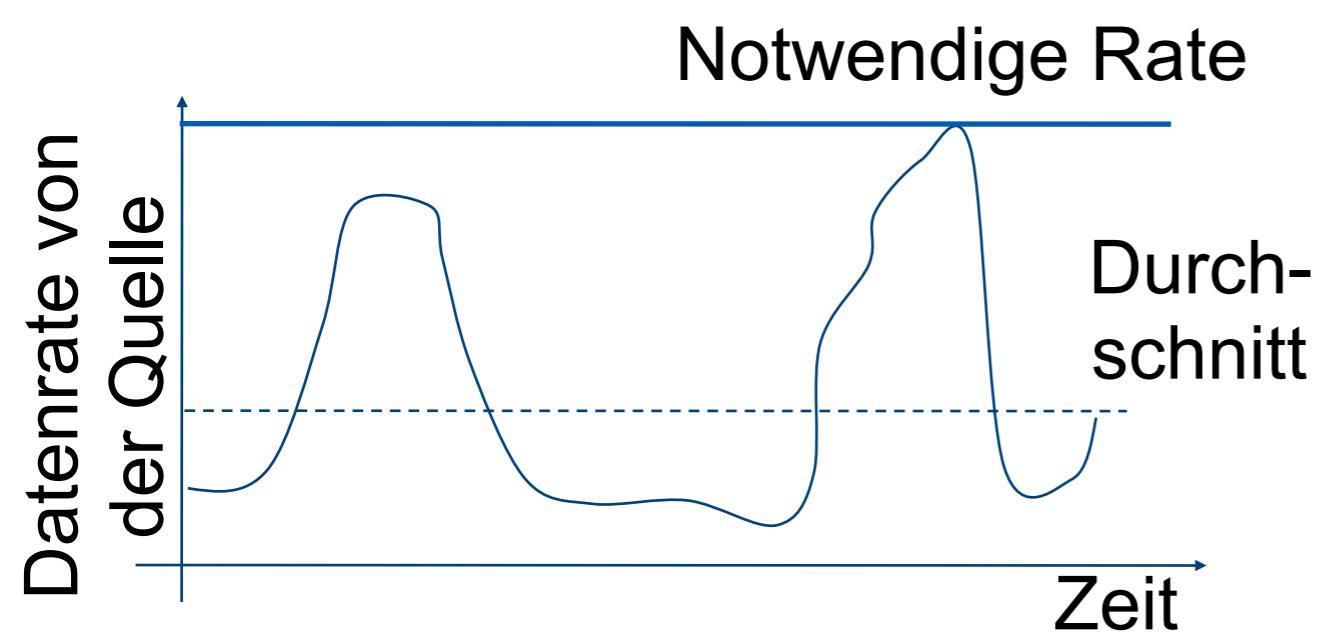
Verkehrsspitzen (bursty traffic)

- Problem: Verkehrsspitzen (bursty traffic)
 - Definition: Großer Unterschied zwischen Spitze und Durchschnitt
 - In Rechnernetzwerken: Spitze/Durchschnitt = 1000/1 nicht ungewöhnlich



Verkehrsspitzen und statisches Multiplexen

- Leitung für statisches Multiplexen:
 - entweder
 - Genügend große Kapazität um mit dem Peak fertig zu werden
 - Verschwendungen, da die Durchschnittsraten den Kanal nicht auslasten wird
 - oder
 - Ausgelegt für Durchschnittsraten
 - Versehen mit Warteschlangen (queue)
 - Vergrößerung der Verzögerung (delay) der Pakete



Verkehrsspitzen und statisches Multiplexen - Verzögerung

- Vergleich der Verzögerung
- Ausgangsfall:
 - Kein Multiplexing
 - Einfacher Datenquelle mit Durchschnittsrate ρ (bits/s) und der Leitungskapazität C bits/s
 - Sei T die Verzögerung
- Multiplex-Fall
 - Die Datenquelle wird in N Quellen unterteilt mit der selben Datenrate
 - Statischer Multiplex über die selbe Leitung
 - Dann ergibt sich (im wesentlichen) die Verzögerung: $N T$
- Schluss: Statisches Multiplexen vergrößert den Delay eines Pakets in der Regel um den Faktor N
 - Grund: Bei einer Verkehrsspitze sind $n-1$ Kanäle leer

Der Mediumzugriff in der Sicherungsschicht

- Statisches Multiplexen
- Dynamische Kanalbelegung
 - Kollisionsbasierte Protokolle
 - Kollisionsfreie Protokolle (contention-free)
 - Protokolle mit beschränkten Wettbewerb (limited contention)

Dynamische Kanalzuweisung – MAC

- Statisches Multiplexing ist nicht geeignet für Datenverbindung mit Spitzen
- Alternative: Zuweisung des Slots/Kanals an die Verbindung mit dem größten Bedarf
 - Dynamische Medium-Belegung
 - statt fester
- Der Mediumzugriff wird organisiert:
 - Mediumszugriff-Protokoll (Medium Access Control protocol - MAC)

Annahmen

- Stationsmodell (terminal model)
 - N unabhängige Stationen möchten eine Leitung/Ressource teilen
 - Mögliches Lastmodell:
 - Wahrscheinlichkeit, dass ein Paket im Intervall der Länge Δt erzeugt wird ist $\lambda \Delta t$ für eine Konstante λ
- Eine Leitung/Kanal
 - für alle Stationen
 - Keine weitere Verbindungen möglich
- Collision assumption
 - Nur ein einfacher Frame kann auf dem Kanal übertragen werden
 - Zwei (oder mehr) sich zeitlich überschneidende Frames kollidieren und werden gelöscht
 - Noch nicht einmal Teile kommen an

Annahmen

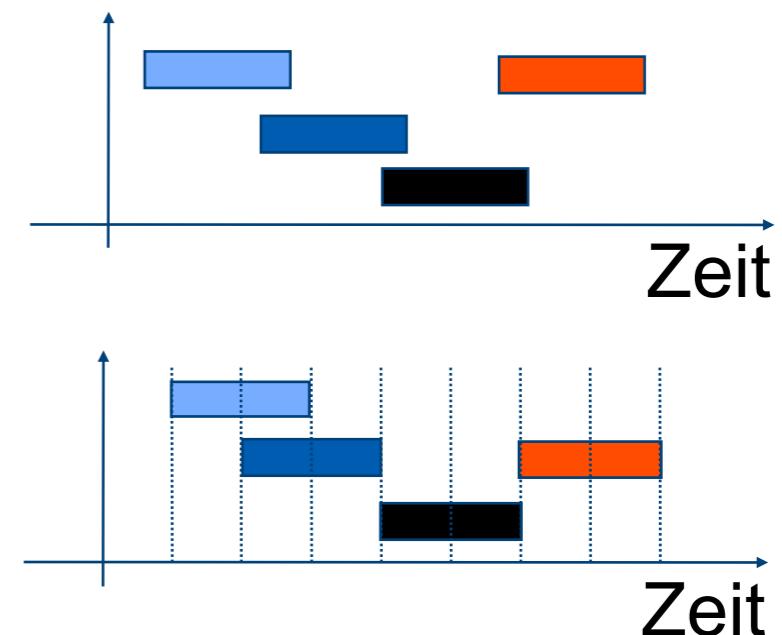
■ Zeitmodelle

- Kontinuierlich

- Übertragungen können jeder Zeit beginnen (keine zentrale Uhr)

- Diskret (Slotted time)

- Die Zeitachse ist in Abschnitte (slots) unterteilt
 - Übertragungen können nur an Abschnittsgrenzen starten
 - Slots können leer (idle), erfolgreich (mit Übertragung) sein oder eine Kollision beinhalten



■ Träger-Messung (Carrier Sensing)

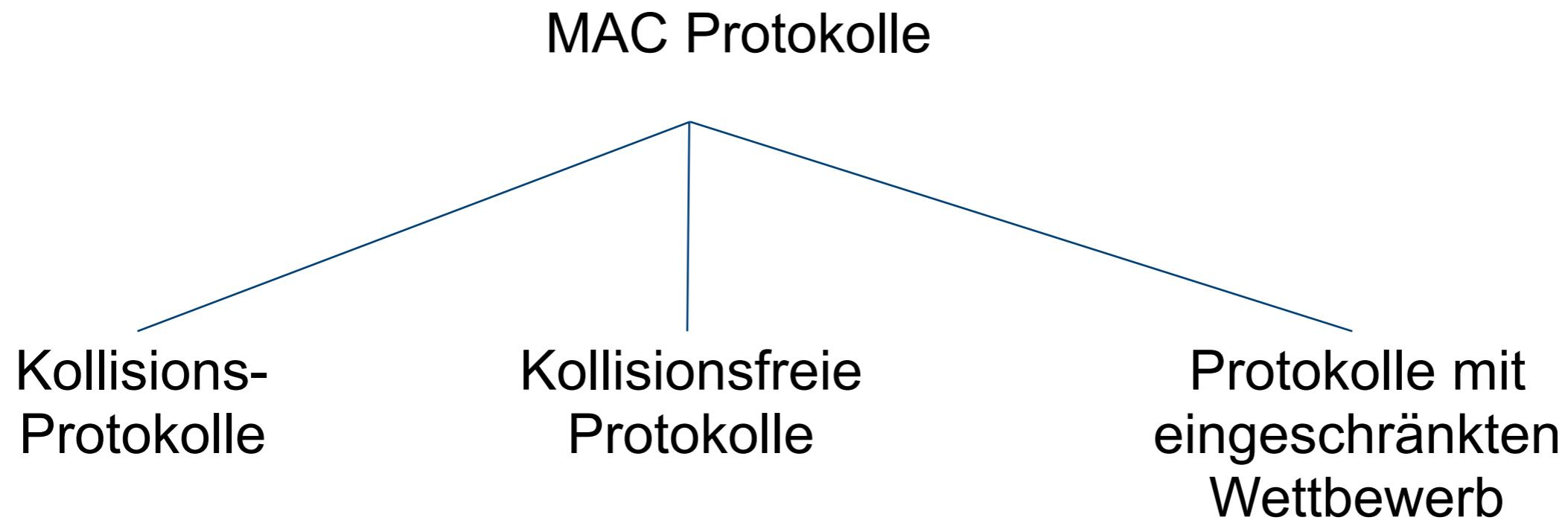
- Stationen können erkennen ob der Kanal momentan von anderen Stationen verwendet wird
 - Nicht notwendigerweise zuverlässig

Bewertung des Verhaltens

- Methoden zur Bewertung der Effizienz einer Kanalzuweisung
- Durchsatz (throughput)
 - Anzahl Pakete pro Zeiteinheit
 - Besonders bei großer Last wichtig
- Verzögerung (delay)
 - Zeit für den Transport eines Pakets
 - Muss bei geringer Last gut sein
- Gerechtigkeit (fairness)
 - Gleichbehandlung aller Stationen
 - Fairer Anteil am Durchsatz und bei Delay

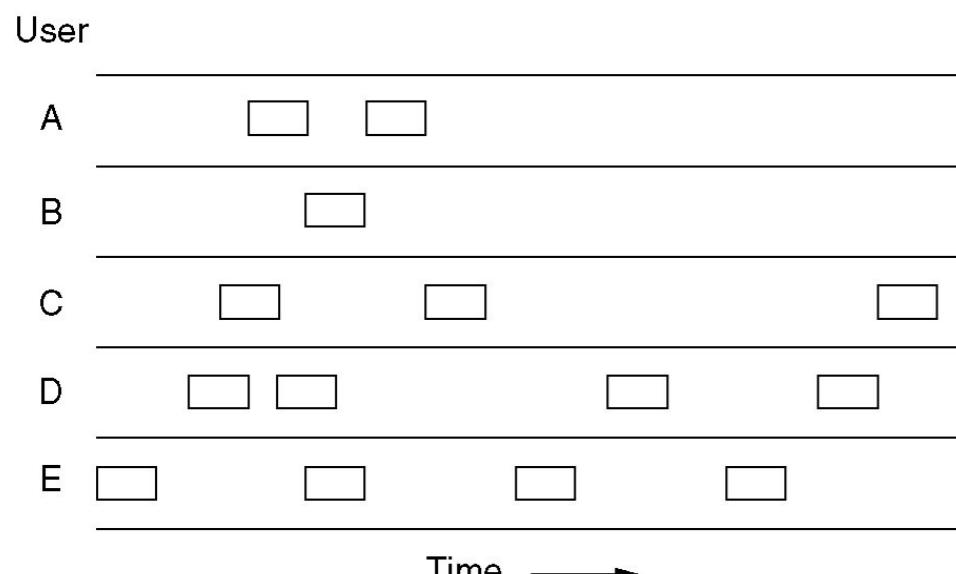
Mögliche MAC-Protokolle

- Unterscheidung: Erlaubt das Protokoll Kollisionen?
 - Als Systementscheidung
 - Die unbedingte Kollisionsvermeidung kann zu Effizienzeinbußen führen

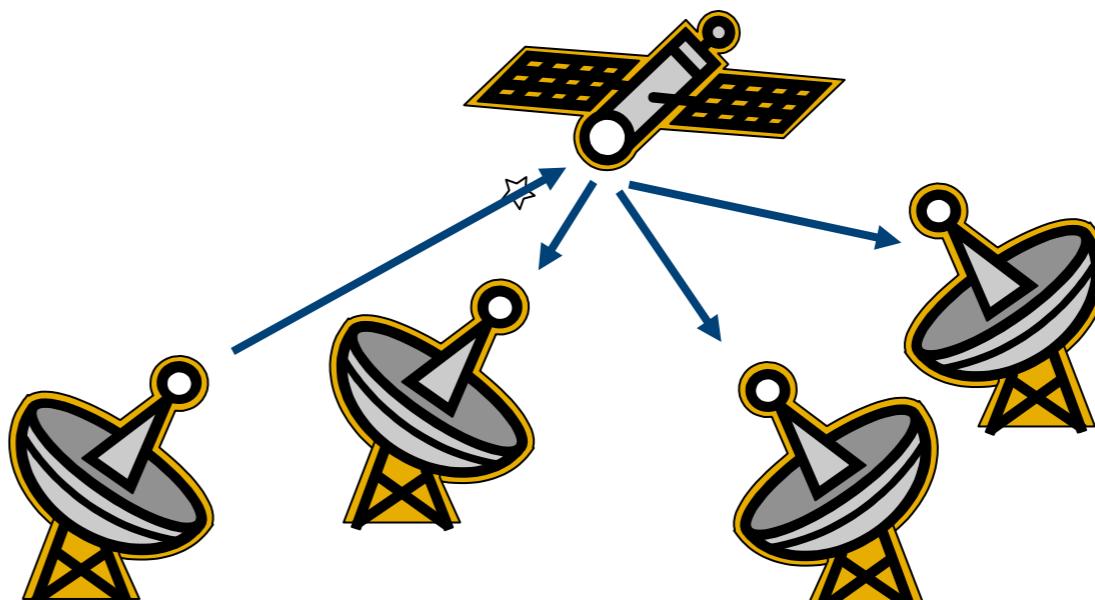


System mit Kollisionen: **Contention System**

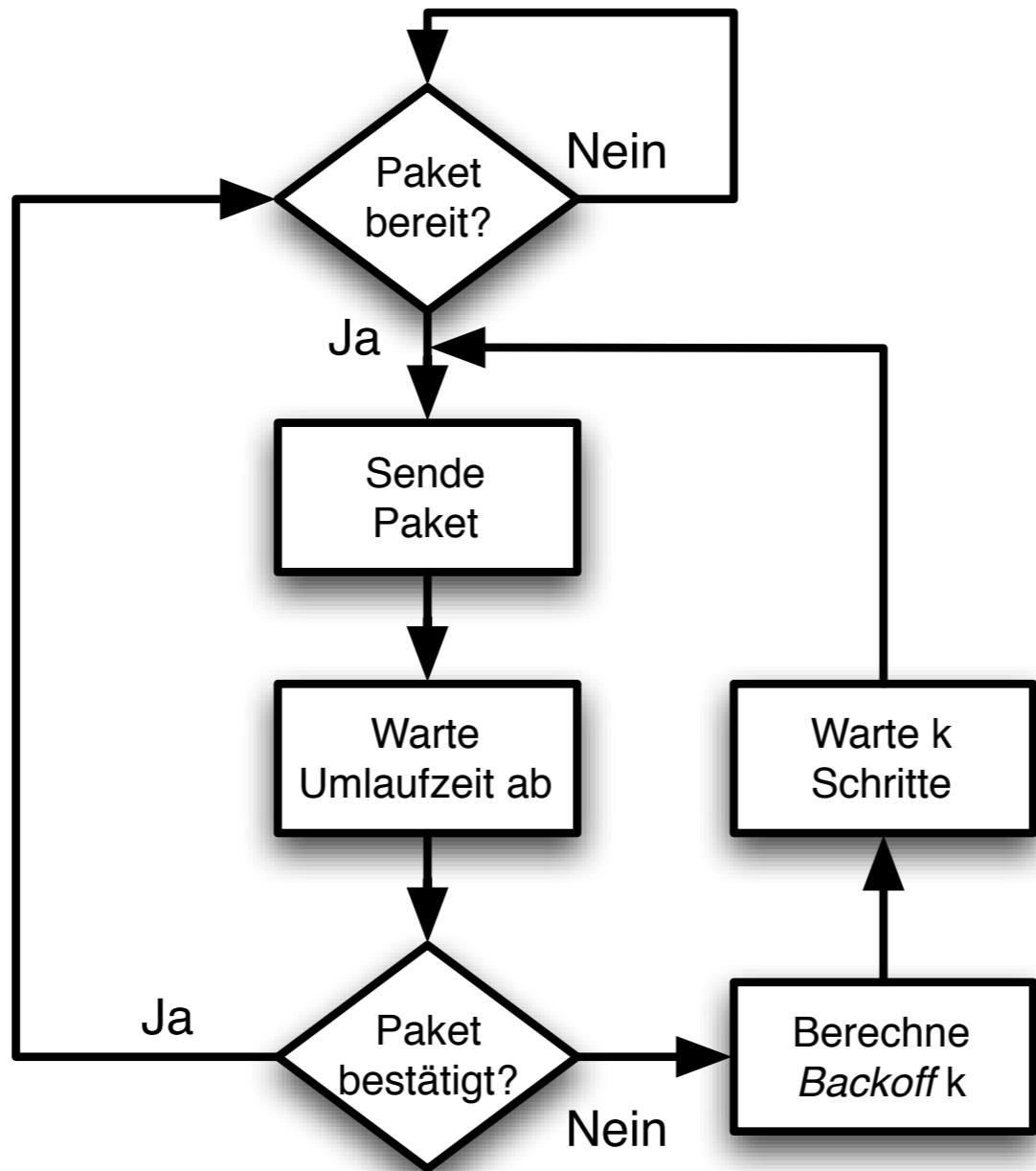
- Algorithmus
 - Sobald ein Paket vorhanden ist, wird es gesendet
- Ursprung
 - 1970 by Abramson et al., University of Hawaii
 - Ziel: Funkverbindung zwischen den Instituten auf den Inseln



Pakete werden zu beliebigen Zeiten übertragen



ALOHA – Analyse



ALOHA – Analyse

- Vorteile
 - Einfach
 - Keine Koordination notwendig
- Nachteile
 - Kollisionen
 - Sender überprüft den Kanalzustand nicht
 - Sender hat keine direkte Methode den Sende-Erfolg zu erfahren
 - Bestätigungen sind notwendig
 - Diese können auch kollidieren

ALOHA – Effizienz

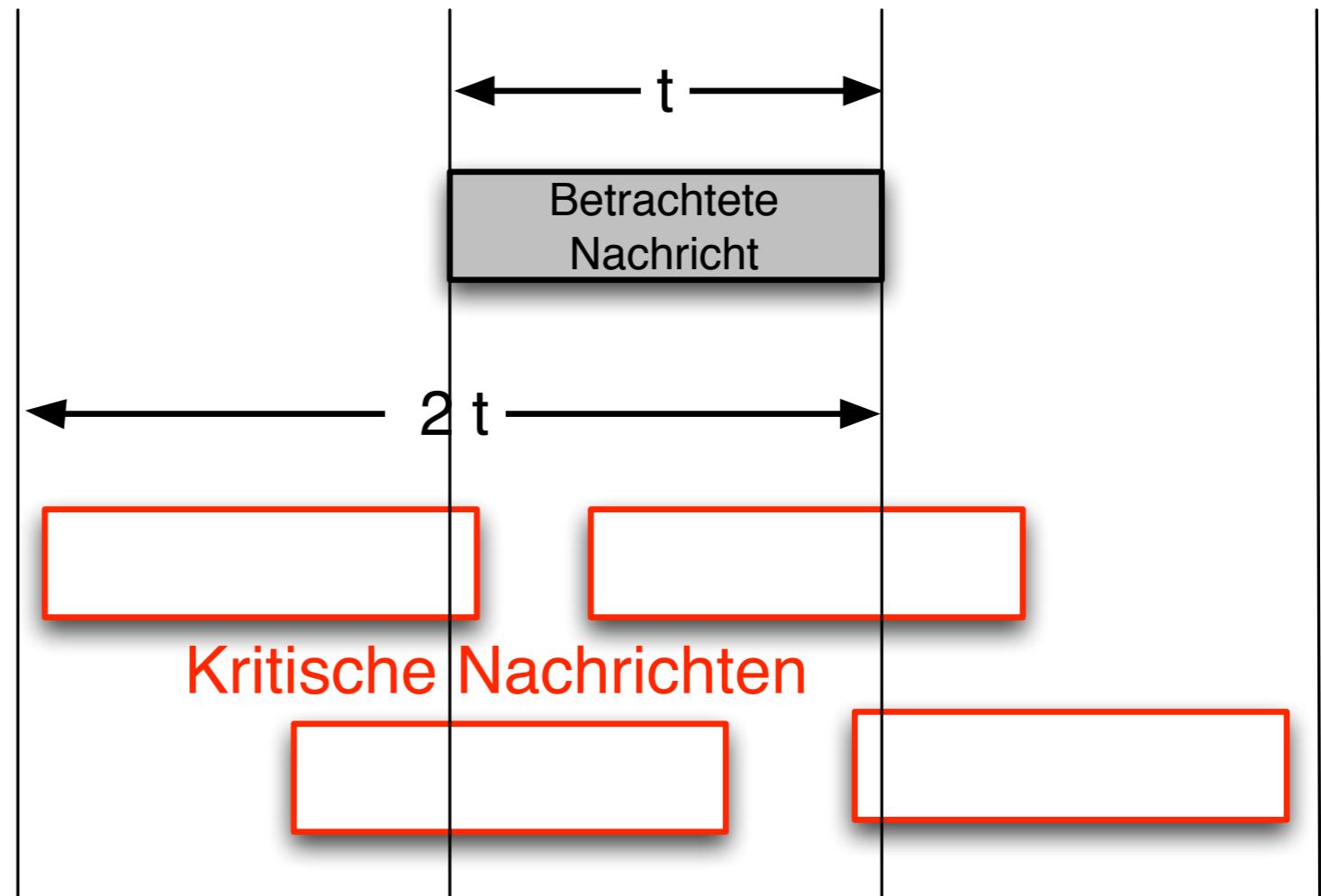
- Betrachte Poisson-Prozess zur Erzeugung von Paketen
 - Entsteht durch “unendlich” viele Stationen, die sich gleich verhalten
 - Zeit zwischen zwei Sende-Versuchen ist exponentiell verteilt
 - Sei G der Erwartungswert der Übertragungsversuche pro Paketlänge
 - Alle Pakete haben gleiche Länge
 - Dann gilt

$$P[k \text{ Versuche}] = \frac{G^k}{k!} e^{-G}$$

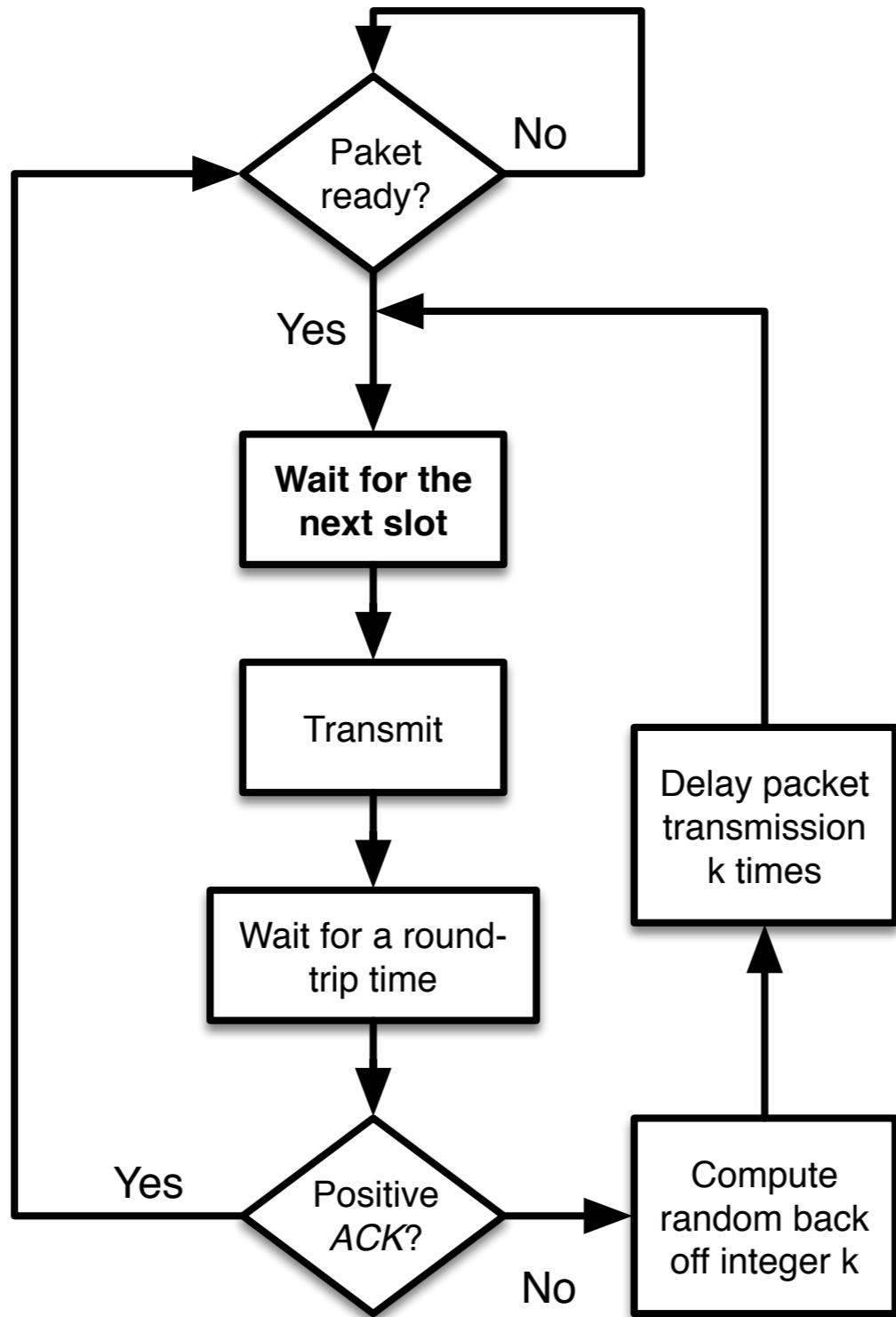
- Um eine erfolgreiche Übertragung zu erhalten, darf keine Kollision mit einem anderen Paket erfolgen
- Wie lautet die Wahrscheinlichkeit für eine solche Übertragung?

ALOHA – Effizienz

- Ein Paket X wird gestört, wenn
 - ein Paket kurz vor X startet
 - wenn ein Paket kurz vor dem Ende von X startet
- Das Paket wird erfolgreich übertragen, wenn in einem Zeitraum von zwei Paketen kein (anderes) Paket übertragen wird
- Durchsatz:
 - $S(G) = Ge^{-2G}$
 - Optimal für $G=1/2$, $S=1/e$



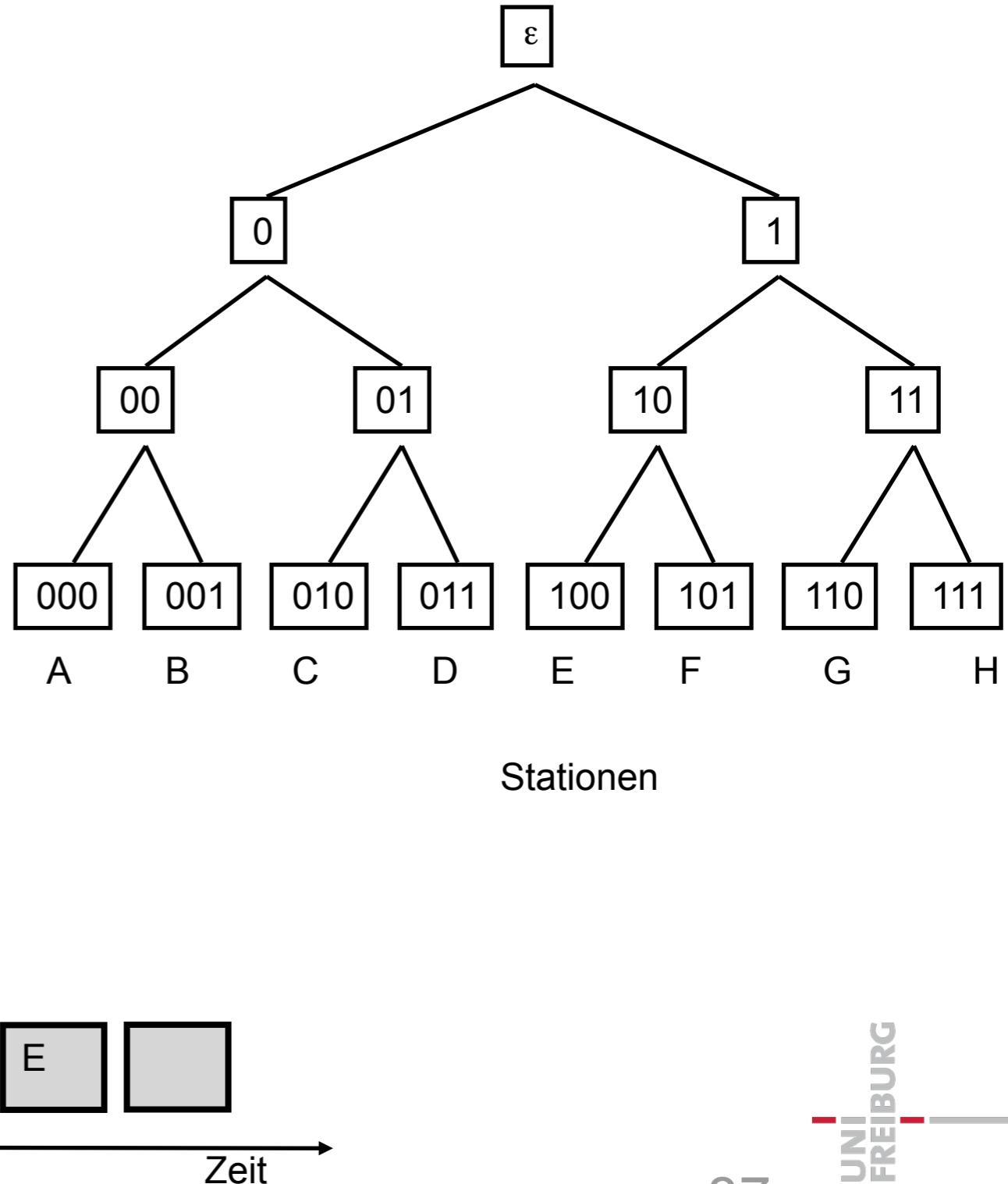
Slotted ALOHA



Adaptives Baumprotokoll

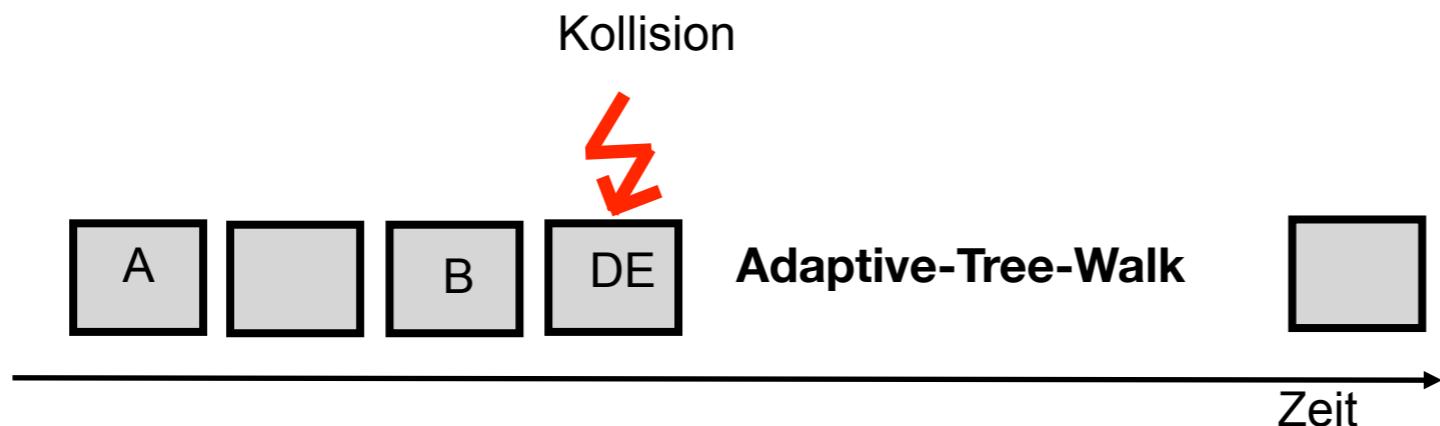
Voraussetzung

- Adaptives Baumprotokoll (adaptive tree walk)
- Ausgangspunkt:
 - Binäre, eindeutige Präsentation aller Knoten (ID)
 - Dargestellt in einem Baum
 - Synchronisiertes Protokoll
 - Drei Typen können unterschieden werden:
 - Keine Station sendet
 - Genau eine Station sendet
 - Kollision: mindestens zwei Stationen senden



■ Basis-Algorithmus

- Jeder Algorithmus sendet sofort (slotted Aloha)
- Falls eine Kollision auftritt,
 - akzeptiert keine Station mehr neue Paket aus der Vermittlungsschicht
 - Führe Adaptive-Tree-Walk(ε) aus



Adaptives Baumprotokoll

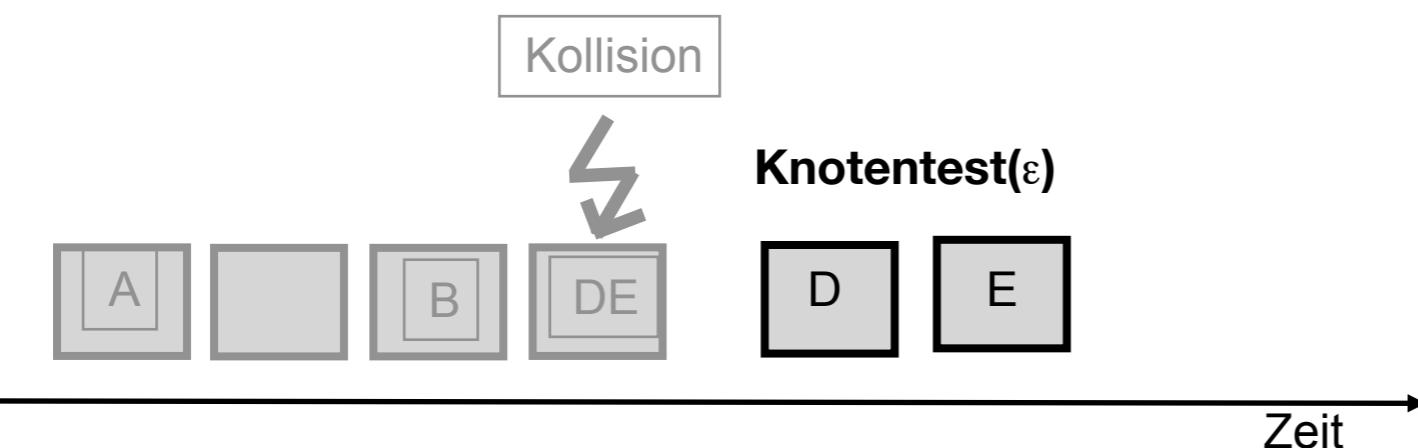
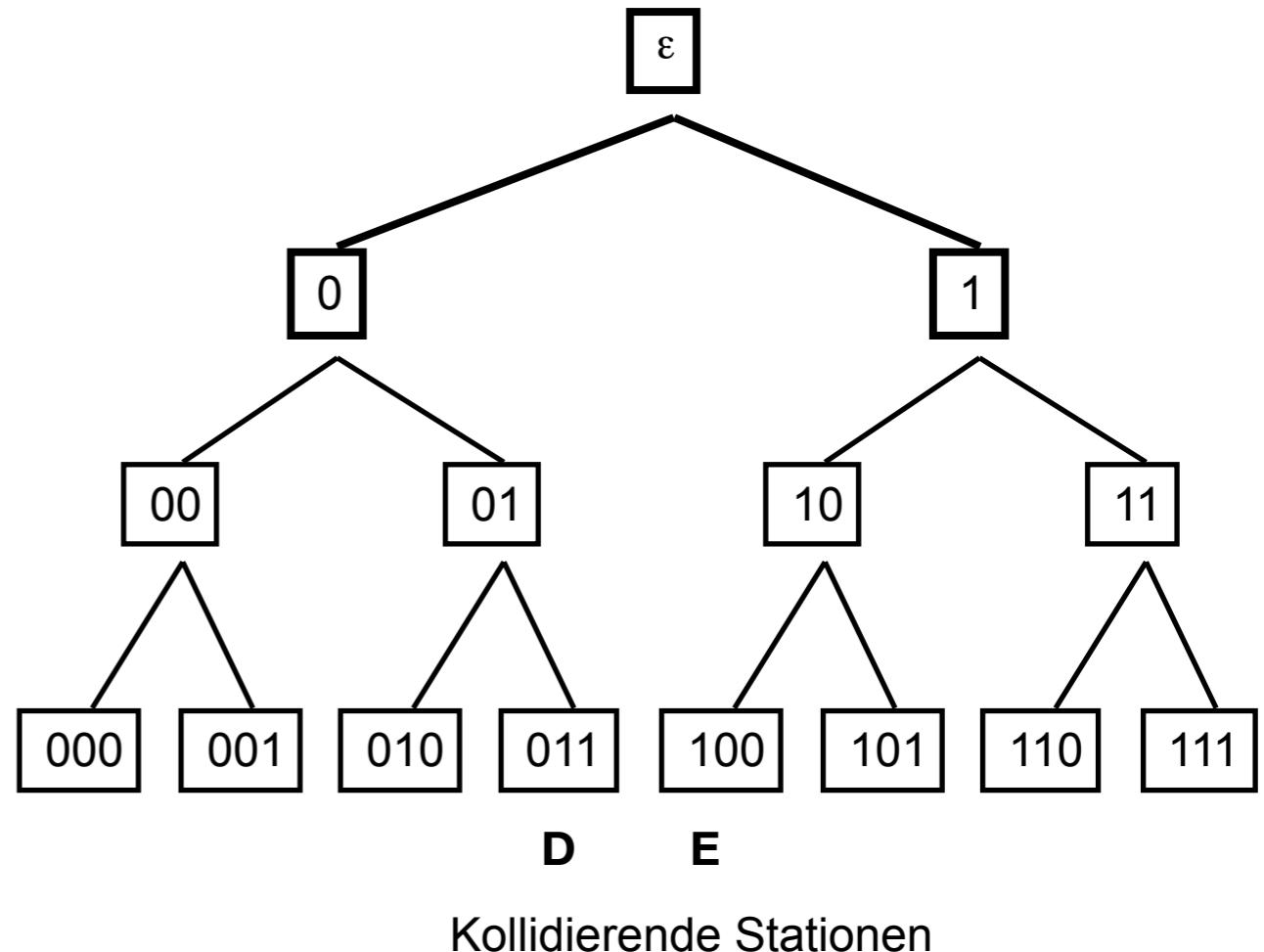
Knoten-Test

Algorithmus Knoten-Test

- für Knoten u des Baums und
- kollidierende Menge S von Stationen

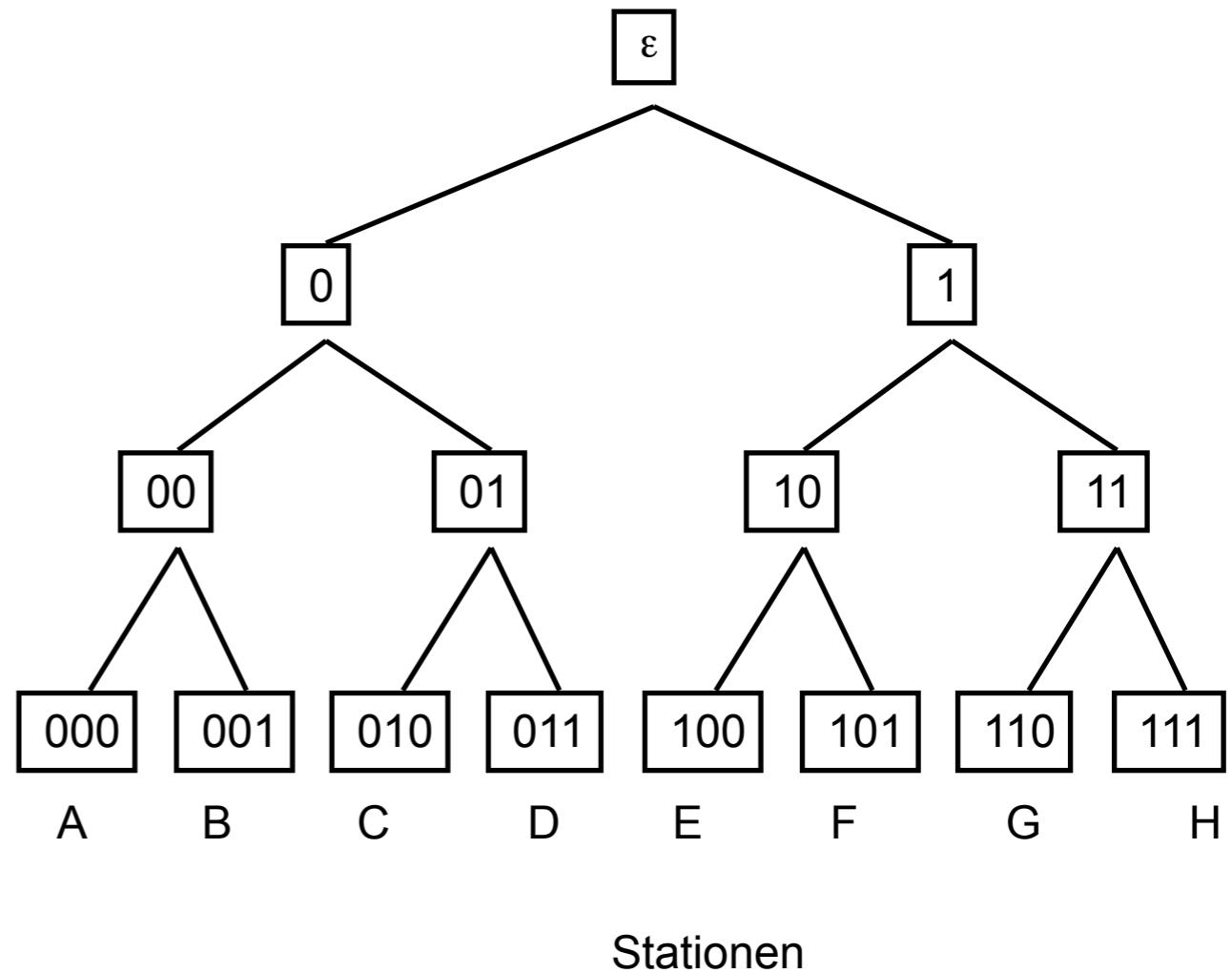
Knoten-Test(u)

- Betrachte zwei Slots pro Knoten des Baums
- Im ersten Slot senden alle Knoten aus S , die mit ID u_0 anfangen
- Im zweiten Slot senden alle Knoten aus S , die mit ID u_1 anfangen



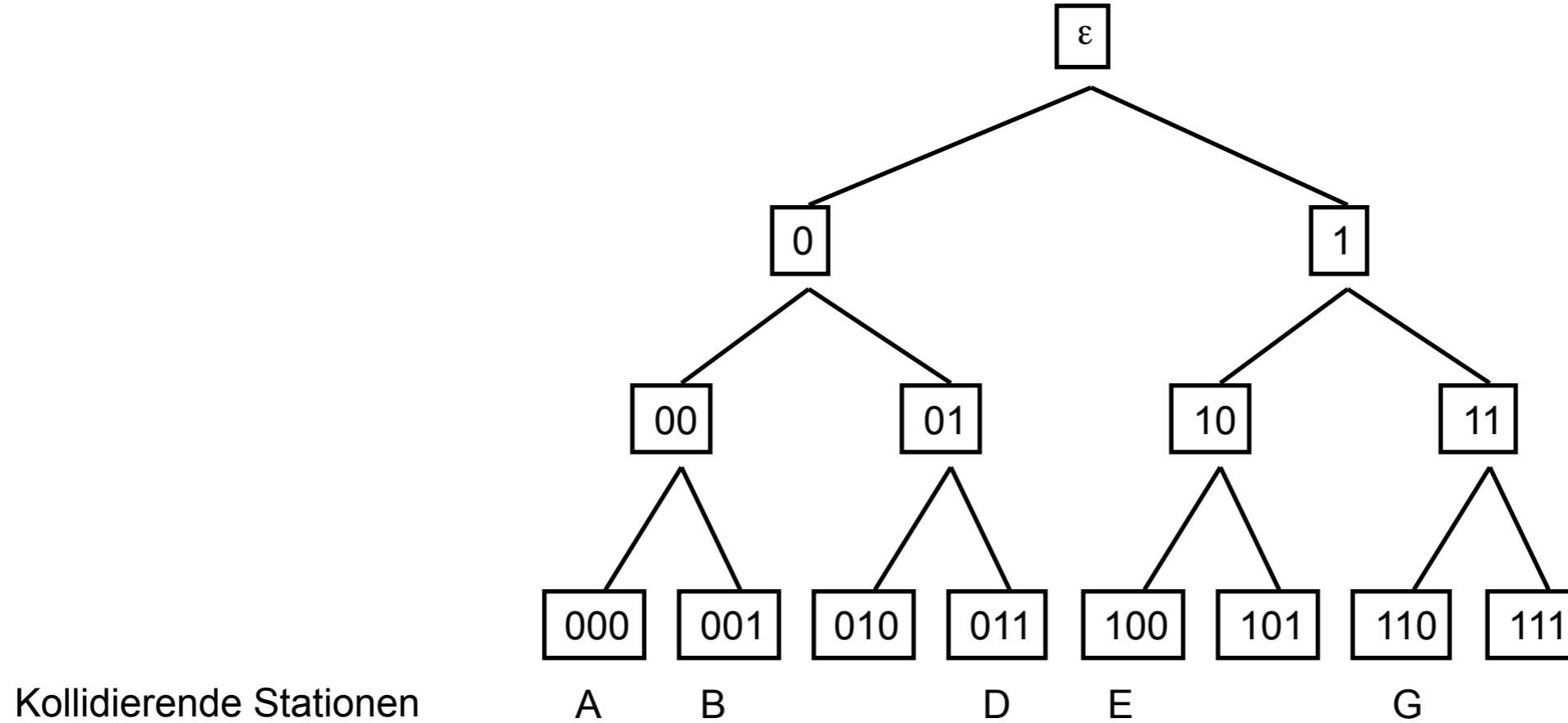
Adaptives Baumprotokoll Kern-Algorithmus

- Algorithmus Knoten-Test
 - für Knoten u des Baums und
 - kollidierende Menge S von Stationen
- Knoten-Test(u)
 - Betrachte zwei Slots pro Knoten des Baums
 - Im ersten Slot senden alle Knoten aus S , die mit ID u_0 anfangen
 - Im zweiten Slot senden alle Knoten aus S , die mit ID u_1 anfangen
- Adaptive Tree Walk(x)
 - Führe Knoten-Test(x) aus
 - Falls Kollision im ersten Slot,
 - führe Adaptive-Tree-Walk(x_0) aus
 - Falls Kollision im zweiten Slot,
 - Führe Adaptive-Tree-Walk(x_1) aus



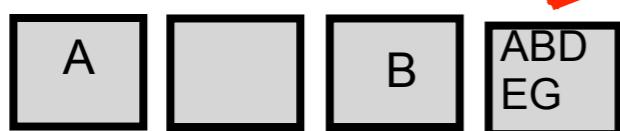
Adaptives Baumprotokoll

Beispiel (1)



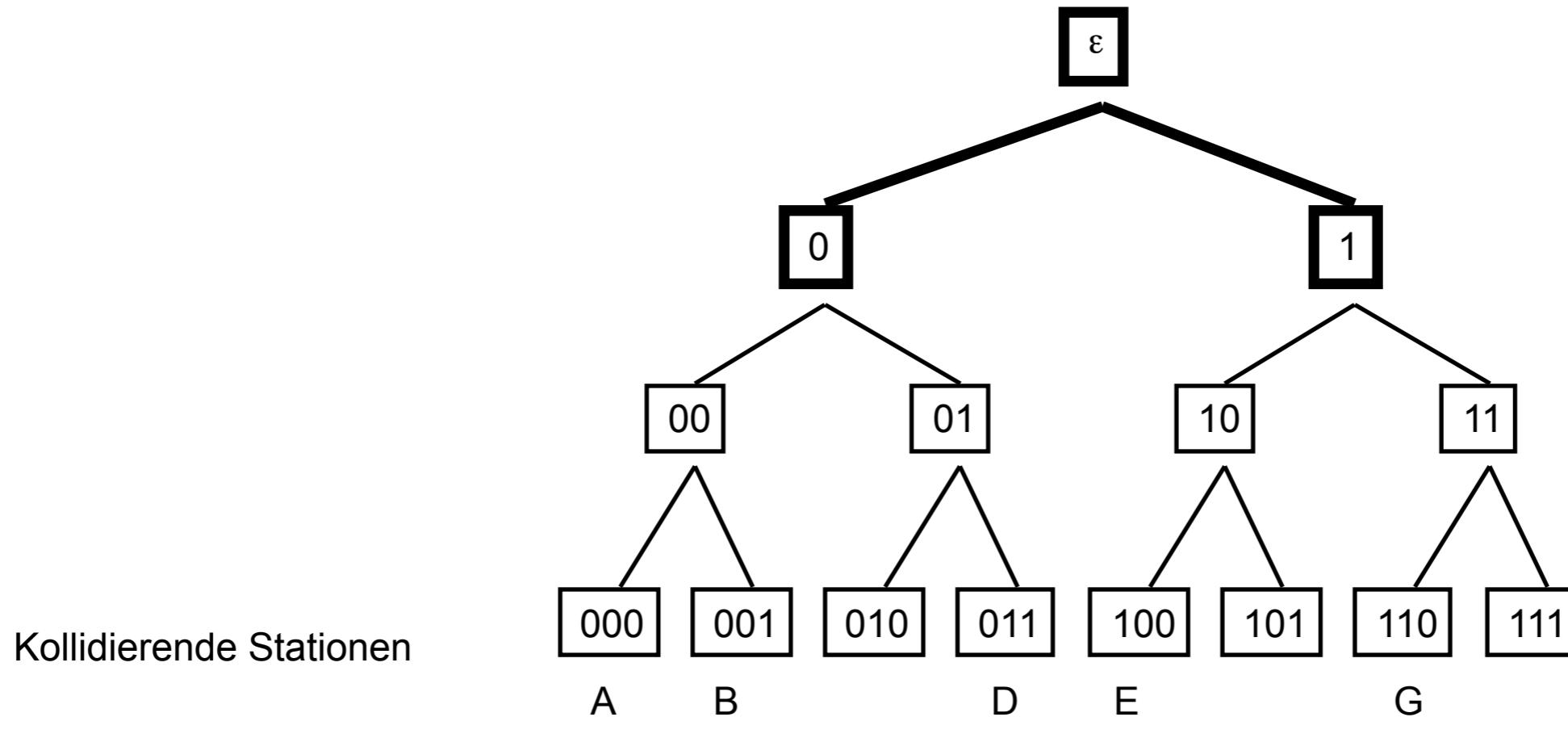
Kollision

Adaptive-Tree-Walk



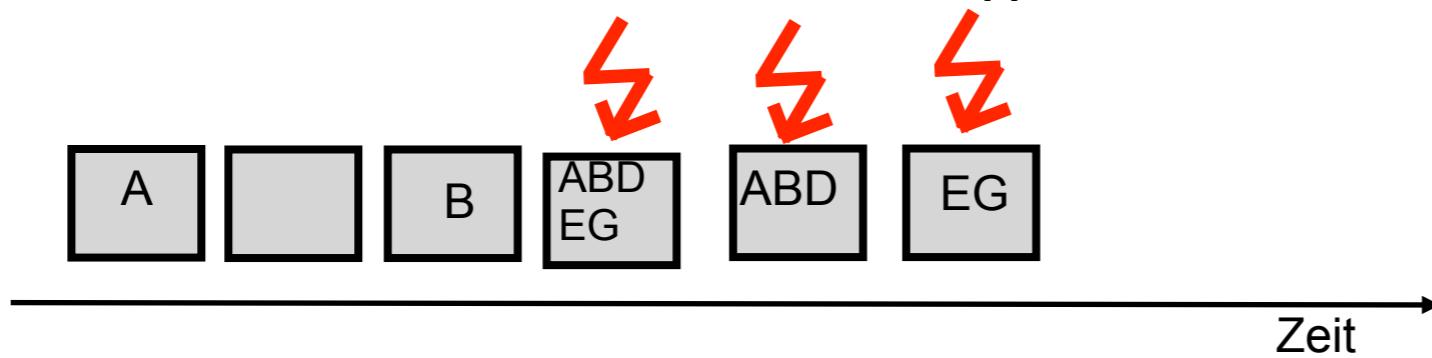
Zeit

Adaptives Baumprotokoll Beispiel (2)

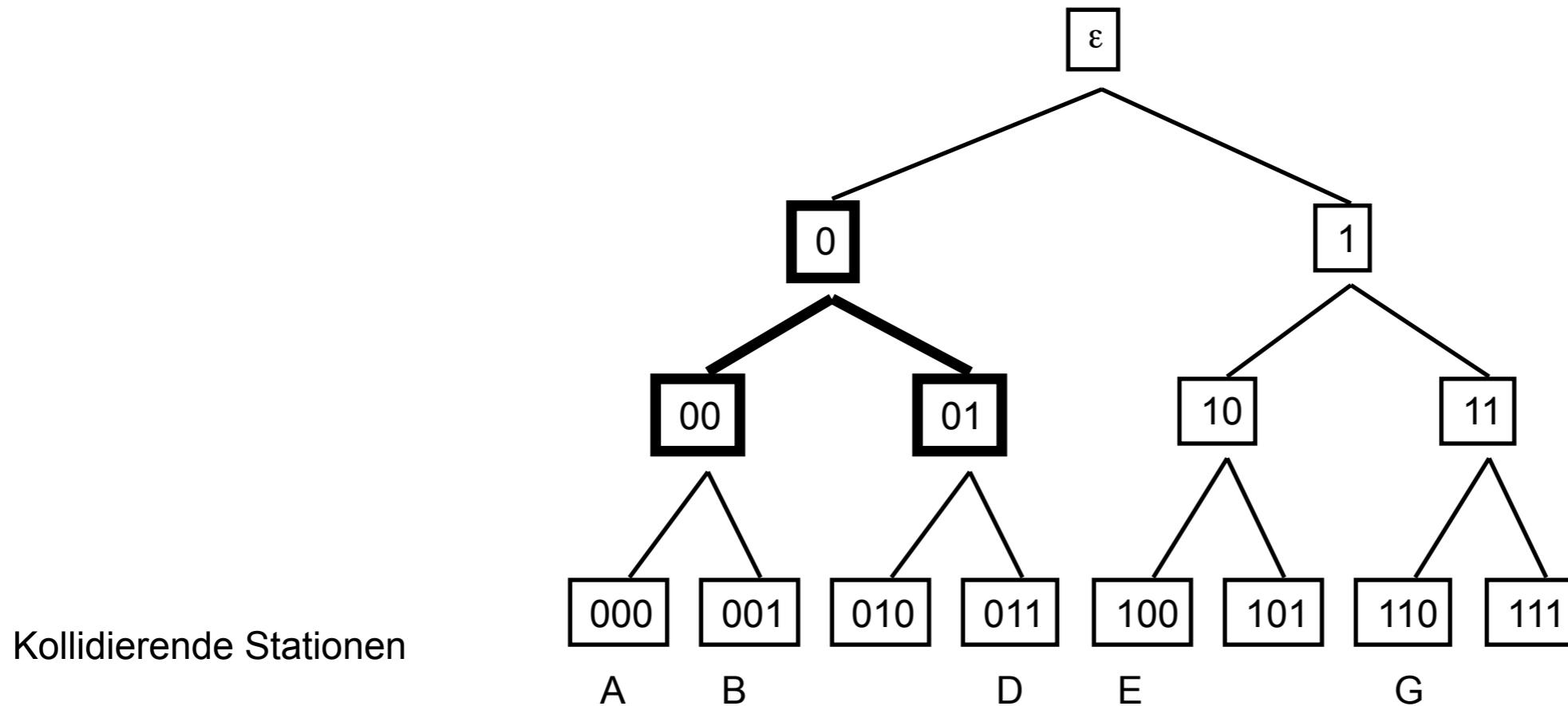


Adaptive-Tree-Walk

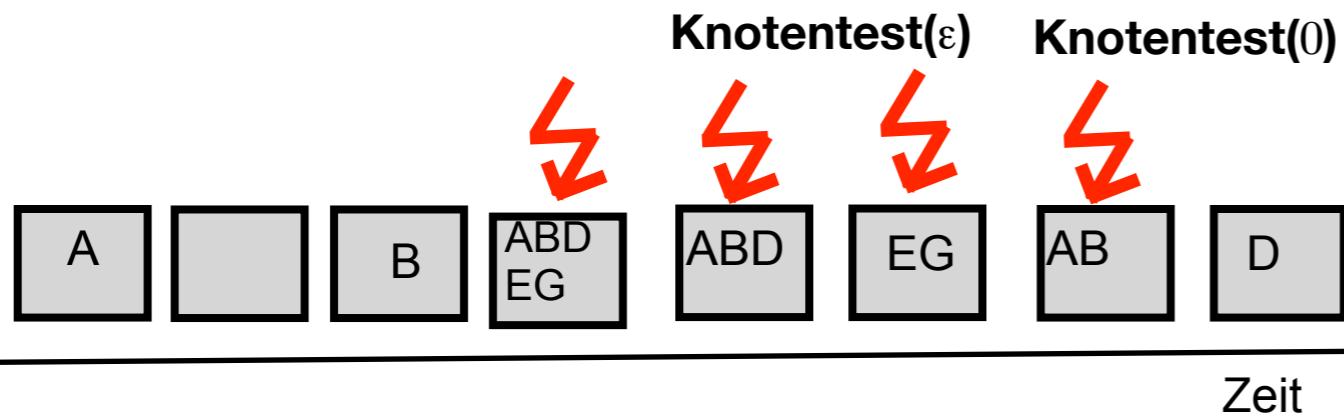
Knotentest(ϵ)



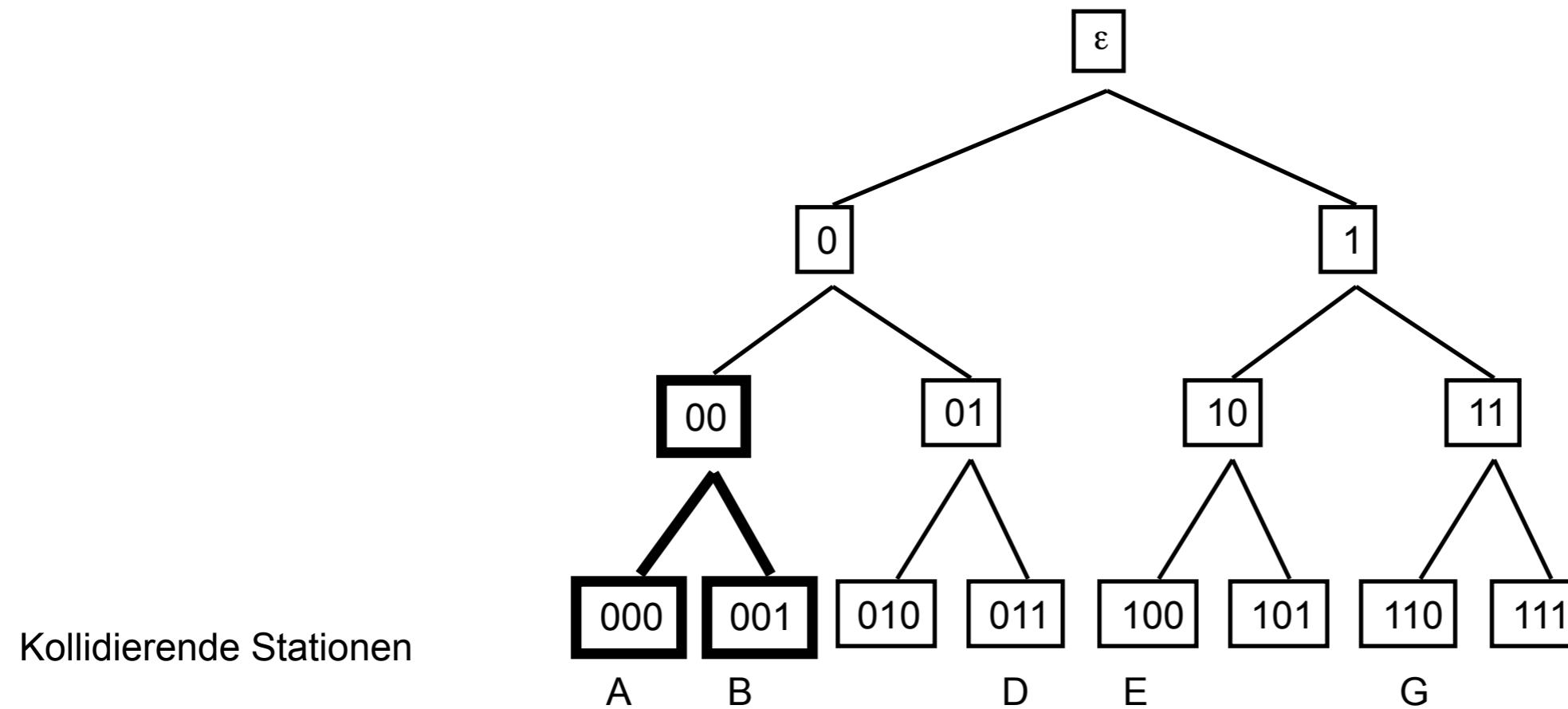
Adaptives Baumprotokoll Beispiel (3)



Adaptive-Tree-Walk

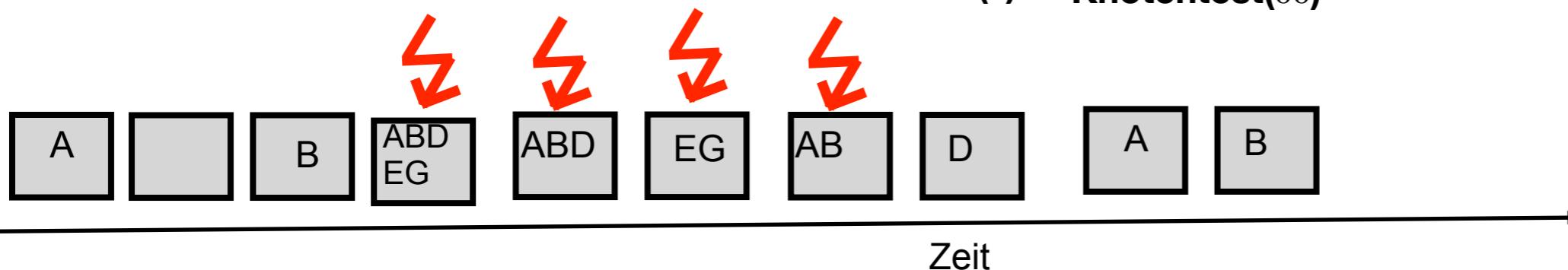


Adaptives Baumprotokoll Beispiel (4)

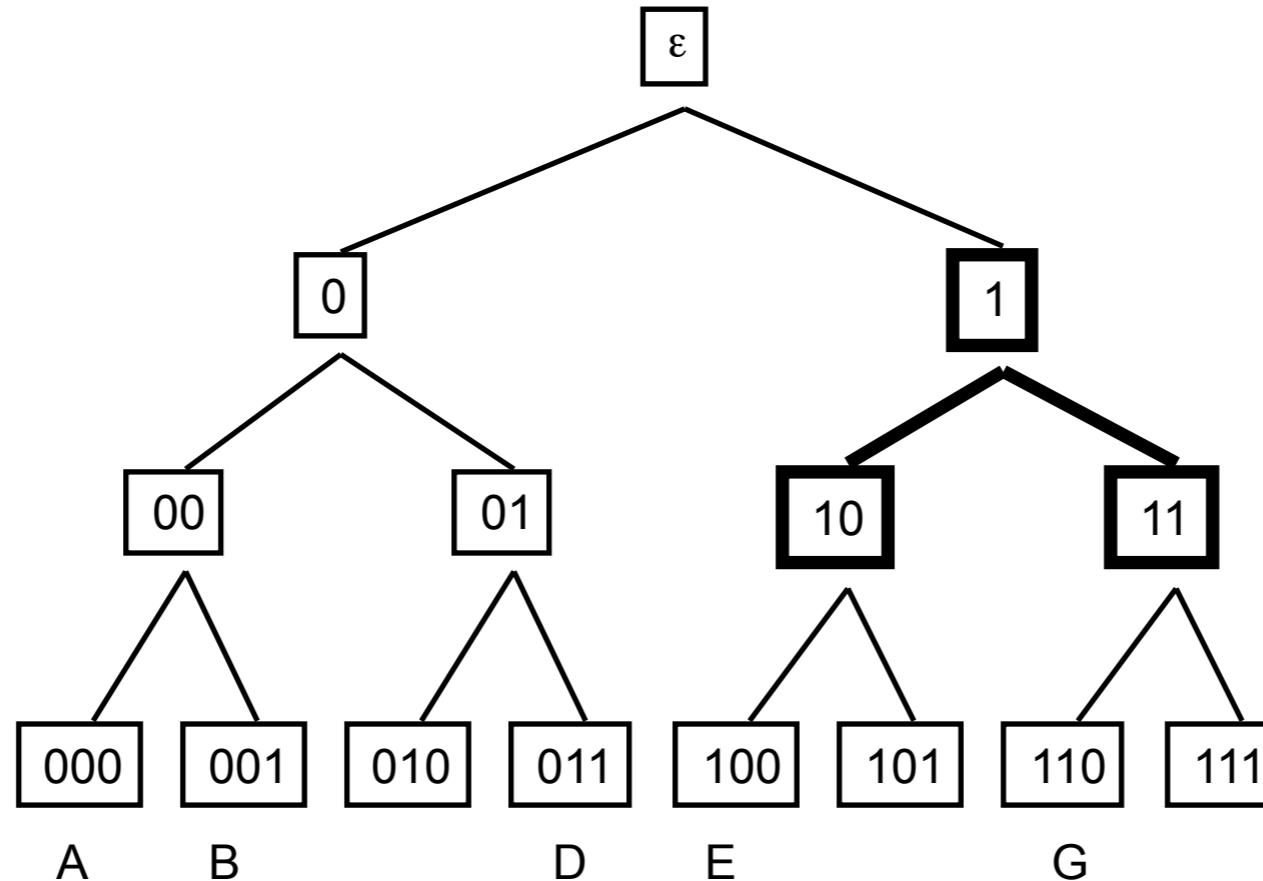


Adaptive-Tree-Walk

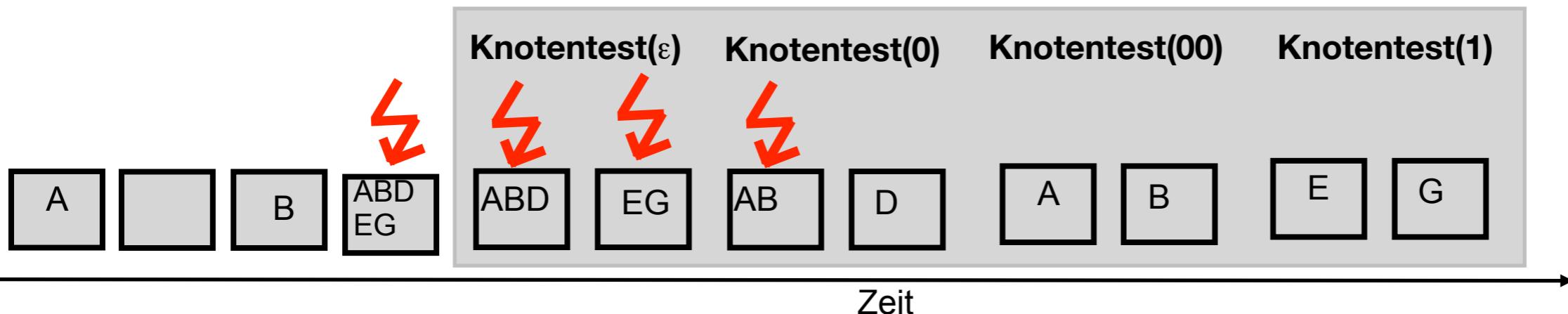
Knotentest(ε) Knotentest(0) Knotentest(00)



Adaptives Baumprotokoll Beispiel (5)



Adaptive-Tree-Walk



Systeme II

3. Die Datensicherungsschicht

Christian Schindelhauer

Technische Fakultät

Rechnernetze und Telematik

Albert-Ludwigs-Universität Freiburg

Systeme II

4. Die Vermittlungsschicht

Christian Schindelhauer

Technische Fakultät

Rechnernetze und Telematik

Albert-Ludwigs-Universität Freiburg

Version 13.06.2017

Circuit Switching oder Packet Switching

■ Circuit Switching

- Etablierung einer Verbindung zwischen lokalen Benutzern durch Schaltstellen
 - mit expliziter Zuordnung von realen Schaltkreisen
 - oder expliziter Zuordnung von virtuellen Ressourcen, z.B. Slots
- Quality of Service einfach, außer bei
 - Leitungsaufbau
 - Leitungsdauer
- Problem
 - Statische Zuordnung
 - Ineffiziente Ausnutzung des Kommunikationsmedium bei dynamischer Last
- Anwendung
 - Telefon
 - Telegraf
 - Funkverbindung

Circuit Switching oder Packet Switching

■ Packet Switching

- Grundprinzip von IP
 - Daten werden in Pakete aufgeteilt und mit Absender/Ziel-Information unabhängig versandt
- Problem: Quality of Service
 - Die Qualität der Verbindung hängt von einzelnen Paketen ab
 - Entweder Zwischenspeichern oder Paketverlust
- Vorteil:
 - Effiziente Ausnutzung des Mediums bei dynamischer Last

■ Resümee

- Packet Switching hat Circuit Switching in praktisch allen Anwendungen abgelöst
- Grund:
 - Effiziente Ausnutzung des Mediums

Taktik der Schichten

■ Transport

- muss gewisse Flusskontrolle gewährleisten
- z.B. Fairness zwischen gleichzeitigen Datenströmen

■ Vermittlung

- Quality of Service (virtuelles Circuit Switching)

■ Sicherung

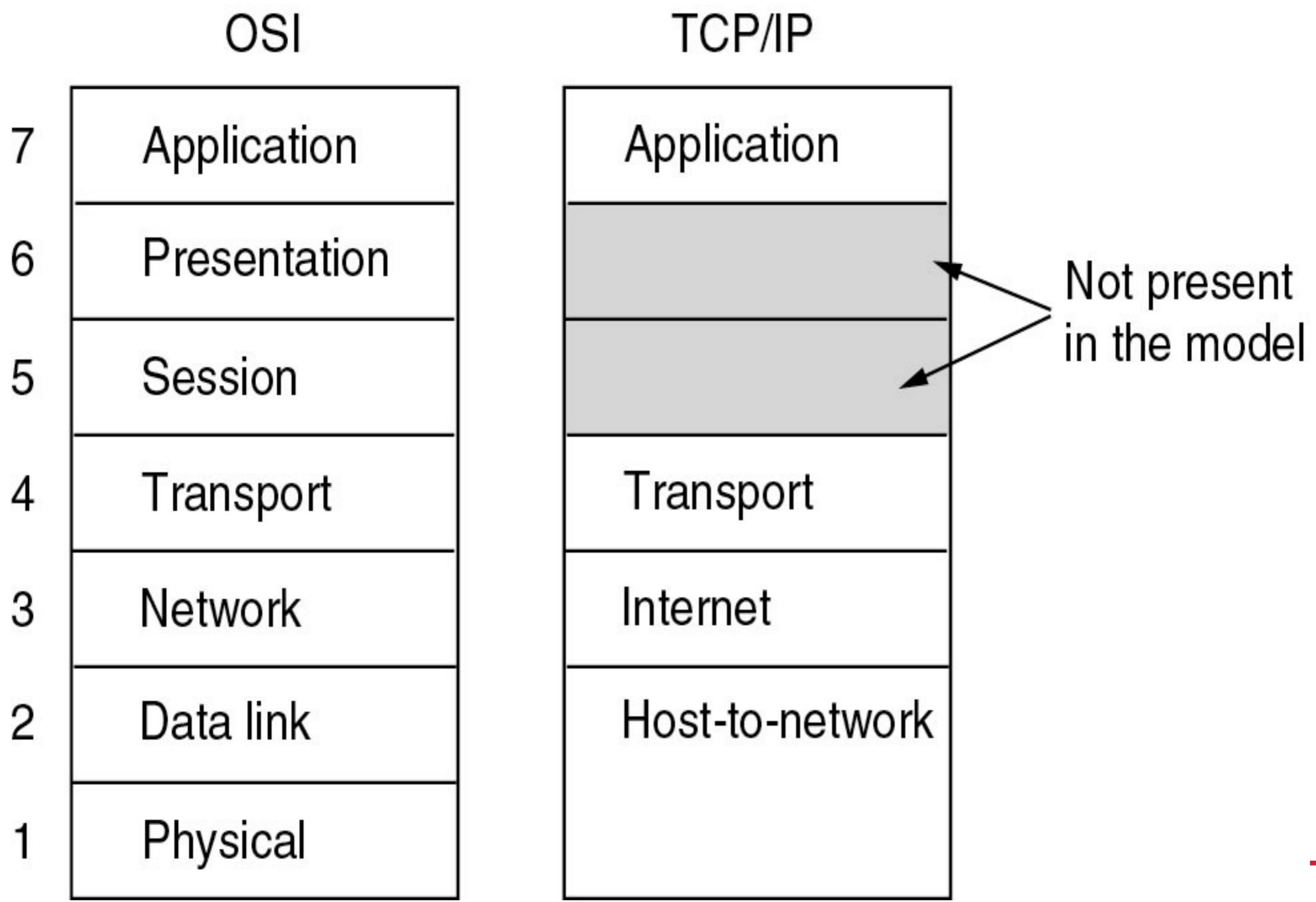
- Flusskontrolle zur Auslastung des Kanals

Layer	Policies
Transport	<ul style="list-style-type: none">• Retransmission policy• Out-of-order caching policy• Acknowledgement policy• Flow control policy• Timeout determination
Network	<ul style="list-style-type: none">• Virtual circuits versus datagram inside the subnet• Packet queueing and service policy• Packet discard policy• Routing algorithm• Packet lifetime management
Data link	<ul style="list-style-type: none">• Retransmission policy• Out-of-order caching policy• Acknowledgement policy• Flow control policy

Die Schichtung des Internets - TCP/IP-Layer

Anwendung	Application	Telnet, FTP, HTTP, SMTP (E-Mail), ...
Transport	Transport	TCP (Transmission Control Protocol) UDP (User Datagram Protocol)
Vermittlung	Network	IP (Internet Protocol) + ICMP (Internet Control Message Protocol) + IGMP (Internet Group Management Protocol)
Verbindung	Host-to-network	LAN (z.B. Ethernet, Token Ring etc.)

OSI versus TCP/IP



Warum eine Vermittlungsschicht

- Lokale Netzwerke können nicht nur über Hubs, Switches oder Bridges verknüpft werden
 - Hubs: Kollisionen nehmen überhand
 - Switches:
 - Routen-Information durch Beobachtung der Daten ineffizient
 - Broadcast aller Nachrichten schafft Probleme
 - Es gibt über 100 Mio. lokale Netzwerke im Internet...
- Zur Beförderung von Paketen in großen Netzwerken braucht man Routeninformationen
 - Wie baut man diese auf?
 - Wie leitet man Pakete weiter?
- Das Internet-Protokoll ist im wesentlichen ein Vermittlungsschichtprotokoll

Routing-Tabelle und Paket-Weiterleitung

■ IP-Routing-Tabelle

- enthält für Ziel (Destination) die Adresse des nächsten Rechners (Gateway)
- Destination kann einen Rechner oder ganze Sub-nets beschreiben
- Zusätzlich wird ein Default-Gateway angegeben

■ Packet Forwarding

- früher Packet Routing genannt
- IP-Paket (datagram) enthält Start-IP-Adresse und Ziel-IP-Adresse
 - Ist Ziel-IP-Adresse = eigene Rechneradresse dann Nachricht ausgeliefert
 - Ist Ziel-IP-Adresse in Routing-Tabelle dann leite Paket zum angegeben Gateway
 - Ist Ziel-IP-Subnetz in Routing-Tabelle dann leite Paket zum angegeben Gateway
 - Ansonsten leite zum Default-Gateway

Paket-Weiterleitung im Internet Protokoll

- IP-Paket (datagram) enthält unter anderen
 - TTL (Time-to-Live): Anzahl der Hops
 - Start-IP-Adresse
 - Ziel-IP-Adresse
- Behandlung eines Pakets
 - Verringere TTL (Time to Live) um 1
 - Falls TTL $\neq 0$ dann Packet-Forwarding aufgrund der Routing-Tabelle
 - Falls TTL = 0 oder bei Problemen in Packet-Forwarding:
 - Lösche Paket
 - Falls Paket ist kein ICMP-Paket dann
 - Sende ICMP-Paket mit
 - Start= aktuelle IP-Adresse und
 - Ziel = alte Start-IP-Adresse

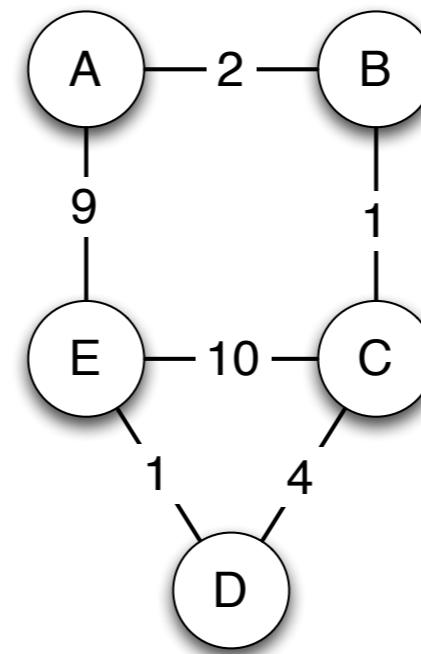
Statisches und Dynamisches Routing

- **Forwarding:**
 - Weiterleiten von Paketen
- **Routing:**
 - Erstellen Routen, d.h.
 - Erstellen der Routing-Tabelle
- **Statisches Routing**
 - Tabelle wird manuell erstellt
 - sinnvoll für kleine und stabile LANs
- **Dynamisches Routing**
 - Tabellen werden durch Routing-Algorithmus erstellt
 - Zentraler Algorithmus, z.B. Link State
 - Einer/jeder kennt alle Information, muss diese erfahren
 - Dezentraler Algorithmus, z.B. Distance Vector
 - arbeitet lokal in jedem Router
 - verbreitet lokale Information im Netzwerk

Distance Vector Routing Protocol

■ Distance Table Datenstruktur

- Jeder Knoten besitzt eine
 - Zeile für jedes mögliches Ziel
 - Spalte für jeden direkten Nachbarn



Distance Table für A

von A	über		Routing Tabellen Eintrag
	B	E	
nach B	2	15	B
C	3	14	B
D	7	10	B
E	8	9	E

■ Verteilter Algorithmus

- Jeder Knoten kommuniziert nur mit seinem Nachbarn

■ Asynchroner Betrieb

- Knoten müssen nicht Informationen austauschen in einer Runde

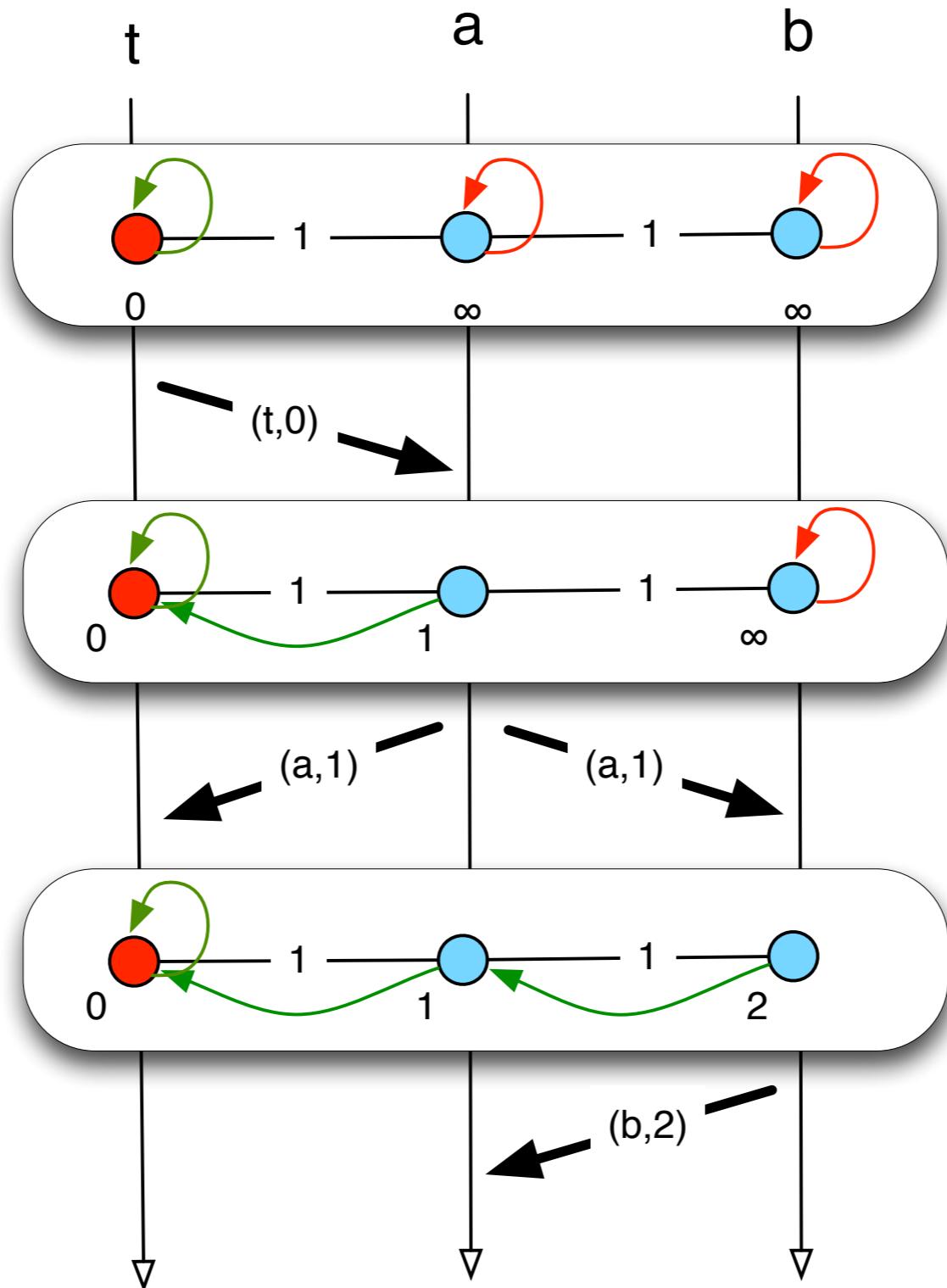
■ Selbst Terminierend

- läuft bis die Knoten keine Informationen mehr austauschen

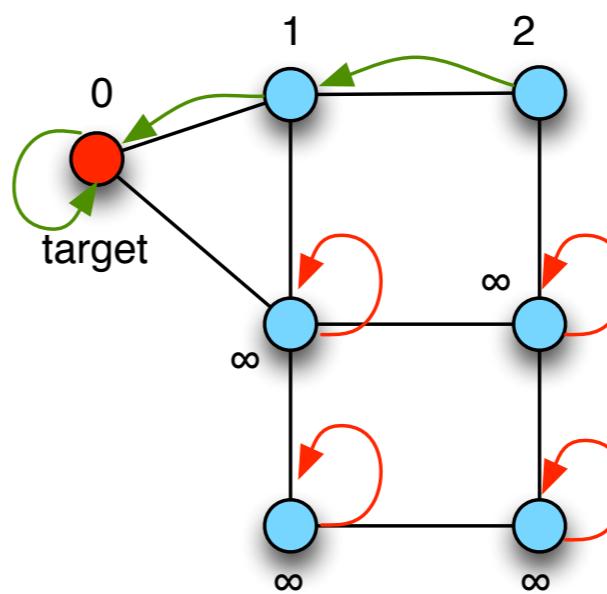
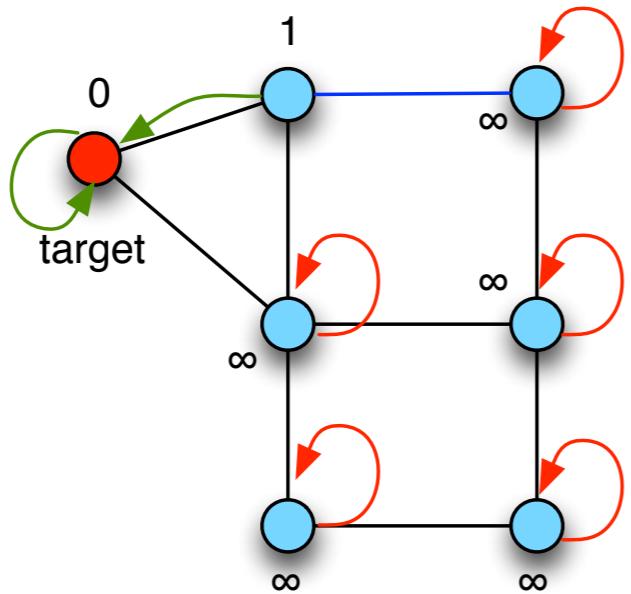
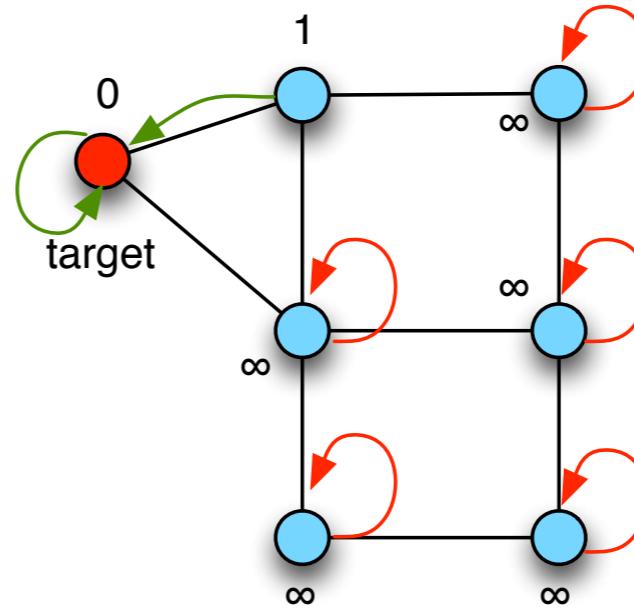
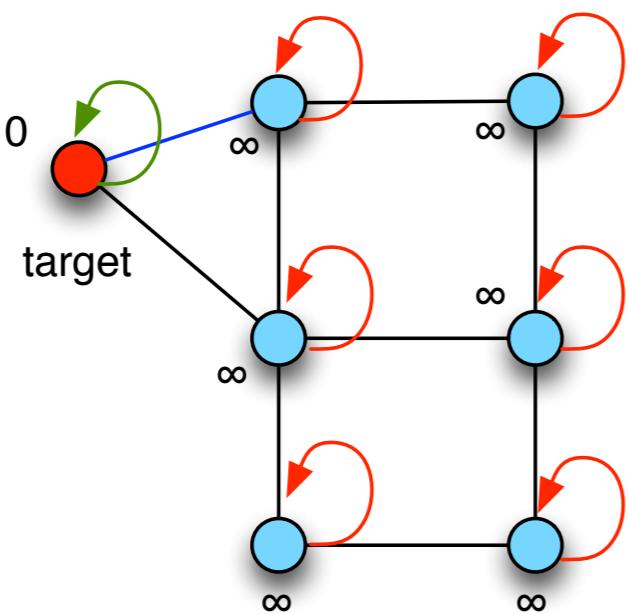
Distance Table für C

von C	über			Routing Tabellen Eintrag
	B	D	E	
nach A	3	11	18	B
B	1	9	16	B
D	6	4	11	D
E	7	5	10	D

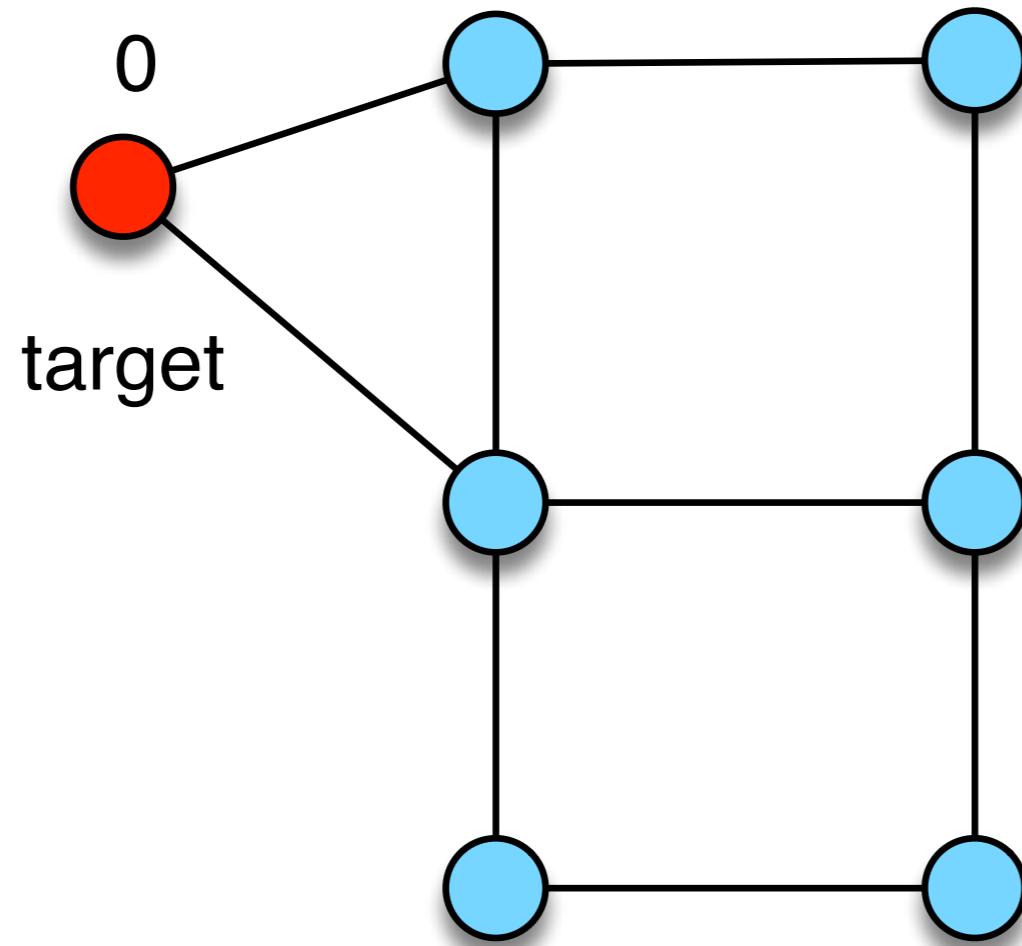
Beispiel für Distance-Vector für Ziel t



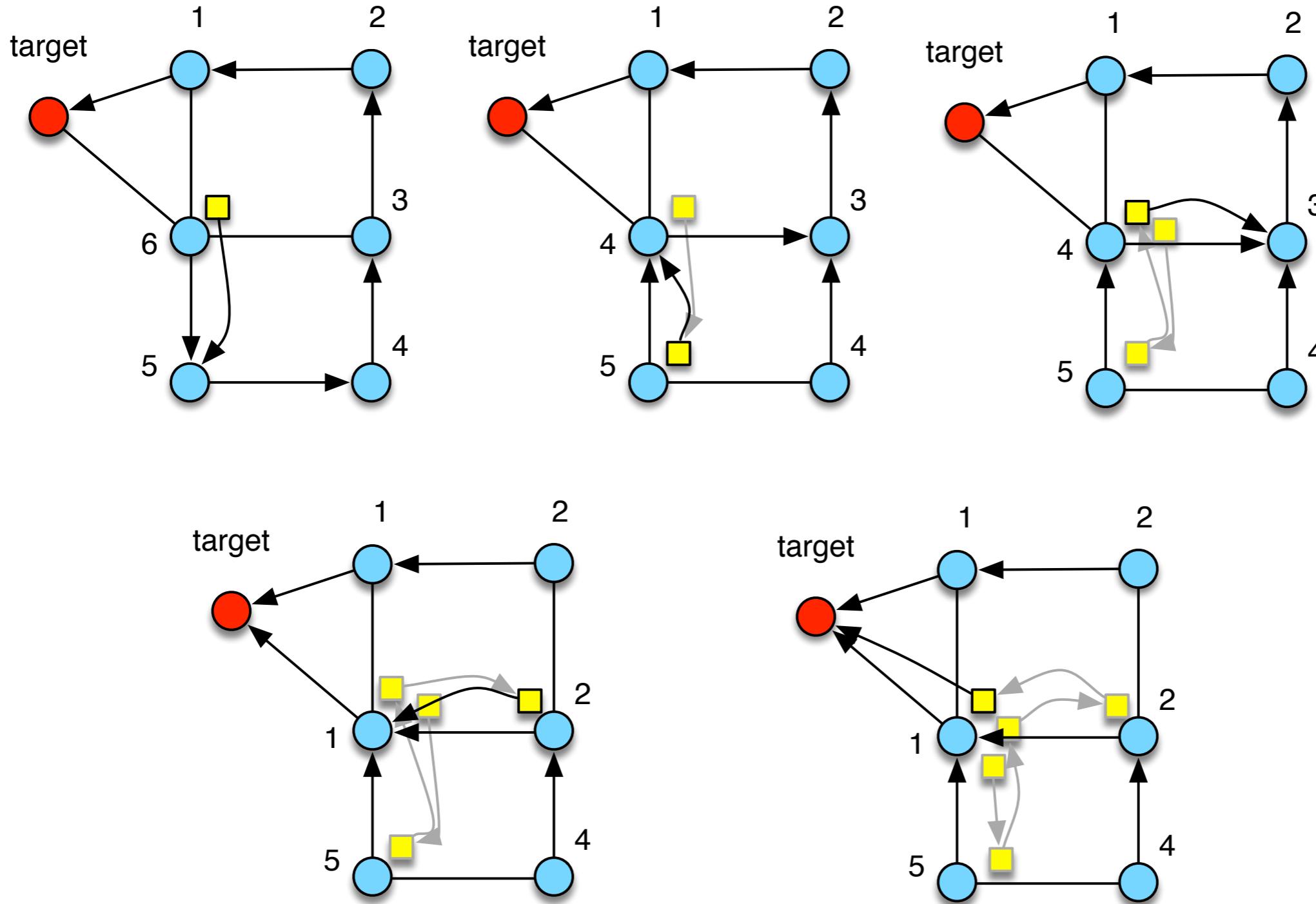
Distance-Vector für ein Ziel



Distance-Vector für ein Ziel

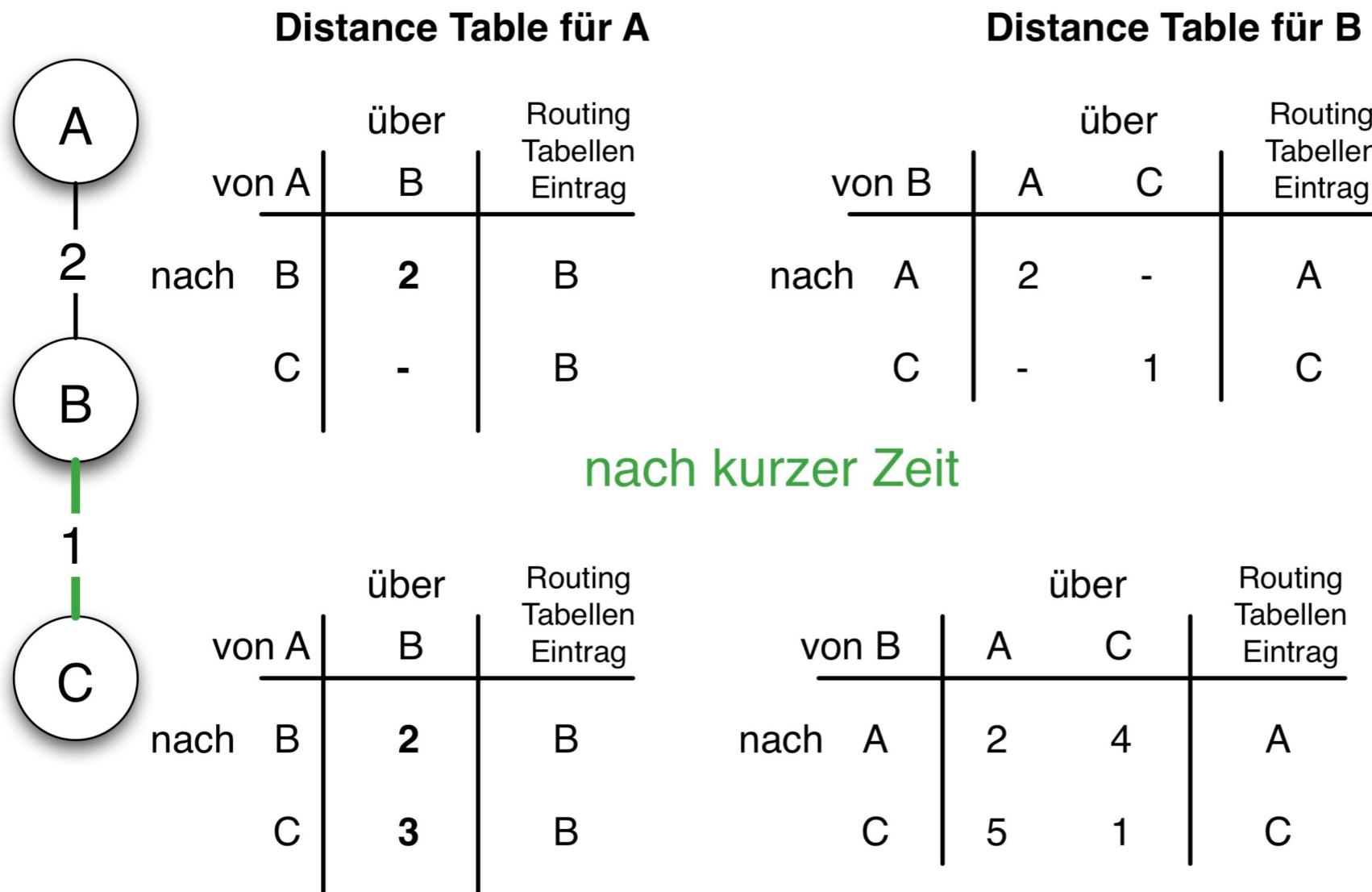


Irrlichter im Routing



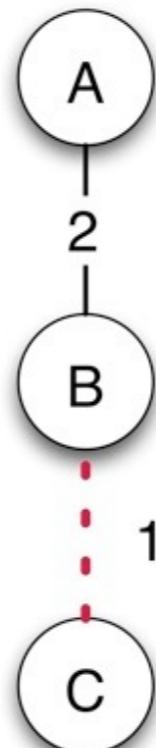
Das “Count to Infinity” - Problem

- Gute Nachrichten verbreiten sich schnell
 - Neue Verbindung wird schnell veröffentlicht



Das “Count to Infinity” - Problem

- Schlechte Nachrichten verbreiten sich langsam



- Verbindung fällt aus
- Nachbarn erhöhen wechselseitig ihre Entfernung
- “Count to Infinity”-Problem

		über	Routing Tabellen Eintrag	
		von A	B	
nach	B	2	B	
	C	3	B	

		über	Routing Tabellen Eintrag	
		von B	A	
nach	A	2	-	A
	C	5	-	A

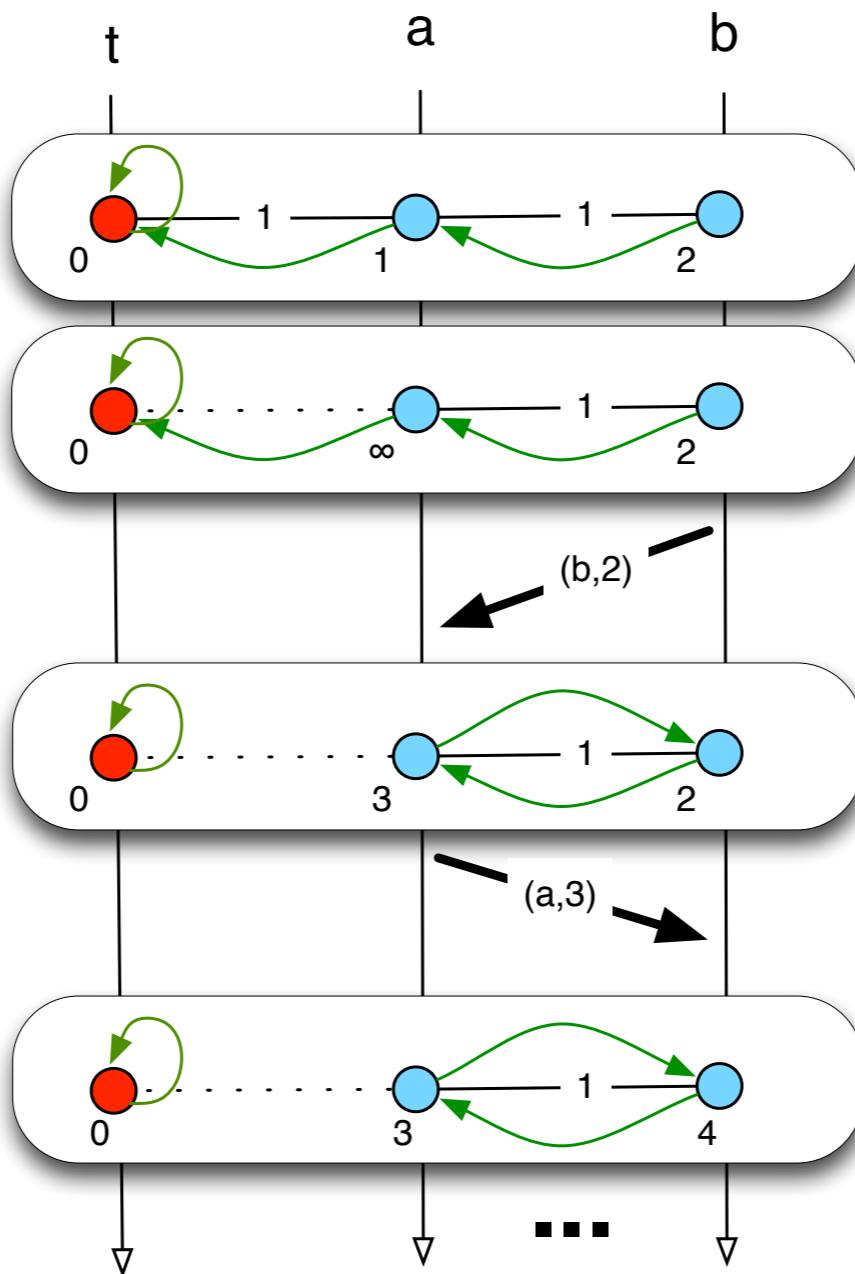
		über	Routing Tabellen Eintrag	
		von A	B	
nach	B	2	B	
	C	7	B	

		über	Routing Tabellen Eintrag	
		von B	A	
nach	A	2	-	A
	C	5	-	A

		über	Routing Tabellen Eintrag	
		von A	B	
nach	B	2	B	
	C	7	B	

		über	Routing Tabellen Eintrag	
		von B	A	
nach	A	2	-	A
	C	9	-	A

Das “Count to Infinity” - Problem für Ziel t



Link-State Protocol

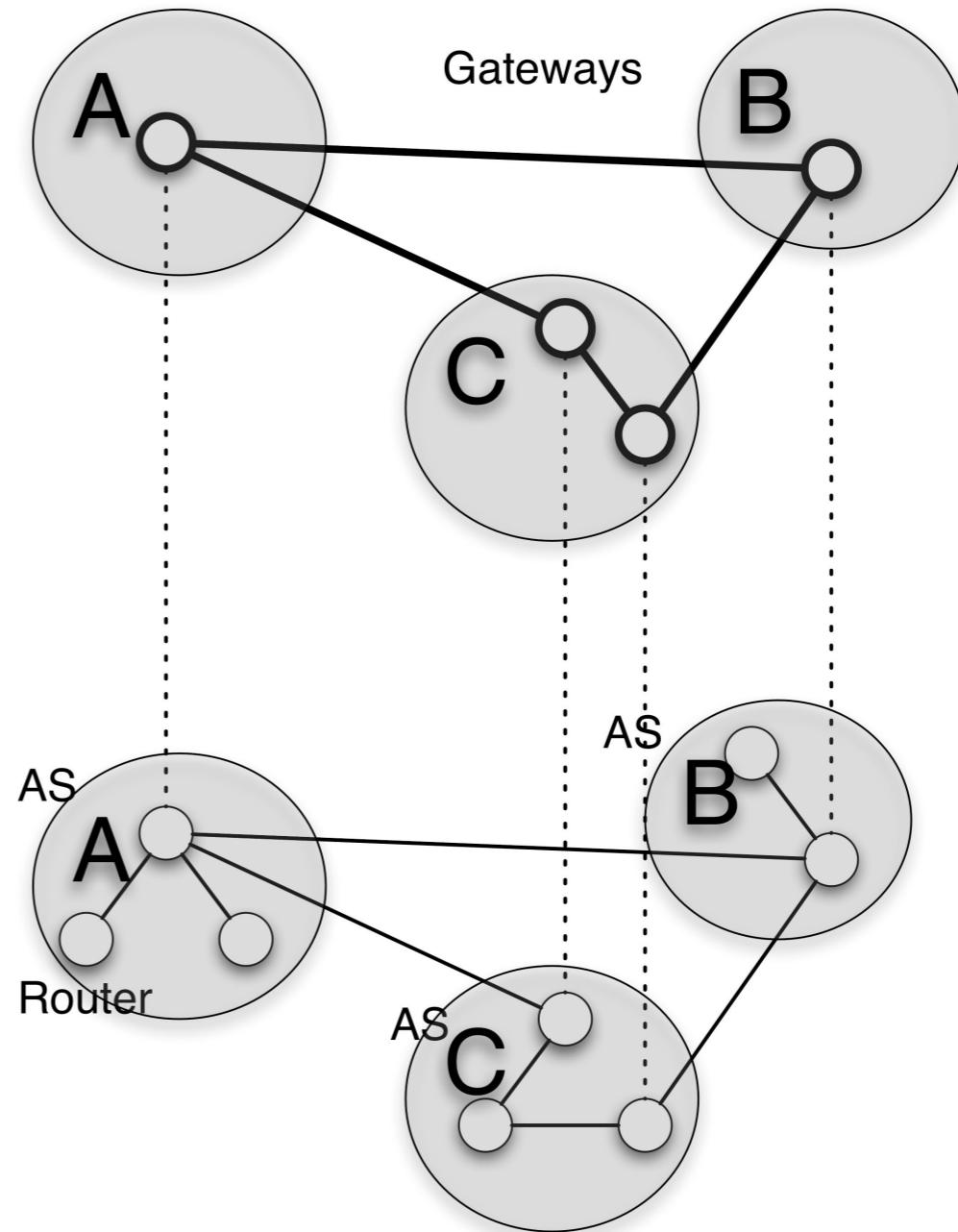
- Link State Router
 - tauschen Information mittels Link State Packets (LSP) aus
 - Jeder verwendet einen eigenen Kürzeste-Wege-Algorithmus zu Anpassung der Routing-Tabelle
- LSP enthält
 - ID des LSP erzeugenden Knotens
 - Kosten dieses Knotens zu jedem direkten Nachbarn
 - Sequenznr. (SEQNO)
 - TTL-Feld für dieses Feld (time to live)
- Verlässliches Fluten (Reliable Flooding)
 - Die aktuellen LSP jedes Knoten werden gespeichert
 - Weiterleitung der LSP zu allen Nachbarn
 - bis auf den Knoten der diese ausgeliefert hat
 - Periodisches Erzeugen neuer LSPs
 - mit steigender SEQNOs
 - Verringern der TTL bei jedem Weiterleiten

Die Grenzen des flachen Routing

- Link State Routing
 - benötigt $O(g n)$ Einträge für n Router mit maximalen Grad g
 - Jeder Knoten muss an jeden anderen seine Informationen senden
- Distance Vector
 - benötigt $O(g n)$ Einträge
 - kann Schleifen einrichten
 - Konvergenzzeit steigt mit Netzwerkgröße
- Im Internet gibt es mehr als 10^7 Router
 - damit sind diese so genannten flachen Verfahren nicht einsetzbar
- Lösung:
 - Hierarchisches Routing

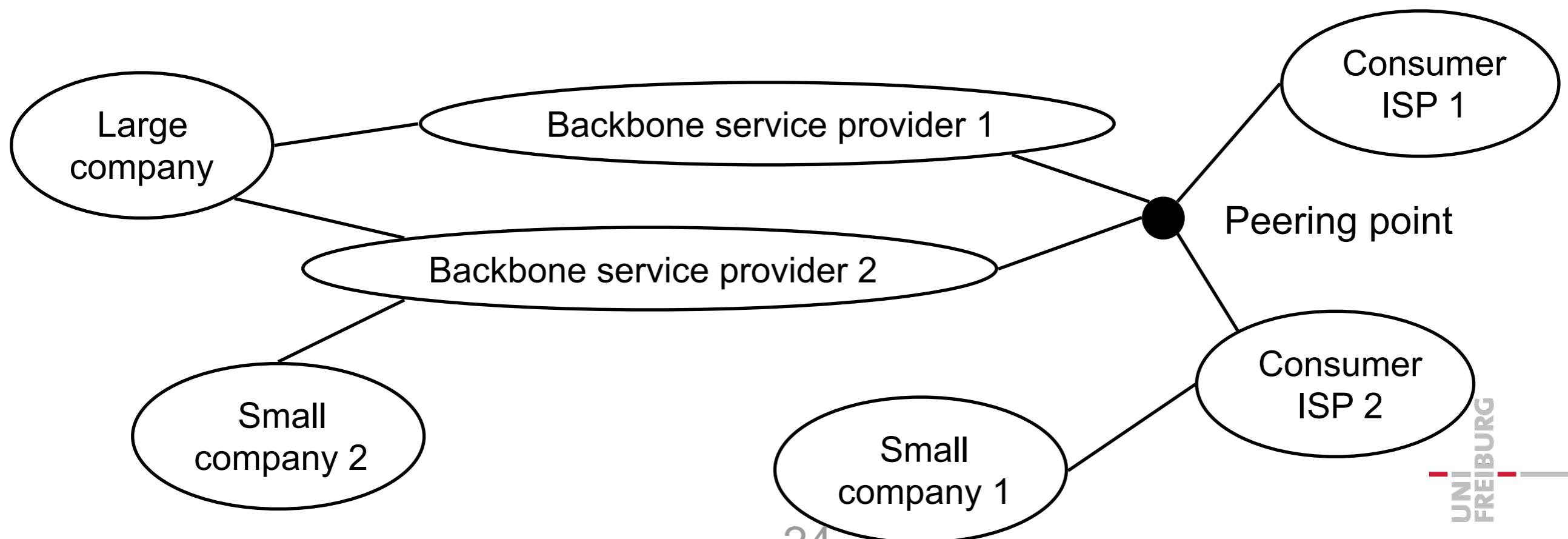
AS, Intra-AS und Inter-AS

- Autonomous System (AS)
 - liefert ein zwei Schichten-Modell des Routing im Internet
 - Beispiele für AS:
 - uni-freiburg.de
- Intra-AS-Routing (Interior Gateway Protocol)
 - ist Routing innerhalb der AS
 - z.B. RIP, OSPF, IGRP, ...
- Inter-AS-Routing (Exterior Gateway Protocol)
 - Übergabepunkte sind Gateways
 - ist vollkommen dezentrales Routing
 - Jeder kann seine Optimierungskriterien vorgeben
 - z.B. EGP (früher), BGP



Typen autonomer Systeme

- Stub-AS
 - Nur eine Verbindung zu anderen AS
- Multihomed AS
 - Verbindungen zu anderen ASen
 - weigert sich aber Verkehr für andere zu befördern
- Transit AS
 - Mehrere Verbindungen
 - Leitet fremde Nachrichten durch (z.B. ISP)



Intra-AS: RIP

Routing Information Protocol

- Distance Vector Algorithmus
 - Distanzmetrik = Hop-Anzahl
- Distanzvektoren
 - werden alle 30s durch Response-Nachricht (advertisement) ausgetauscht
- Für jedes Advertisement
 - Für bis zu 25 Zielnetze werden Routen veröffentlicht per UDP
- Falls kein Advertisement nach 180s empfangen wurde
 - Routen über Nachbarn werden für ungültig erklärt
 - Neue Advertisements werden zu den Nachbarn geschickt
 - Diese antworten auch mit neuen Advertisements
 - falls die Tabellen sich ändern
 - Rückverbindungen werden unterdrückt um Ping-Pong-Schleifen zu verhindern (poison reverse) gegen Count-to-Infinity-Problem
 - Unendliche Distanz = 16 Hops

Intra-AS OSPF (Open Shortest Path First)

- “open” = öffentlich verfügbar
- Link-State-Algorithmus
 - LS Paket-Verbreitung
 - Topologie wird in jedem Knoten abgebildet
 - Routenberechnung mit Dijkstras Algorithmus
- OSPF-Advertisement
 - per TCP, erhöht Sicherheit (security)
 - werden in die gesamte AS geflutet
 - Mehrere Wege gleicher Kosten möglich

Intra-AS

Hierarchisches OSPF

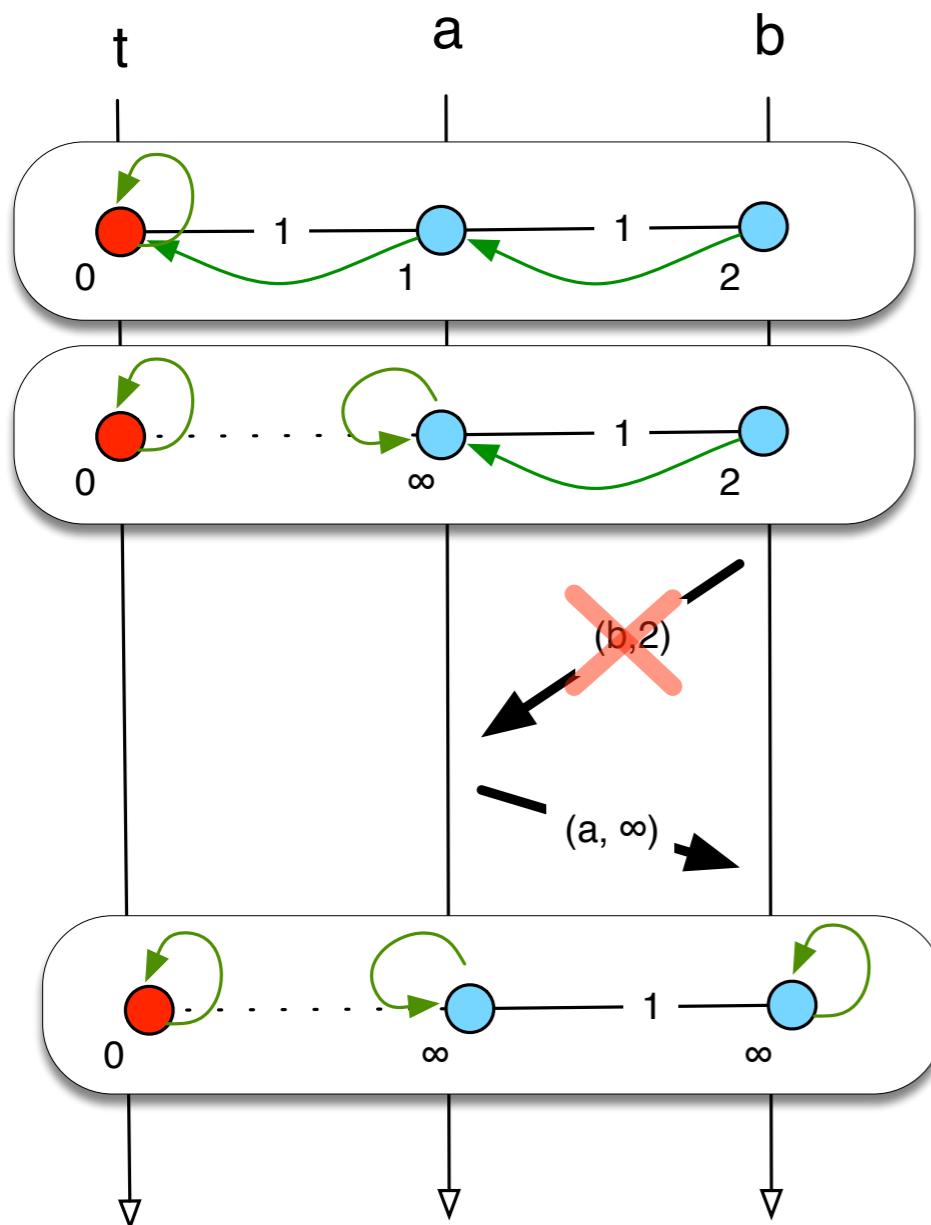
- Für große Netzwerke zwei Ebenen:
 - Lokales Gebiet und Rückgrat (backbone)
 - Lokal: Link-state advertisement
 - Jeder Knoten berechnet nur in Richtung zu den Netzen in anderen lokalen Gebieten
- Local Area Border Router:
 - Fassen die Distanzen in das eigene lokale Gebiet zusammen
 - Bieten diese den anderen Area Border Routern an (per Advertisement)
- Backbone Routers
 - verwenden OSPF beschränkt auf das Rückgrat (backbone)
- Boundary Routers:
 - verbinden zu anderen AS

Intra-AS: IGRP (Interior Gateway Routing Protocol)

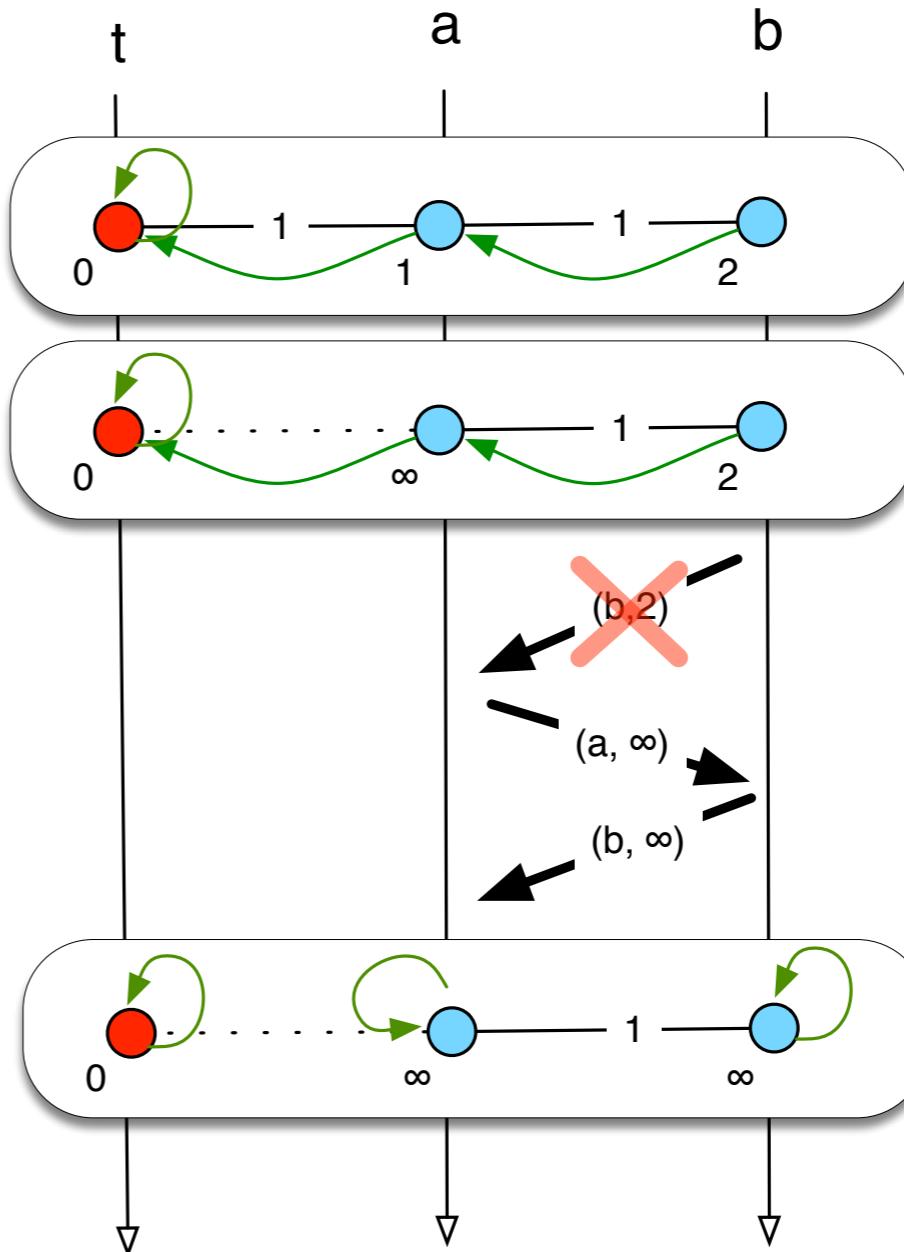
- CISCO-Protokoll, Nachfolger von RIP (1980er)
- Distance-Vector-Protokoll, wie RIP
 - Hold Down
 - weggefallene Verbindungen werden mit Entfernung „unendlich“ angeboten (100 = unendlich)
 - Split Horizon
 - Advertisements werden nicht auf dem angebotenen Pfad weitergegeben
 - Poison Reverse
 - weggefallene Verbindungen werden sofort mit Entfernung „unendlich“ allen Nachbarn angeboten
- Verschiedene Kostenmetriken
 - Delay, Bandwidth, Reliability, Load etc.
- Verwendet TCP für den Austausch von Routing Updates

Lösungen für Count-to-Infinity

Split Horizon



Poison Reverse



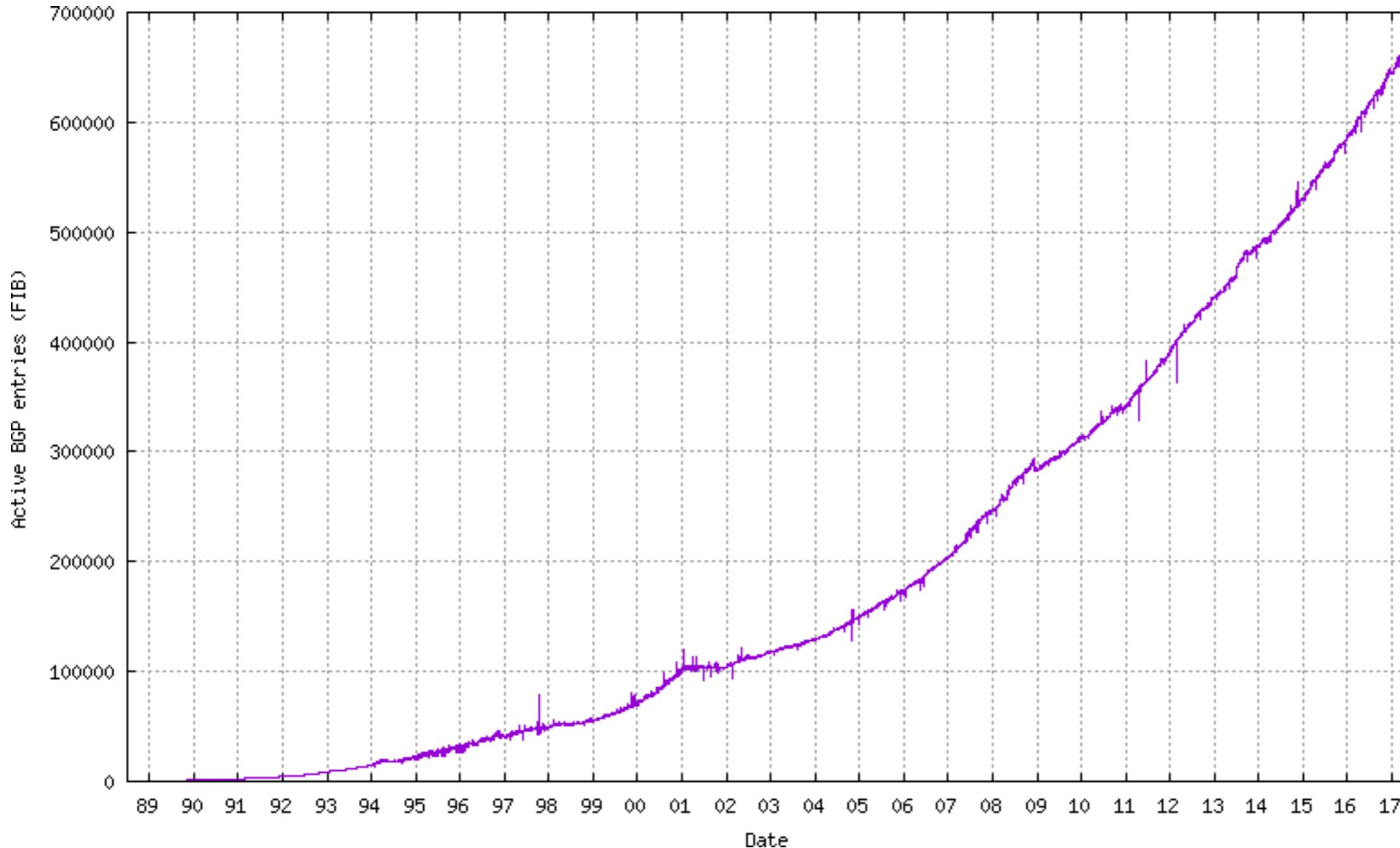
Inter-AS-Routing

- Inter-AS-Routing ist schwierig...
 - Organisationen können Durchleitung von Nachrichten verweigern
 - Politische Anforderungen
 - Weiterleitung durch andere Länder?
 - Routing-Metriken der verschiedenen autonomen Systeme sind oftmals unvergleichbar
 - Wegeoptimierung unmöglich!
 - Inter-AS-Routing versucht wenigstens Erreichbarkeit der Knoten zu ermöglichen
 - Größe: momentan müssen Inter-Domain-Router mehr als 300.000 Einträge verwalten

Inter-AS: BGPv4 (Border Gateway Protocol)

- Ist faktisch der Standard
- Path-Vector-Protocol
 - ähnlich wie Distance Vector Protocol
 - es werden aber ganze Pfade zum Ziel gespeichert
 - jeder Border Gateway teilt all seinen Nachbarn (peers) den gesamten Pfad (Folge von ASen) zum Ziel mit (advertisement) (per TCP)
- Falls Gateway X den Pfad zum Peer-Gateway W sendet
 - dann kann W den Pfad wählen oder auch nicht
 - Optimierungskriterien:
 - Kosten, Politik, etc.
 - Falls W den Pfad von X wählt, dann publiziert er
 - $\text{Path}(W,Z) = (W, \text{Path } (X,Z))$
- Anmerkung
 - X kann den eingehenden Verkehr kontrollieren durch Senden von Advertisements
 - Sehr kompliziertes Protokoll

BGP-Routing Tabellengröße 1989-2017



<http://bgp.potaroo.net/as1221/bgp-active.html>

Broadcast & Multicast

■ Broadcast routing

- Ein Paket soll (in Kopie) an alle ausgeliefert werden
- Lösungen:
 - Fluten des Netzwerks
 - Besser: Konstruktion eines minimalen Spannbaums

■ Multicast routing

- Ein Paket soll an eine gegebene Teilmenge der Knoten ausgeliefert werden (in Kopie)
- Lösung:
 - Optimal: Steiner Baum Problem (bis heute nicht lösbar)
 - Andere (nicht-optimale) Baum-konstruktionen

IP Multicast

Motivation

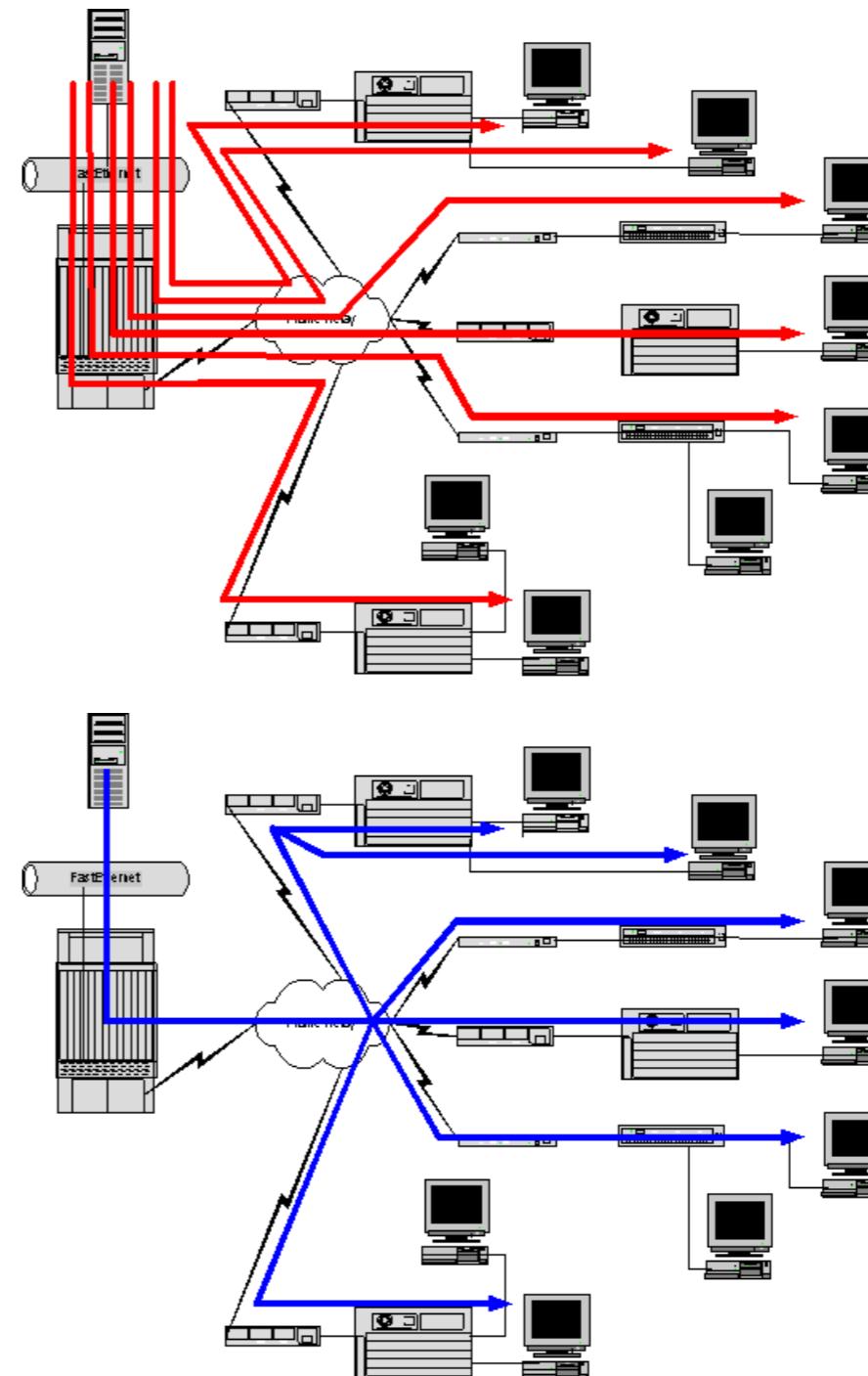
- Übertragung eines Stroms an viele Empfänger

Unicast

- Strom muss mehrfach einzeln übertragen werden
- Bottleneck am Sender

Multicast

- Strom wird über die Router vervielfältigt
- Kein Bottleneck mehr



Funktionsprinzip

- IPv4 Multicast-Adressen
 - in der Klasse D (außerhalb des CIDR - Classless Interdomain Routings)
 - 224.0.0.0 - 239.255.255.255
 - in IPv6 mit Präfix FF
- Hosts melden sich per IGMP bei der Adresse an
 - IGMP = Internet Group Management Protocol
 - Nach der Anmeldung wird der Multicast-Tree aktualisiert
- Source sendet an die Multicast-Adresse
 - Router duplizieren die Nachrichten an den Routern
 - und verteilen sie in die Bäume
- Angemeldete Hosts erhalten diese Nachrichten
 - bis zu einem Time-Out
 - oder bis sie sich abmelden
- Achtung:
 - Kein TCP, nur UDP
 - Viele Router lehnen die Beförderung von Multicast-Nachrichten ab
 - Lösung: Tunneln

Routing Protokolle

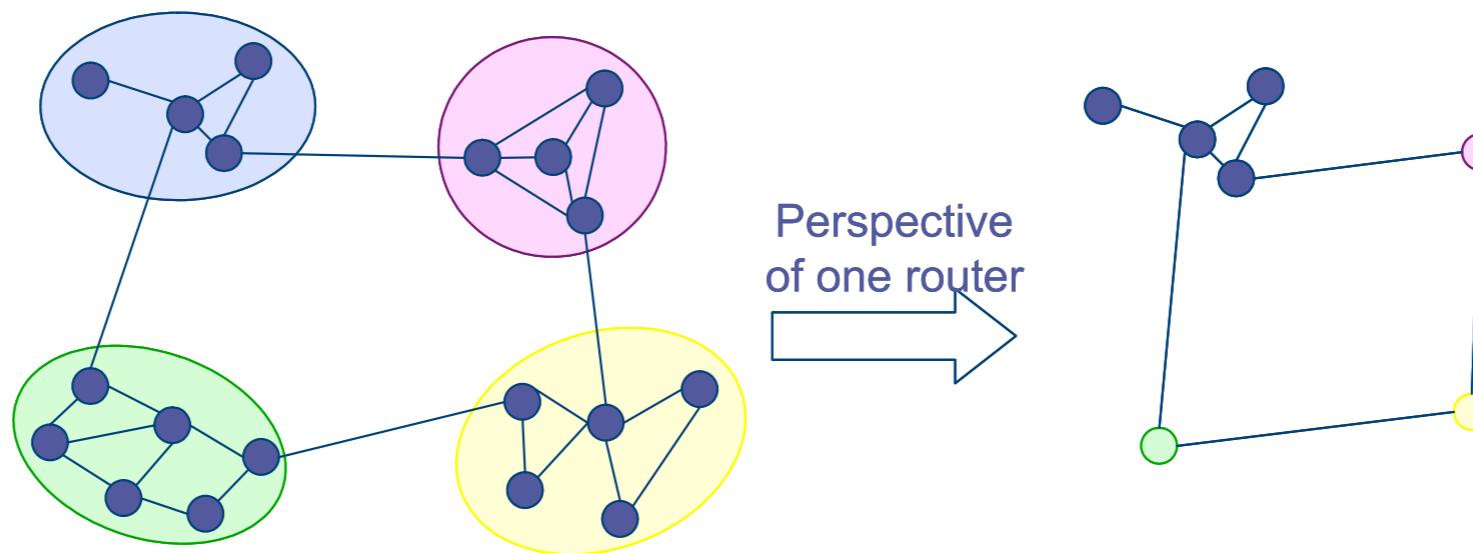
- Distance Vector Multicast Routing Protocol (DVMRP)
 - jahrelang eingesetzt in MBONE (insbesondere in Freiburg)
 - Eigene Routing-Tabelle für Multicast
- Protocol Independent Multicast (PIM)
 - im Sparse Mode (PIM-SM)
 - aktueller Standard
 - beschneidet den Multicast Baum
 - benutzt Unicast-Routing-Tabellen
 - ist damit weitestgehend protokollunabhängig
- Voraussetzung PIM-SM:
 - benötigt Rendevous-Point (RP) in ein-Hop-Entfernung
 - RP muss PIM-SM unterstützen
 - oder Tunneling zu einem Proxy in der Nähe eines RP

Warum so wenig IP Multicast?

- Trotz erfolgreichen Einsatz
 - in Video-Übertragung von IETF-Meetings
 - MBONE (Multicast Backbone)
- gibt es wenig ISP welche IP Multicast in den Routern unterstützen
- Zusätzlicher Wartungsaufwand
 - Schwierig zu konfigurieren
 - Verschiedene Protokolle
- Gefahr von Denial-of-Service-Attacken
 - Implikationen größer als bei Unicast
- Transport-Protokoll
 - Nur UDP einsetzbar
 - Zuverlässige Protokolle
 - Vorwärtsfehlerkorrektur
 - Oder proprietäre Protokolle in den Routern (z.B. CISCO)
- Marktsituation
 - Endkunden fragen kaum Multicast nach (benutzen lieber P2P-Netzwerke)
 - Wegen einzelner Dateien und weniger Abnehmer erscheint ein Multicast wenig erstrebenswert (Adressenknappheit!)

Adressierung und Hierarchisches Routing

- Flache (MAC-) Adressen haben keine Strukturinformation



- Hierarchische Adressen
 - Routing wird vereinfacht wenn Adressen hierarchische Routing-Struktur abbilden
 - Group-ID_n:Group-ID_{n-1}:...:Group-ID₁:Device-ID

■ IP-Adressen

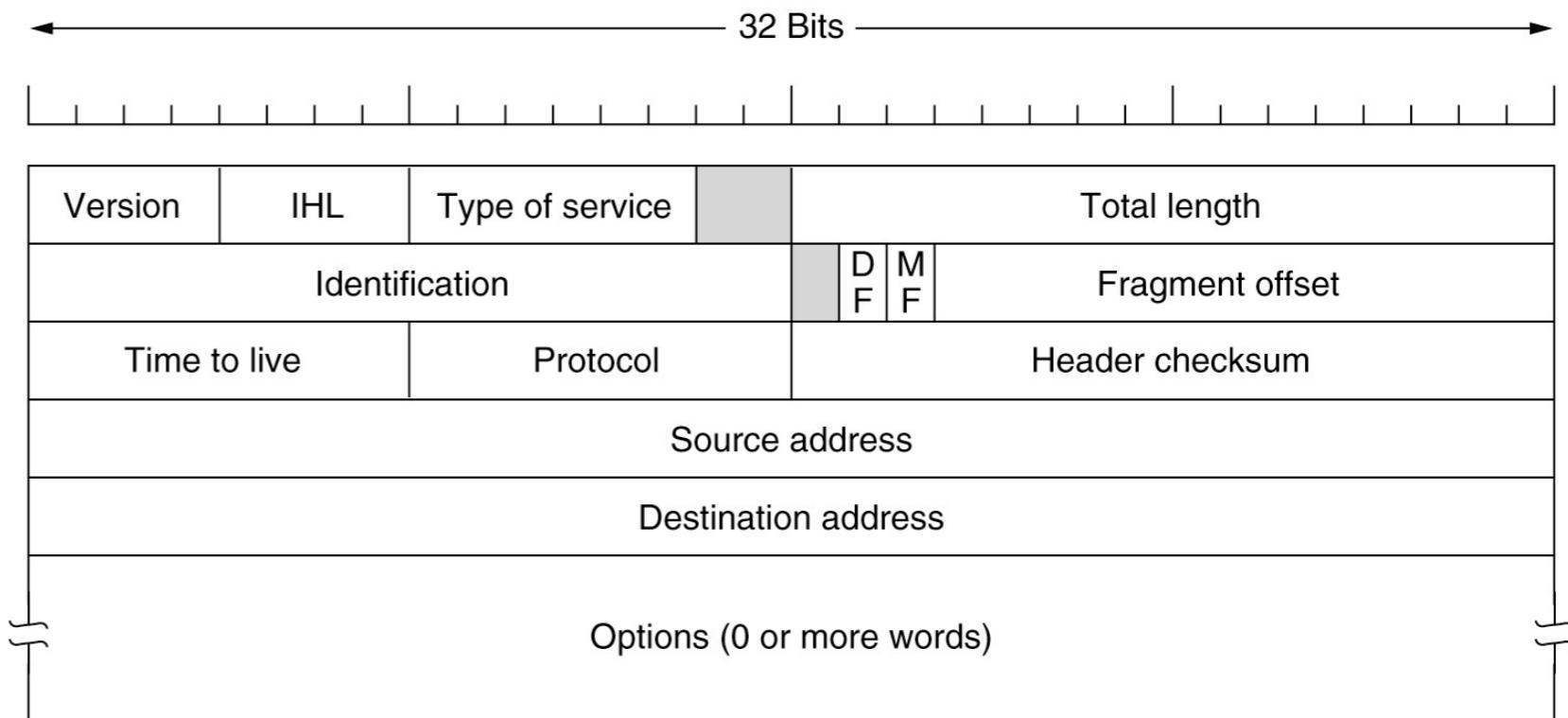
- Jedes Interface in einem Netzwerk hat weltweit eindeutige IP-Adresse
- 32 Bits unterteilt in Net-ID und Host-ID
- Net-ID vergeben durch Internet Network Information Center
- Host-ID durch lokale Netzwerkadministration

■ Domain Name System (DNS)

- Ersetzt IP-Adressen wie z.B. 132.230.167.230 durch Namen wie z.B. falcon.informatik.uni-freiburg.de und umgekehrt
- Verteilte robuste Datenbank

IPv4-Header (RFC 791)

- Version: 4 = IPv4
- IHL: IP Headerlänge
 - in 32 Bit-Wörtern (>5)
- Type of Service
 - Optimiere delay, throughput, reliability, monetary cost
- Checksum (nur für IP-Header)
- Source and destination IP-address
- Protocol, identifiziert passendes Protokoll
 - Z.B. TCP, UDP, ICMP, IGMP
- Time to Live:
 - maximale Anzahl Hops

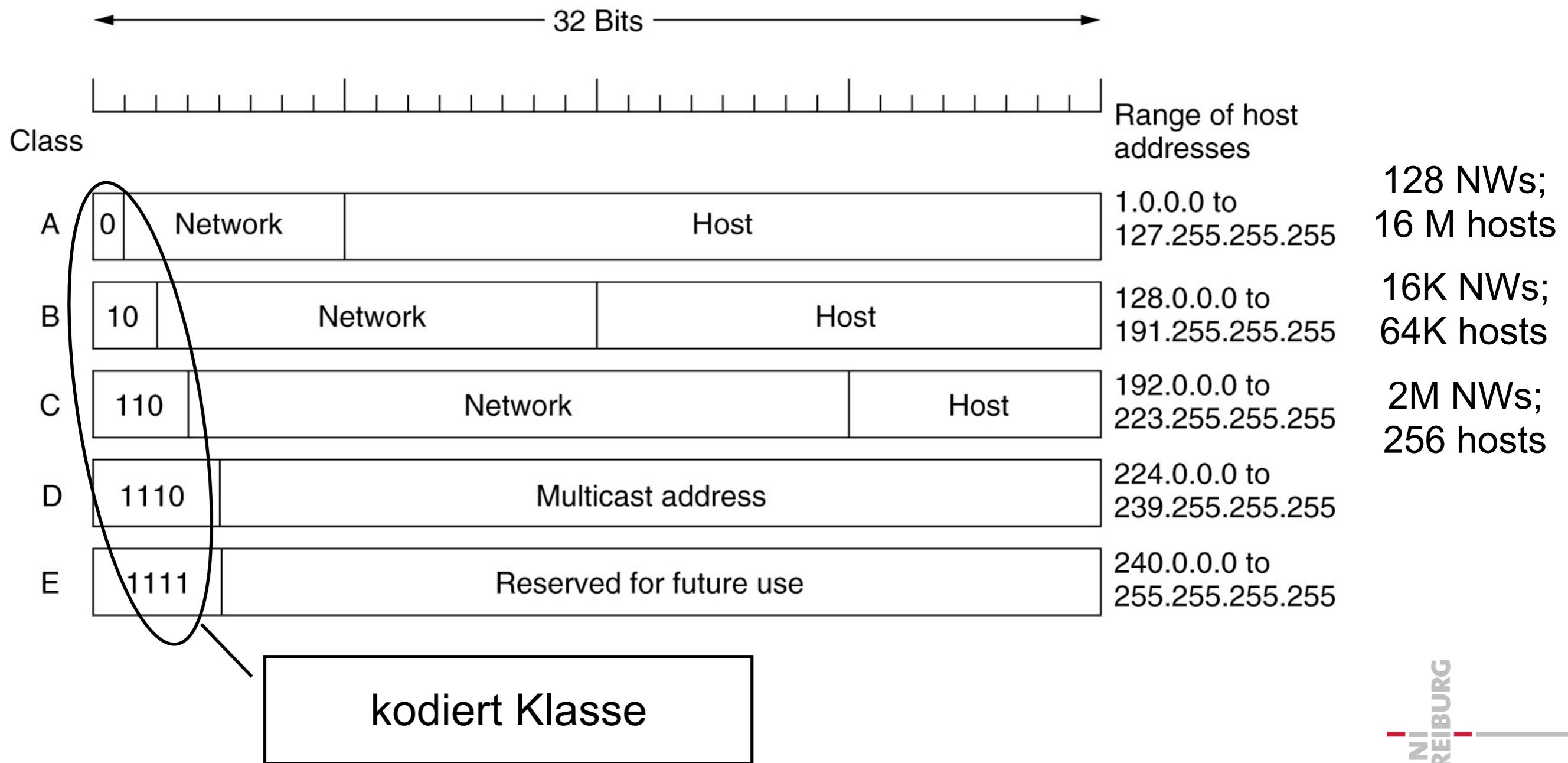


Internet IP Adressen bis 1993

- IP-Adressen unterscheiden zwei Hierarchien
 - Netzwerk-Interfaces
 - Netzwerke
 - Verschiedene Netzwerkgrößen
 - Netzwerkklassen:
 - Groß - mittel - klein
(Klasse A, B, and C)
- Eine IP-Adresse hat 32 Bits
 - Erster Teil: Netzwerkadresse
 - Zweiter Teil: Interface

IP-Klassen bis 1993

- Klassen A, B, and C
- D für multicast; E: “reserved”



IPv4-Adressen

- Bis 1993 (heutzutage veraltet)
 - 5 Klassen gekennzeichnet durch Präfix
 - Dann Subnetzpräfix fester Länge und Host-ID (Geräteteil)
- Seit 1993
 - Classless Inter-Domain-Routing (CIDR)
 - Die Netzwerk-Adresse und die Host-ID (Geräteteil) werden variabel durch die Netzwerkmaske aufgeteilt.
 - Z.B.:
 - Die Netzwerkmaske 1111111.1111111.1111111.00000000
 - Besagt, dass die IP-Adresse
 - 10000100. 11100110. 10010110. 11110011
 - Aus dem Netzwerk 10000100. 11100110. 10010110
 - den Host 11110011 bezeichnet
- Route aggregation
 - Die Routing-Protokolle BGP, RIP v2 und OSPF können verschiedene Netzwerke unter einer ID anbieten
 - Z.B. alle Netzwerke mit Präfix 10010101010* werden über Host X erreicht

Umwandlung in MAC-Adressen: ARP

- Address Resolution Protocol (ARP)
- Umwandlung: IP-Adresse in MAC-Adresse
 - Broadcast im LAN, um nach Rechner mit passender IP-Adresse zu fragen
 - Knoten antwortet mit MAC-Adresse
 - Router kann dann das Paket dorthin ausliefern
- IPv6:
 - Funktionalität durch Neighbor Discovery Protocol (NDP)
 - Informationen werden per ICMPv6 ausgetauscht

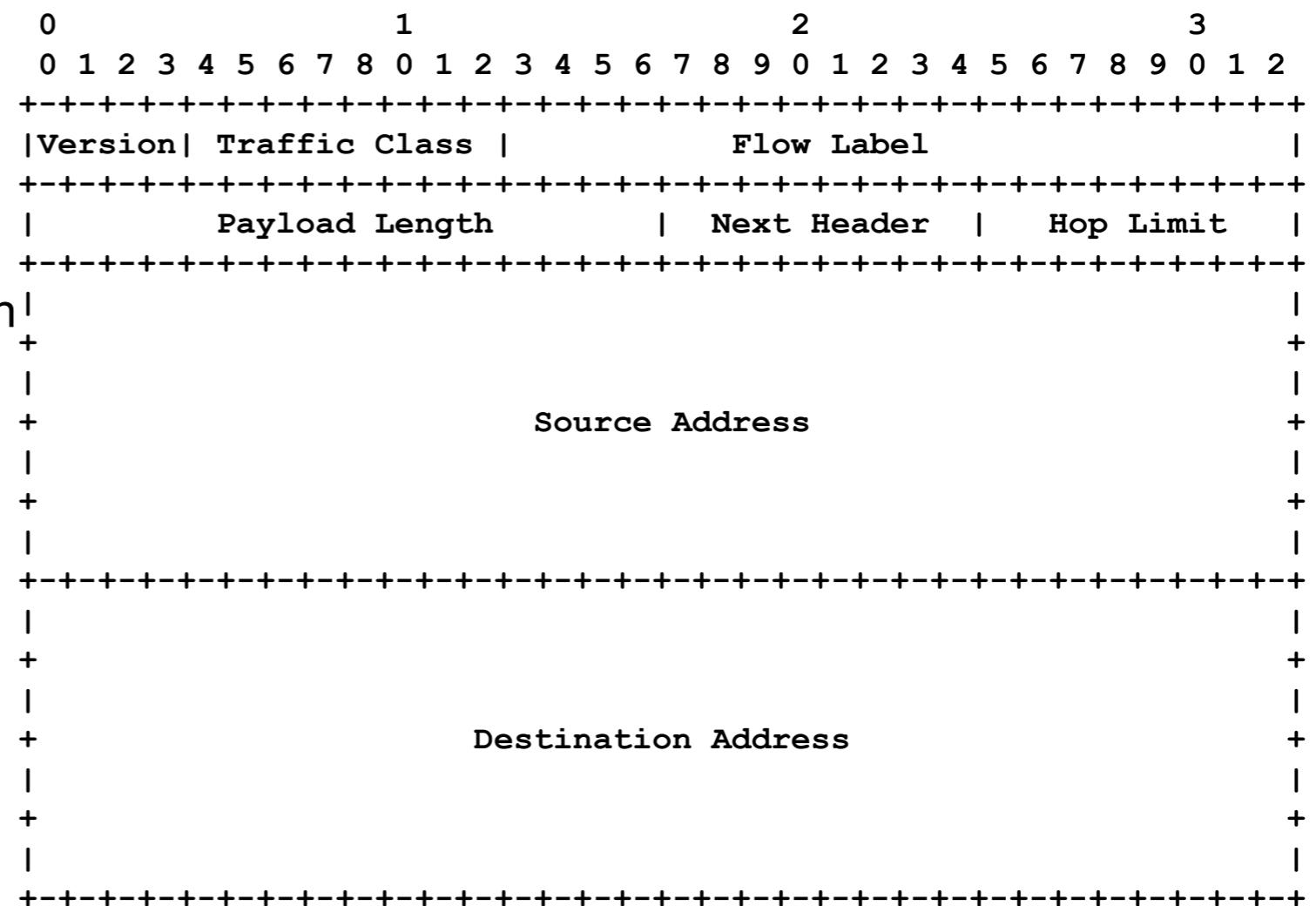
- Wozu IPv6:
 - Freie IPv4-Adressen sind seit 31.01.2011 nicht mehr vorhanden
 - Zwar gibt es 4 Milliarden in IPv4 (32 Bit)
 - Diese sind aber statisch organisiert in Netzwerk- und Host-ID
 - Adressen für Funktelefone, Kühlschränke, Autos, Tastaturen, etc...
 - Autokonfiguration
 - DHCP, Mobile IP, Umnummerierung
 - Neue Dienste
 - Sicherheit (IPSec)
 - Qualitätssicherung (QoS)
 - Multicast
 - Anycast
 - Vereinfachungen für Router
 - keine IP-Prüfsummen
 - Keine Partitionierung von IP-Paketen

Lösung der Adressenknappheit: DHCP

- DHCP (Dynamic Host Configuration Protocol)
 - Manuelle Zuordnung (Bindung an die MAC-Adresse, z.B. für Server)
 - Automatische Zuordnung (feste Zuordnung, nicht voreingestellt)
 - Dynamische Zuordnung (Neuvergabe möglich)
- Einbindung neuer Rechner ohne Konfiguration
 - Rechner „holt“ sich die IP-Adresse von einem DHCP-Server
 - Dieser weist dem Rechner die IP-Adressen dynamisch zu
 - Nachdem der Rechner das Netzwerk verlässt, kann die IP-Adresse wieder vergeben werden
 - Bei dynamischer Zuordnung, müssen IP-Adressen auch „aufgefrischt“ werden
 - Versucht ein Rechner eine alte IP-Adresse zu verwenden,
 - die abgelaufen ist oder
 - schon neu vergeben ist
 - Dann werden entsprechende Anfragen zurückgewiesen
 - Problem: Stehlen von IP-Adressen

IPv6-Header (RFC 2460)

- Version: 6 = IPv6
- Traffic Class
 - Für QoS (Prioritätsvergabe)
- Flow Label
 - Für QoS oder Echtzeitanwendungen
- Payload Length
 - Größe des Rests des IP-Pakets (Datagramms)
- Next Header (wie bei IPv4: protocol)
 - Z.B. ICMP, IGMP, TCP, EGP, UDP, Multiplexing, ...
- Hop Limit (Time to Live)
 - maximale Anzahl Hops
- Source Address
- Destination Address
 - 128 Bit IPv6-Adresse



IPsec (RFC 2401)

- Schutz vor Replay-Attacken
- IKE (Internet Key Exchange) Protokoll
 - Vereinbarung einer Security Association
 - Identifikation, Festlegung von Schlüsseln, Netzwerke, Erneuerungszeiträume für Authentifizierung und IPsec Schlüssel
 - Erzeugung einer SA im Schnellmodus (nach Etablierung)
- Encapsulating Security Payload (ESP)
 - IP-Kopf unverschlüsselt, Nutzdaten verschlüsselt, mit Authentifizierung
- IPsec im Transportmodus (für direkte Verbindungen)
 - IPsec Header zwischen IP-Header und Nutzdaten
 - Überprüfung in den IP-Routern (dort muss IPsec vorhanden sein)
- IPsec im Tunnelmodus (falls mindestens ein Router dazwischen ist)
 - Das komplette IP-Paket wird verschlüsselt und mit dem IPsec-Header in einen neuen IP-Header verpackt
 - Nur an den Enden muss IPsec vorhanden sein.
- IPsec ist Bestandteil von IPv6
- Rückport nach IPv4

Firewalls

- Typen von Firewalls
 - Host-Firewall
 - Netzwerk-Firewall
- Netzwerk-Firewall
 - unterscheidet
 - Externes Netz
(Internet - feindselig)
 - Internes Netz
(LAN - vertrauenswürdig)
 - Demilitarisierte Zone
(vom externen Netz erreichbare Server)
- Host-Firewall
 - z.B. Personal Firewall
 - kontrolliert den gesamten Datenverkehr eines Rechners
 - Schutz vor Attacken von außerhalb und von innen (Trojanern)

Firewalls - Methoden

- Paketfilter
 - Sperren von Ports oder IP-Adressen
 - Content-Filter
 - Filtern von SPAM-Mails, Viren, ActiveX oder JavaScript aus HTML-Seiten
- Proxy
 - Transparente (extern sichtbare) Hosts
 - Kanalisierung der Kommunikation und möglicher Attacken auf gesicherte Rechner
- NAT, PAT
 - Network Address Translation
 - Port Address Translation
- Bastion Host
- Proxy

Firewalls: Begriffe

- **(Network) Firewall**
 - beschränkt den Zugriff auf ein geschütztes Netzwerk aus dem Internet
- **Paket-Filter**
 - wählen Pakete aus dem Datenfluss in oder aus dem Netzwerk aus
 - Zweck des Eingangsfilter:
 - z.B. Verletzung der Zugriffskontrolle
 - Zweck des Ausgangsfilter:
 - z.B. Trojaner
- **Bastion Host**
 - ist ein Rechner an der Peripherie, der besonderen Gefahren ausgesetzt ist
 - und daher besonders geschützt ist
- **Dual-homed host**
 - Normaler Rechner mit zwei Interfaces (verbindet zwei Netzwerke)

Firewalls: Begriffe

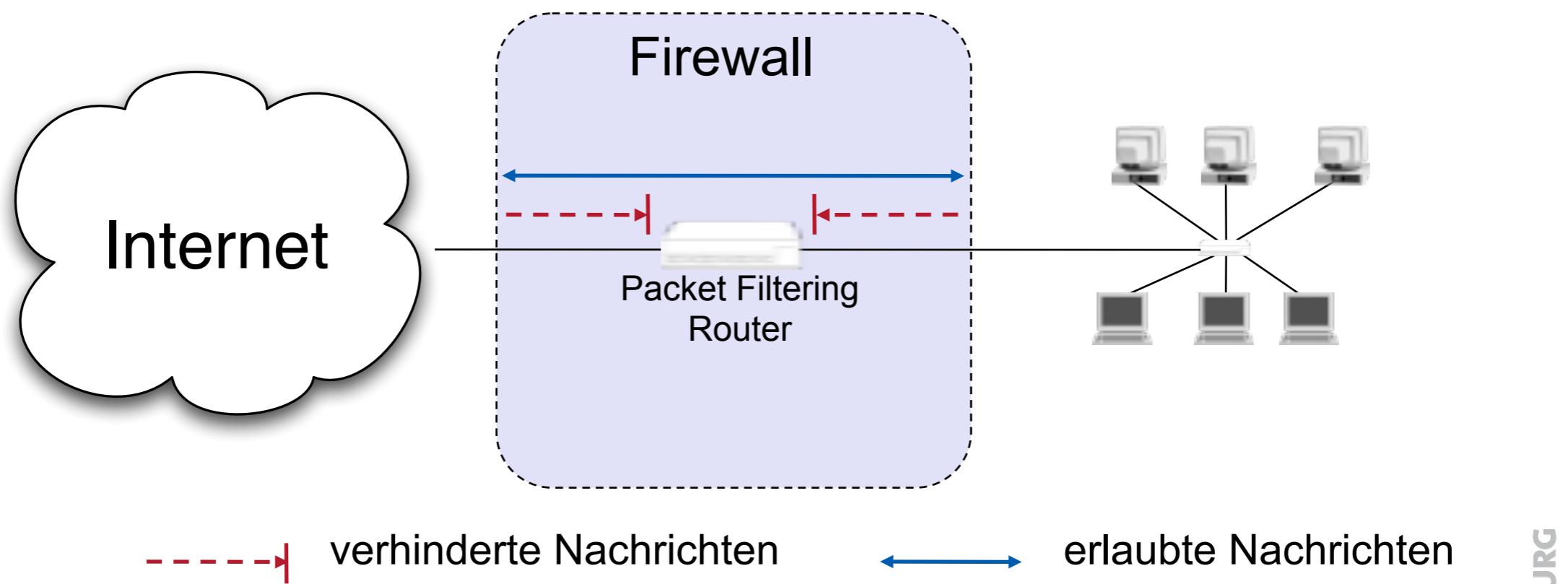
- Proxy (Stellvertreter)
 - Spezieller Rechner, über den Anfragen umgeleitet werden
 - Anfragen und Antworten werden über den Proxy geleitet
 - Vorteil
 - Nur dort müssen Abwehrmaßnahmen getroffen werden
- Perimeter Network:
 - Ein Teilnetzwerk, das zwischen gesicherter und ungesicherter Zone eine zusätzliche Schutzschicht bietet
 - Synonym demilitarisierte Zone (DMZ)

- NAT (Network Address Translation)
- Basic NAT (Static NAT)
 - Jede interne IP wird durch eine externe IP ersetzt
- Hiding NAT = PAT (Port Address Translation) = NAPT (Network Address Port Translation)
 - Das Socket-Paar (IP-Adresse und Port-Nummer) wird umkodiert

- **Verfahren**
 - Die verschiedenen lokalen Rechner werden in den Ports kodiert
 - Diese werden im Router an der Verbindung zum WAN dann geeignet kodiert
 - Bei ausgehenden Paketen wird die LAN-IP-Adresse und ein kodierter Port als Quelle angegeben
 - Bei eingehenden Paketen (mit der LAN-IP-Adresse als Ziel), kann dann aus dem kodierten Port der lokale Rechner und der passende Port aus einer Tabelle zurückgerechnet werden
- **Sicherheitsvorteile**
 - Rechner im lokalen Netzwerk können nicht direkt angesprochen werden
 - Löst auch das Problem knapper IPv4-Adressen
 - NAT nicht üblich für IPv6
 - Lokale Rechner können nicht als Server dienen
- **DHCP (Dynamic Host Configuration Protocol)**
 - bringt ähnliche Vorteile

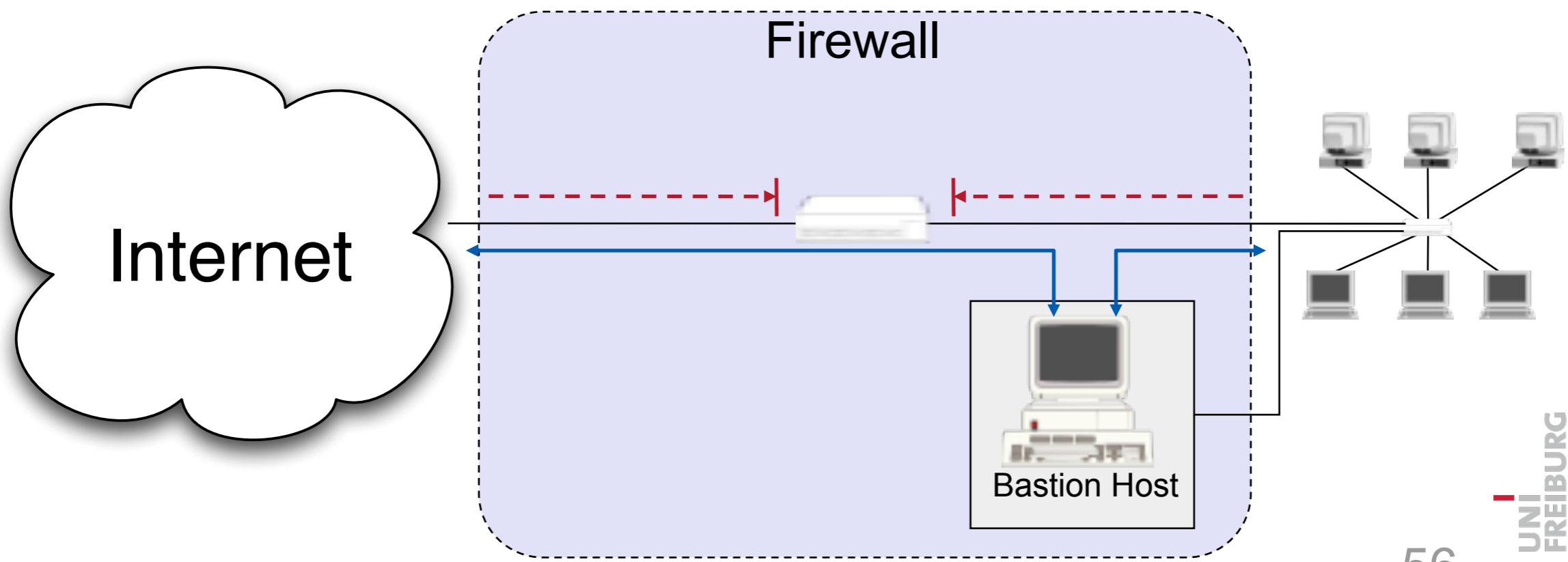
Firewall-Architektur Einfacher Paketfilter

- Realisiert durch
 - Eine Standard-Workstation (e.g. Linux PC) mit zwei Netzwerk-Interfaces und Filter-Software oder
 - Spezielles Router-Gerät mit Filterfähigkeiten



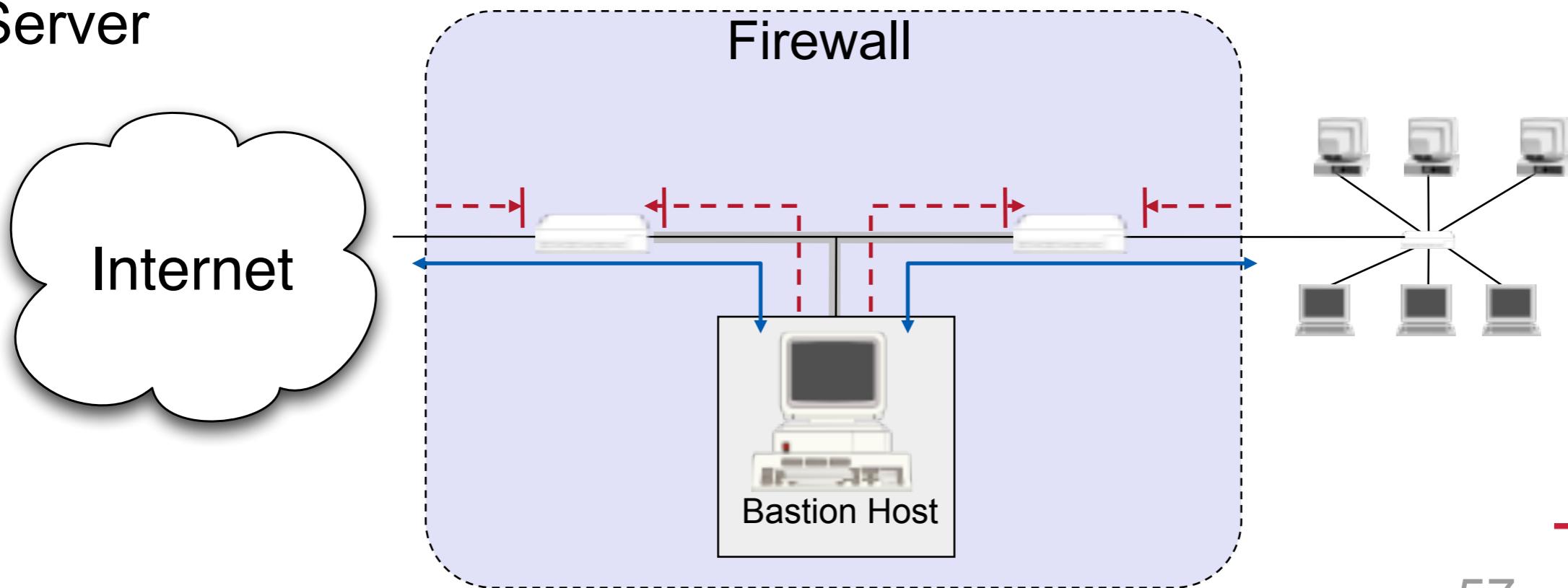
Firewall-Architektur Screened Host

- Screened Host
- Der Paketfilter
 - erlaubt nur Verkehr zwischen Internet und dem Bastion Host und
 - Bastion Host und geschützten Netzwerk
- Der Screened Host bietet sich als Proxy an
 - Der Proxy Host hat die Fähigkeiten selbst Angriffe abzuwehren



Firewall-Architektur Screened Subnet

- Perimeter network zwischen Paketfiltern
- Der innere Paketfilter schützt das innere Netzwerk, falls das Perimeter-Network in Schwierigkeiten kommt
 - Ein gehackter Bastion Host kann so das Netzwerk nicht ausspionieren
- Perimeter Netzwerke sind besonders geeignet für die Bereitstellung öffentlicher Dienste, z.B. FTP, oder WWW-Server

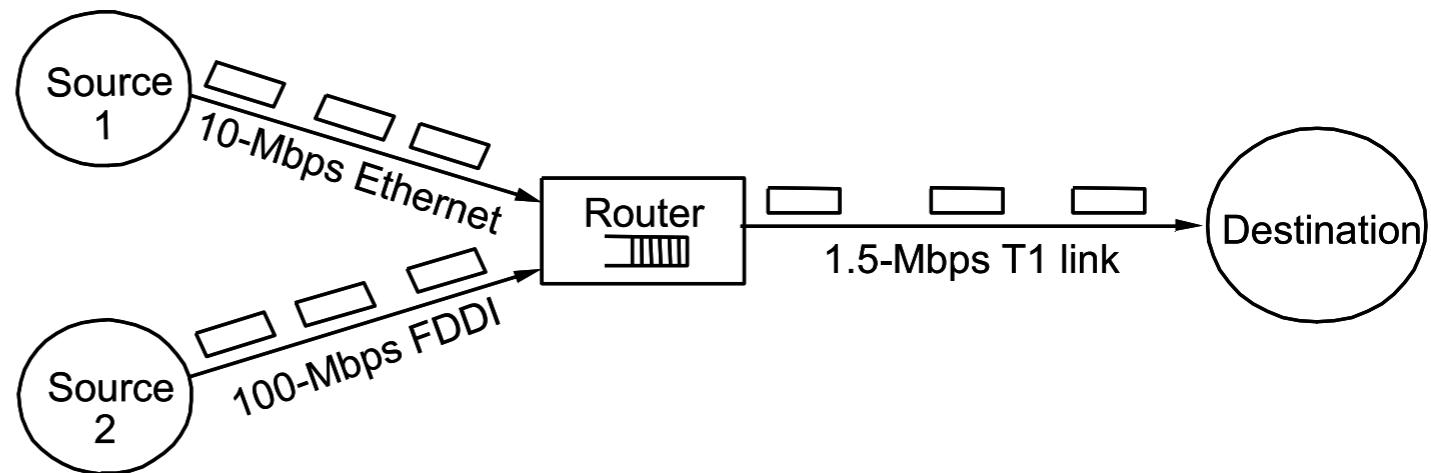


Firewall und Paketfilter

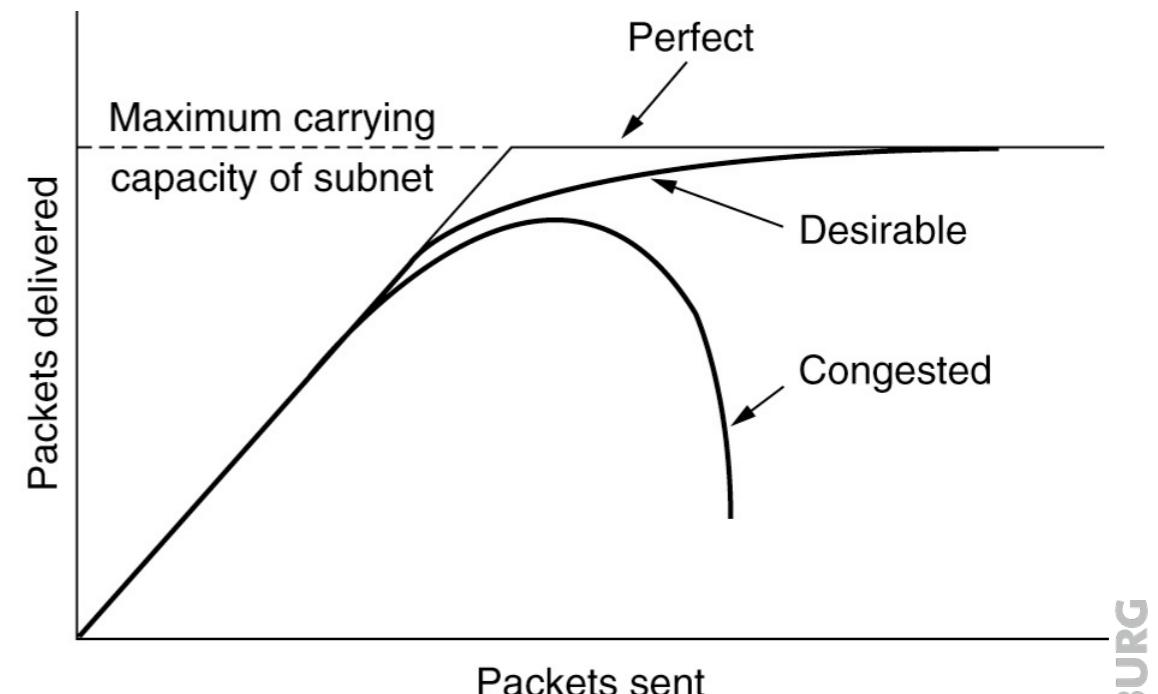
- Fähigkeiten von Paketfilter
 - Erkennung von Typ möglich (Demultiplexing-Information)
- Verkehrskontrolle durch
 - Source IP Address
 - Destination IP Address
 - Transport protocol
 - Source/destination application port
- Grenzen von Paketfiltern (und Firewalls)
 - Tunnel-Algorithmen sind aber mitunter nicht erkennbar
 - Möglich ist aber auch Eindringen über andere Verbindungen
 - z.B. Laptops, UMTS, GSM, Memory Sticks

Congestion Control Stauvermeidung

- Jedes Netzwerk hat eine eingeschränkte Übertragungs-Bandbreite



- Wenn mehr Daten in das Netzwerk eingeleitet werden, führt das zum
 - Datenstau (congestion) oder gar
 - Netzwerkzusammenbruch (congestive collapse)
- Folge: Datenpakete werden nicht ausgeliefert



Schneeballeffekt

- Congestion control soll Schneeballeffekte vermeiden
 - Netzwerküberlast führt zu Paketverlust (Pufferüberlauf, ...)
 - Paketverlust führt zu Neuversand
 - Neuversand erhöht Netzwerklast
 - Höherer Paketverlust
 - Mehr neu versandte Pakete
 - ...

Anforderungen an Congestion Control

- Effizienz
 - Verzögerung klein
 - Durchsatz hoch
- Fairness
 - Jeder Fluss bekommt einen fairen Anteil
 - Priorisierung möglich
 - gemäß Anwendung
 - und Bedarf

Mittel der Stauvermeidung

- Erhöhung der Kapazität
 - Aktivierung weiterer Verbindungen, Router
 - Benötigt Zeit und in der Regel den Eingriff der Systemadministration
- Reservierung und Zugangskontrolle
 - Verhinderung neuen Verkehrs an der Kapazitätsgrenze
 - Typisch für (Virtual) Circuit Switching
- Verringerung und Steuerung der Last
 - (Dezentrale) Verringerung der angeforderten Last bestehender Verbindungen
 - Benötigt Feedback aus dem Netzwerk
 - Typisch für Packet Switching
 - wird in TCP verwendet

Orte und Maße

- Router- oder Host-orientiert
 - Messpunkt (wo wird der Stau bemerkt)
 - Steuerung (wo werden die Entscheidungen gefällt)
 - Aktion (wo werden Maßnahmen ergriffen)
- Fenster-basiert oder Raten-basiert
 - Rate: x Bytes pro Sekunde
 - Fenster: siehe Fenstermechanismen in der Sicherungsschicht
 - wird im Internet verwendet

Routeraktion: Paket löschen

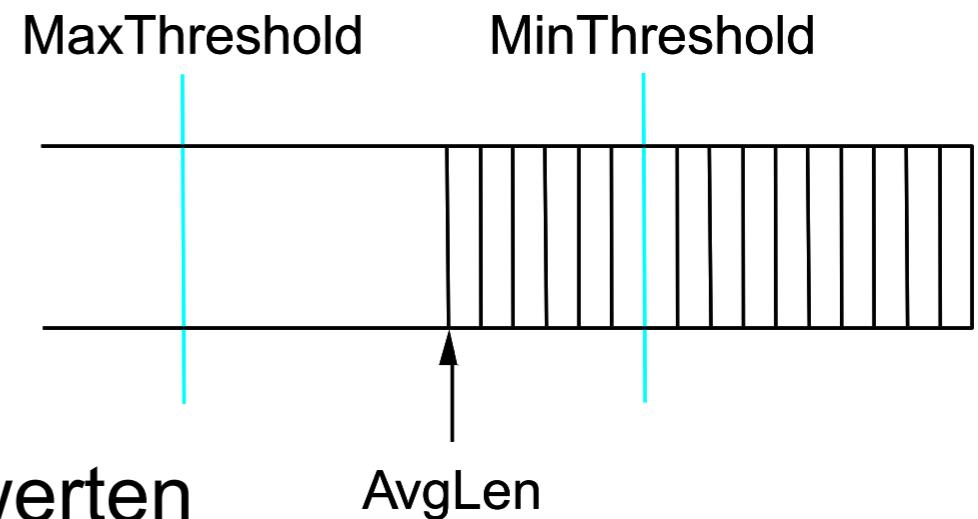
- Bei Pufferüberlauf im Router
 - muss (mindestens) ein Paket gelöscht werden
- Das zuletzt angekommene Paket löschen (*drop-tail queue*)
 - Intuition: “Alte” Pakete sind wichtiger als neue (Wein)
 - z.B. für go-back-n-Strategie
- Ein älteres Paket im Puffer löschen
 - Intuition: Für Multimedia-Verkehr sind neue Pakete wichtiger als alte (Milch)

Paketverlust erzeugt implizites Feedback

- Paketverlust durch Pufferüberlauf im Router erzeugt Feedback in der Transportschicht beim Sender durch ausstehende Bestätigungen
 - Internet
- Annahme:
 - Paketverlust wird hauptsächlich durch Stau ausgelöst
- Maßnahme:
 - Transport-Protokoll passt Senderate an die neue Situation an

Proaktive Methoden

- Pufferüberlauf deutet auf Netzwerküberlast hin
- Idee: Proaktives Feedback = Stauvermeidung (Congestion avoidance)



- Aktion bereits bei kritischen Anzeigewerten
- z.B. bei Überschreitung einer Puffergröße
- z.B. wenn kontinuierlich mehr Verkehr eingeht als ausgeliefert werden kann
- ...
- Router ist dann in einem Warn-Zustand

Proactive Aktion: Pakete drosseln (Choke packets)

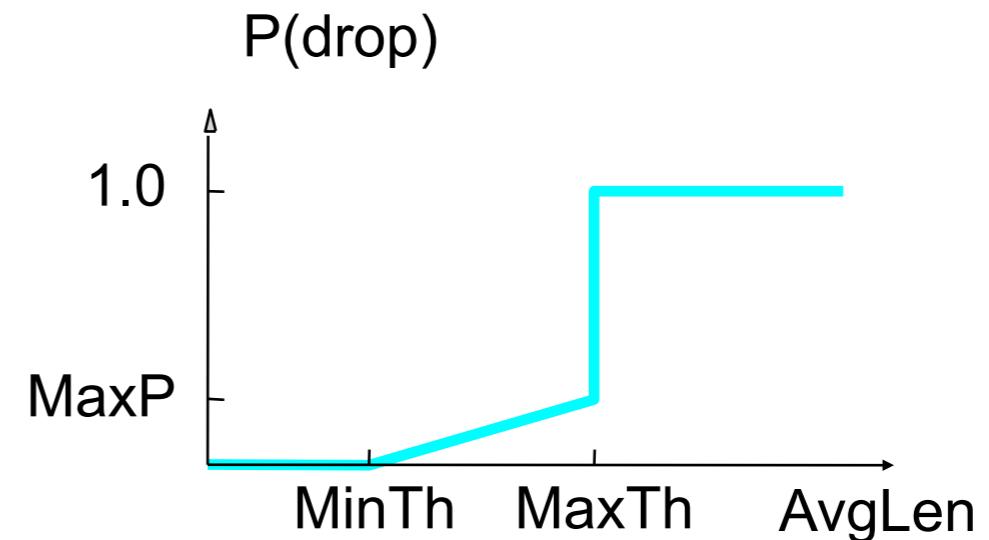
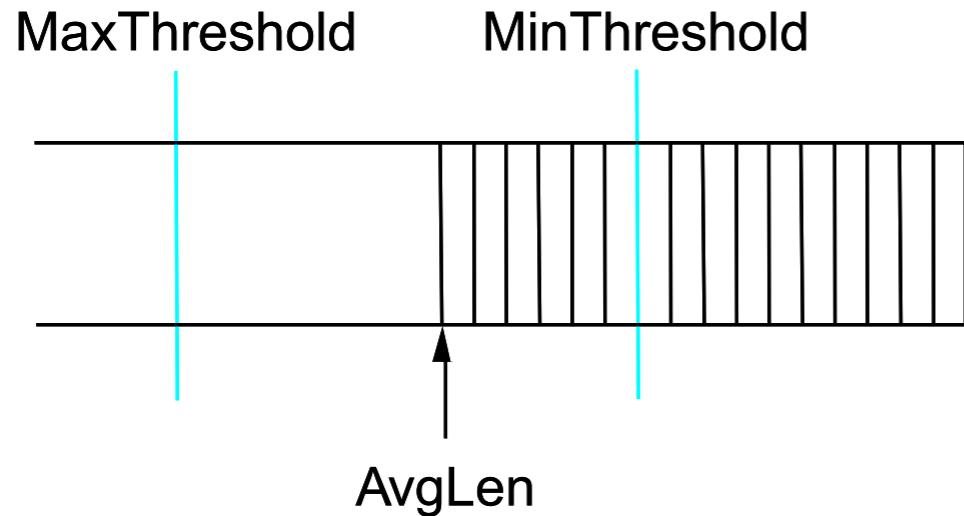
- Wenn der Router in dem Warnzustand ist:
 - Sendet er Choke-Pakete (Drossel-Pakete) zum Sender
- Choke-Pakete fordern den Sender auf die Sende-Rate zu verringern
- Problem:
 - Im kritischen Zustand werden noch mehr Pakete erzeugt
 - Bis zur Reaktion beim Sender vergrößert sich das Problem

Proaktive Aktion: Warnbits

- Wenn der Router in dem Warnzustand ist:
 - Sendet er Warn-Bits in allen Paketen zum Ziel-Host
- Ziel-Host sendet diese Warn-Bits in den Bestätigungs-Bits zurück zum Sender
 - Quelle erhält Warnung und reduziert Sende-Rate

Proaktive Aktion: Random early detection (RED)

- Verlorene Pakete werden als Indiz aufgefasst
- Router löschen Pakete willkürlich im Warnzustand
- Löschrate kann mit der Puffergröße steigen



Reaktion des Senders

- Raten-basierte Protokolle
 - Reduzierung der Sende-Rate
 - Problem: Um wieviel?
- Fenster-basierte Protokolle:
 - Verringerung des Congestion-Fensters
 - z.B. mit AIMD (additive increase, multiplicative decrease)

Systeme II

4. Die Vermittlungsschicht

Christian Schindelhauer

Technische Fakultät

Rechnernetze und Telematik

Albert-Ludwigs-Universität Freiburg

Systeme II

5. Die Transportschicht

Christian Schindelhauer

Technische Fakultät

Rechnernetze und Telematik

Albert-Ludwigs-Universität Freiburg

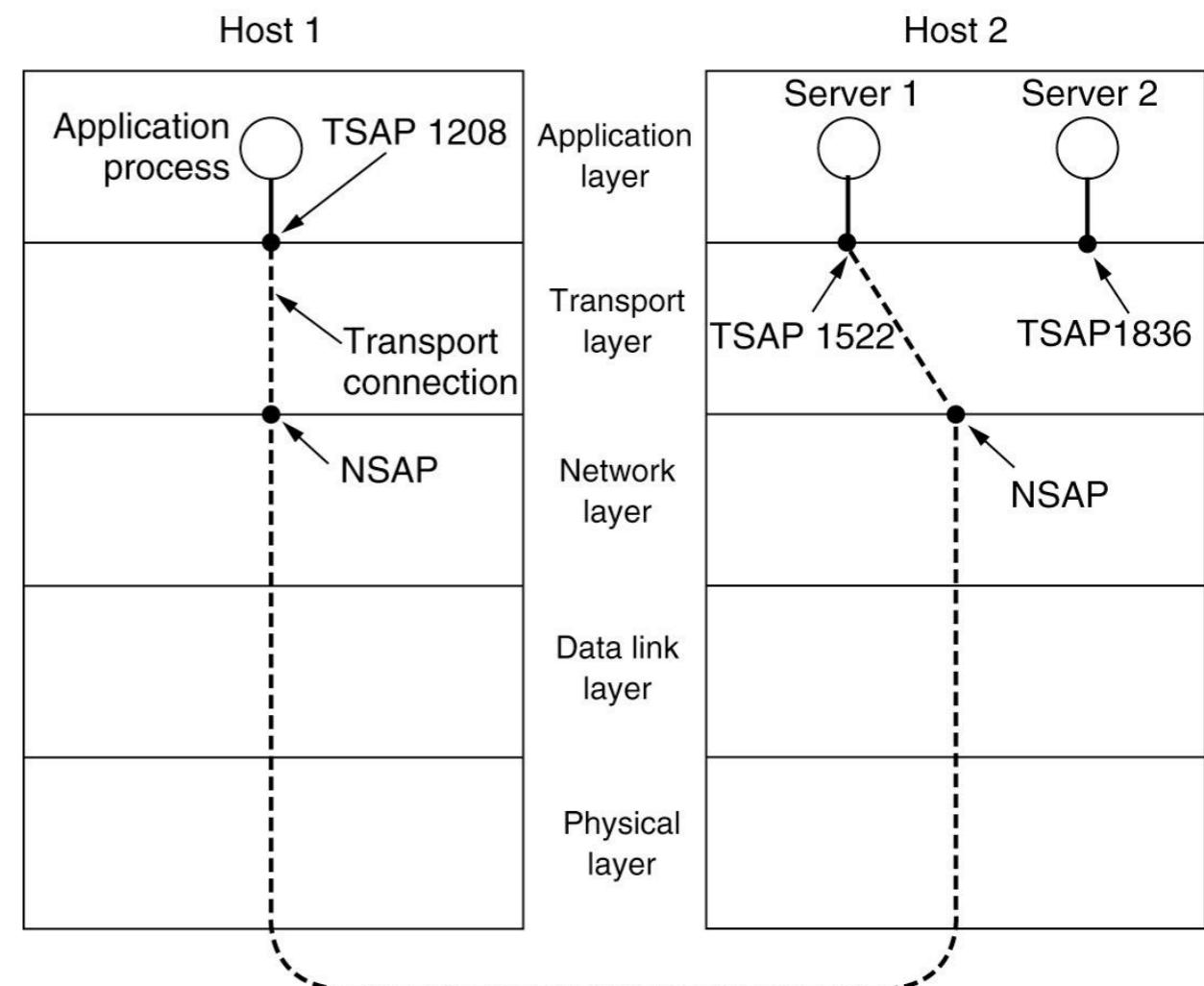
Version 26.06.2017

Dienste der Transportschicht

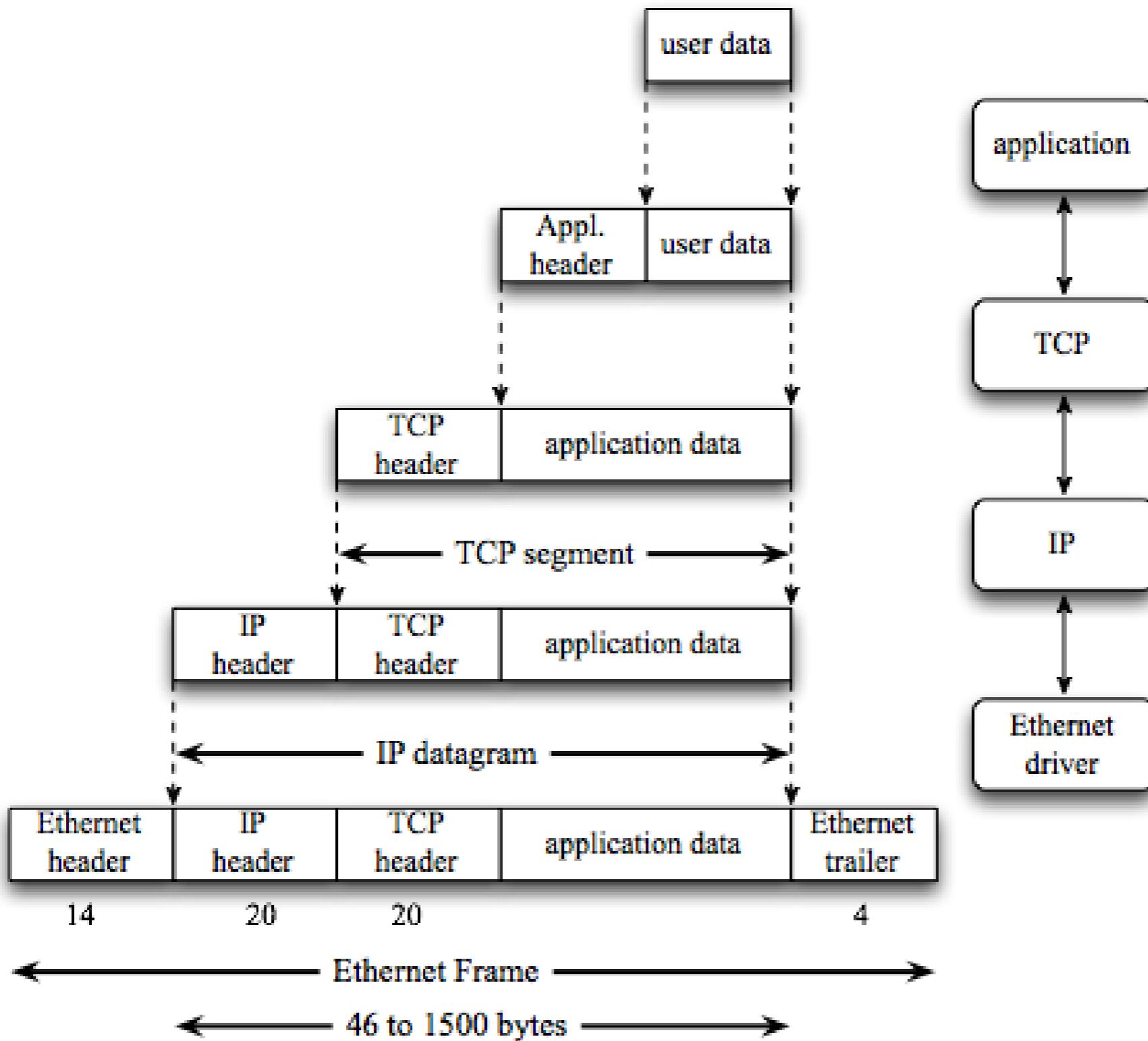
- Verbindungslos oder Verbindungsorientiert
 - Beachte: Sitzungsschicht im ISO/OSI-Protokoll
- Zuverlässig oder unzuverlässig
 - Best effort oder Quality of Service
 - Fehlerkontrolle
- Mit oder ohne Congestion Control
- Möglichkeit verschiedener Punkt-zu-Punktverbindungen
 - Stichwort: Demultiplexen
- Interaktionsmodelle
 - Byte-Strom, Nachrichten, „Remote Procedure Call“

Multiplex in der Transportschicht

- Die Netzwerkschicht leitet Daten an die Transportschicht unkontrolliert weiter
- Die Transportschicht muss sie den verschiedenen Anwendungen zuordnen:
 - z.B. Web, Mail, FTP, ssh, ...
 - In TCP/UDP durch Port-Nummern
 - z.B. Port 80 für Web-Server

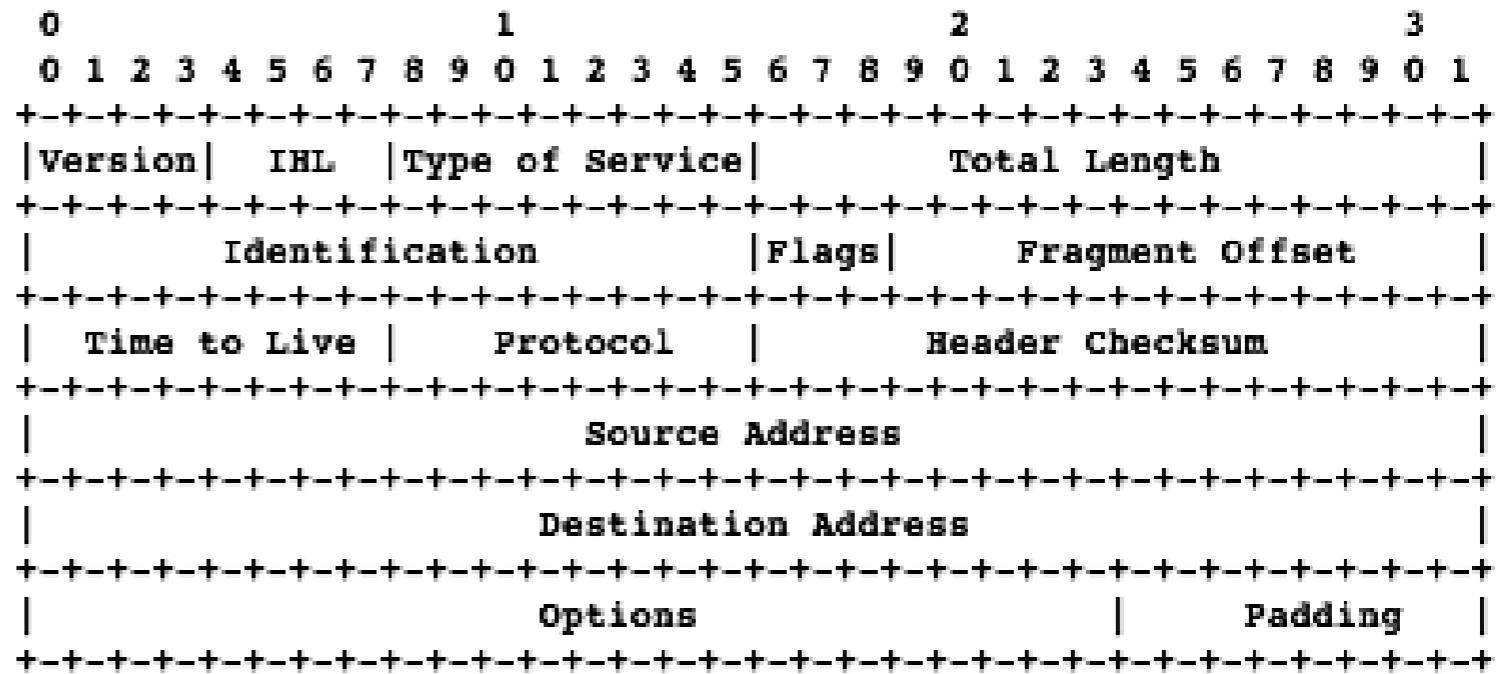


Datenkapselung

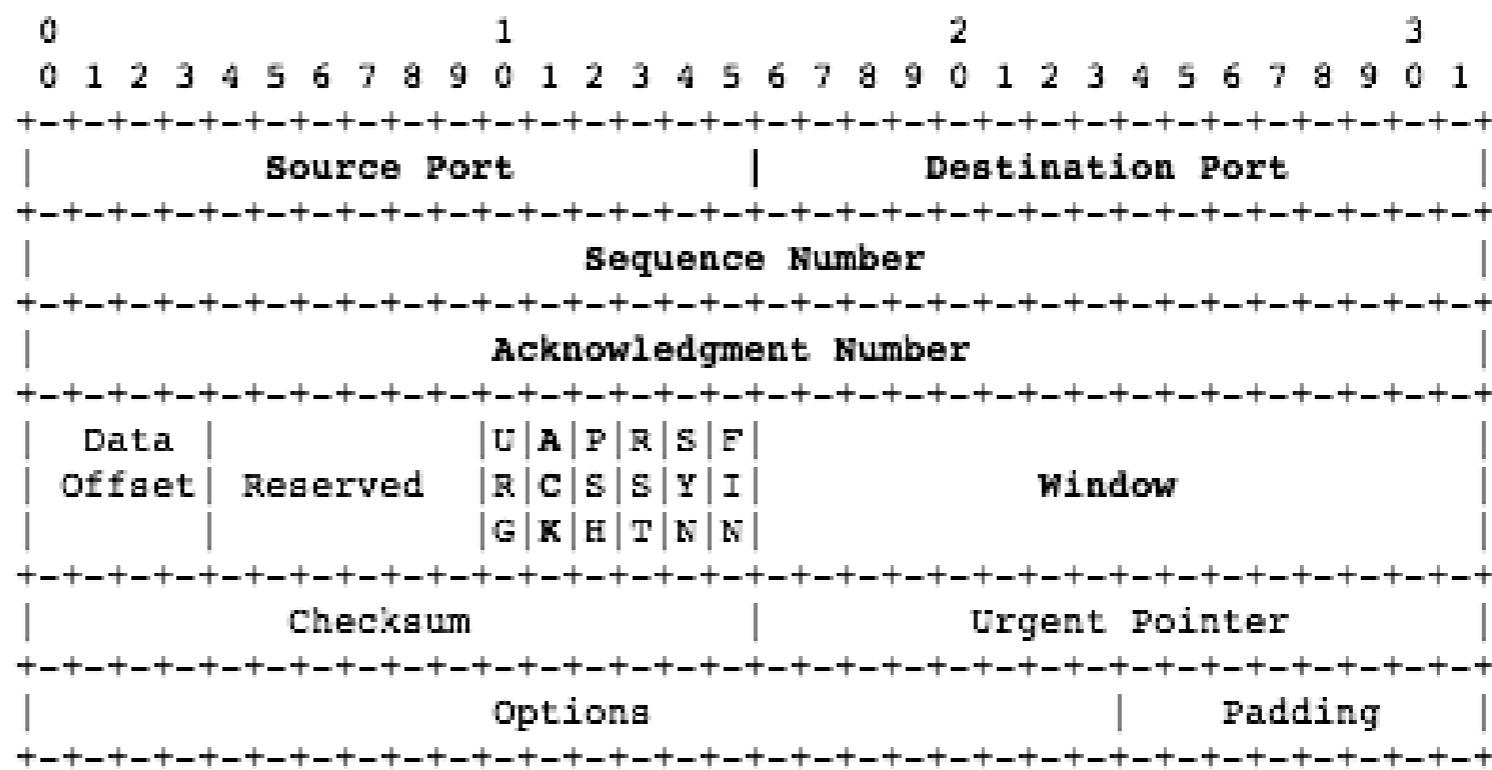


IP-Header (RFC 791)

- Version: 4 = IPv4
- IHL: Headerlänge
 - in 32 Bit-Wörter (>5)
- Type of Service
 - Optimiere delay, throughput, reliability, monetary cost
- Checksum (nur für IP-Header)
- Source and destination IP-address
- Protocol, identifiziert passendes Protokoll
 - Z.B. TCP, UDP, ICMP, IGMP
- Time to Live:
 - maximale Anzahl Hops



TCP-Header



Transportschicht (transport layer)

- TCP (transmission control protocol)
 - Erzeugt zuverlässigen Datenfluß zwischen zwei Rechnern
 - Unterteilt Datenströme aus Anwendungsschicht in Pakete
 - Gegenseite schickt Empfangsbestätigungen (Acknowledgments)
- UDP (user datagram protocol)
 - Einfacher unzuverlässiger Dienst zum Versand von einzelnen Päckchen
 - Wandelt Eingabe in ein Datagramm um
 - Anwendungsschicht bestimmt Paketgröße
- Versand durch Netzwerkschicht
- Kein Routing: End-to-End-Protokolle

- TCP ist ein verbindungsorientierter, zuverlässiger Dienst für bidirektionale Byteströme
- TCP ist verbindungsorientiert
 - Zwei Parteien identifiziert durch Socket: IP-Adresse und Port
(TCP-Verbindung eindeutig identifiziert durch Socketpaar)
 - Kein Broadcast oder Multicast
 - Verbindungsauftbau und Ende notwendig
 - Solange Verbindung nicht (ordentlich) beendet, ist Verbindung noch aktiv

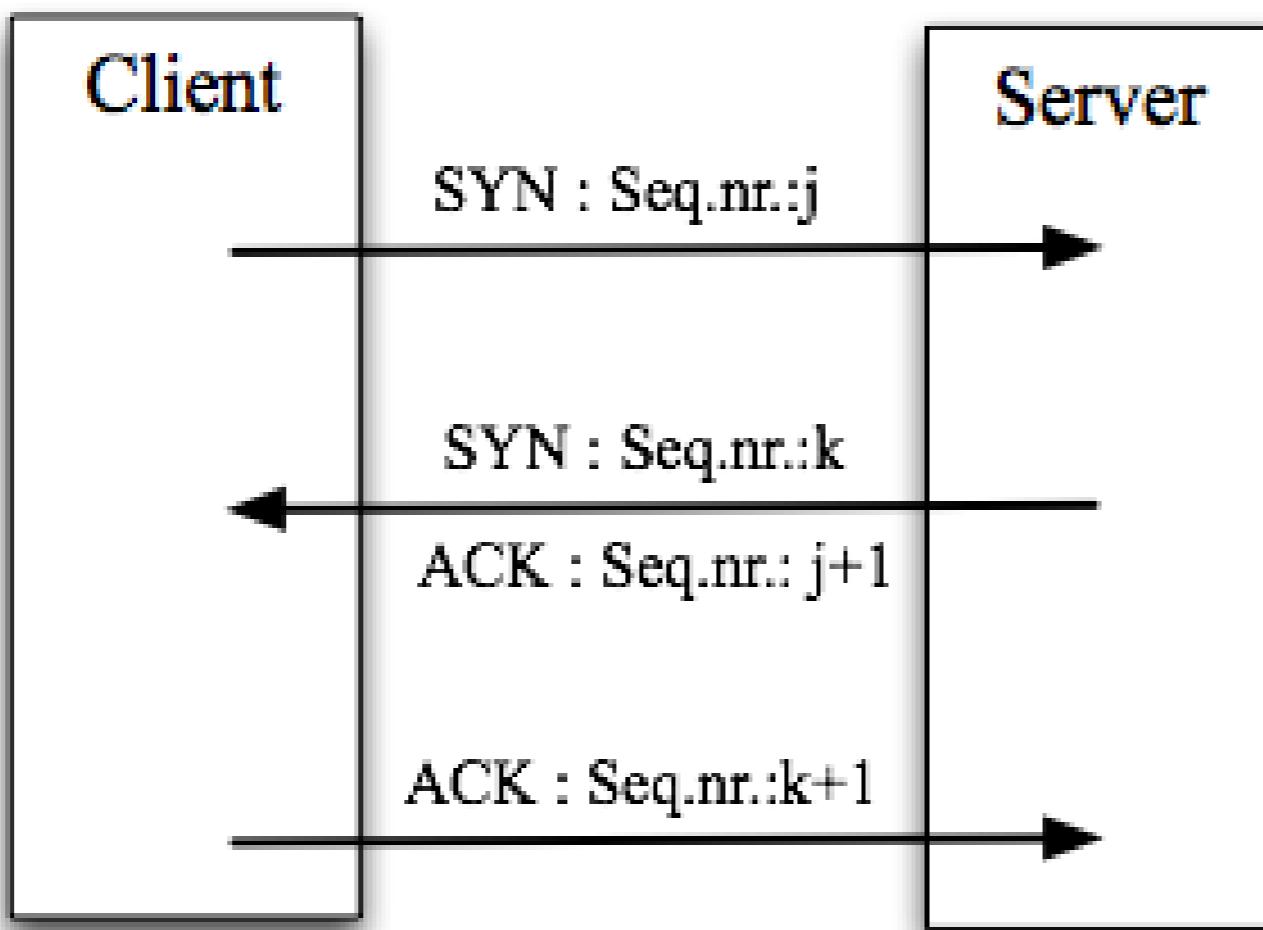
TCP (II)

- TCP ist ein verbindungsorientierter, zuverlässiger Dienst für bidirektionale Byteströme
- TCP ist zuverlässig
 - Jedes Datenpaket wird bestätigt (acknowledgment)
 - Erneutes Senden von unbestätigten Datenpakete
 - Checksum für TCP-Header und Daten
 - TCP nummeriert Pakete und sortiert beim Empfänger
 - Löscht duplizierte Pakete

- TCP ist ein verbindungsorientierter, zuverlässiger Dienst für bidirektionale Byteströme
- TCP ist ein Dienst für bidirektionale Byteströme
 - Daten sind zwei gegenläufige Folgen aus einzelnen Bytes (=8 Bits)
 - Inhalt wird nicht interpretiert
 - Zeitverhalten der Datenfolgen kann verändert werden
 - Versucht zeitnahe Auslieferung jedes einzelnen Datenbytes
 - Versucht Übertragungsmedium effizient zu nutzen
 - = wenig Pakete

TCP-Verbindungsauftbau

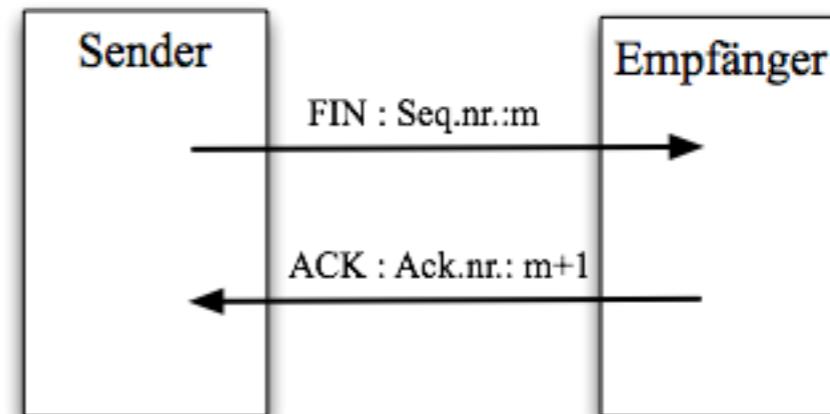
- In der Regel Client-Server-Verbindungen
 - Dann Aufbau mit drei TCP-Pakete (=Segmente)
 - Mit ersten SYN-Segment auch Übermittlung der MSS (maximum segment size)



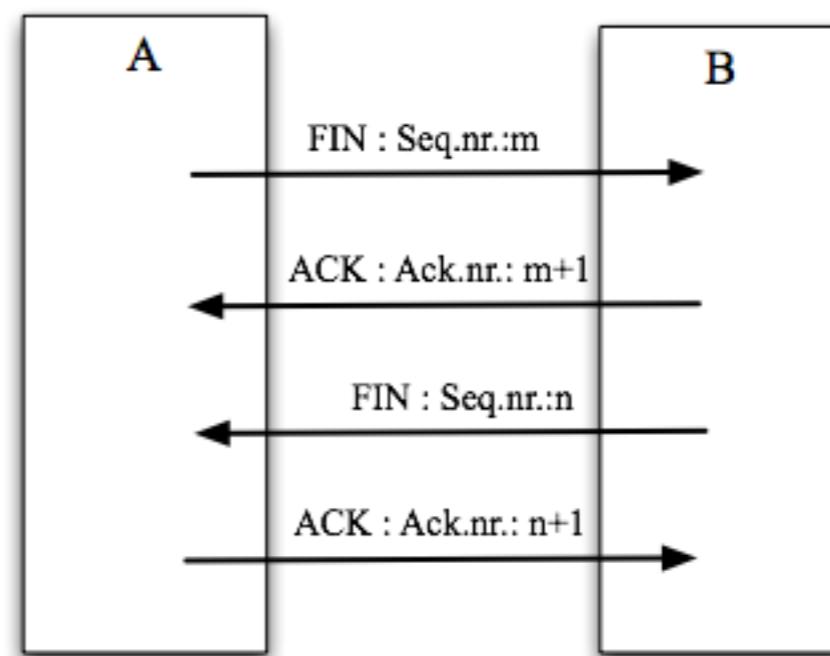
TCP-Verbindungssende

■ Half-Close

- Sender kündigt Ende mit FIN-Segment an und wartet auf Bestätigung
- In Gegenrichtung kann weitergesendet werden

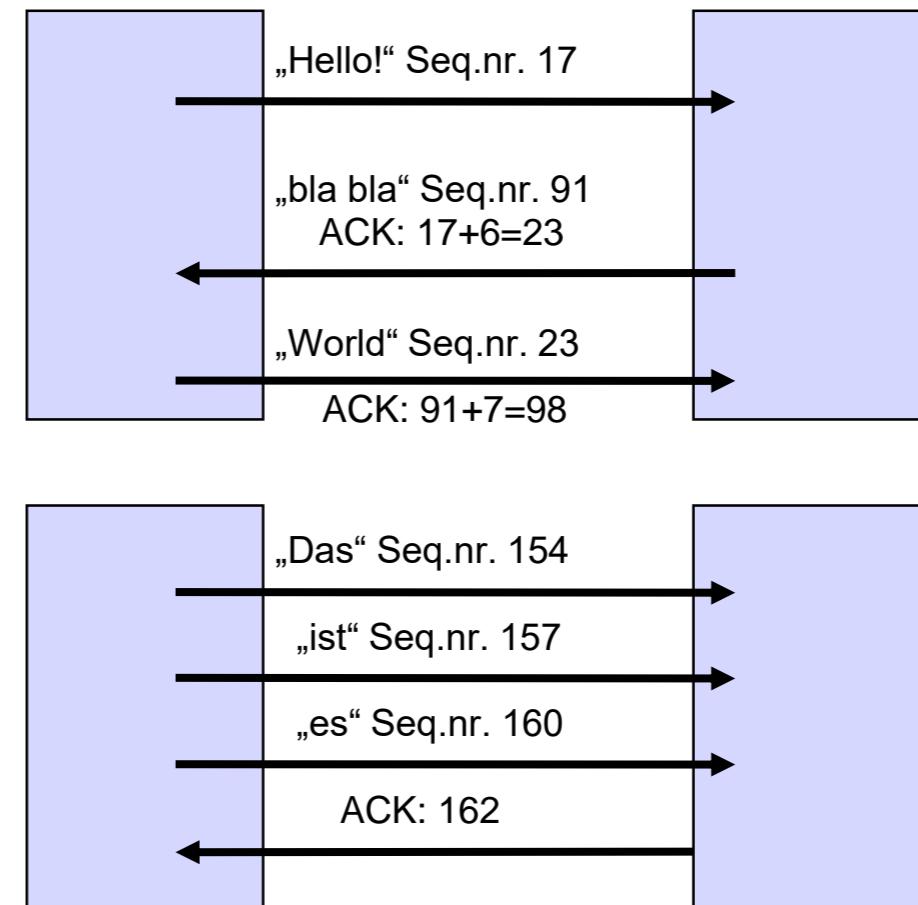


■ 2 Half-Close beenden TCP-Verbindung



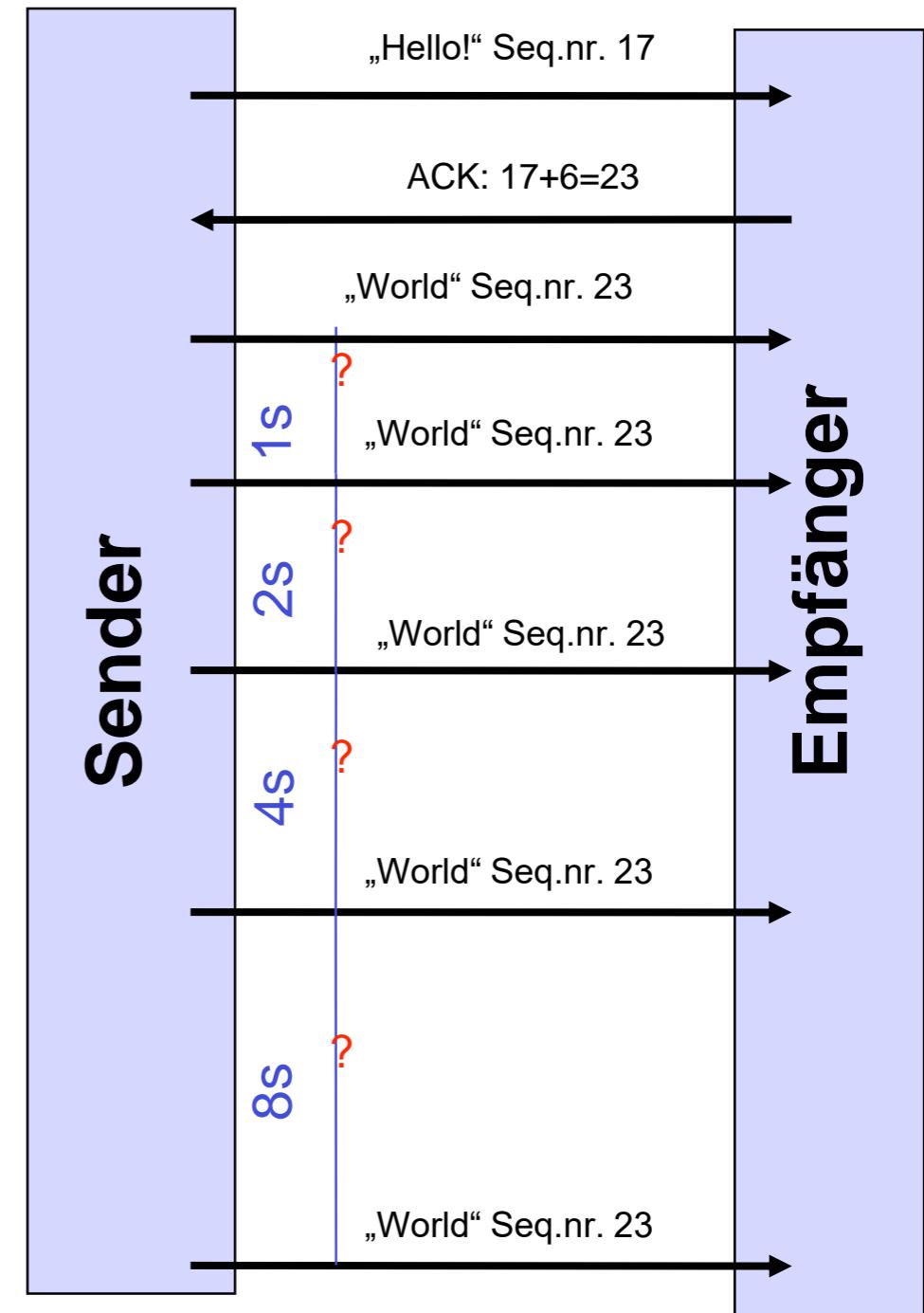
Bestätigungen

- Huckepack-Technik
 - Bestätigungen „reiten“ auf den Datenpaket der Gegenrichtung
- Eine Bestätigungssegment kann viele Segmente bestätigen
 - Liegen keine Daten an, werden Acks verzögert



Exponentielles Zurückweichen

- Retransmission Timeout (RTO)
 - regelt Zeitraum zwischen Senden von Datenduplikaten, falls Bestätigung ausbleibt
- Wann wird ein TCP-Paket nicht bestätigt?
 - Wenn die Bestätigung wesentlich länger benötigt, als die durchschnittliche Umlaufzeit (RTT/round trip time)
 - 1. Problem: Messung der RTT
 - 2. Problem: Bestätigung kommt, nur spät
 - Sender
 - Wartet Zeitraum gemäß RTO
 - Sendet Paket nochmal und setzt
 - $RTO \leftarrow 2 \text{ RTO}$ (bis $RTO = 64$ Sek.)
- Neuberechnung von RTO, wenn Pakete bestätigt werden



Schätzung der Umlaufzeit (RTT/Round Trip Time)

- TCP-Paket gilt als nicht bestätigt, wenn Bestätigung „wesentlich“ länger dauert als RTO
 - RTT nicht on-line berechenbar (nur rückblickend)
 - RTT schwankt stark
- Daher: Retransmission Timeout Value aus großzügiger Schätzung:
 - RFC 793: ($M :=$ letzte gemessene RTT)
 - $R \leftarrow \alpha R + (1 - \alpha) M,$ wobei $\alpha = 0,9$
 - $RTO \leftarrow \beta R,$ wobei $\beta = 2$
 - Jacobson 88: Schätzung nicht robust genug, daher
 - $A \leftarrow A + g (M - A),$ wobei $g = 1/8$
 - $D \leftarrow D + h (|M - A| - D),$ wobei $h = 1/4$
 - $RTO \leftarrow A + 4D$
- Aktualisierung nicht bei mehrfach versandten Pakete

TCP - Algorithmus von Nagle

- Wie kann man sicherstellen,
 - dass kleine Pakete zeitnah ausgeliefert werden
 - und bei vielen Daten große Pakete bevorzugt werden?
- Algorithmus von Nagle:
 - Kleine Pakete werden nicht versendet, solange Bestätigungen noch ausstehen.
 - Paket ist klein, wenn Datenlänge < MSS
 - Trifft die Bestätigung des zuvor gesendeten Pakets ein, so wird das nächste verschickt.
- Beispiel:
 - Telnet versus ftp
- Eigenschaften
 - Selbst-taktend: Schnelle Verbindung = viele kleine Pakete

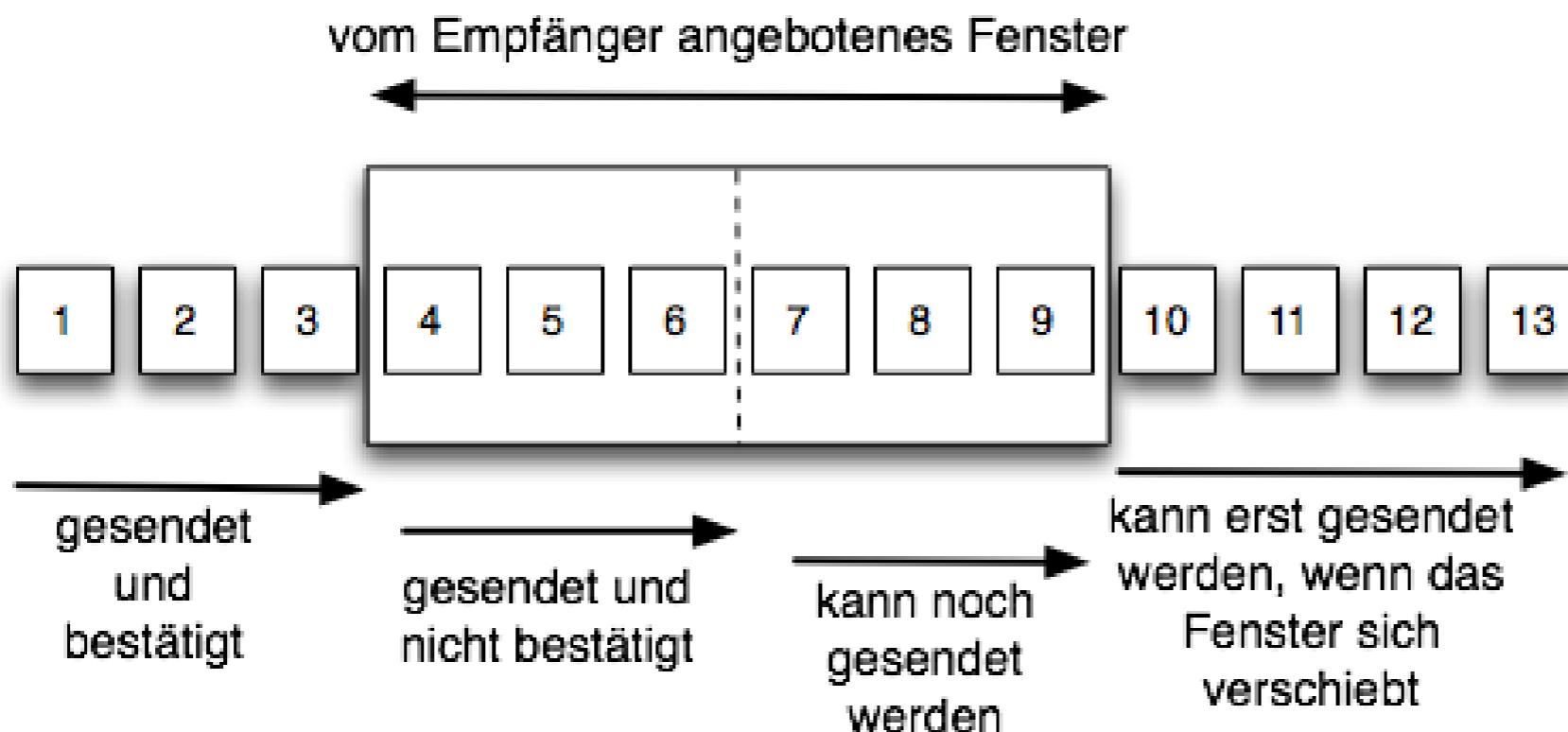
Flusskontrolle

- Problem: Schneller Sender und langsamer Empfänger
 - Der Sender lässt den Empfangspuffer des Empfängers überlaufen
 - Übertragungsbandweite wird durch sinnlosen Mehrfachversand (nach Fehlerkontrolle) verschwendet
- Anpassung der Frame-Sende-Rate an dem Empfänger notwendig



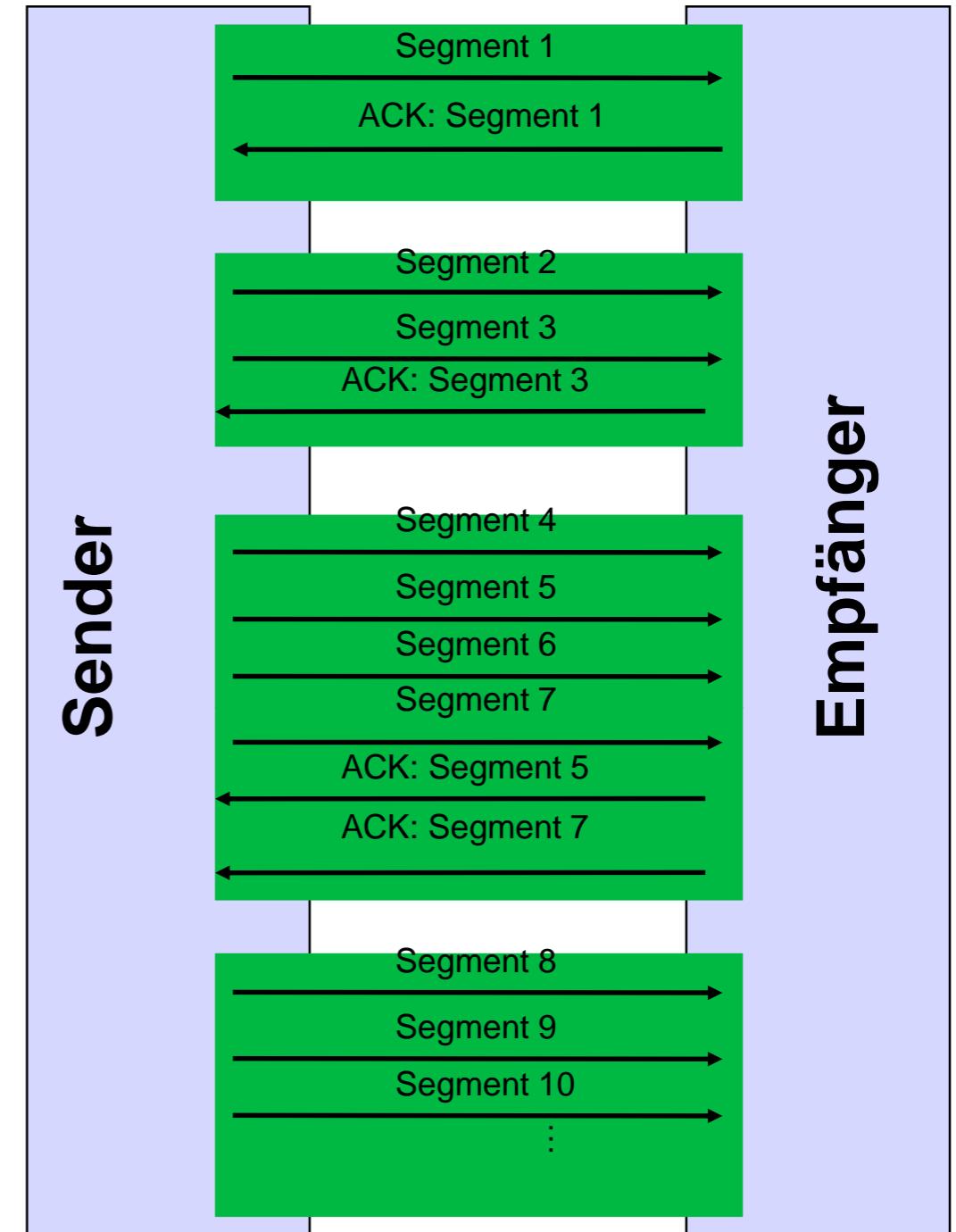
Gleitende Fenster (sliding windows)

- Datenratenanpassung durch Fenster
 - Empfänger bestimmt Fenstergröße (wnd) im TCP-Header der ACK-Segmente
 - Ist Empfangspuffer des Empfängers voll, sendet er wnd=0
 - Andernfalls sendet Empfänger wnd>0
- Sender beachtet:
 - Anzahl unbestätigter gesender Daten \leq Fenstergröße



Slow Start Congestion Fenster

- Sender darf vom Empfänger angebotene Fenstergröße nicht von Anfang wahrnehmen
- 2. Fenster: Congestion-Fenster (cwnd/Congestion window)
 - Von Sender gewählt (FSK)
 - Sendefenster: $\min \{ \text{wnd}, \text{cwnd} \}$
 - S: Segmentgröße
 - Am Anfang:
 - $\text{cwnd} \leftarrow S$
 - Für jede empfangene Bestätigung:
 - $\text{cwnd} \leftarrow \text{cwnd} + S$
 - Solange bis einmal Bestätigung ausbleibt
- „Slow Start“ = Exponentielles Wachstum



TCP Tahoe: Congestion Avoidance

- Jacobson 88:
 - Parameter: cwnd und Slow-Start-Schwellwert (ssthresh=slow start threshold)
 - S = Datensegmentgröße = maximale Segmentgröße
- Verbindungsauftbau:
 - $cwnd \leftarrow S$ $ssthresh \leftarrow 65535$
- Bei Paketverlust, d.h. Bestätigungszeit > RTO,
 - multiplicatively decreasing
 - $cwnd \leftarrow S$ $ssthresh \leftarrow \max \left\{ 2S, \frac{1}{2} \min \{ cwnd, wnd \} \right\}$
- Werden Segmente bestätigt und $cwnd \leq ssthresh$, dann
 - slow start: $cwnd \leftarrow cwnd + S$
- Werden Segmente bestätigt und $cwnd > ssthresh$, dann additively increasing

x: Anzahl Pakete pro RTT

x $\leftarrow 1$

y $\leftarrow \text{max}$

x $\leftarrow 1$

y $\leftarrow x/2$

x $\leftarrow 2 \oplus x$, bis x = y

$$cwnd \leftarrow cwnd + S - \frac{S}{cwnd}$$

x $\leftarrow x + 1$

TCP Tahoe

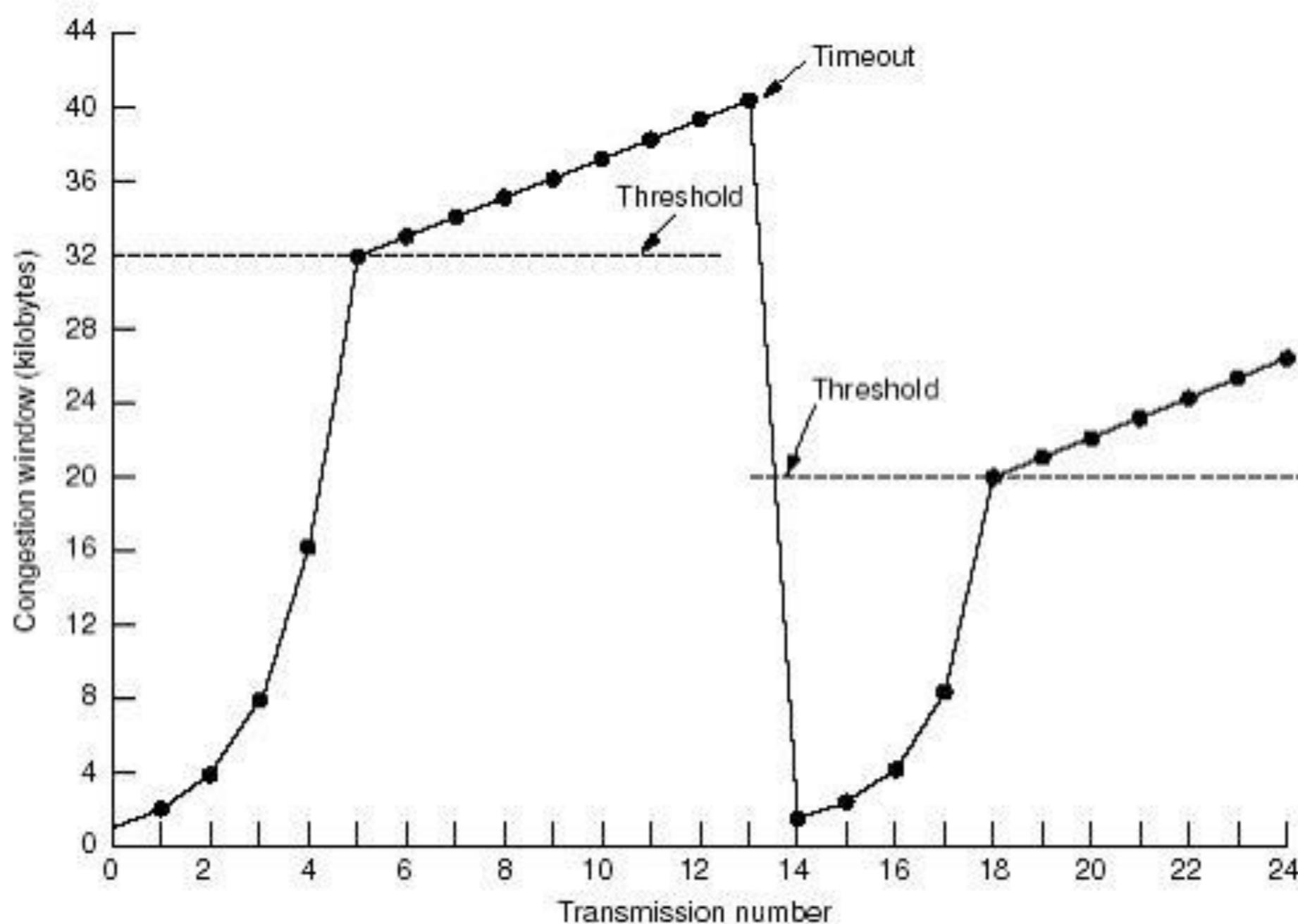


Fig3

pictures from TANENBAUM A. S. Computer Networks 3rd edition

Fast Retransmit und Fast Recovery

- TCP Tahoe [Jacobson 1988]:
 - Geht nur ein Paket verloren, dann
 - Wiederversand Paket + Restfenster
 - Und gleichzeitig Slow Start
 - Fast retransmit
 - Nach drei Bestätigungen desselben Pakets (triple duplicate ACK),
 - sende Paket nochmal, starte mit Slow Start
- TCP Reno [Stevens 1994]
 - Nach Fast retransmit:
 - $ssthresh \leftarrow \min(wnd, cwnd)/2$
 - $cwnd \leftarrow ssthresh + 3 S$
 - Fast recovery nach Fast retransmit
 - Erhöhe Paketrate mit jeder weiteren Bestätigung
 - $cwnd \leftarrow cwnd + S$
 - Congestion avoidance: Trifft Bestätigung von $P+x$ ein:
 - $cwnd \leftarrow ssthresh$

$$\begin{array}{|c|}\hline y \leftarrow x/2 \\ \hline x \leftarrow y + 3 \\ \hline\end{array}$$

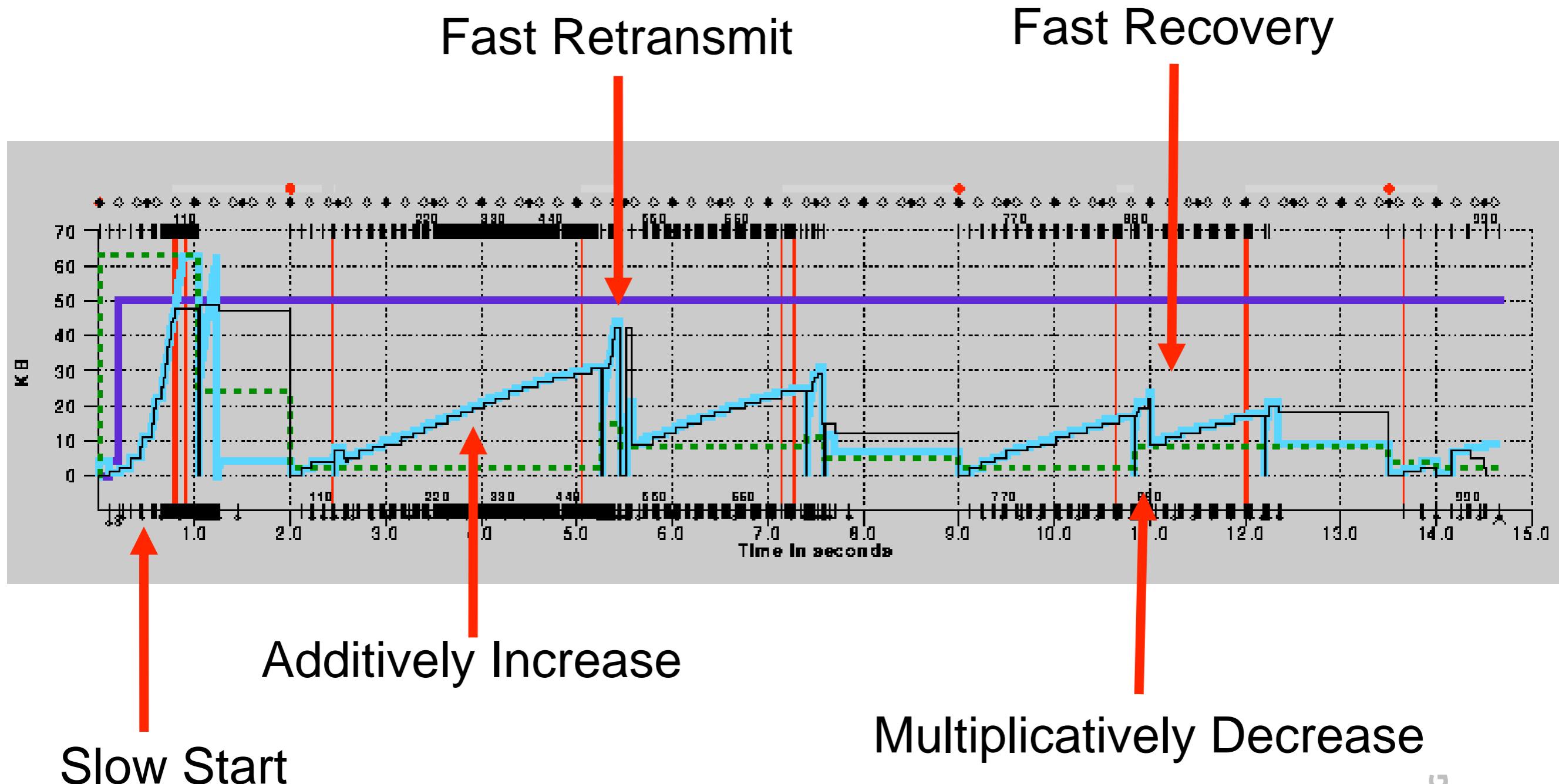
Stauvermeidungsprinzip: AIMD

- Kombination von TCP und Fast Recovery verhält sich im wesentlichen wie folgt:

$$x \leftarrow 1$$

- Verbindungsaufbau:
- Bei Paketverlust, MD:m $x \leftarrow x/2$ decreasing
- Werden Segmente best. $x \leftarrow x + 1$ additive increasing

Beispiel: TCP Reno in Aktion



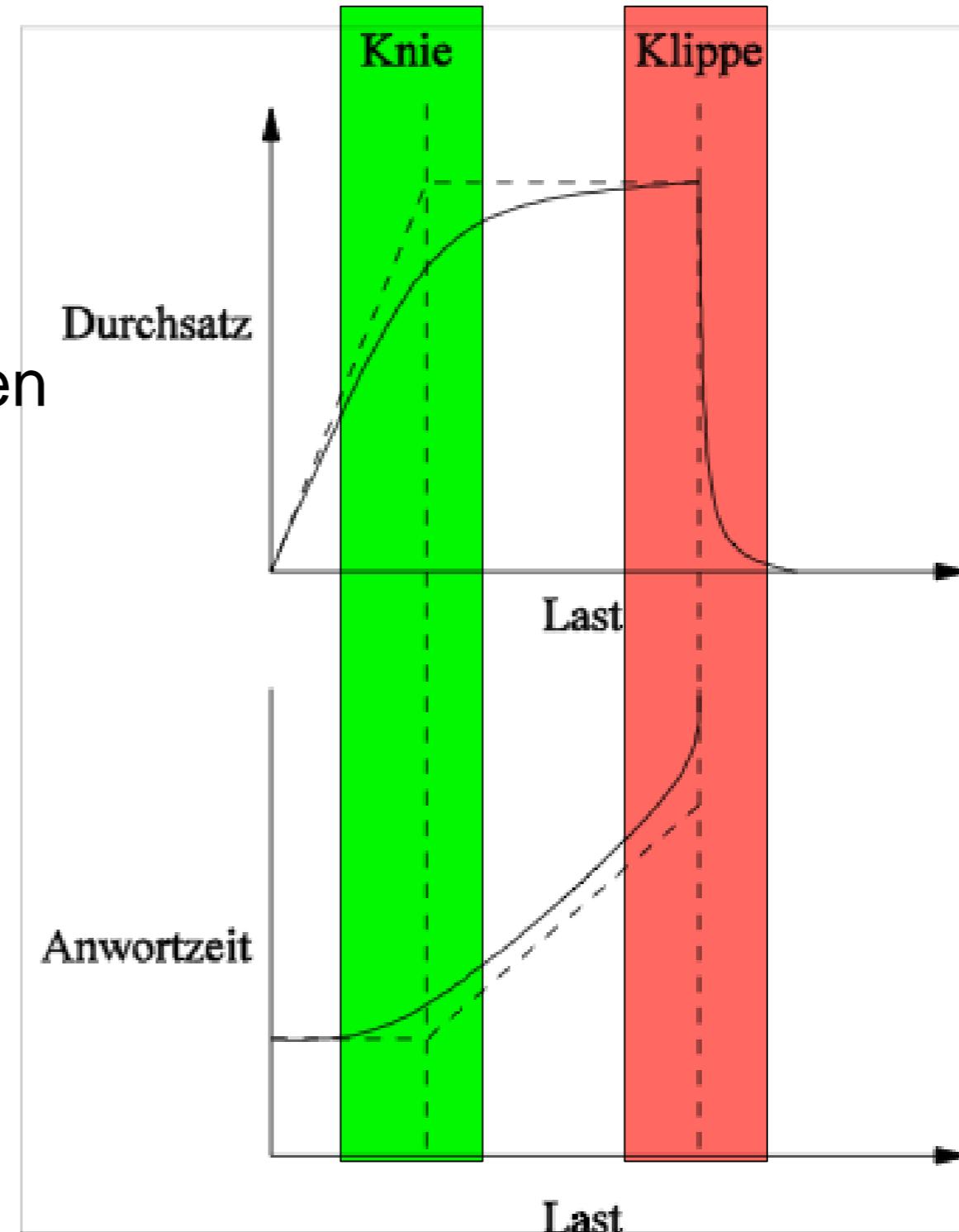
Durchsatz und Antwortzeit

- **Klippe:**

- Hohe Last
- Geringer Durchsatz
- Praktisch alle Daten gehen verloren

- **Knie:**

- Hohe Last
- Hoher Durchsatz
- Einzelne Daten gehen verloren



Ein einfaches Datenratenmodell

- n Teilnehmer, Rundenmodell
 - Teilnehmer i hat Datenrate $x_i(t)$
 - Anfangsdatenrate $x_1(0), \dots, x_n(0)$ gegeben
- Feedback nach Runde t :
 - $y(t) = 0$, falls $\sum_{i=1}^n x_i(t) \leq K$
 - $y(t) = 1$, falls $\sum_{i=1}^n x_i(t) > K$
 - wobei K ist Kielast
- Jeder Teilnehmer aktualisiert in Runde $t+1$:
 - $x_i(t+1) = f(x_i(t), y(t))$
 - Increase-Strategie $f_0(x) = f(x, 0)$
 - Decrease-Strategie $f_1(x) = f(x, 1)$
- Wir betrachten lineare Funktionen:

$$f_0(x) = a_I + b_I x \quad \text{und} \quad f_1(x) = a_D + b_D x .$$

Lineare Datenratenanpassung

■ Interessante Spezialfälle:

- AIAD: Additive Increase
Additive Decrease

$$f_0(x) = a_I + x \quad \text{und} \quad f_1(x) = a_D + x ,$$

wobei $a_I > 0$ und $a_D < 0$.

- MIMD: Multiplicative
Increase/Multiplicative
Decrease

$$f_0(x) = b_I x \quad \text{und} \quad f_1(x) = b_D x ,$$

wobei $b_I > 1$ und $b_D < 1$.

- AIMD: Additive Increase
Multiplicative Decrease

$$f_0(x) = a_I + x \quad \text{und} \quad f_1(x) = b_D x ,$$

wobei $a_I > 0$ und $b_D < 1$.

Fairness und Effizienz

■ Effizienz

- Last:

$$X(t) := \sum_{i=1}^n x_i(t)$$

- Maß

$$|X(t) - K|$$

■ Fairness: Für $x=(x_1, \dots, x_n)$:

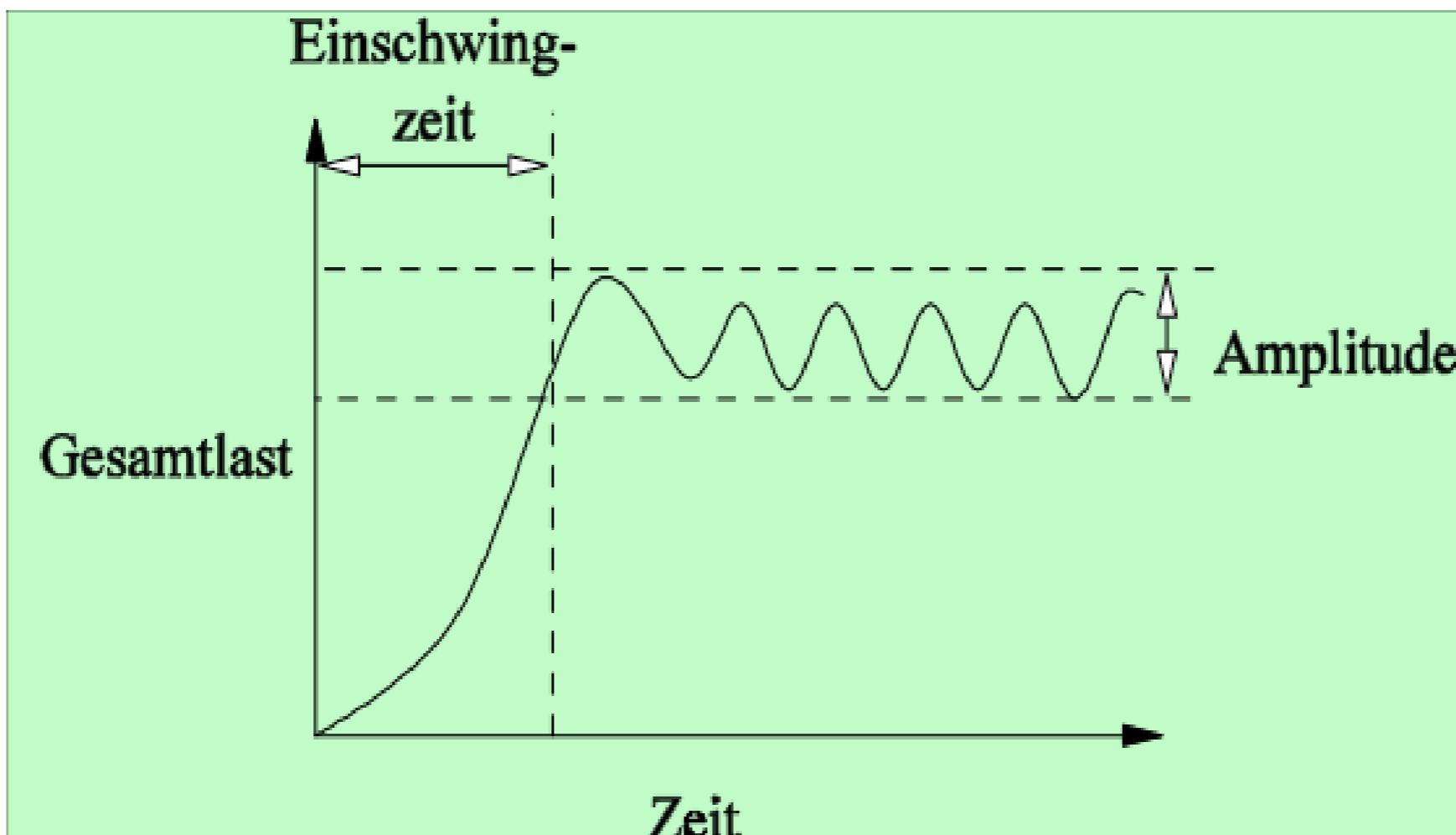
$$F(x) = \frac{(\sum_{i=1}^n x_i)^2}{n \sum_{i=1}^n (x_i)^2}.$$

- $1/n \leq F(x) \leq 1$
- $F(x) = 1 \leftrightarrow$ absolute Fairness
- Skalierungsunabhängig
- Kontinuierlich, stetig, differenzierbar
- Falls k von n fair, Rest 0, dann $F(x) = k/n$

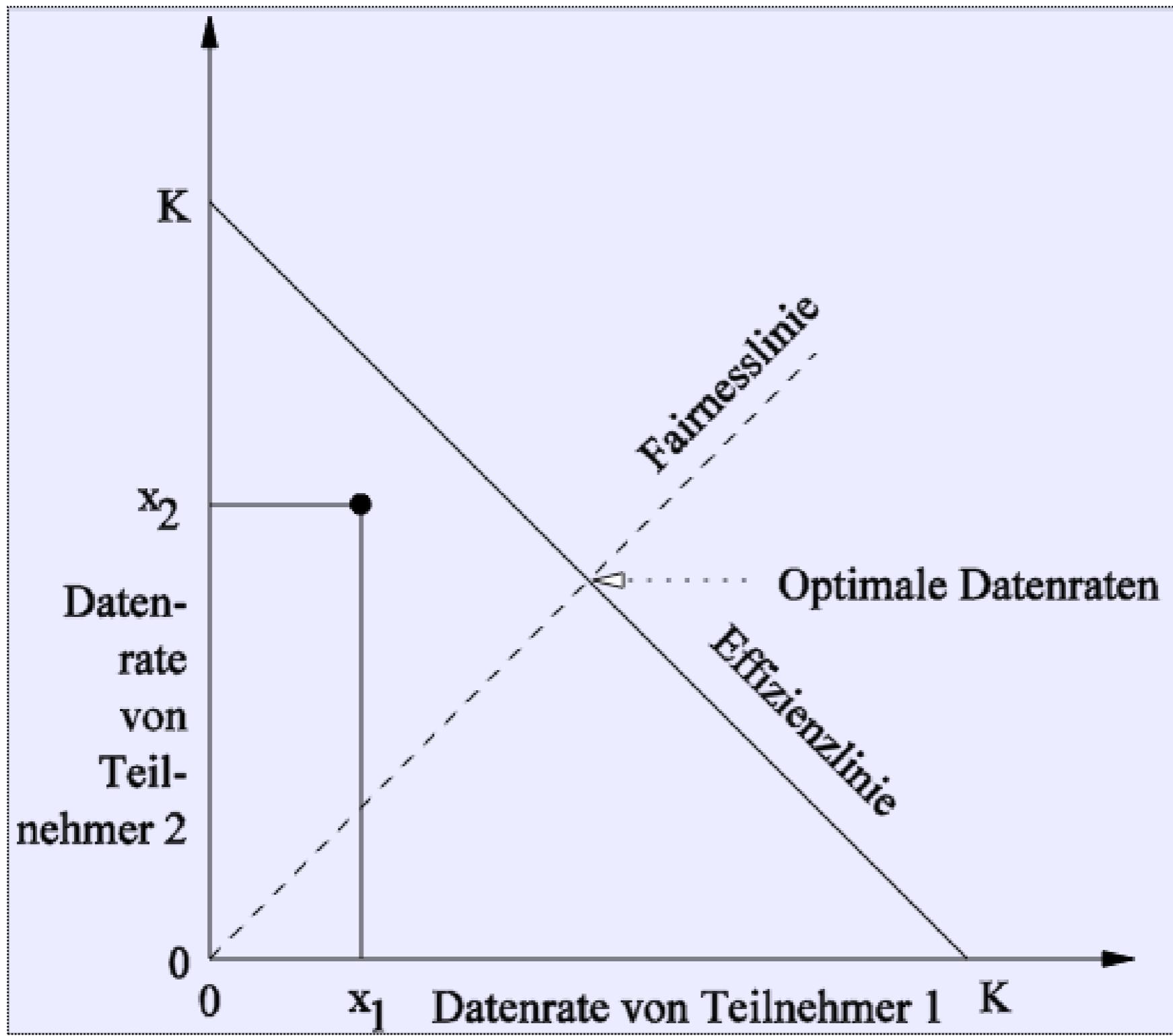
•

Konvergenz

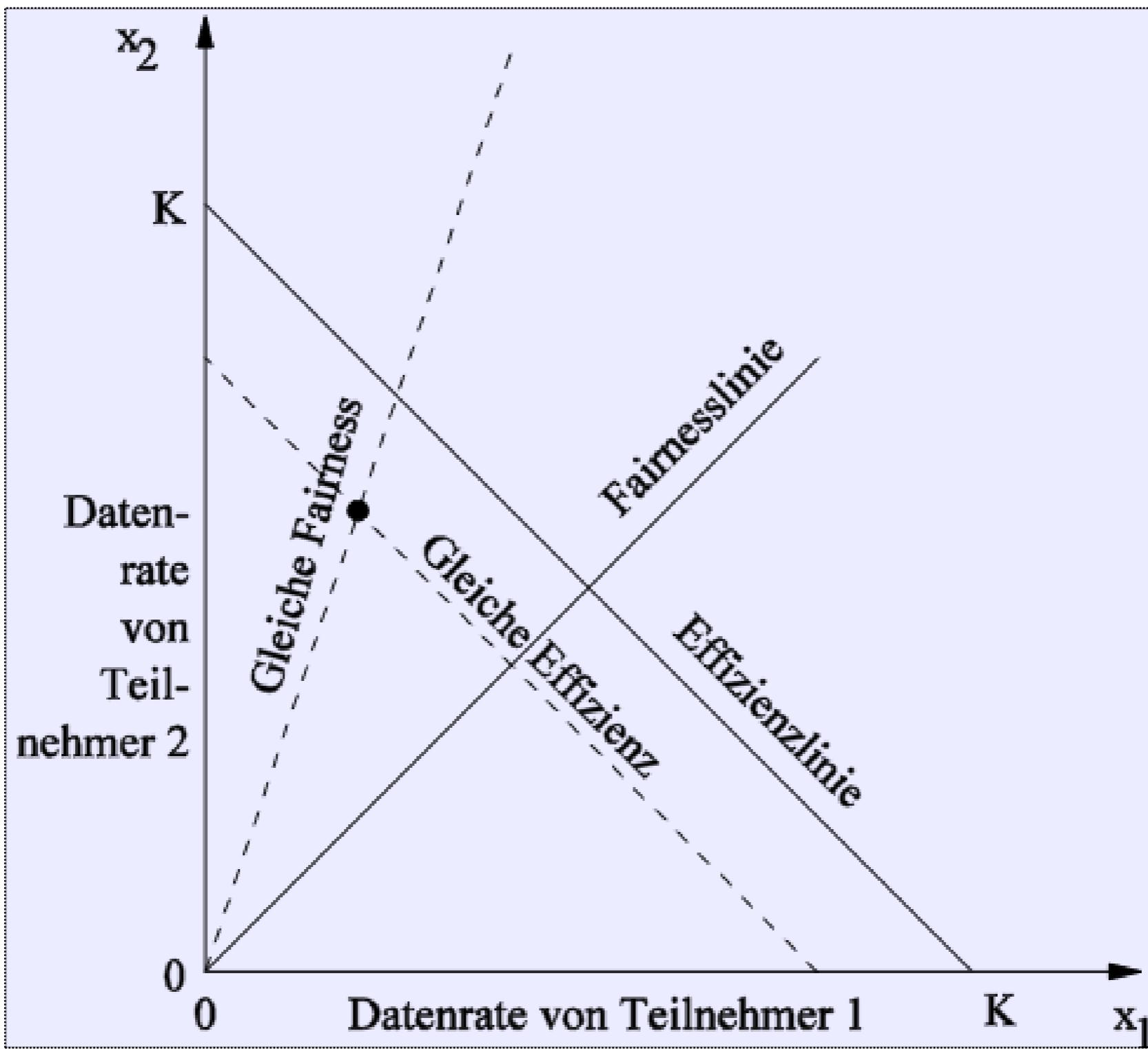
- Konvergenz unmöglich
- Bestenfalls Oszillation um Optimalwert
 - Oszillationsamplitude A
 - Einschwingzeit T



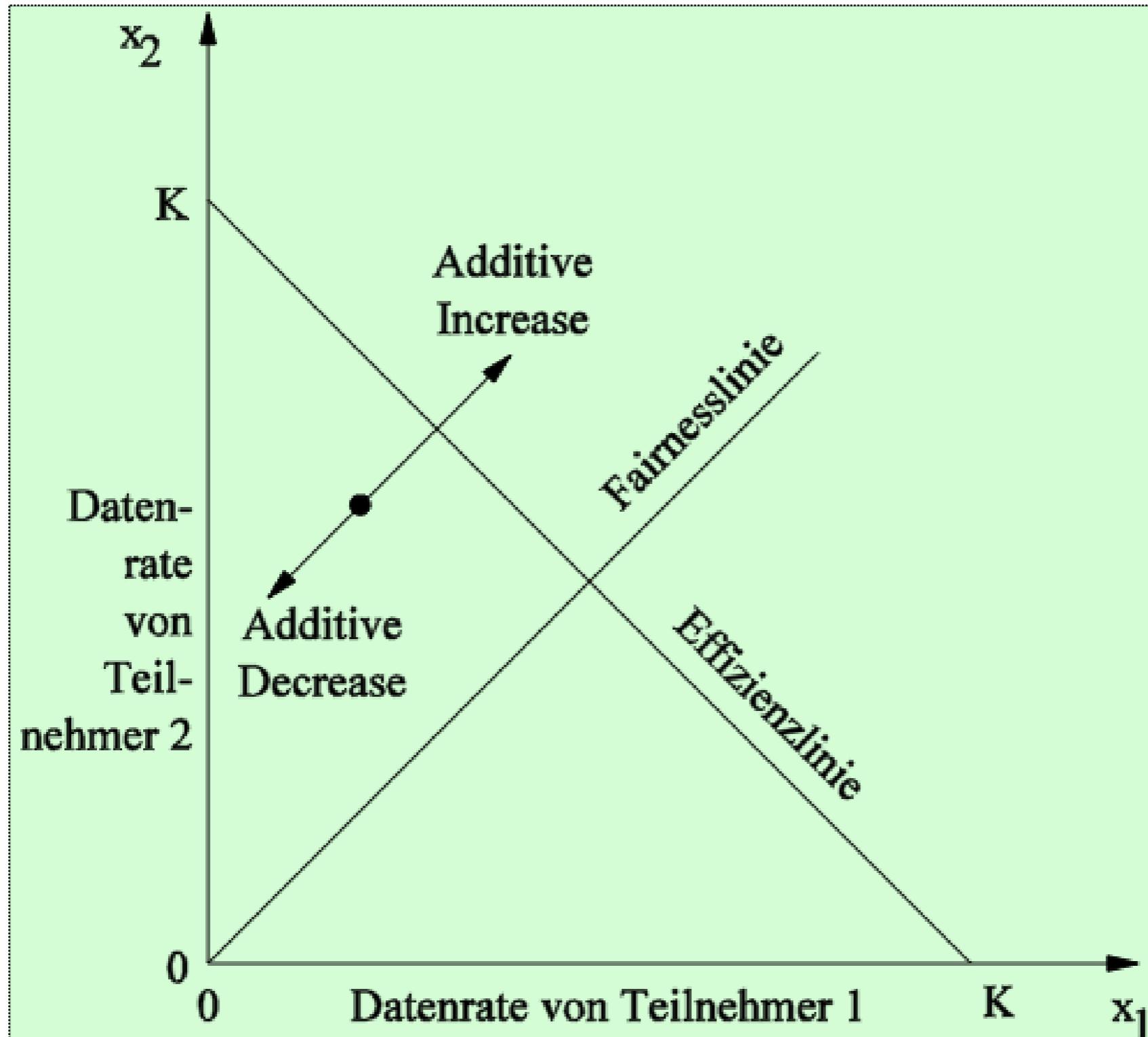
Vektordarstellung (I)



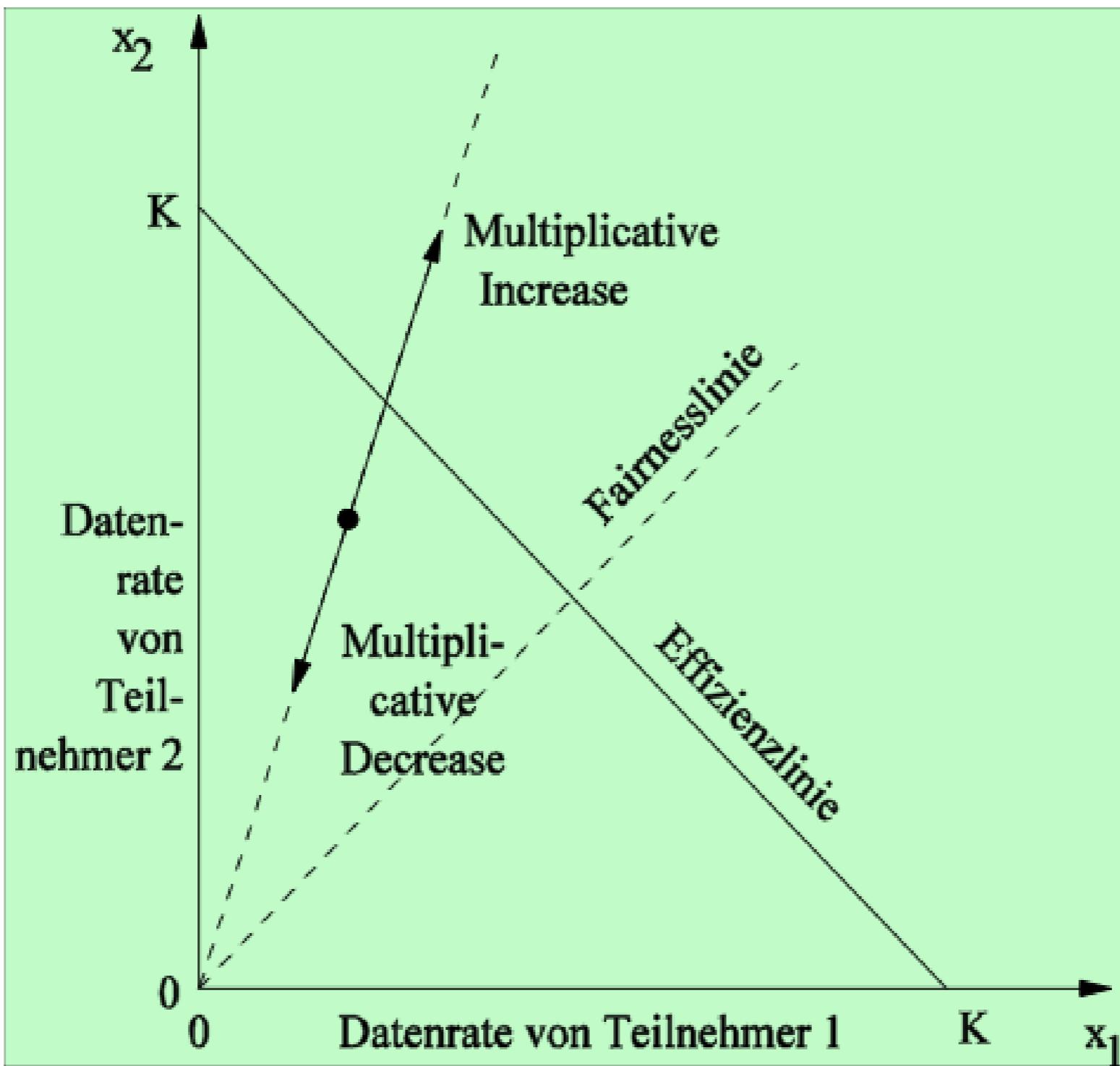
Vektordarstellung (II)



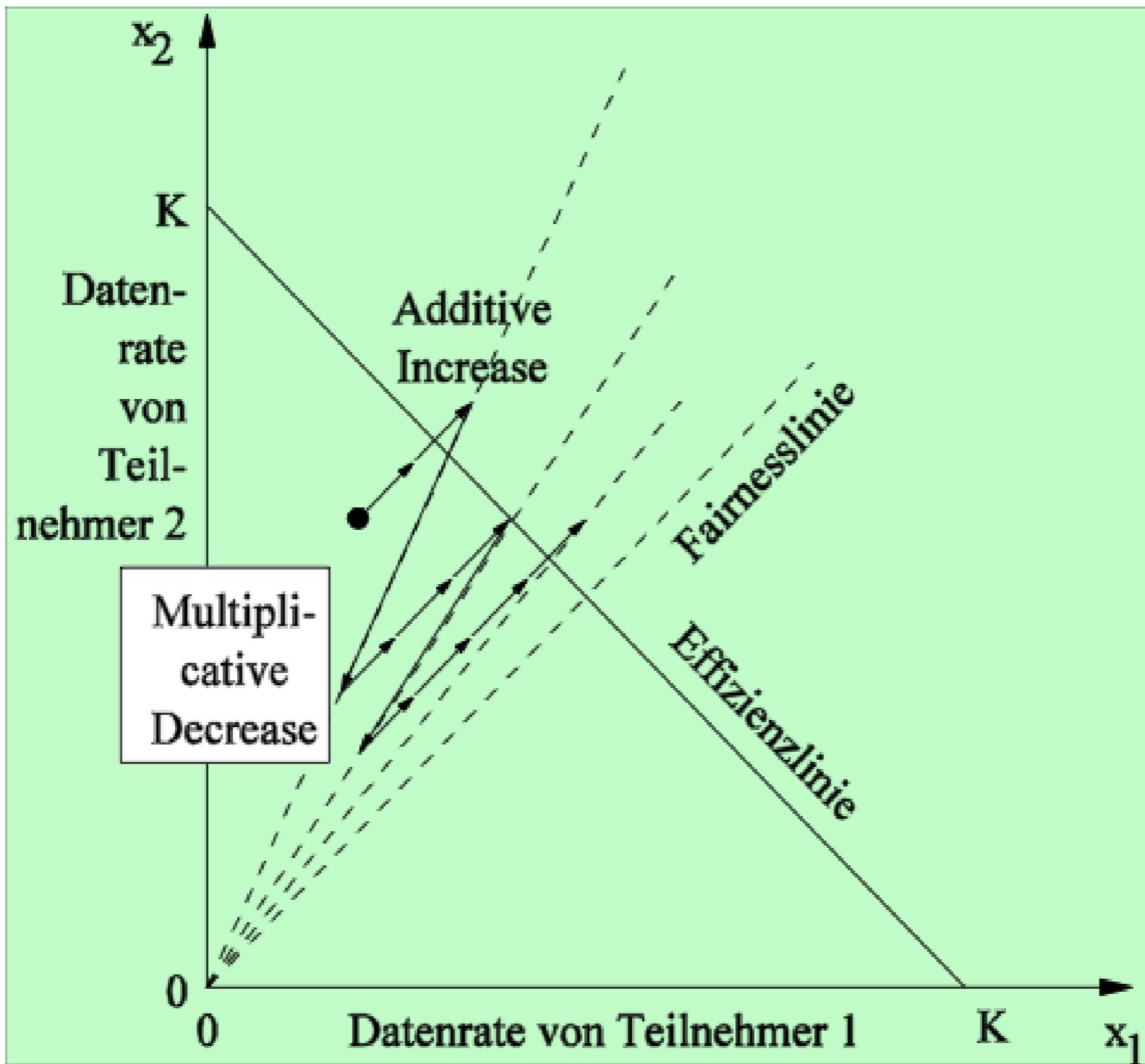
AIAD Additive Increase/ Additive Decrease



MIMD: Multiplicative Incr./ Multiplicative Decrease



AIMD: Additively Increase/ Multiplicatively Decrease



Probleme mit TCP Reno

- Verbindungen mit großer RTT werden diskriminiert
- Warum?
 - Auf jeden Router konkurrieren TCP-Verbindungen
 - Paketverluste halbieren Umsatz (MD)
 - Wer viele Router hat, endet mit sehr kleinen Congestion-Window
- Außerdem:
 - Kleinere RTT ist schnellere Update-Zeit
 - Daher steigt die Rate (AI) auf kurzen Verbindungen schneller
 - Mögliche Lösung:
 - konstante Datenratenanpassung statt Fenster-basierte Anpassung

TCP Vegas

- RTT-basiertes Protokoll als Nachfolger von TCP Reno
 - “L. Brakmo and L. Peterson, “TCP Vegas: End-to-End Congestion Avoidance on a Global Internet”, IEEE Journal on Selected Areas of Communications, vol. 13, no. 8, October 1995, pp. 1465–1480.
- Bessere Effizienz
- Geringere Paketverluste
- Aber:
 - TCP Vegas und TCP Reno gegeneinander unfair

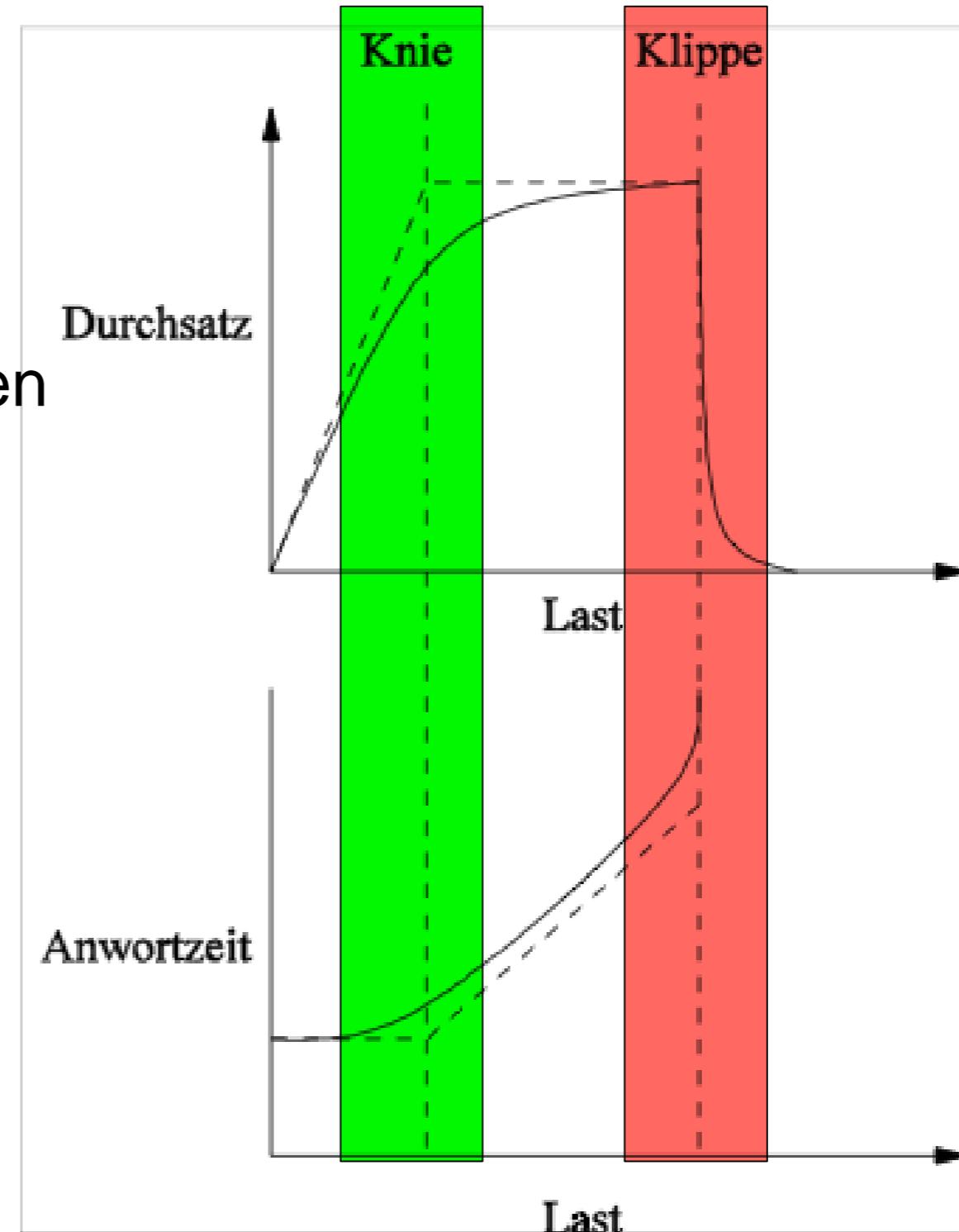
Durchsatz und Antwortzeit

- **Klippe:**

- Hohe Last
- Geringer Durchsatz
- Praktisch alle Daten gehen verloren

- **Knie:**

- Hohe Last
- Hoher Durchsatz
- Einzelne Daten gehen verloren



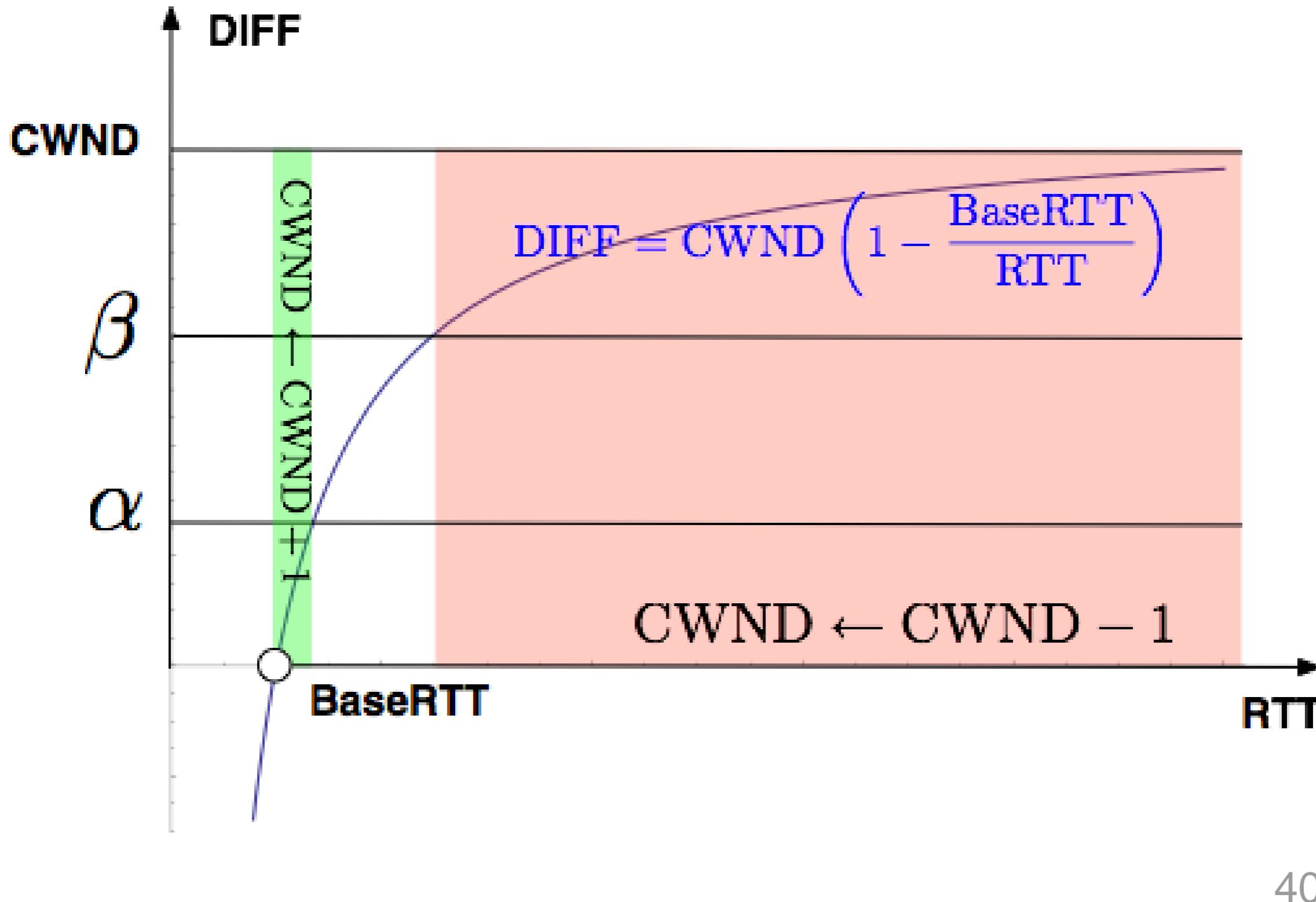
TCP Vegas-Algorithmus

- TCP Stauvermeidung basierend auf Delay
 - RTT (round trip time)
- Wurde implementiert in Linux, FreeBSD
- Ziel
 - Mehr Fairness
- TCP Vegas ist TCP Reno-freundlich
 - Im Konflikt mit TCP Reno gibt Vegas nach
- Literatur
 - MLA Brakmo, Lawrence S., and Larry L. Peterson. "TCP Vegas: End to end congestion avoidance on a global Internet." *IEEE Journal on selected Areas in communications* 13.8 (1995): 1465-1480.
 - Mo, Anantharam, Walrand, „Analysis and Comparison of TCP Reno and Vegas“, IEEE Proc. InfoCom 1999

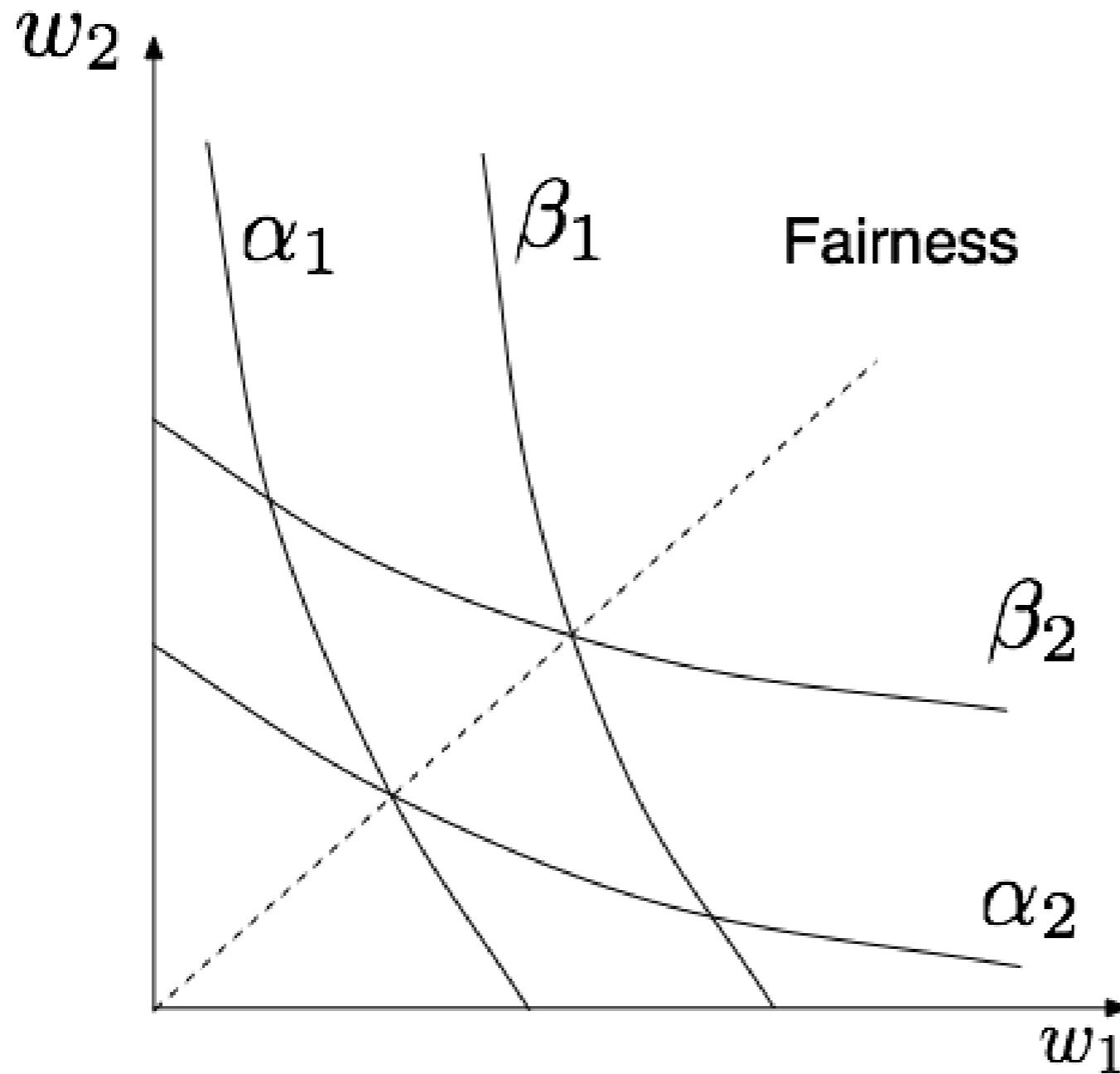
TCP Vegas-Algorithmus

- Parameter
 - geschätzte Umlaufzeit: RTT
 - minimale Umlaufzeit: $BaseRTT$
 - wirkliche Datenrate: $Actual = CWND/RTT$
 - erwartete Datenrate: $Expected = CWND/BaseRTT$
 - $Diff = (Expected - Actual) \cdot BaseRTT$
 - Programmparameter: $0 \leq \alpha < \beta$
- Wenn $Diff \leq \alpha$ (d.h. $Actual \approx Expected$)
 - Last ist gering
 - $CWND \leftarrow CWND + 1$
- Wenn $Diff > \beta$, (d.h. $Actual << Expected$)
 - Last ist zu hoch
 - $CWND \leftarrow CWND - 1$
- Sonst keine Aktion: $CWND \leftarrow CWND$

TCP Vegas - Abhangigkeit von RTT



Fenster-Anpassung in Vegas



■ TCP

- reagiert dynamisch auf die zur Verfügung stehende Bandweite
- Faire Aufteilung der Bandweite
 - Im Idealfall: n TCP-Verbindungen erhalten einen Anteil von $1/n$

■ Zusammenspiel mit anderen Protokollen

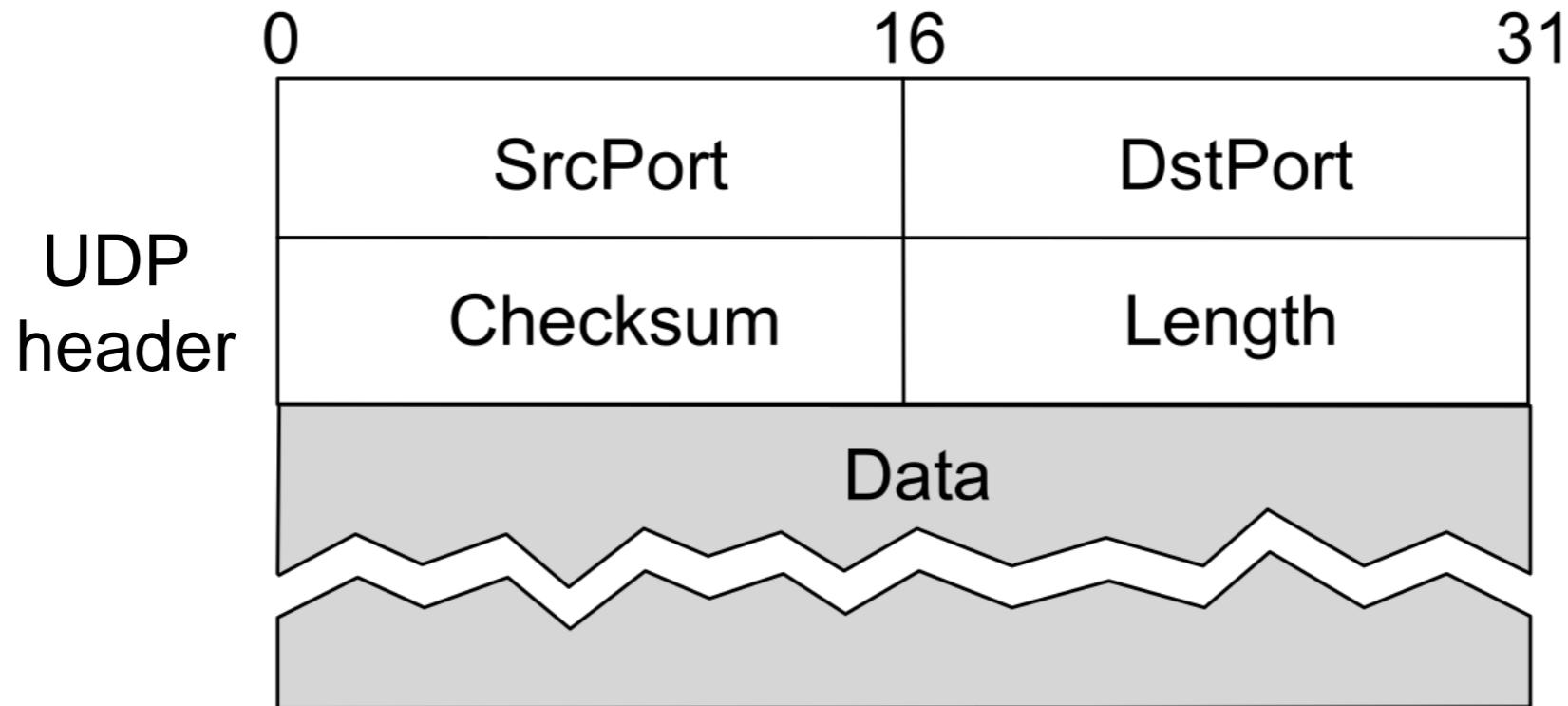
- Reaktion hängt von der Last anderer Transportprotokolle ab
 - z.B. UDP hat keine Congestion Control
- Andere Protokolle können jeder Zeit eingesetzt werden
- UDP und andere Protokoll können TCP Verbindungen unterdrücken

■ Schlussfolgerung

- Transport-Protokolle müssen TCP-kompatibel sein (TCP friendly)

UDP

- User Datagram Protocol (UDP)
 - ist ein unzuverlässiges, verbindungsloses Transportprotokoll für Pakete
- Hauptfunktion:
 - Demultiplexing von Paketen aus der Vermittlungsschicht
- Zusätzlich (optional):
 - Checksum aus UDP Header + Daten



- TCP erzeugt zuverlässigen Byte-Strom
 - Fehlerkontrolle durch “GoBack-N”
- Congestion control
 - Fensterbasiert
 - AIMD, Slow start, *Congestion Threshold*
 - Flusskontrolle durch *Window*
 - Verbindungsauftakt
 - Algorithmus von Nagle

Systeme II

5. Die Transportschicht

Christian Schindelhauer

Technische Fakultät

Rechnernetze und Telematik

Albert-Ludwigs-Universität Freiburg

Systeme II

6. Die Anwendungsschicht

Christian Schindelhauer

Technische Fakultät

Rechnernetze und Telematik

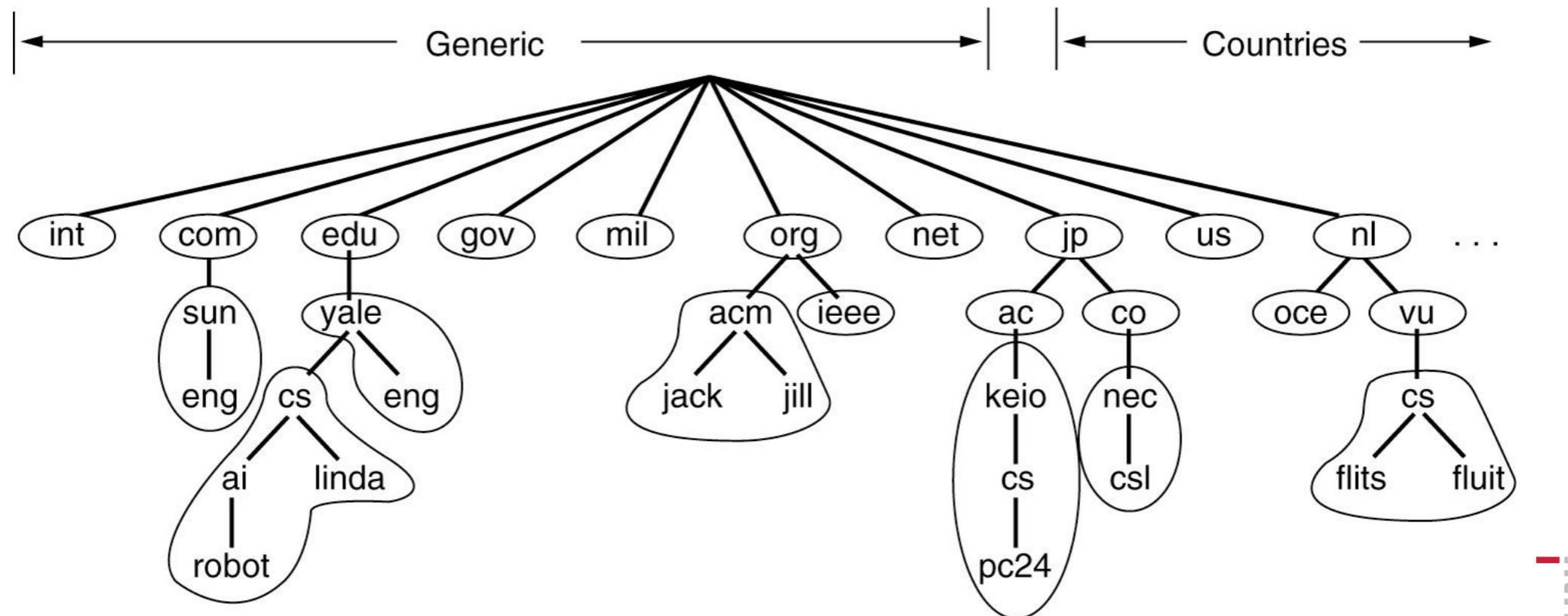
Albert-Ludwigs-Universität Freiburg

Version 06.07.2017

- Menschen kommen mit den 4-Byte IPv4-Adressen nicht zurecht:
 - 209.85.148.102 für Google
 - 132.230.2.100 für Uni Freiburg
 - Was bedeuten?
 - 77.87.229.75
 - 132.230.150.170
- Besser: Natürliche Wörter für IP-Adressen
 - Z.B. www.get-free-beer.de
 - oder www.uni-freiburg.de
- Das Domain Name System (DNS) übersetzt solche Adressen in IP-Adressen

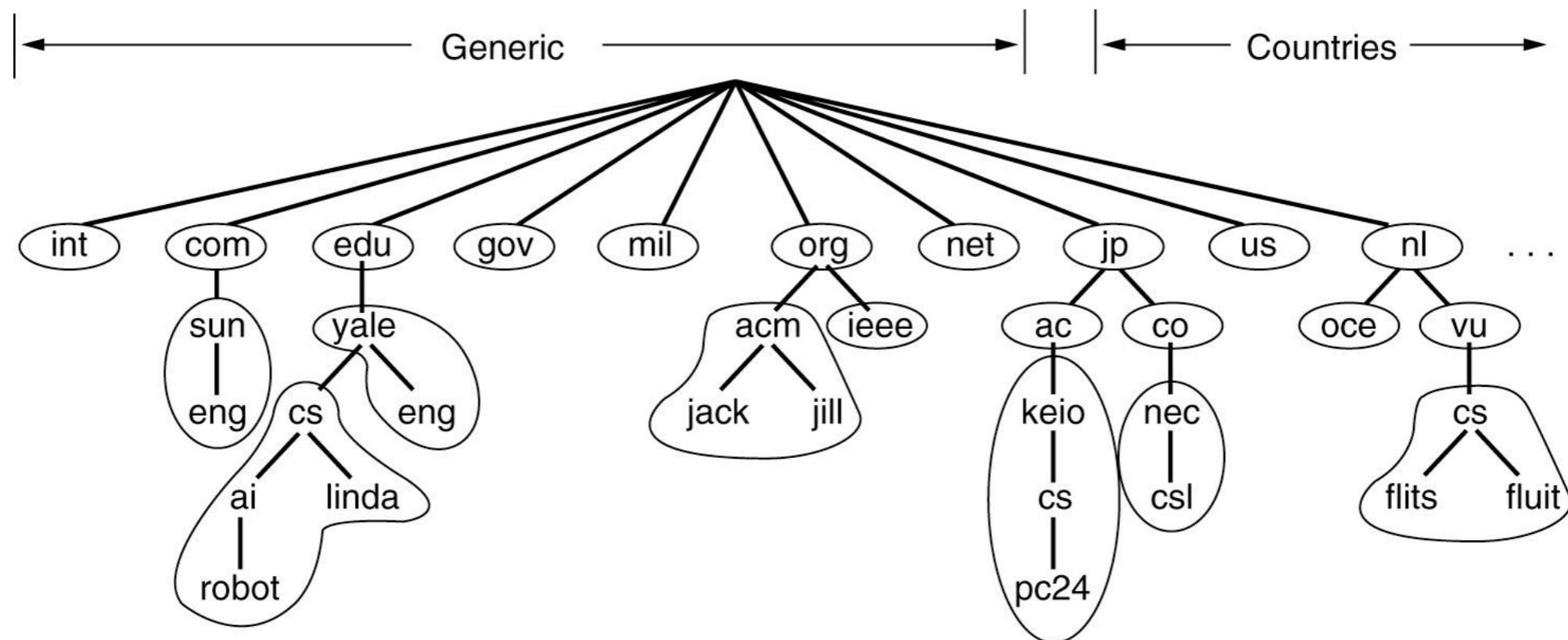
DNS – Architektur

- DNS bildet Namen auf Adressen ab
 - Eigentlich: Namen auf Ressourcen-Einträge
- Namen sind hierarchisch strukturiert in einen Namensraum
 - Max. 63 Zeichen pro Komponente, insgesamt 255 Zeichen
 - In jeder Domain kontrolliert der Domain-Besitzer den Namensraum darunter
- Die Abbildung geschieht durch Name-Server



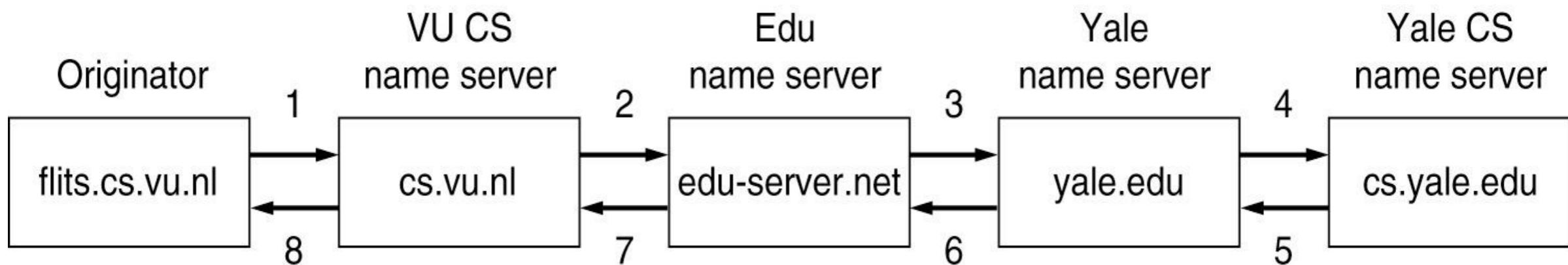
DNS Name Server

- Der Namensraum ist in Zonen aufgeteilt
- Jede Zone hat einen *Primary Name Server* mit maßgeblicher Information
 - Zusätzlich *Secondary Name Server* für Zuverlässigkeit
- Jeder Name Server kennt
 - seine eigene Zone
 - Name-Server der darunterliegenden Bereiche
 - Bruder-Name-Server oder zumindestens einen Server, der diese kennt

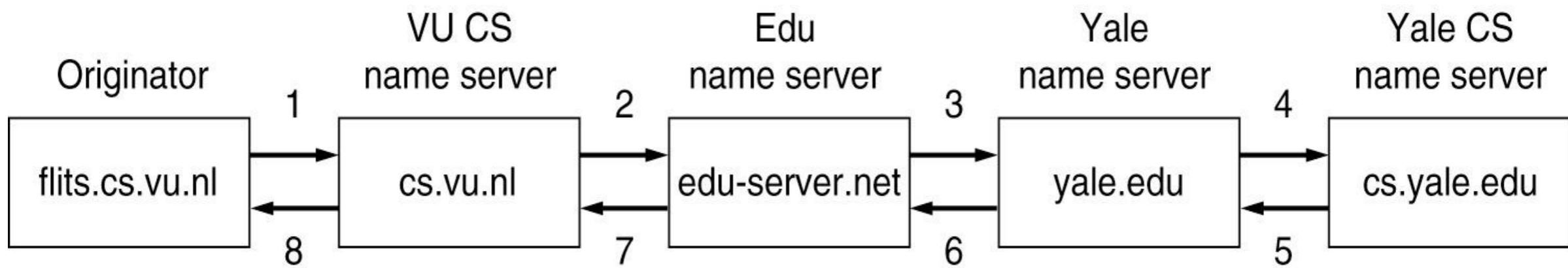
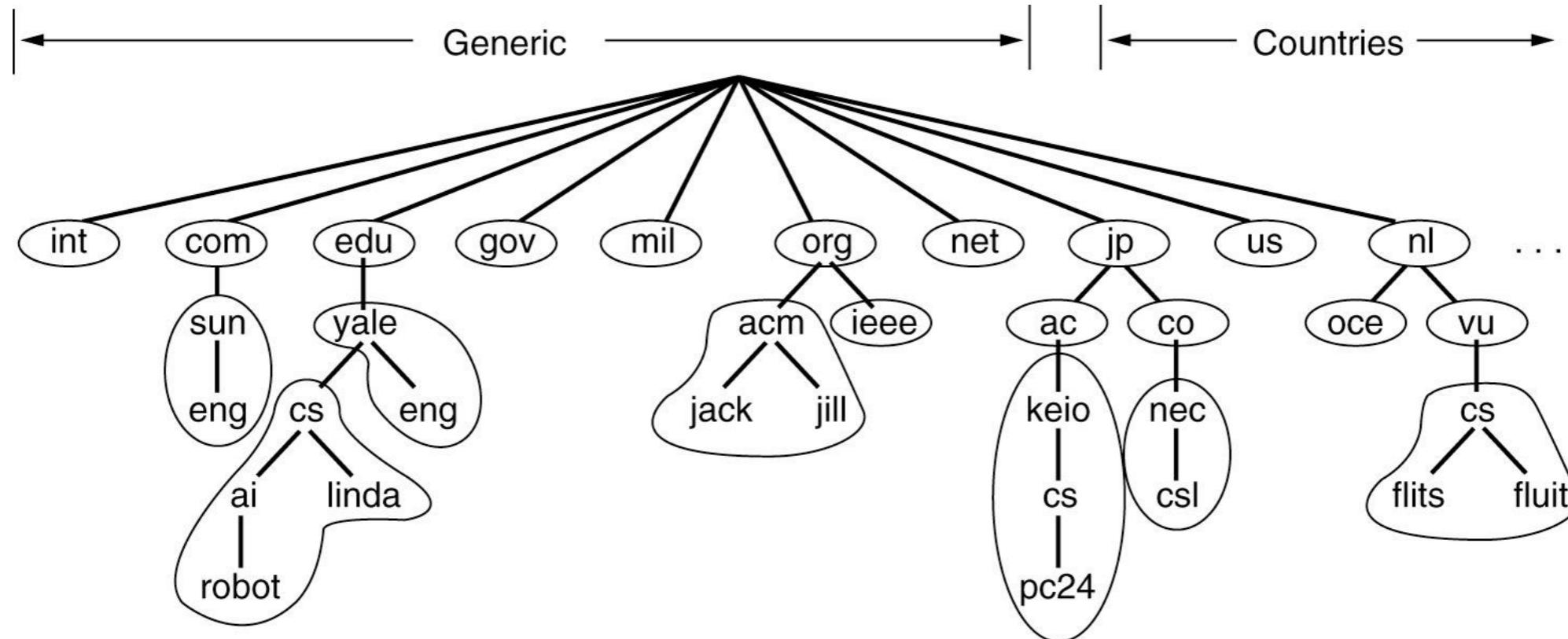


DNS Anfragebearbeitung

- Anfragen von einem End-System werden zu den vorkonfigurierten Name-Server geschickt
 - Soweit möglich, antwortet dieser Name-Server
 - Falls nicht, wird die Anfrage zu dem bestgeeigneten Name-Server weitergereicht
 - Die Antworten werden durch die Zwischen-Server zurückgeschickt
- Server darf Antworten speichern (cachen)
 - Aber nur für eine bestimmte Zeit

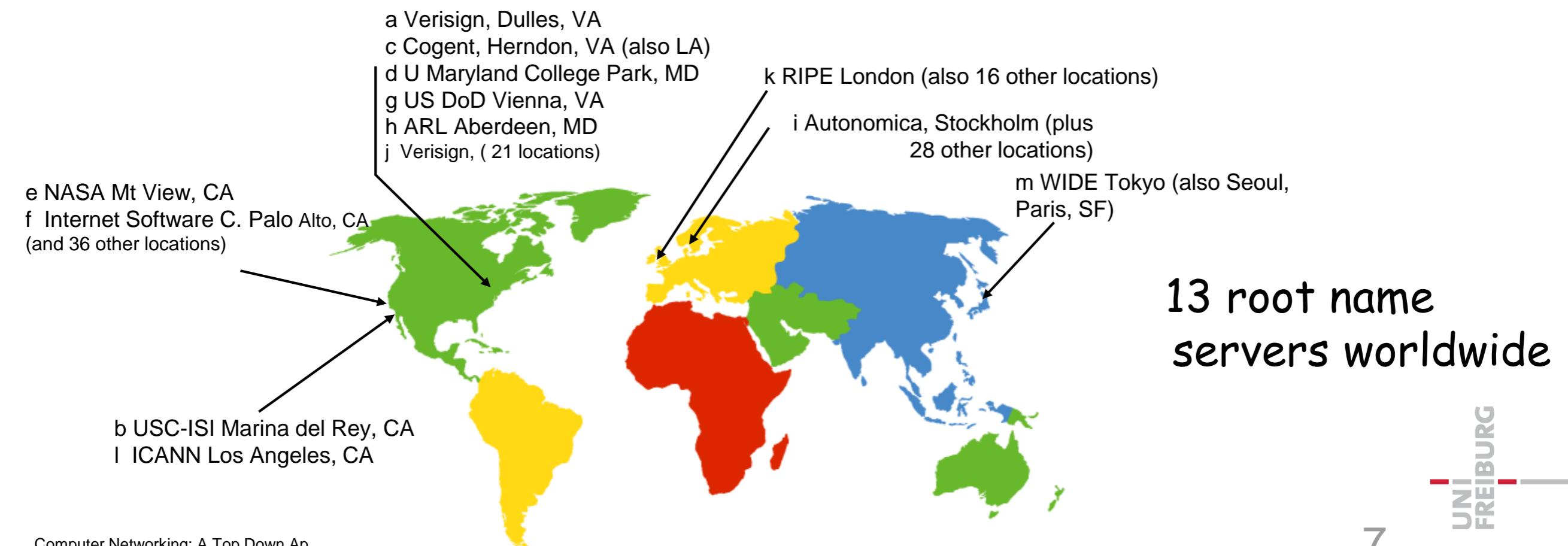


Beispiel



DNS Root Name Servers

- wird von lokalen Name-Server kontaktiert, wenn der Name nicht aufgelöst werden kann
- Root Name Server:
 - wird kontaktiert vom Name-Server falls die Zuordnung der Namen nicht bekannt ist.
 - erhält die Zuordnung
 - gibt die Zuordnung an den lokalen Name-Server weiter



TLD und autorisierte Server

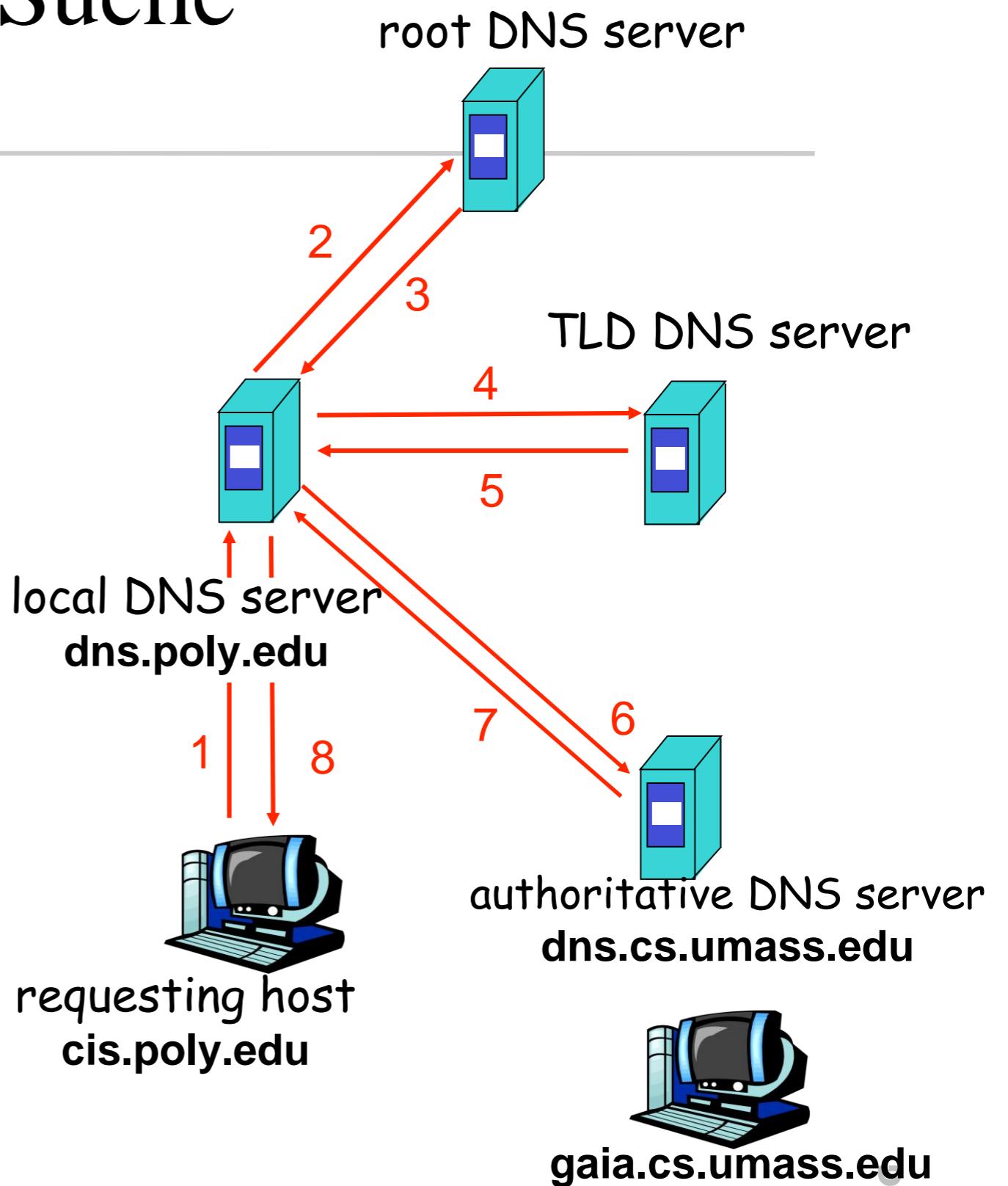
- Top-Level Domain (TLD) Server
 - verantwortlich für com, org, net, edu, etc, und alle Top-Level-Country-Domains uk, fr, ca, jp.
 - Network Solutions unterhält Server für *com* TLD
 - Educause für *edu* TLD
- Autorisierte DNS Servers:
 - DNS-Server von Organisationen
 - welche verantwortlich für die Zuordnung von IP-Adresse zu Hostnamen sind
 - können von den Organisationen oder Service-Provider unterhalten werden

Local Name Server

- Jeder ISP hat einen lokalen Name-Server
 - Default Name Server
- Jede DNS-Anfrage wird zum lokalen Name-Server geschickt
 - fungiert als Proxy und leitet Anfragen in die Hierarchie weiter

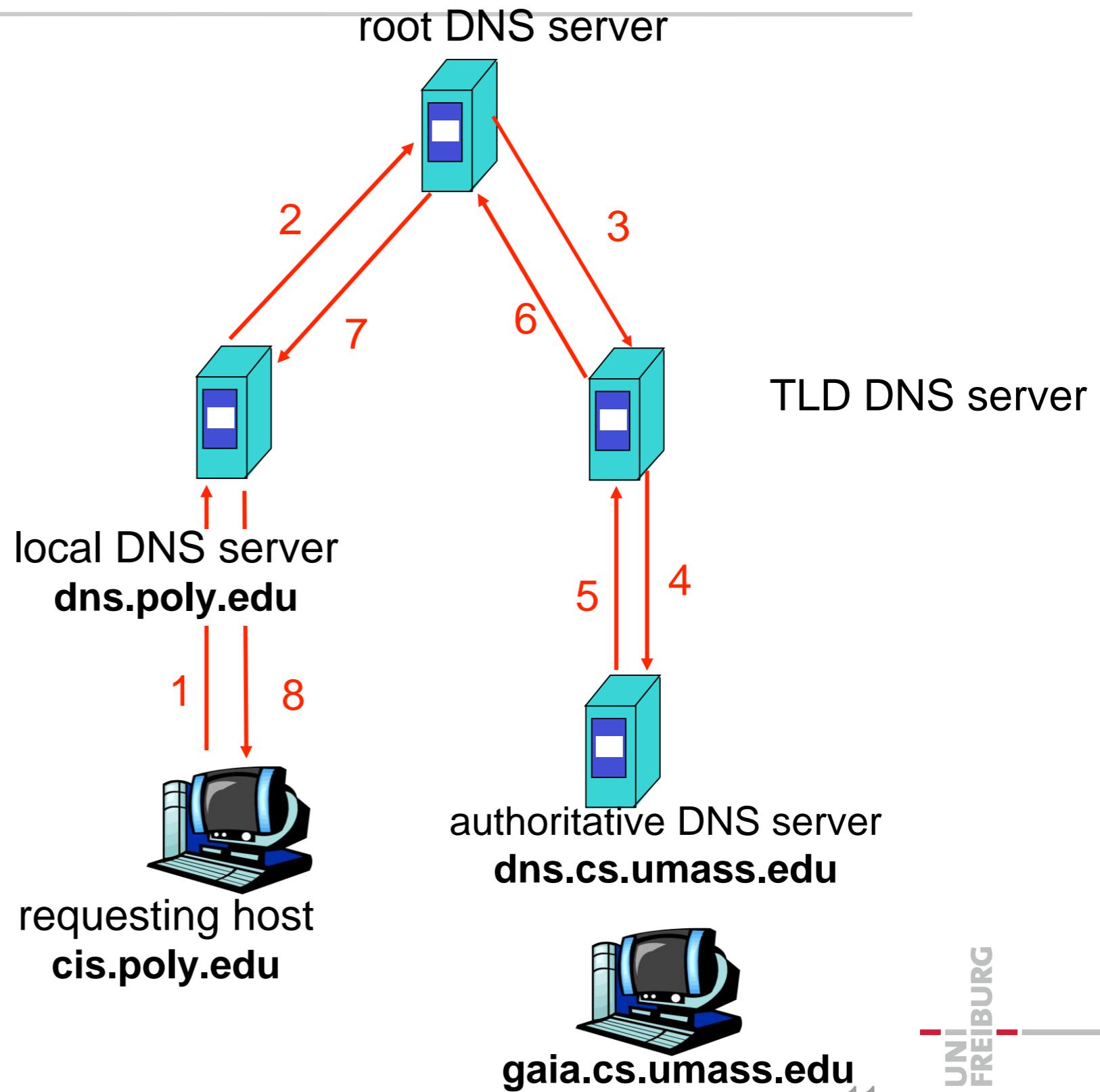
DNS Iterative Suche

- Rechner bei cis.poly.edu fragt nach IP address für gaia.cs.umass.edu
- Iterative Anfrage
 - Angefragte Server antworten
 - mit IP-Adresse
 - oder mit dem Namen des nächsten Servers
 - Lokaler DNS-Server ist selbst für Suche verantwortlich



DNS Rekursive Suche

- Jeder angefragte Server ist für die Namensauflösung zuständig
- Anfrage wird rekursive weitergeleitet und dann zurück gegeben



- Sobald ein Name-Server einen Namen kennen lernt, speichert er die Zuordnung
 - Cache-Einträge haben einen Time-Out und werden nach einer gewissen Zeit gelöscht
 - TLD-Servers werden in lokalen Name-Servern gespeichert
 - Daher werden Root-Name-Server nicht oft besucht
- Update und Benachrichtigungsmechanismus von IETF festgelegt
 - RFC 2136
 - <http://www.ietf.org/html.charters/dnsind-charter.html>

DNS-Einträge

- DNS: verteilte Datenbank speichert Resource Records (RR)
- RR Format: (Name, Wert, Typ, TTL)
- Typ = A
 - Name = hostname
 - Wert = IP-Adresse
- Typ = NS
 - Name = domain (z.B. uni-freiburg.de)
 - Wert = hostname eines autorisierten Name-Servers für diese Domain
- Typ = CNAME
 - Name = Alias für einen „kanonischen“ (wirklichen) Namen
 - z.B. www.ibm.com ist in Wirklichkeit servereast.backup2.ibm.com
 - Wert ist kanonischer Name
- Typ = MX
 - Wert ist der Name des Mailservers

DNS Resource Record

- Ressourcen-Einträge: Informationen über Domains, einzelne Hosts,...
- Inhalt:
 - Domain_name: Domain(s) des Eintrags
 - Time_to_live: Gültigkeit (in Sekunden)
 - Class: Im Internet immer "IN"
 - Type: Siehe Tabelle
 - Value: z.B. IP-Adresse

Type	Meaning	Value
SOA	Start of Authority	Parameters for this zone
A	IP address of a host	32-Bit integer
MX	Mail exchange	Priority, domain willing to accept e-mail
NS	Name Server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
HINFO	Host description	CPU and OS in ASCII
TXT	Text	Uninterpreted ASCII text

DNS-Protokoll

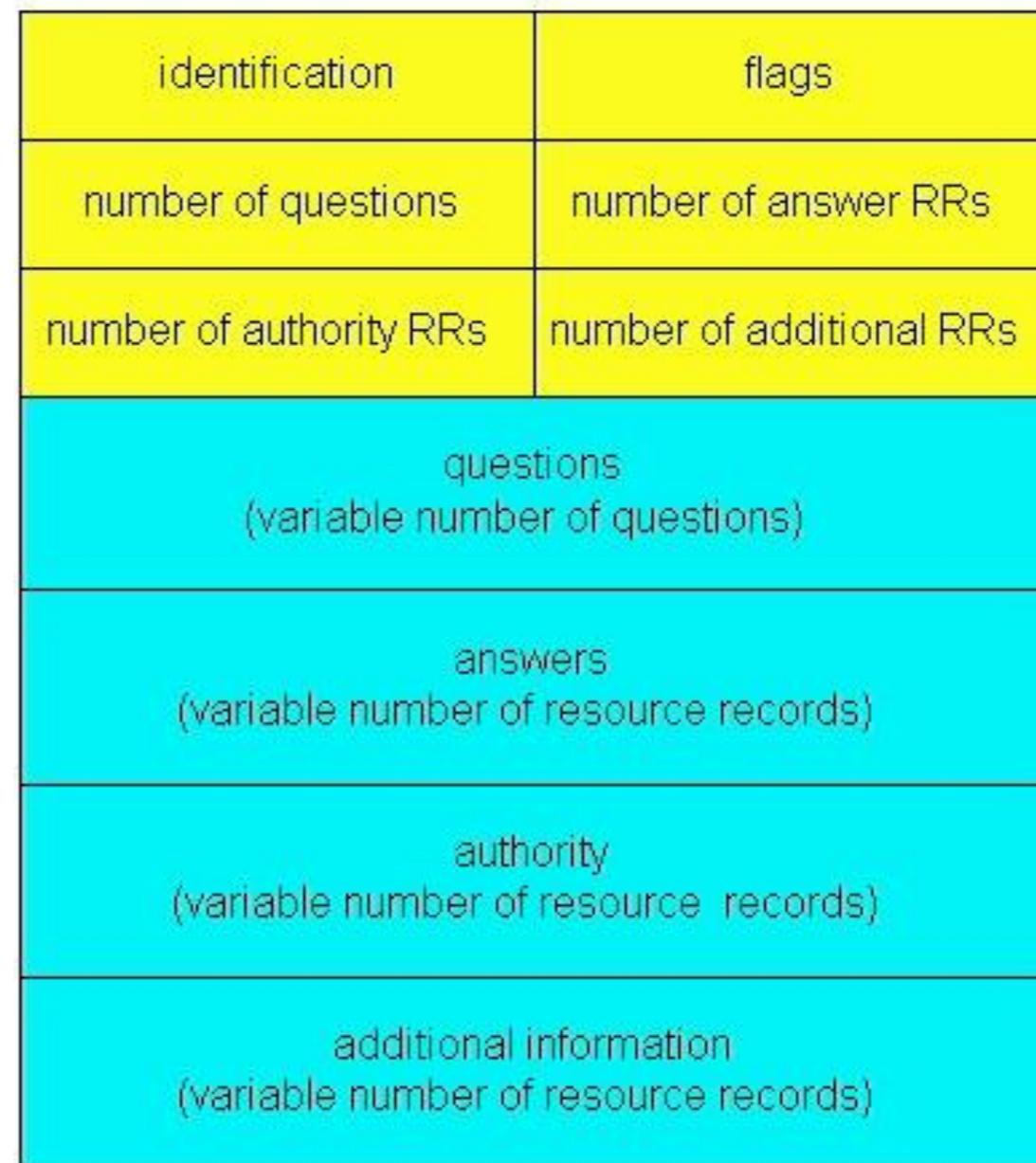
- Anfrage und Antwort im selben Format

Nachrichten-Header

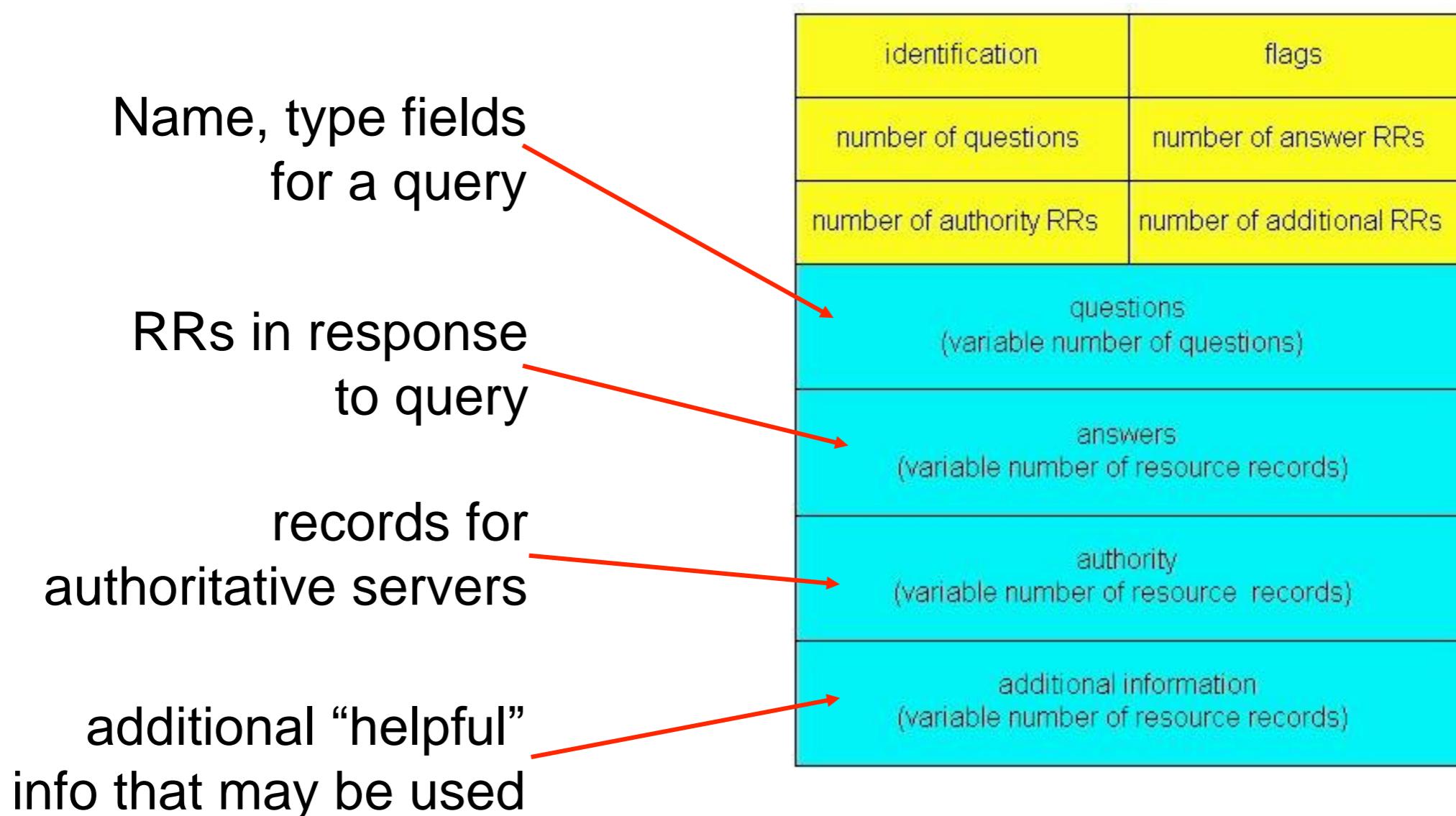
- ID, 16 Bit für Anzahl der Anfragen, Anzahl der Antworten, ...

Flags:

- Query oder Reply
- Rekursion gewünscht
- Rekursion verfügbar
- Antwort ist autorisiert



DNS-Protokoll und Nachrichten



Dynamisches DNS

- Problem
 - Zeitlich zugewiesene IP-Adressen
 - z.B. durch DHCP
- Dynamisches DNS
 - Sobald ein Knoten eine neue IP-Adresse erhält, registriert dieser diese beim DNS-Server, der für diesen Namen zuständig ist
 - Kurze time-to-live-Einträge sorgen für eine zeitnahe Anpassung
 - da sonst bei Abwesenheit die Anfragen an falsche Rechner weitergeleitet werden
- Anwendung
 - Registrierung einer Domain für den Otto Normalverbraucher
 - Siehe www.dyndns.com

DNS Security Extensions

- Cache Poisoning
 - Falsche Einträge werden in DNS-Server eingebracht
 - weitergeleitete Einträge werden gecacht und können zu falschen Auskünften führen
- DNSSEC
 - implementiert seit 2010
 - Zuständiger Master-Server unterschreibt seine Einträge digital (mit Hilfe eines Public-Key-Kryptosystems)
 - Ursprüngliche Information bleibt unverschlüsselt
- Schlüsselverwaltung
 - Gegenseitiges unterschreiben der Public-Keys
 - Aufwand wird gemildert durch „Chain of Trust“
 - Hierarchisches Kette von unterschriebenen Schlüsseln
- Diskussion
 - aufwändiger DNS-Antworten
 - Sicherheitslücken bleiben bestehen

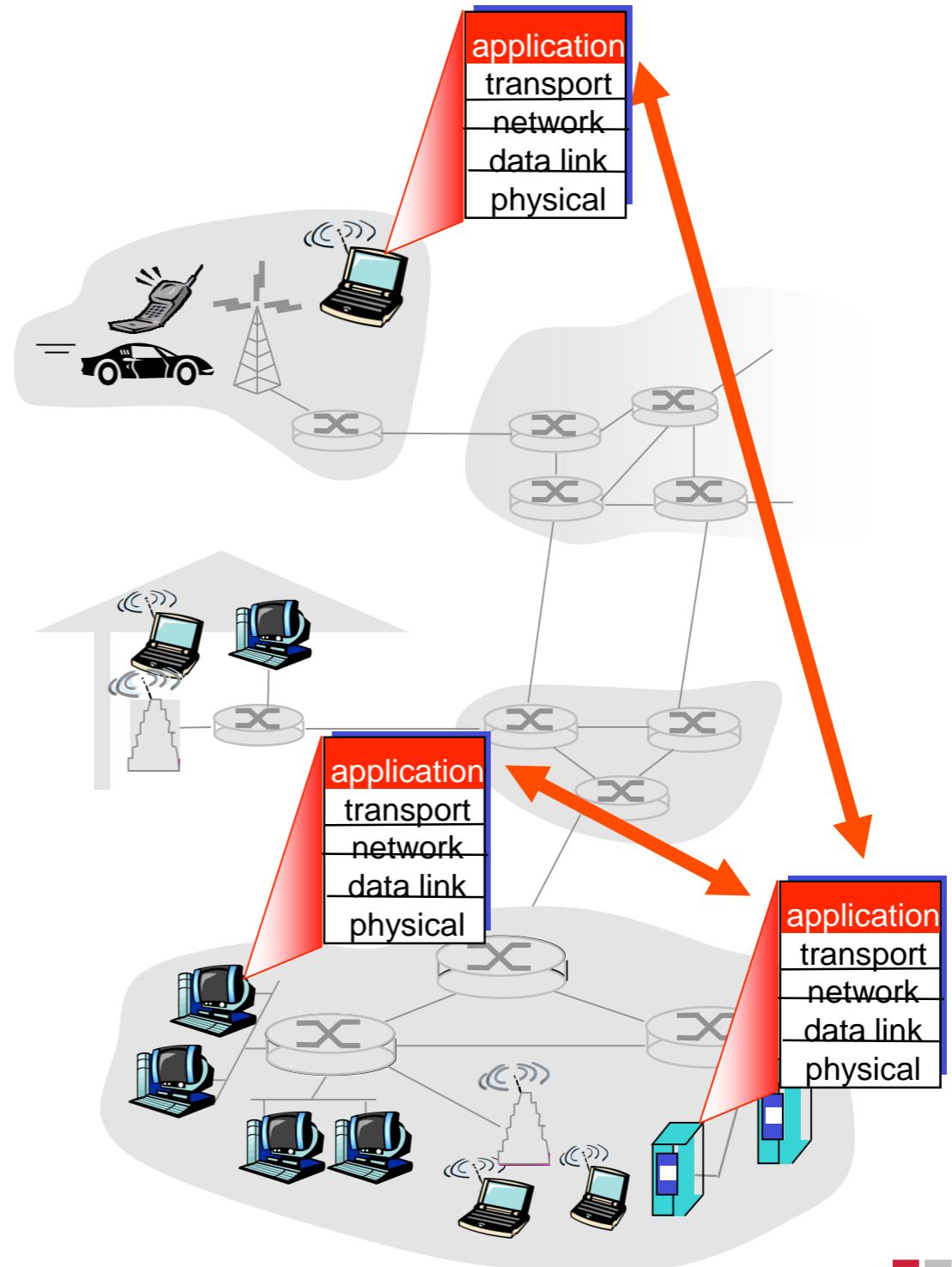
- Aspekte der Programmierung im Internet aus der Sicht der Anwendung
- Anforderungen an die Transportschicht
- Client-Server-Prinzip
- Peer-to-Peer-Prinzip
- Beispiel-Protokolle:
 - HTTP
 - SMTP / POP3 / IMAP
 - DNS
- Programmierung von Netzwerk-Anwendungen

Beispiele Netzwerk-Anwendungen

- E-Mail
- Web
- Instant messaging
- Remote Login
- P2P File Sharing
- Multi-User Network Games
- Video Streaming
- Social Networks
- Voice over IP
- Real-time Video Konferenz
- Grid Computing

Erstellen einer Netzwerk-Anwendung

- Programme laufen auf den End-Punkten
 - kommunizieren über das Netzwerk
 - z.B. Web-Client kommuniziert durch Browser-Software
- Netzwerk-Router
 - werden nicht programmiert!
 - nicht für den Benutzer verfügbar
- Dadurch schnelle Programm-Entwicklung möglich
 - gleiche Umgebung
 - schnelle Verbreitung



Kommunikationsformen in der Anwendungsschicht

- Client-server
 - beinhaltet auch Data Centers & Cloud Computing
- Peer-to-peer (P2P)
- Hybride Verbindung von Client-Server und P2P

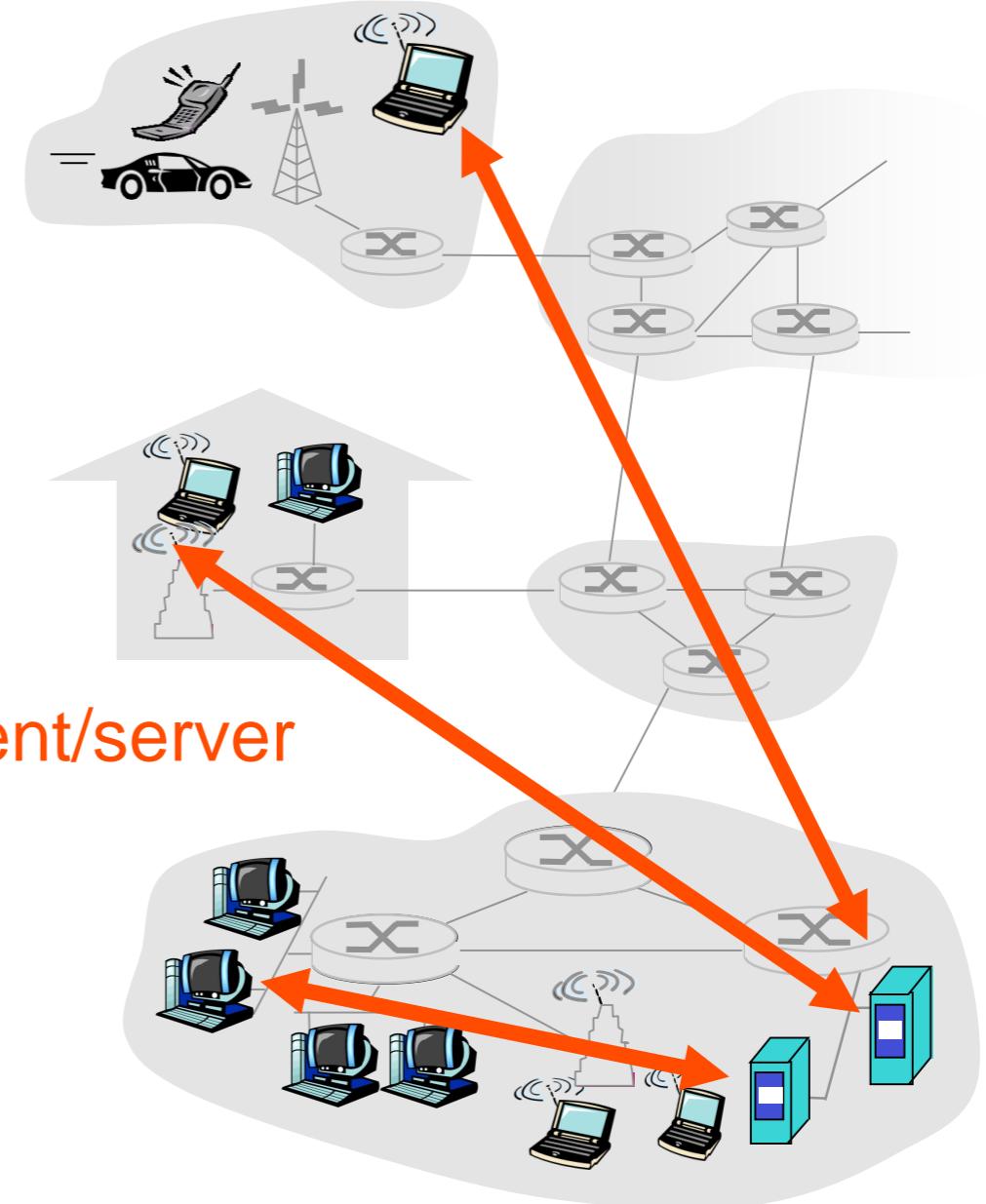
Client-Server-Architektur

■ Server

- allzeit verfügbarer Host
- permanente IP-Address
 - oder per DNS ansprechbar
- Server-Farms wegen Skalierung

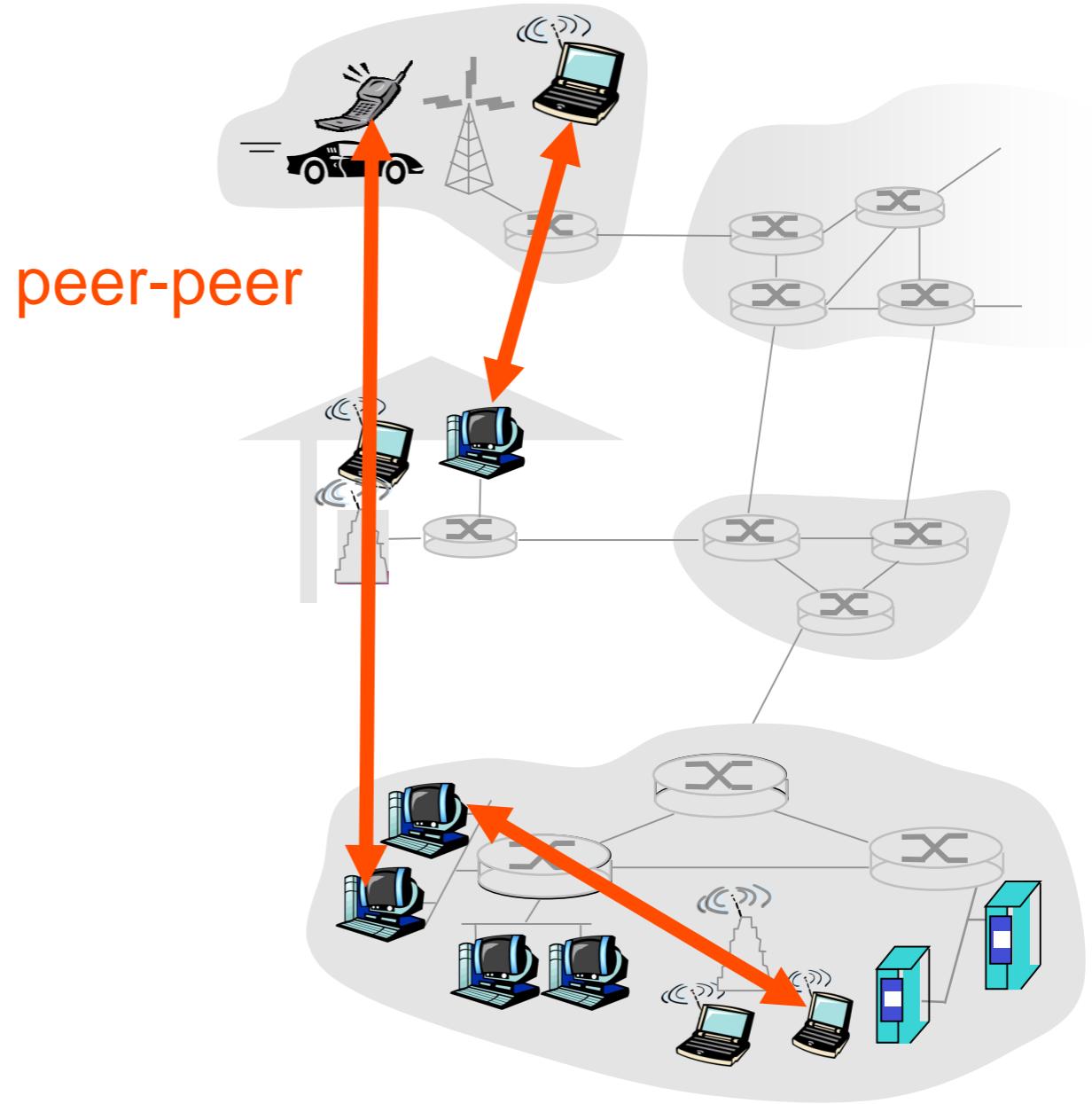
■ Client

- kommuniziert mit dem Server
- möglicherweise nicht durchgängig verbunden
- evtl. dynamische IP-Adresse
- Clients kommunizieren **nicht** miteinander



Peer-to-Peer-Architektur

- Ohne Server
- End-Systeme kommunizieren direkt
- Peers
 - sind nur zeitlich begrenzt online
 - verändern von Zeit zu Zeit ihre IP-Adresse
- Hochskalierbar, aber schwer zu handhaben



Hybrid aus Client-Server und Peer-to-Peer

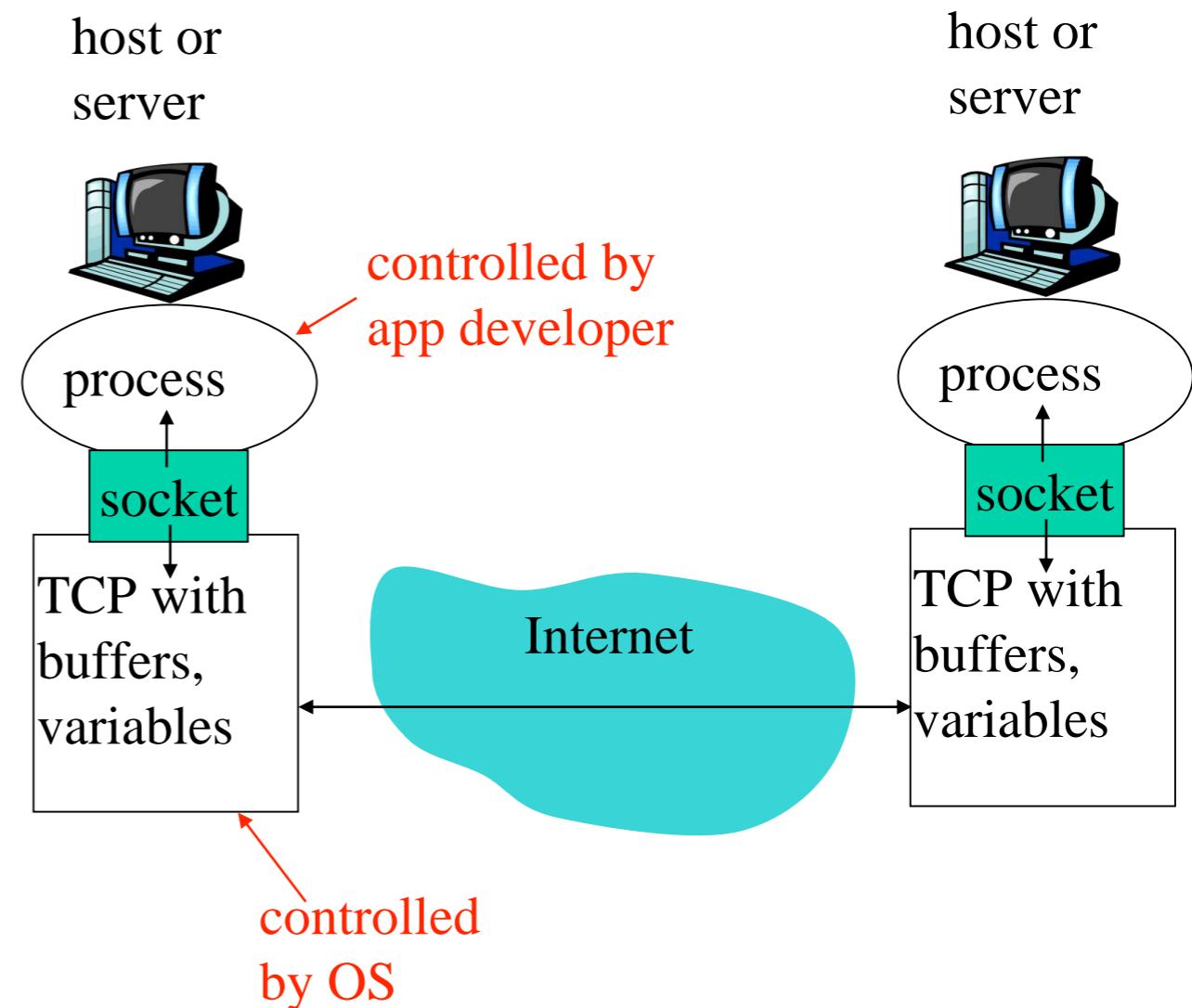
- z.B. Skype
 - Voice-over-IP P2P
 - Server für Anmeldung und Verzeichnis
 - Telefonie und Video-Verbindung Direktverbindung
- Instant Messaging
 - Chat zwischen zwei Benutzern ist P2P
 - Zentraler Service:
 - Client-Anwesenheit
 - Suche und Zuordnung der IP-Adresse
 - Benutzer registrieren die IP-Adresse, sobald online
 - Benutzer fragen beim Server nach IP-Adresse der Partner

Kommunizierende Prozesse

- Prozess: Programm auf einem Rechner (Host)
 - innerhalb des selben Rechners kommunizieren Prozesse durch Inter-Prozess-Kommunikation
 - über OS
- Prozesse in verschiedenen Rechnern
 - kommunizieren durch Nachrichten
- Client-Prozess
 - Initiiert die Kommunikation
- Server-Prozess
 - wartet auf Client-Kontakt
- P2P
 - haben Client und Server-Prozesse

Sockets

- Prozesse senden und empfangen Nachrichten über Sockets (Steckdosen)
- Sockets mit Türen vergleichbar
- Sender-Prozess
 - schiebt die Nachricht zur Tür hinaus
 - vertraut auf die Transport-Infrastruktur, dass die eine Seite der Tür mit der anderen verbündet
- API
 - Wahl des Transport-Protokolls
 - kann bestimmte Parameter wählen



Anwendungsschicht-Programm beschreibt

- Nachrichtentyp
 - z.B. Request, Response
- Nachrichten-Syntax
 - Nachrichtenfelder und Zuordnung
- Nachrichten-Semantik
 - Bedeutung der Felder
- Regeln für das Senden und Empfangen von Nachrichten
- Public-domain Protokolle
 - definiert in RFC
 - für Kompatibilität
 - z.B. HTTP, SMTP, BitTorrent
- Proprietäre Protokolle
 - z.B. Skype, ppstream

Welchen Transport-Service braucht eine Anwendung?

- Datenverlust
 - einige Anwendungen (z.B. Audio) tolerieren gewissen Verlust
 - andere (z.B. Dateitransfer, Telnet) benötigen 100% verlässlichen Datentransport
- Timing
 - einige Anwendungen (z.B. Internet Telefonie, Spiele) brauchen geringen Delay
- Durchsatz (throughput)
 - einige Anwendungen (z.B. Multimedia) brauchen Mindestdurchsatz
 - andere (“elastische Anwendungen”) passen sich dem Durchsatz an
- Sicherheit
- Verschlüsselung, Datenintegrität

Web und HTTP

- Web-Seiten (web page) besteht aus Objekten
- Objekte sind HTML-Datei, JPEG-Bild, Java-Applet, Audio-Datei,...
- Web-Seite besteht aus Base HTML-Datei mit einigen referenzierten Objekten
- Jedes Objekt wird durch eine URL adressiert
 - Beispiel URL:

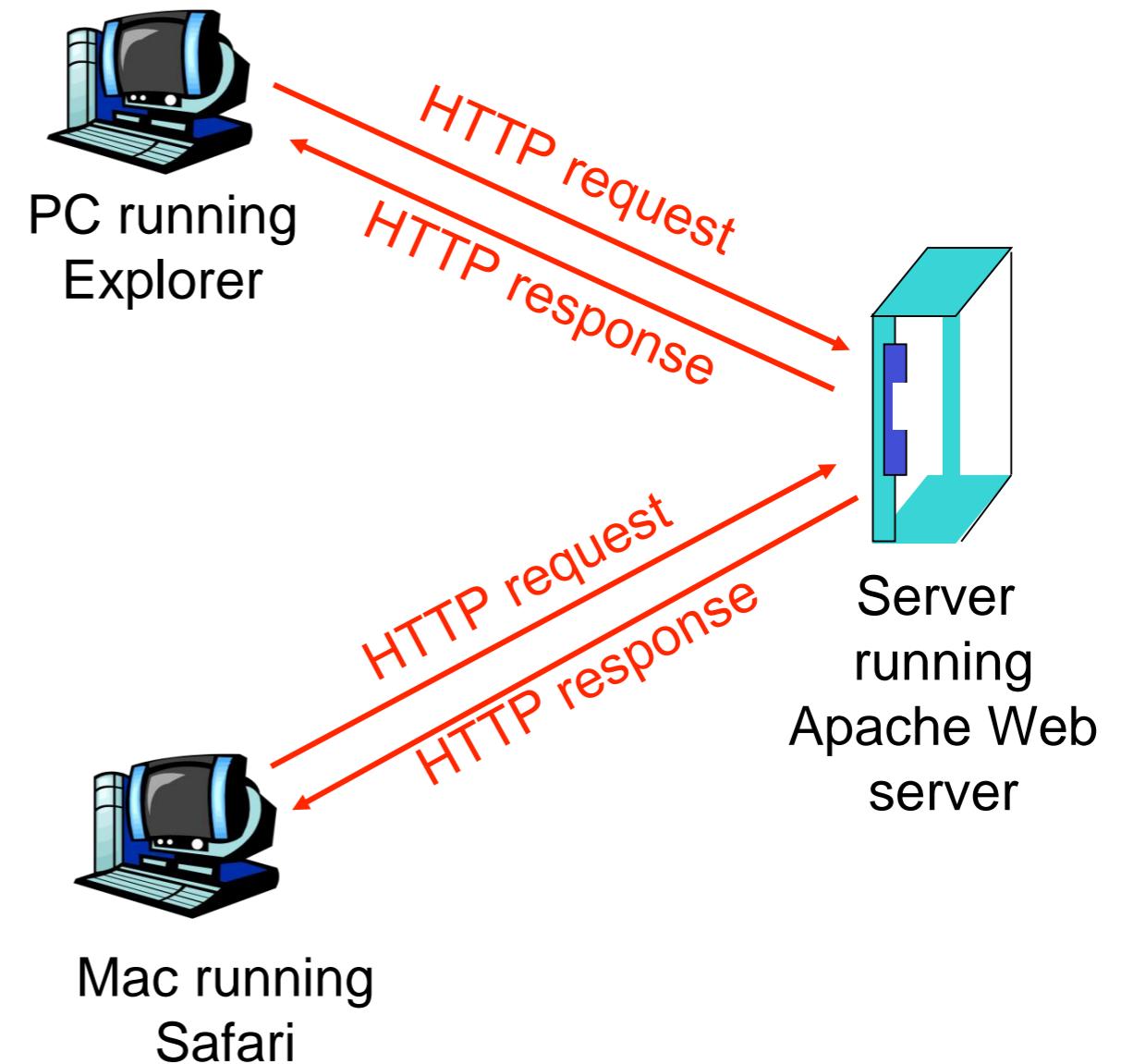
www.someschool.edu/someDept/pic.gif

host name

path name

HTTP-Überblick

- HTTP: Hypertext Transfer Protocol
 - Anwendungsschicht-Protokoll des Webs
- Client/Server-Modell
 - Client
 - Browser fragt an
 - erhält und zeigt Web-Objekte an
 - Server
 - Web-Server sendet Objekte als Antwort der Anfrage



HTTP-Überblick

- Verwendet TCP
- Client initiiert TCP-Verbindung
 - erzeugt Socket zum Server auf Port 80
- Server akzeptiert TCP-Verbindung vom Client
- HTTP-Nachrichten
 - zwischen HTTP-Client und HTTP-Server
 - Anwendungsschicht-Protokoll-Nachrichten
- TCP-Verbindung wird geschlossen

HTTP-Überblick

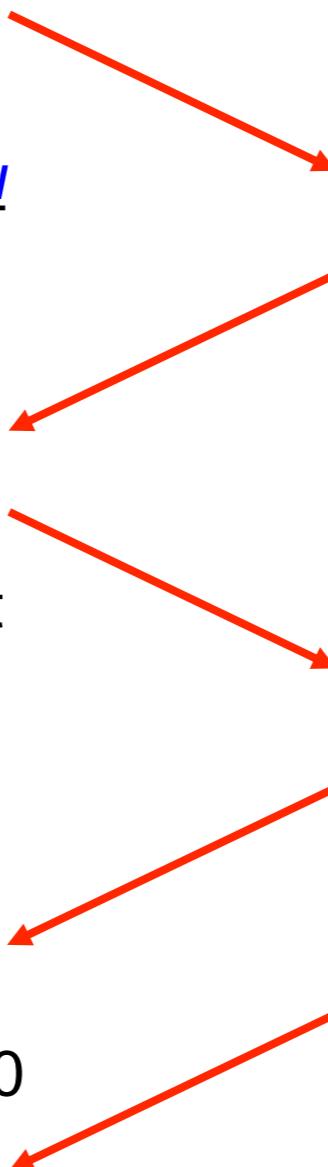
- HTTP ist zustandslos (stateless)
 - Server merkt sich nichts über vorige Anfragen
- Warum?
 - Protokolle mit Zuständen sind komplex
 - Zustände müssen gemerkt und zugeordnet werden
 - falls Server oder Client abstürzen, müssen die möglicherweise inkonsistenten Zustände wieder angepasst werden

HTTP-Verbindungen

- Abbrechende (nicht persistente) HTTP-Verbindung
 - Höchstens ein Objekt wird über eine TCP-Verbindung gesendet
- Weiter bestehende (persistente) HTTP
 - Verschiedene Objekte können über eine bestehende TCP-Verbindung zwischen Client und Server gesendet werden

Nicht-Persistente HTTP-Verbindung

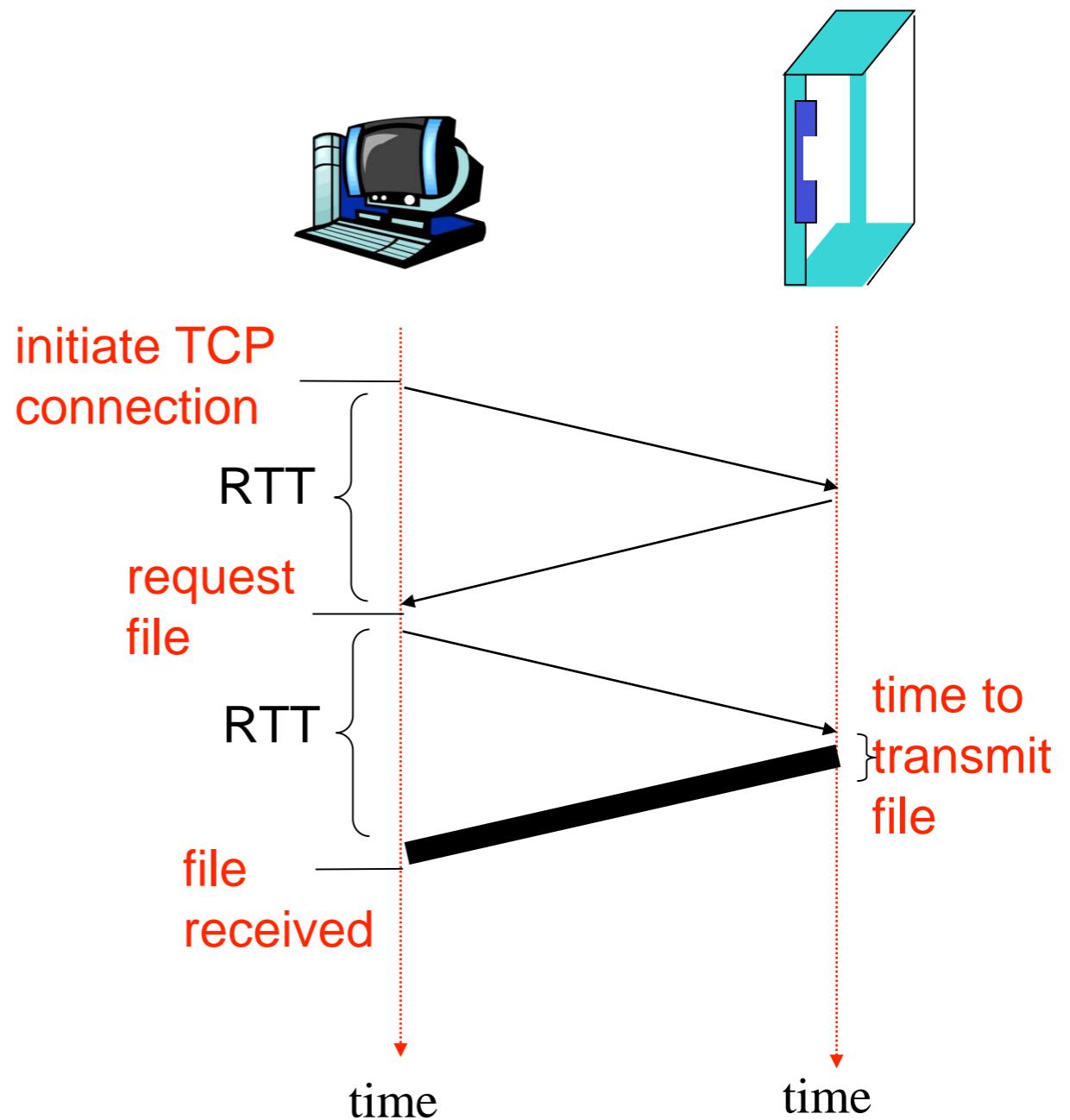
- 1a. HTTP-Client initiiert TCP-Verbindung zum HTTP-Server (Prozess) at www.someSchool.edu on port 80
2. HTTP-Client sendet HTTP Request Message (mit URL) zum TCP-Verbindungs-Socket. Die Nachricht zeigt an, dass der Client das Objekt *someDepartment/home.index* will
5. HTTP-Client erhält die Antwort-Nachricht mit der html-Datei und Zeit des HTML an. Nach dem Parsen der HTML-Datei findet er 10 referenzierte JPEG-Objekte
6. Schritte 1-5 werden für jedes der 10 JPEG-Objekte wiederholt



- 1b. HTTP-Server beim host www.someSchool.edu wartet auf eine TCP-Verbindung auf Port 80. Er akzeptiert die Verbindung und informiert den Client
3. HTTP-Server empfängt die Anfrage-Nachricht und erzeugt eine Response Message mit dem angefragten Objekt und sendet diese Nachricht an seinen Socket
4. HTTP-Server schließt die TCP-Verbindung

Nicht-persistentes HTTP: Antwortzeit

- Umlaufzeit (RTT – Round Trip Time)
 - Zeit für ein Packet von Client zum Server und wieder zurück
- Antwortzeit (Response Time)
 - eine RTT um TCP-Verbindung zu initiieren
 - eine RTT für HTTP Anfrage und die ersten Bytes des HTTP-Pakets
 - Transmit Time: Zeit für Dateiübertragung
- Zeit = 2 RTT + transmit time



Persistentes HTTP

- Nicht-persistentes HTTP
 - benötigt 2 RTTs pro Objekt
 - Betriebssystem-Overhead für jede TCP-Verbindung
 - Browser öffnet oft TCP-Verbindungen parallel um referenzierte Objekte zu laden
- Persistentes HTTP
 - Server lässt die Verbindung nach der Antwortnachricht offen
 - Folgende HTTP-Nachrichten zwischen den gleichen Client/Server werden über die geöffnete Verbindung versandt
 - Client sendet Anfragen, sobald es ein referiertes Objekt findet
 - höchstens eine Umlaufzeit (RTT) für alle referenzierten Objekte

HTTP-Request Nachricht

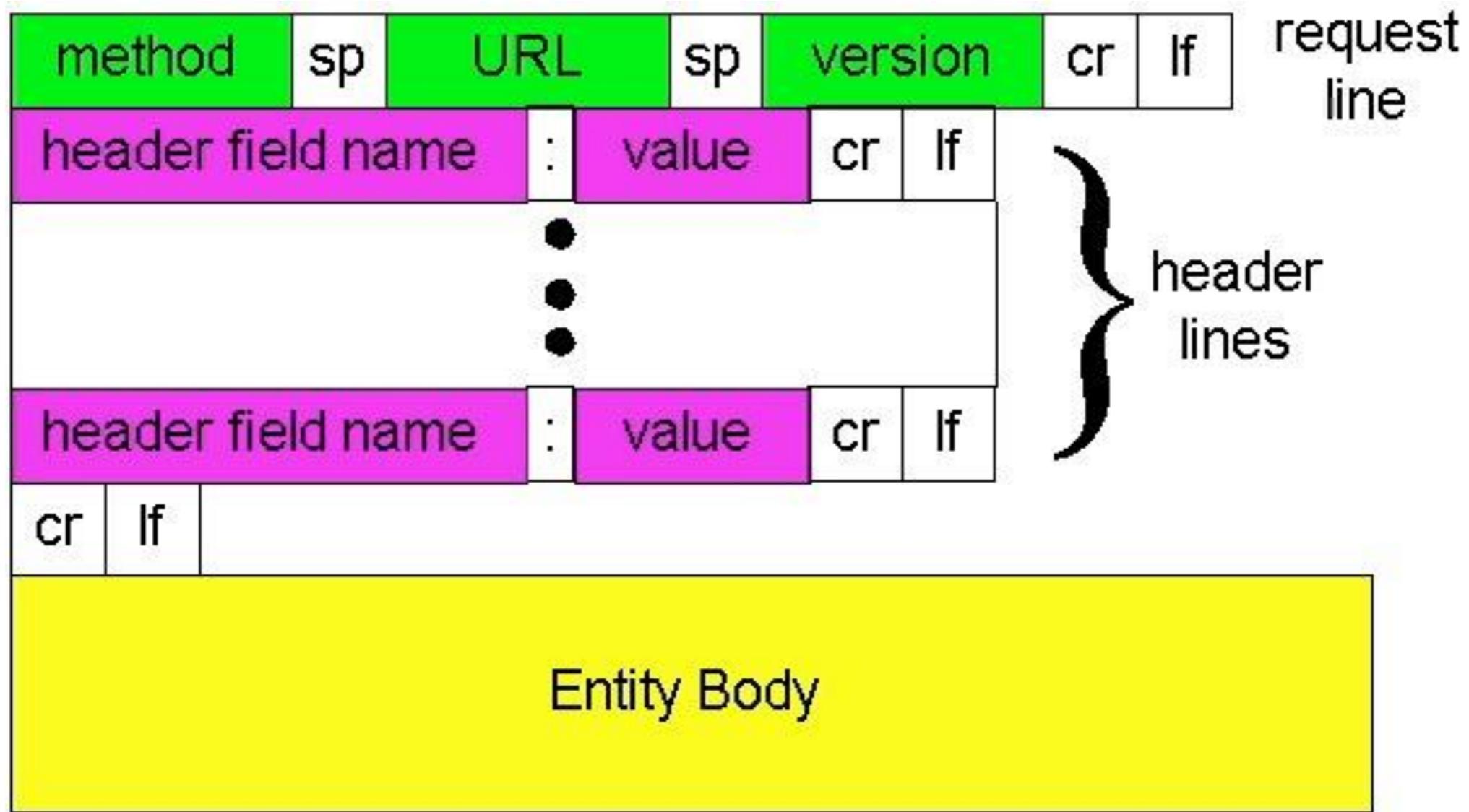
- Zwei Typen der HTTP-Nachricht: request, response
- HTTP-Request Nachricht:
 - ASCII (human-readable format)

Request Zeile
(GET, POST,
HEAD Befehle)

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
User-agent: Mozilla/4.0
Connection: close
Accept-language: fr
```

Extra Zeilenschaltung
zeigt das Ende der
Nachricht an

HTTP-Request Nachricht: Allgemeines Format



Upload

- Post
 - Web-Seiten haben öfters Leerfelder für Eingaben
 - Eingabe wird im *Body* zum Server hochgeladen
- URL-Methode
 - Verwendet GET-Methode
 - Input wird im URL-Feld der Anfrage-Nachricht gesendet:

`www.somesite.com/animalsearch?monkeys&banana`

Methoden

- HTTP/1.0
 - GET
 - POST
 - HEAD
 - fragt den Server nur nach dem Head, nicht nach dem Inhalt (*body*)
- HTTP/1.1
 - GET, POST, HEAD
 - PUT
 - lädt eine Datei im *body*-Feld zum Pfad hoch, der im URL-Feld spezifiziert wurde
 - DELETE
 - löscht Datei, die im URL-Feld angegeben wurde

HTTP-Antwort Nachricht

Status-Zeile
(protocol
status code
status phrase)

HTTP/1.1 200 OK

Kopfzeile

Connection: close
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998
Content-Length: 6821
Content-Type: text/html

Daten, e.g.,
requested
HTML file

data data data data data ...

HTTP per Telnet

1. Telnet zum Web-Server

```
telnet cis.poly.edu 80
```

Öffnet TCP Verbindung auf Port 80 (default HTTP Server-Port) von cis.poly.edu.

2. Eingabe einer GET HTTP Anfrage:

```
GET /~ross/ HTTP/1.1
Host: cis.poly.edu
```

Erzeugt einen minimalen und vollständigen GET-Request zu einem HTTP-Server

3. Was kommt als Antwort vom HTTP server?

HTTP Antwort-Status

- In der ersten Zeile der Client-Antwort-Nachricht (client response)
- Beispiele:
 - 200 OK
 - Anfrage wird beantwortet in dieser Nachricht
 - 301 Moved Permanently
 - neue Adresse für Objekt
 - Adresse folgt in der Nachricht
 - 400 Bad Request
 - Anfrage wird nicht verstanden
 - 404 Not Found
 - Angefragtes Dokument nicht vorhanden
 - 505 HTTP Version Not Supported

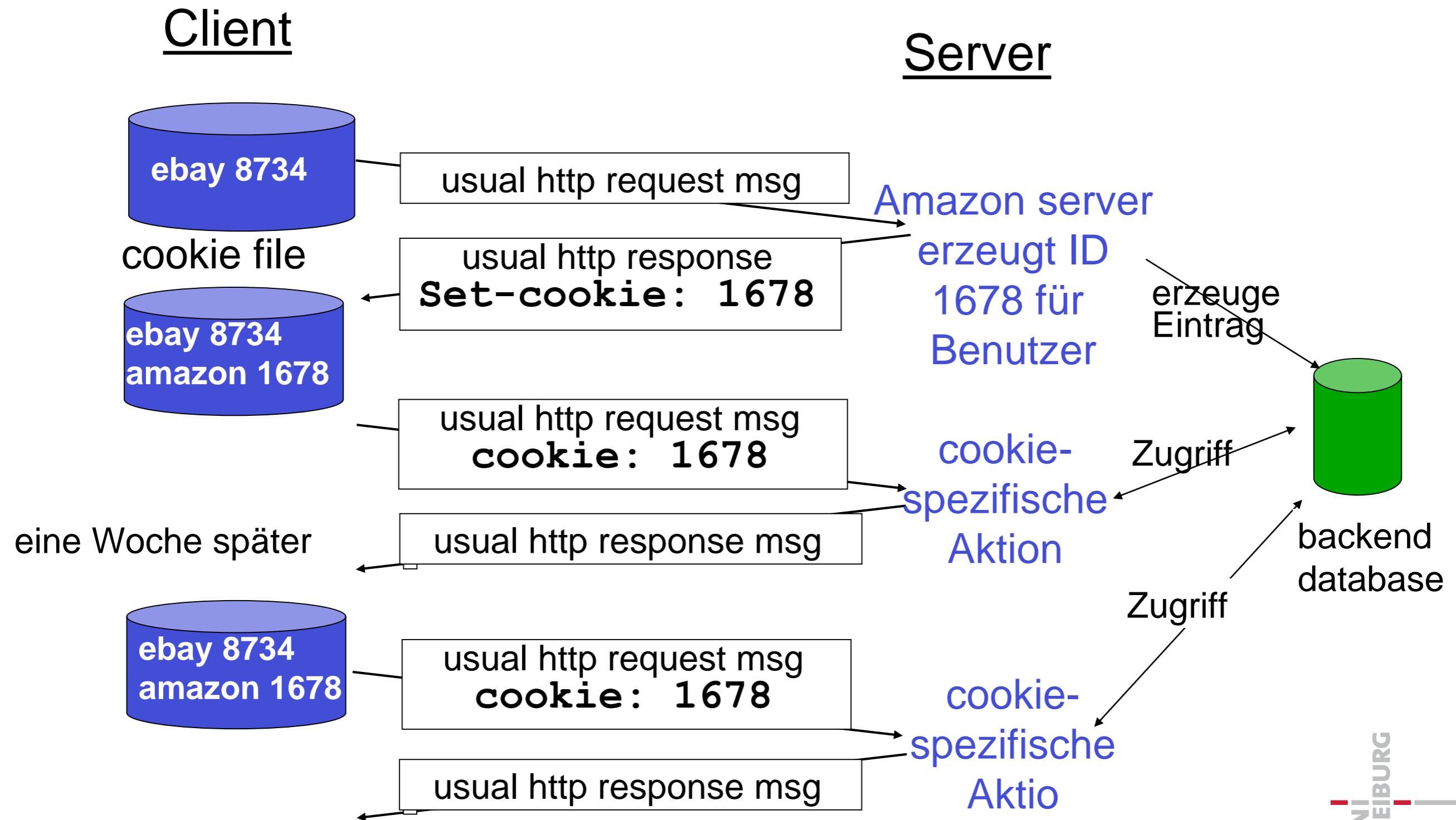
Benutzerstatus: Cookies

- Viele Web-Sites verwenden Cookies
- Vier Komponenten
 - 1) Cookie Kopf-Zeile der HTTP-Antwort-Nachricht (Response Message)
 - 2) Cookie-Kopf-Zeile in HTTP-Anfrage-Nachricht (Request Message)
 - 3) Cookie-Datei auf dem Benutzer-Rechner
 - wird vom Web-Browser des Benutzers unterhalten
 - 4) Datenbank auf der Web-Site (des Servers)

Benutzerstatus: Cookies

- Beispiel:
- Susan
 - surft das Web vom PC
 - besucht E-Commerce-Site *Amazon* zum ersten Mal
 - wenn die HTTP-Anfrage die Site erreicht, erzeugt die Web-Site
 - eindeutige ID
 - Eintrag in der Datenbank des Web-Servers

Cookies: Erzeugen einer Status-Information

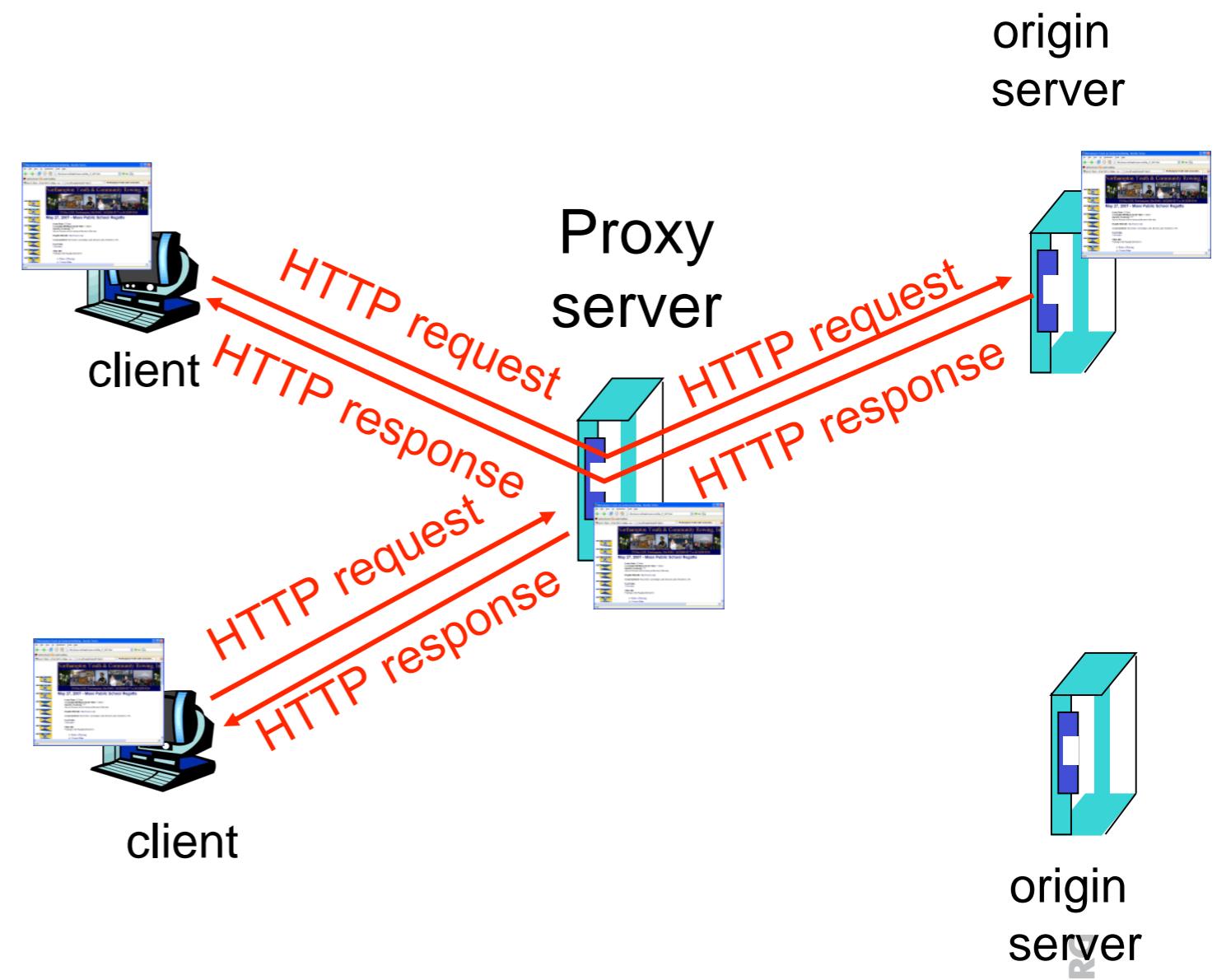


Cookies

- Cookies erlauben
 - Authentifikation
 - Einkaufswagen
 - Empfehlungen
 - Sitzungs-Status des Benutzers (Web Mail)
- Wie man den Status unterhält
 - speichert Zustand zwischen verschiedenen Transaktionen
 - Cookies: HTTP Nachrichten transportieren den Status
- Cookies und Privatsphäre
 - Cookies übergeben der Web-Site eine Menge von Informationen
 - z.B. Name, E-Mail, Kaufverhalten, etc.

Web Caches (Proxy Server)

- Ziel:
 - Client-Anfragen erfüllen ohne den Original-Server zu verwenden
- Benutzer greift auf das Web per Cache zu
 - Hierfür wird Browser konfiguriert
- Browser sendet alle HTTP-Anfragen zum Cache
 - Ist das Objekt im Cache, dann wird das Objekt geliefert
 - ansonsten liefert der Original-Server an den Proxy-Server
 - dieser liefert dann das Objekt an den Client

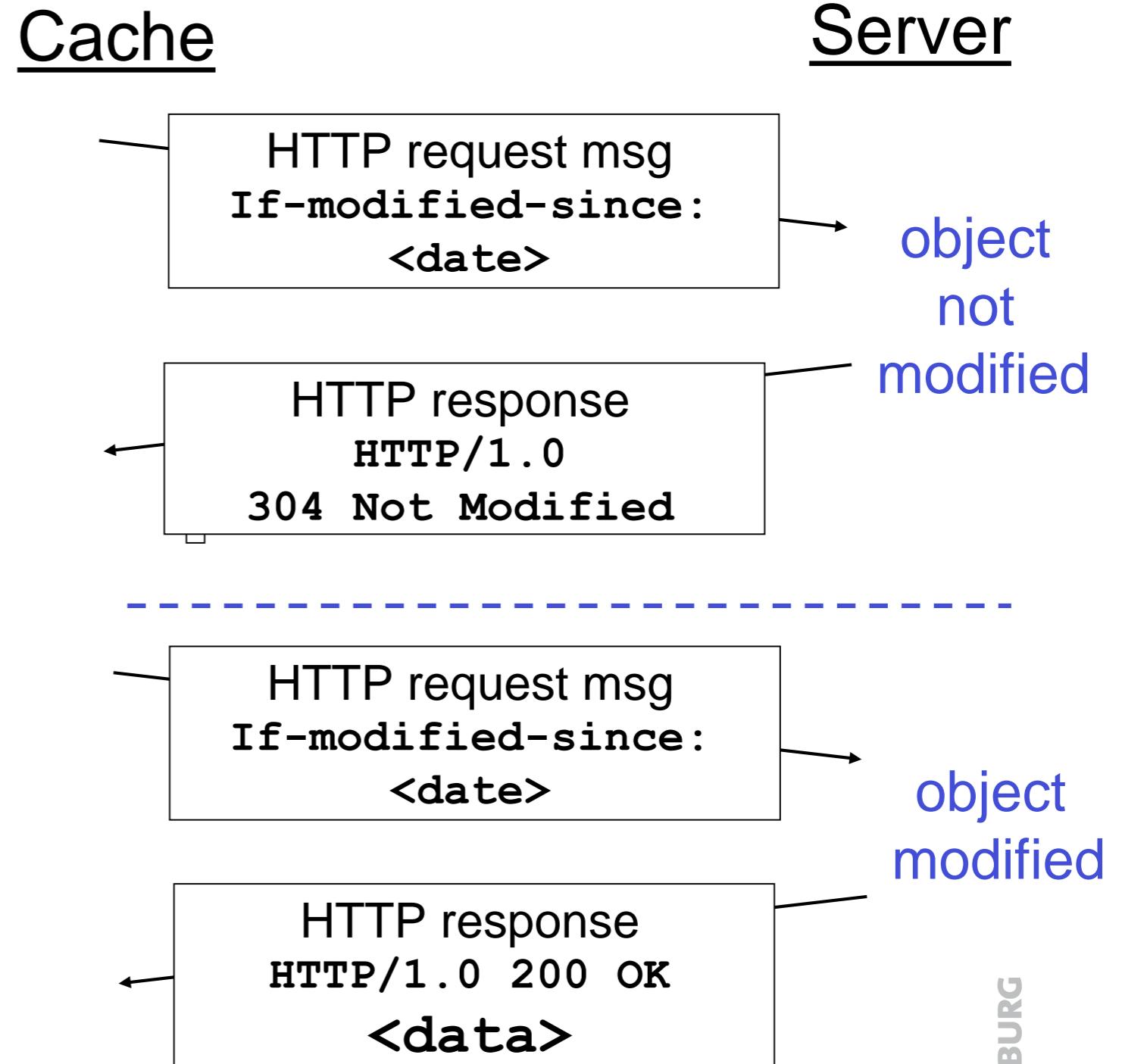


Web-Caching

- Cache fungiert als Client und Server
 - typisch wird der Cache vom ISP (Internet Service Provider) bereit gestellt
- Warum
 - reduziert Antwortzeit für Client-Anfragen
 - reduziert den Verkehr über die Leitungen zu anderen ISPs
 - ermöglicht „kleinen“ Web-Servern effizient Inhalte zu verteilen

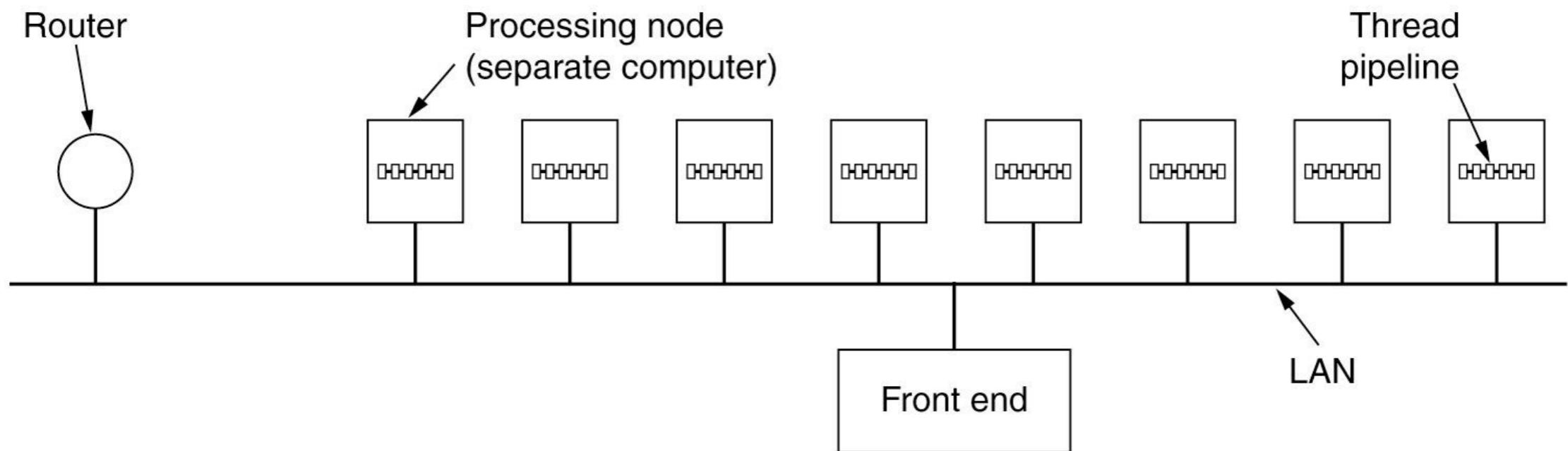
Conditional GET

- Ziel: Objekt soll nicht gesendet werden, falls der Cache die aktuelle Version hat
- Cache: gibt den Zeitstempel der gecachten Kopie einer HTTP-Anfrage
 - If-modified-since: <date>
- Server: Antwort enthält kein Objekt, falls die gecachte Kopie aktuell ist
 - HTTP/1.0 304 Not Modified



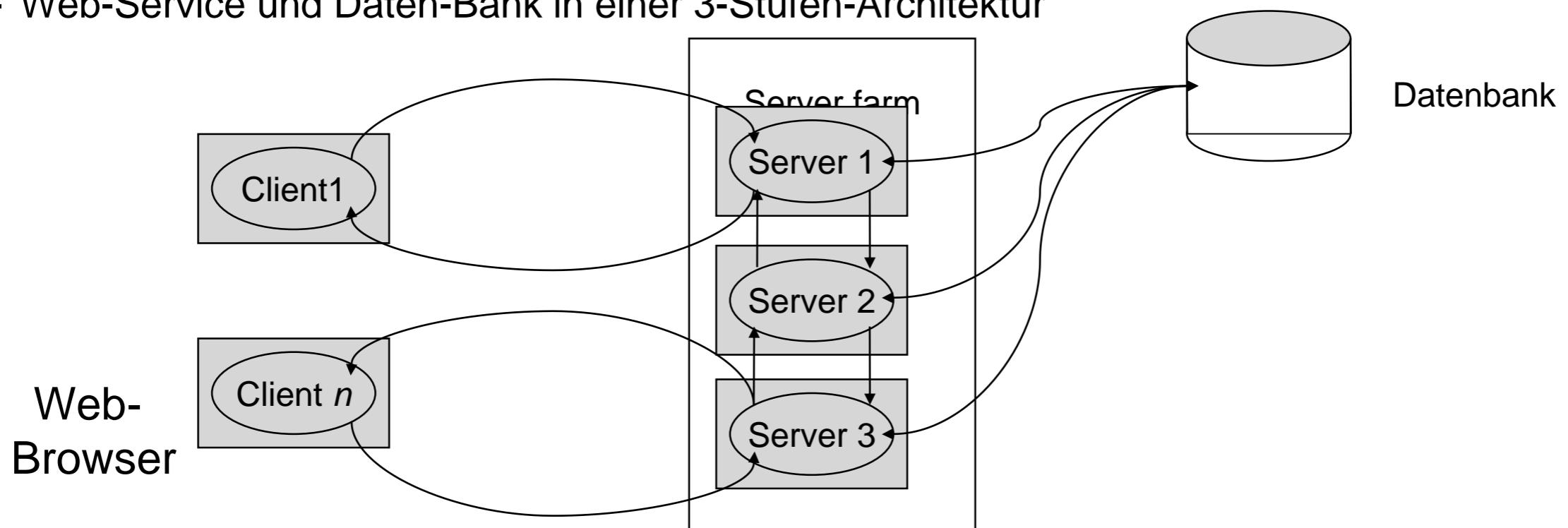
Server-Farm

- Um die Leistungsfähigkeit auf der Server-Seite zu erhöhen
 - wird eine Reihe von Web-Servern eingesetzt
- Front end
 - nimmt Anfragen an
 - reicht sie an separaten Host zur Weiterbearbeitung weiter



Web-Servers und Datenbanken

- Web-Server stellen nicht nur statische Web-Seiten zur Verfügung
 - Web-Seiten werden auch automatisch erzeugt
 - Hierzu wird auf eine Datenbank zurückgegriffen
 - Diese ist nicht statisch und kann durch Interaktionen verändert werden
- Problem:
 - Konsistenz
- Lösung
 - Web-Service und Daten-Bank in einer 3-Stufen-Architektur



Beispiel: Google Data Centers

- Kosten eines Daten-Centers: 600 Mio US\$
- Google investierte 2007 2,4 Mrd. US\$ in Daten-Center
- Jedes Daten-Center verbraucht 50-100 MW

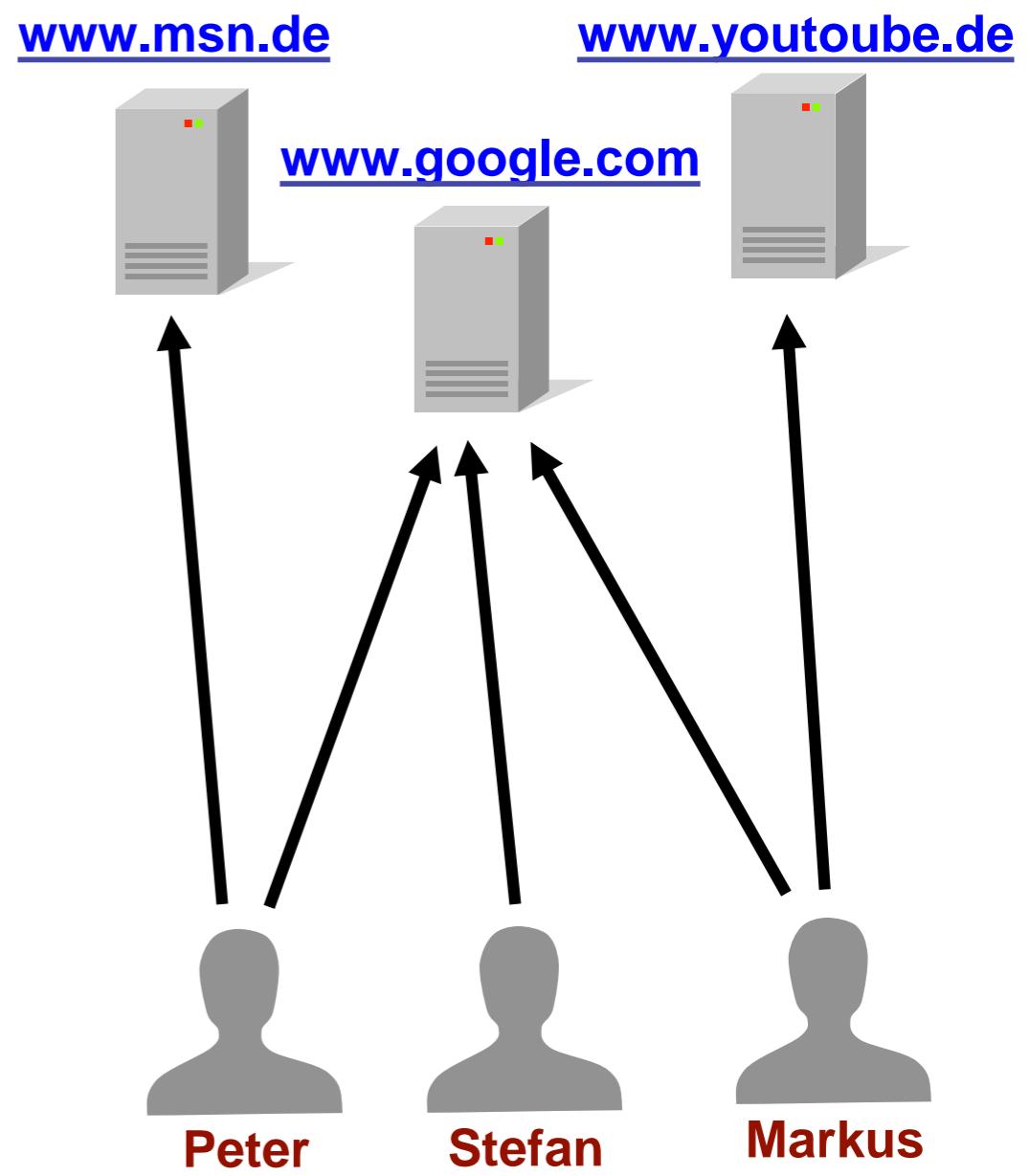


Content Distribution Networks (CDN)

- Eine koordinierte Menge von Caches
 - Die Last großer Web-Sites wird verteilt auf global verteilte Server-Farmen
 - Diese übernehmen Web-Seiten möglichst verschiedener Organisationen
 - z.B. News, Software-Hersteller, Regierungen
 - Beispiele: Akamai, Digital Island
 - Cache-Anfragen werden auf die regional und lastmäßig bestgeeigneten Server umgeleitet
- Beispiel Akamai:
 - Durch verteilte Hash-Tabellen ist die Verteilung effizient und lokal möglich

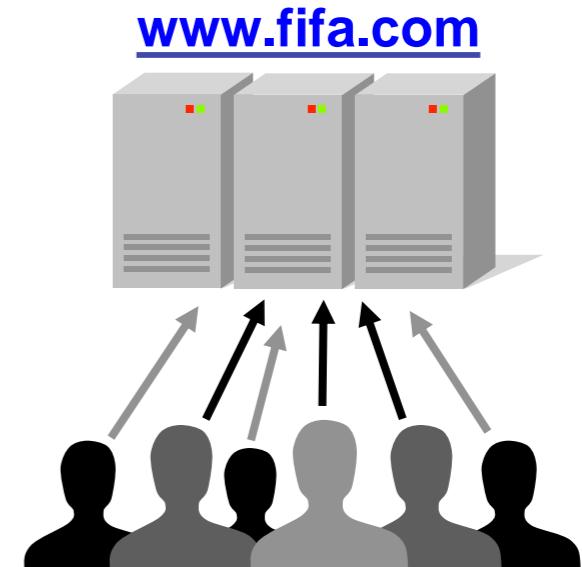
WWW-Lastbalancierung

- Für Surfen im Web typisch:
 - Web-Server bieten Web-Seiten an
 - Web-Clients fordern Web-Seiten an
- In der Regel sind diese Mengen disjunkt
- Eingehende Anforderungen belasten Web-Server hinsichtlich:
 - Übertragungsbandbreite
 - Rechenaufwand (Zeit, Speicher)

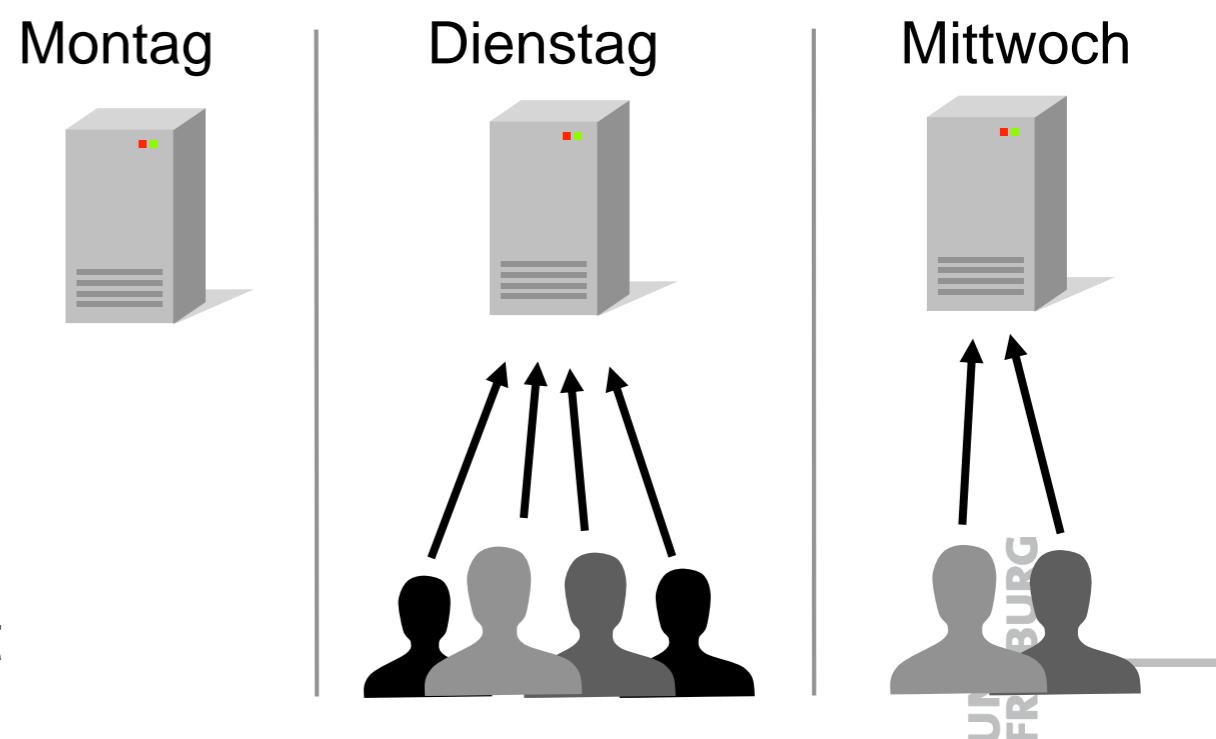


Lastanforderungen

- Einige Web-Server haben immer hohe Lastanforderungen
 - Z.B. Nachrichten-Sites, Suchmaschinen, Web-verzeichnisse
 - Für permanente Anforderungen müssen Server entsprechend ausgelegt werden

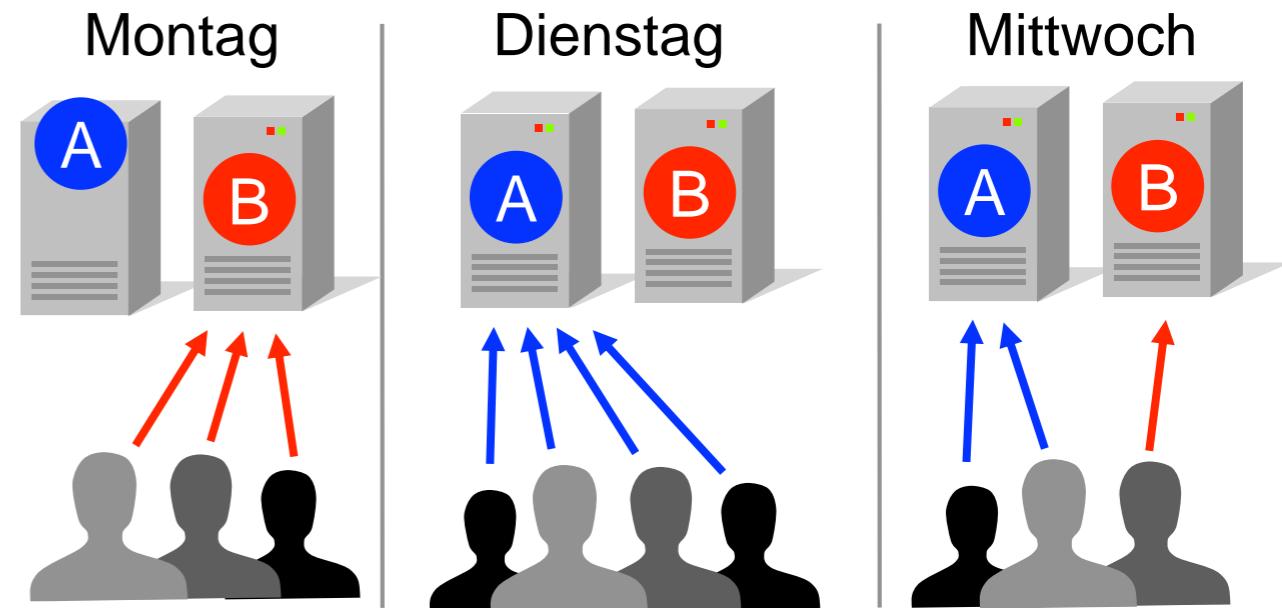


- Andere leiden unter hohen Fluktuationen
 - z. B. bei besonderen Ereignissen:
 - fifa.com (Fussball-EM)
 - t-mobile.de (iPhone 6 Einführung)
 - Server-Erweiterung nicht sinnvoll
 - Bedienung der Anfragen aber erwünscht

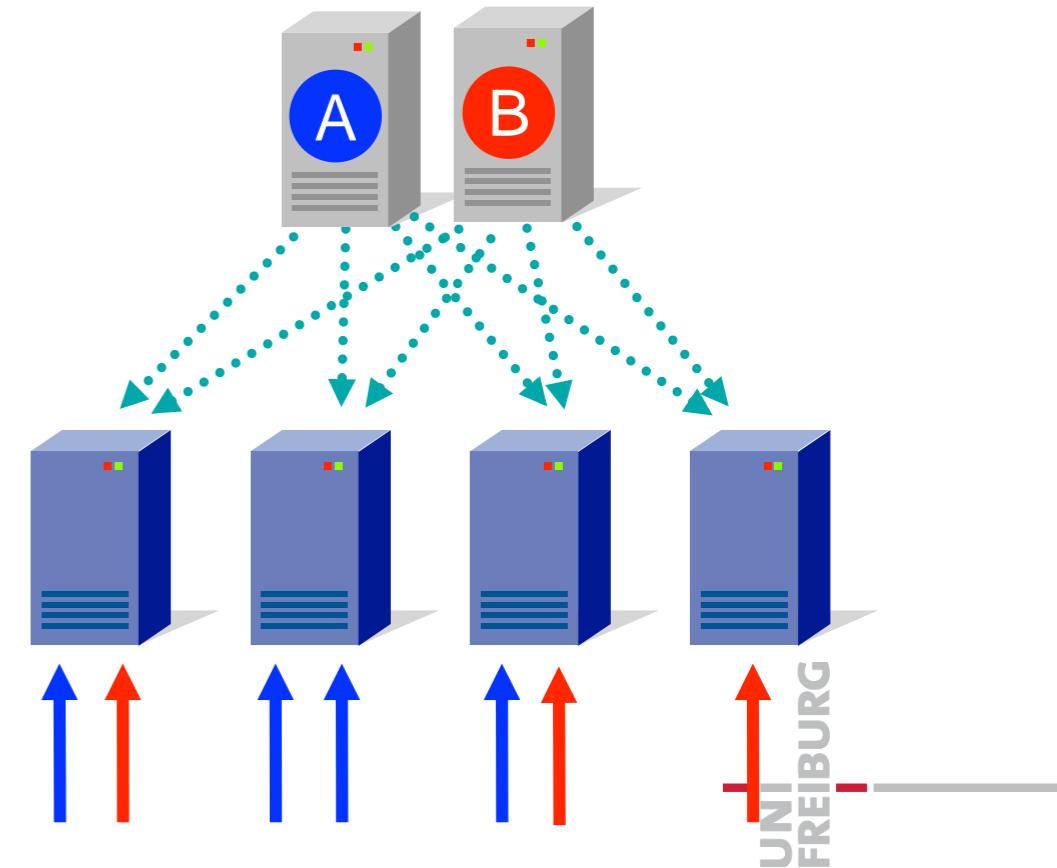


Lastbalancierung im WWW

- Fluktuationen betreffen meistens einzelne Server

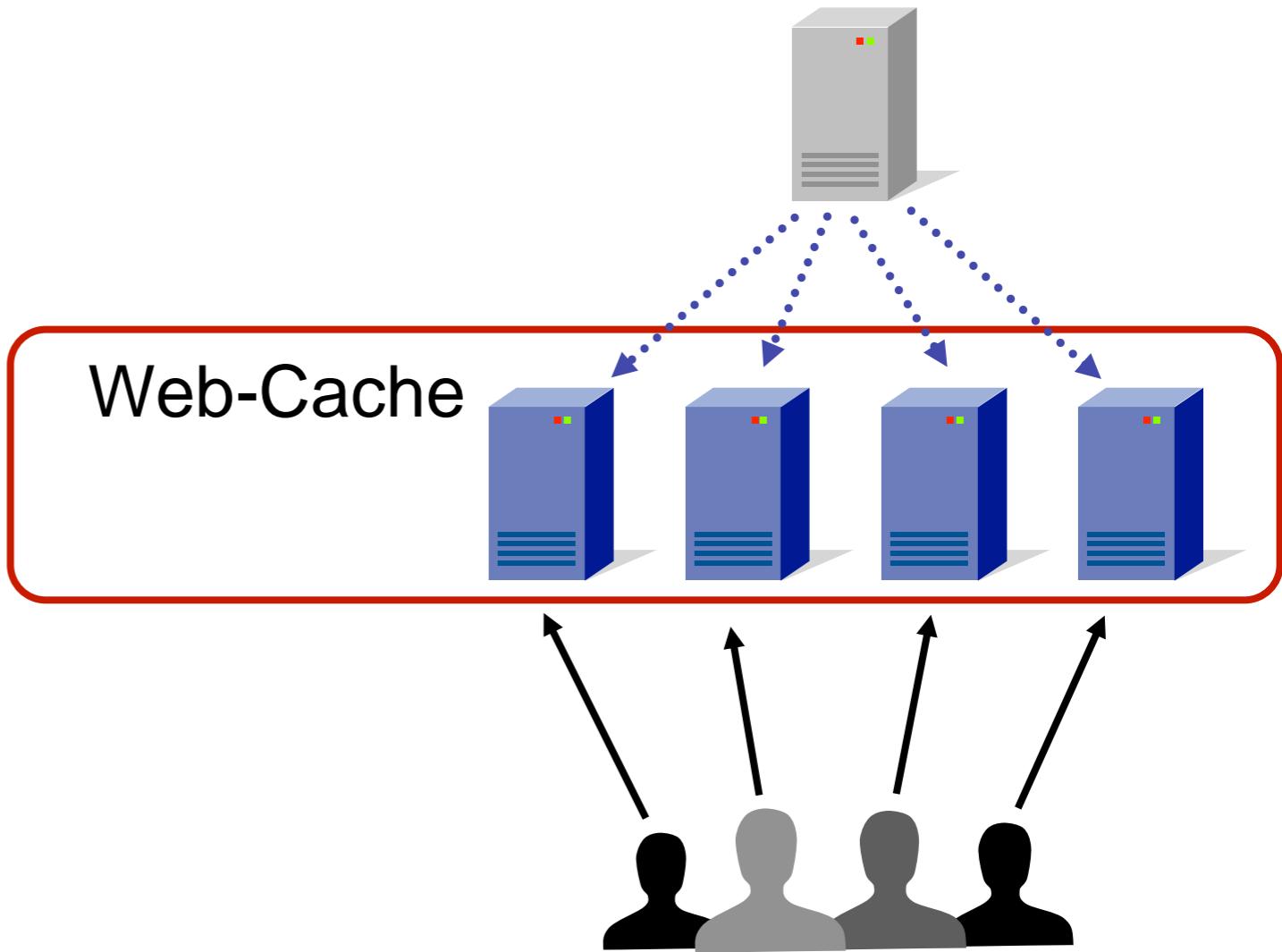


- (Kommerzielle) Lösung
 - Dienstleister bieten Ausweich-(Cache-)Server an
 - Viele Anforderungen werden auf diese Server verteilt
- Aber wie?



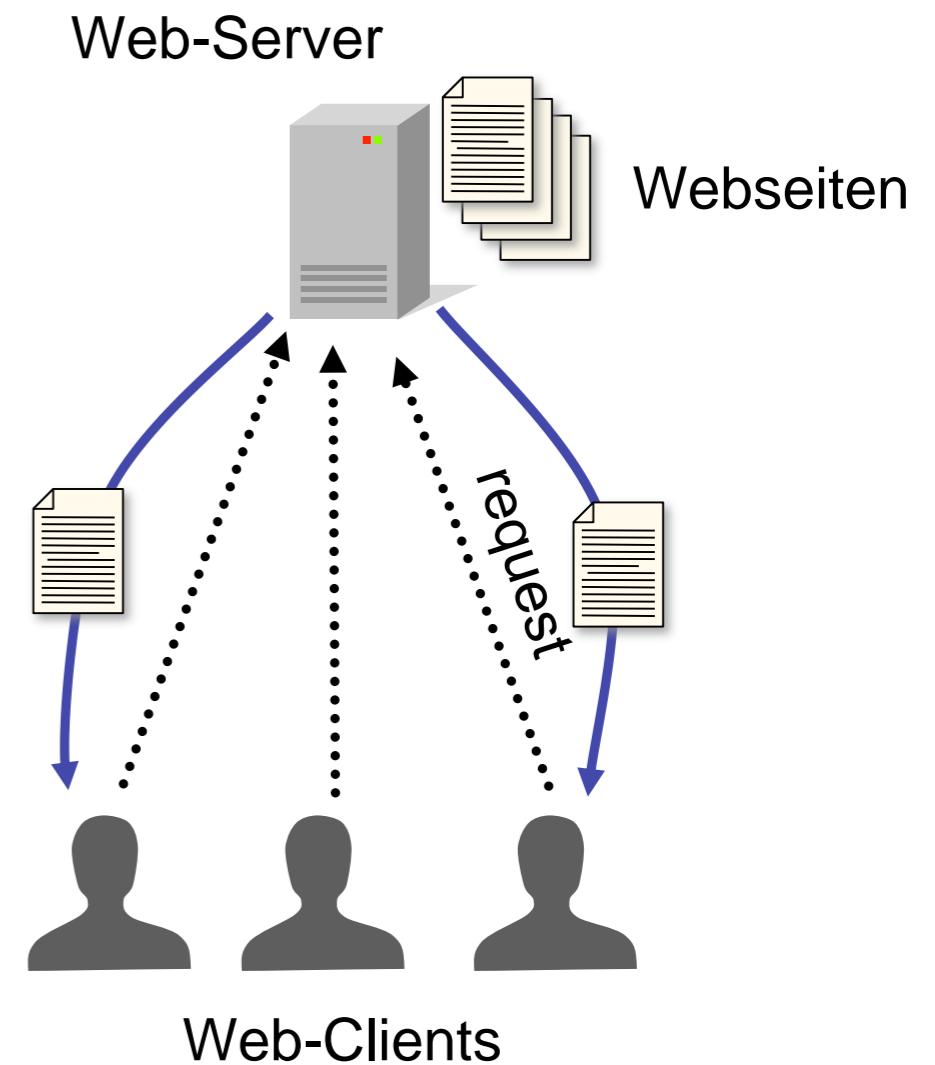
Web-Caching

- Leighton, Lewin, et al.
STOC 97
 - *Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web*
 - Passen bestehende Verfahren für dynamische Hash-Funktionen an WWW-Anforderungen an
- Leighton und Lewin (MIT) gründen Akamai 1997



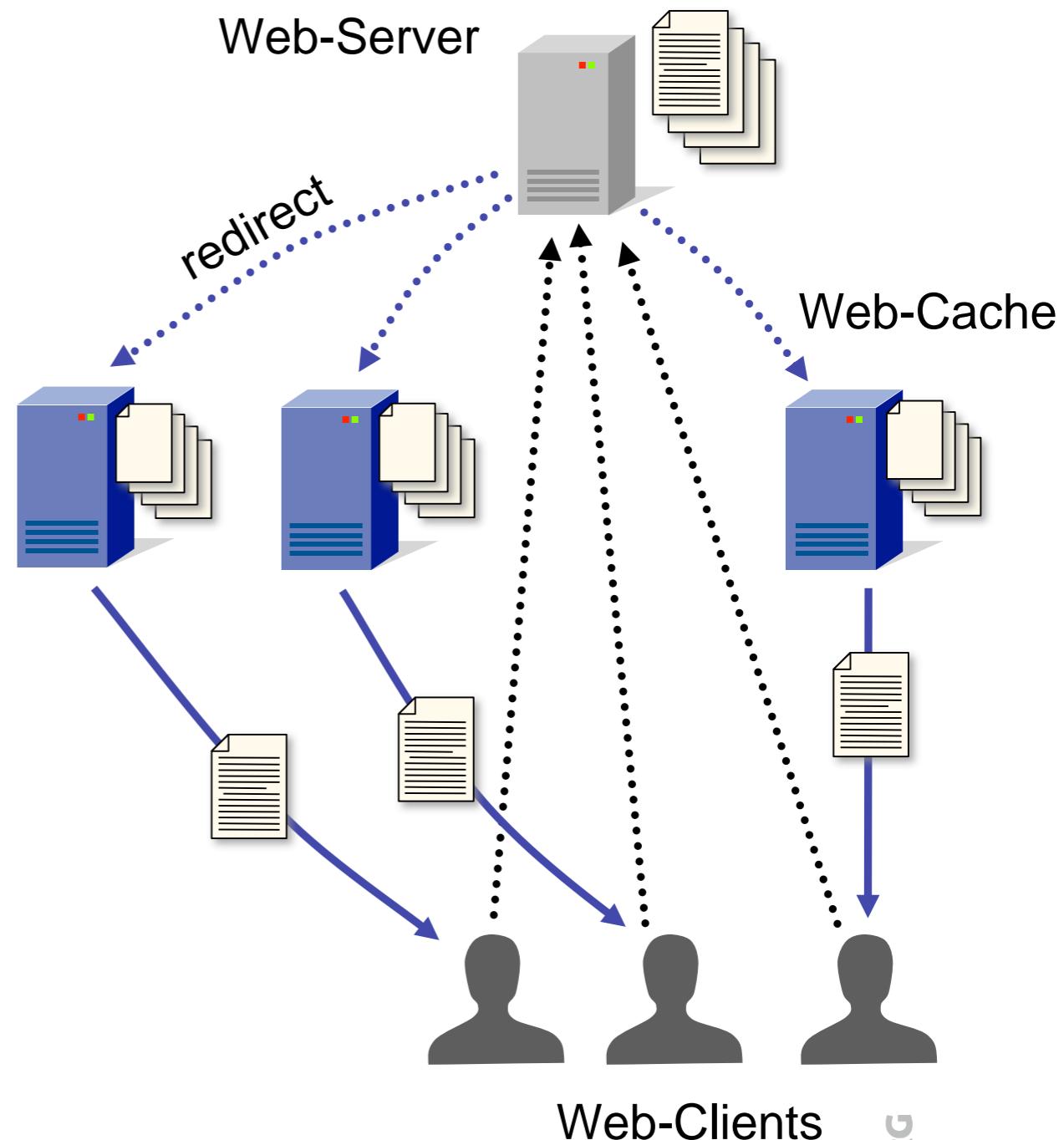
Ausgangssituation

- Ohne Lastbalancierung:
 - Jeder Browser (Web-Client) belegt einen Web-Server für eine Web-Site
- Vorteil:
 - Einfach
- Nachteil:
 - Der Server muss immer für den Worst-Case ausgelegt werden



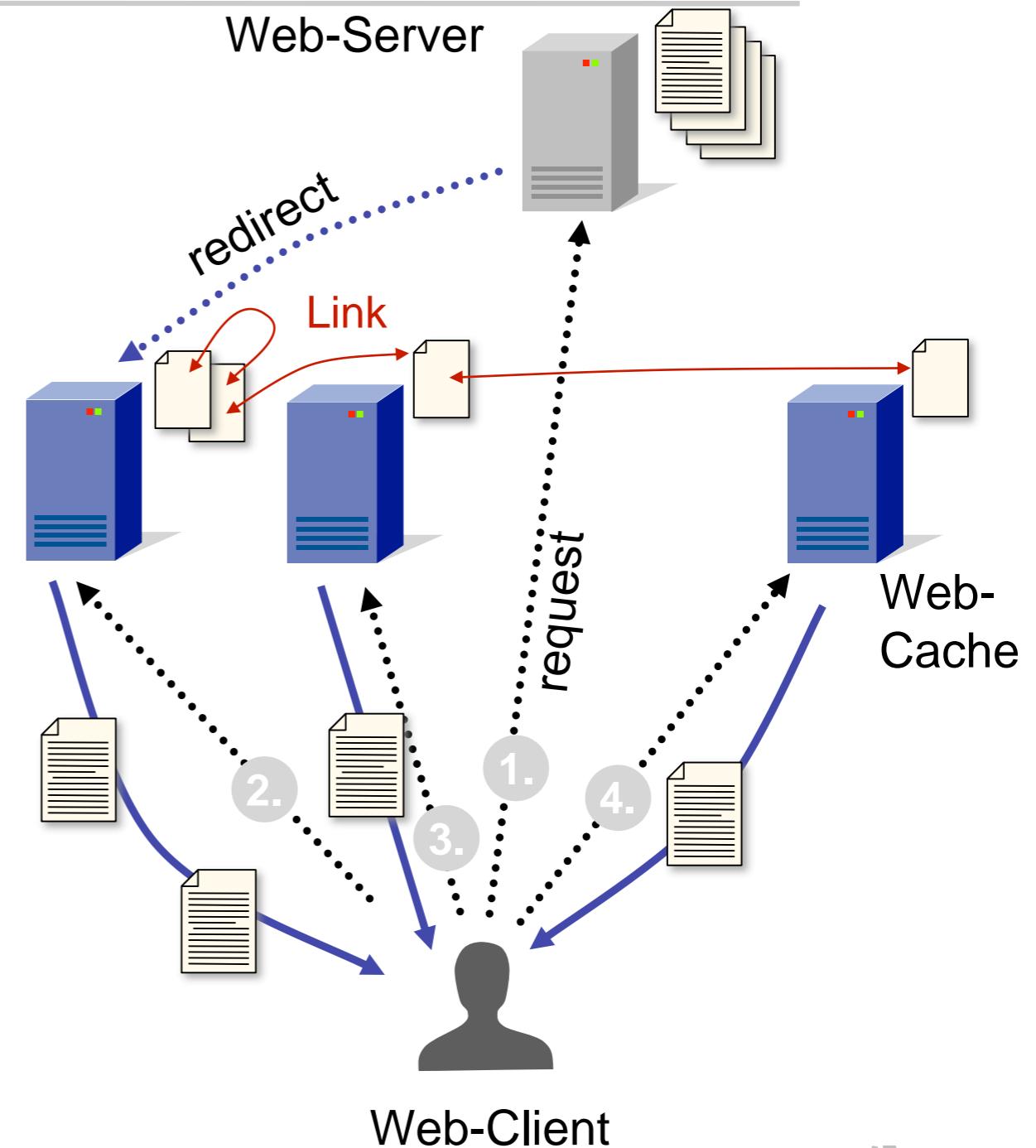
Site Caching

- Ganze Web-Site wird auf verschiedene Web-Caches kopiert
- Browser fragt bei Web-Server nach Seite
- Web-Server leitet Anfrage auf Web-Cache um (redirect)
- Web-Cache liefert Web-Seite aus
- Vorteil:
 - Gute Lastbalancierung für Seitenverteilung
- Nachteil:
 - Bottleneck: Redirect
 - Großer Overhead durch vollständige Web-Site-Replikationen



Proxy Caching

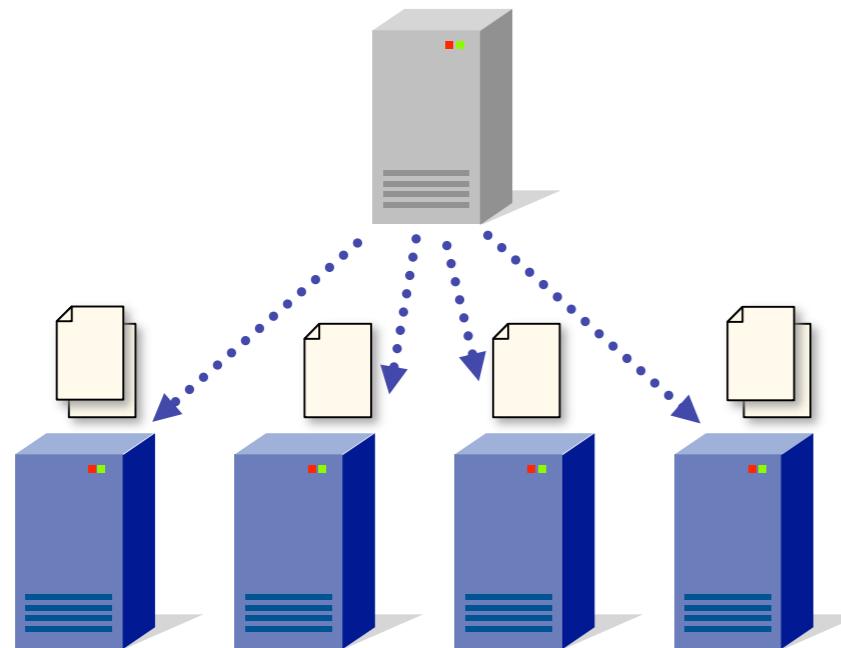
- Jede Web-Seite wird auf einige (wenige) Web-Caches verteilt
- Nur Startanfrage erreicht Web-Server
- Links referenzieren auf Seiten im Web-Cache
- Dann surft der Web-Client nur noch auf den Web-Cache
- Vorteil:
 - Kein Bottleneck
- Nachteil:
 - Lastbalancierung nur implizit möglich
 - Hohe Anforderung an Caching-Algorithmus



Anforderungen an Caching-Algorithmus

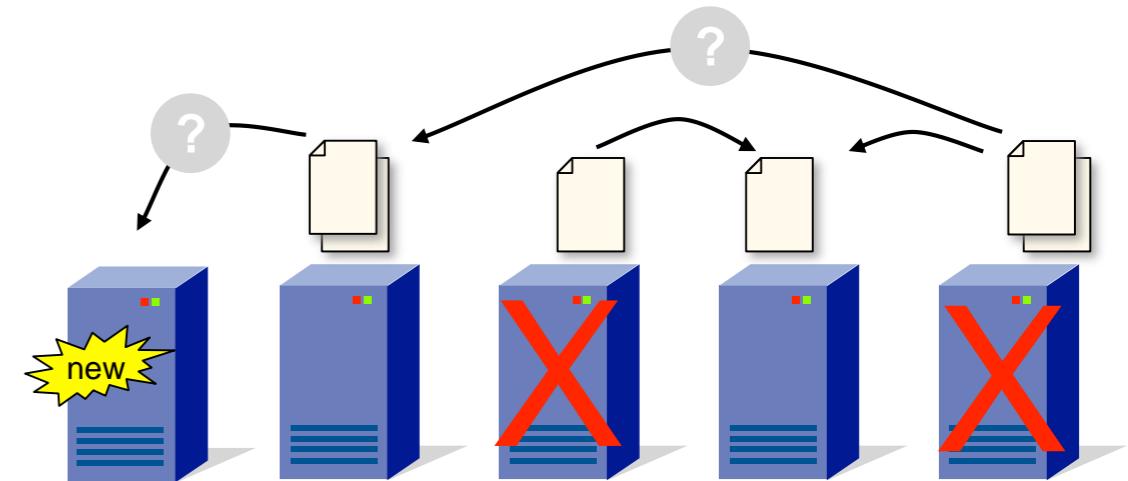
Balance

Gleichmäßige Verteilung der Seiten



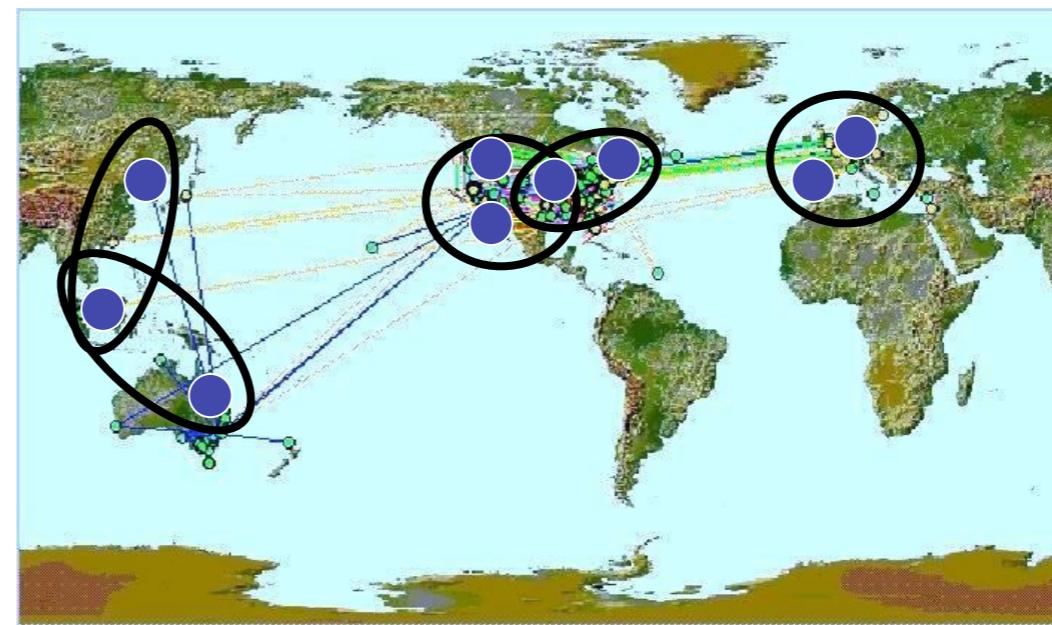
Dynamik

Effizientes Einfügen/Löschen von neuen Web-Cache-Servern



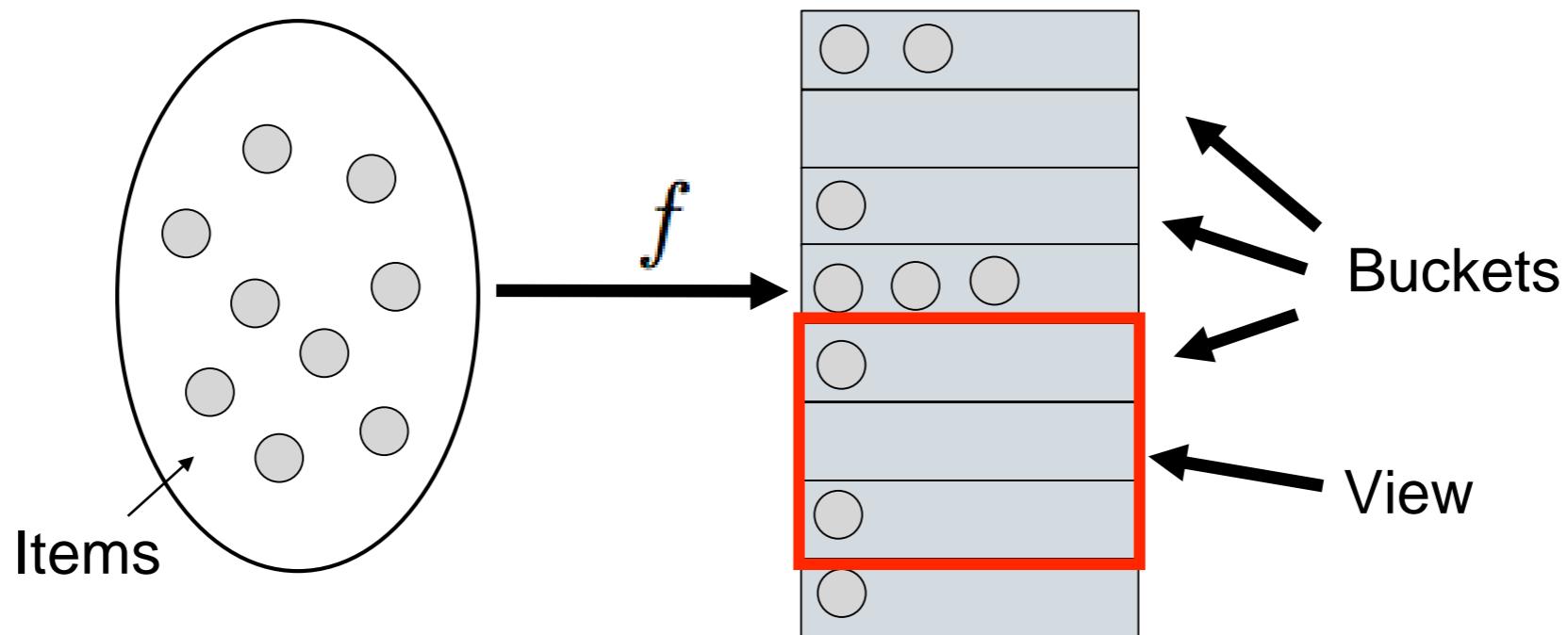
Views

Web-Clients „sehen“ unterschiedliche Menge von Web-Caches



Ranged Hash-Funktionen oder Verteilte Hash-Funktionen

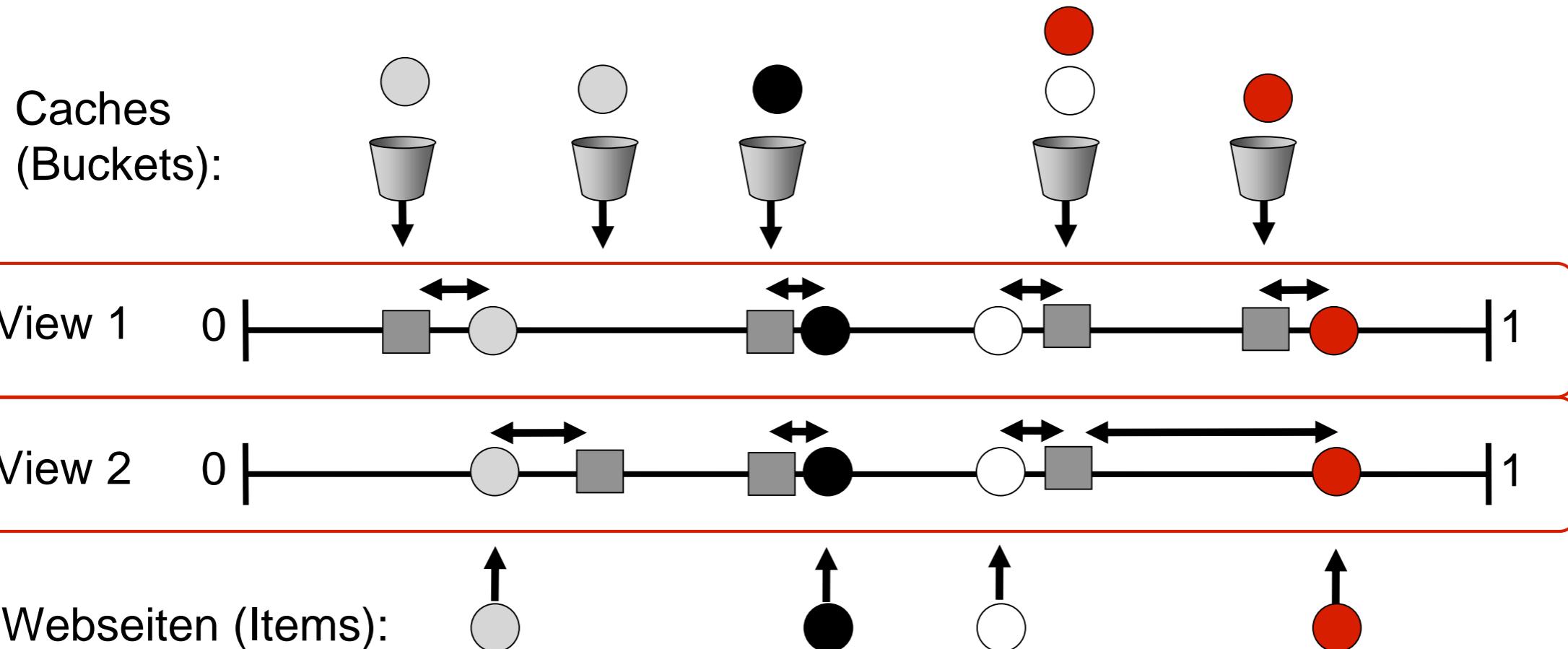
- Gegeben:
 - Elemente (Items)
 - Caches (Buckets)
 - Views: Menge von Caches
- Ranged Hash-Funktion:
 - Zuordnung eines Elements zu einem Cache in einem View



Anforderungen an Ranged Hash-Funktionen

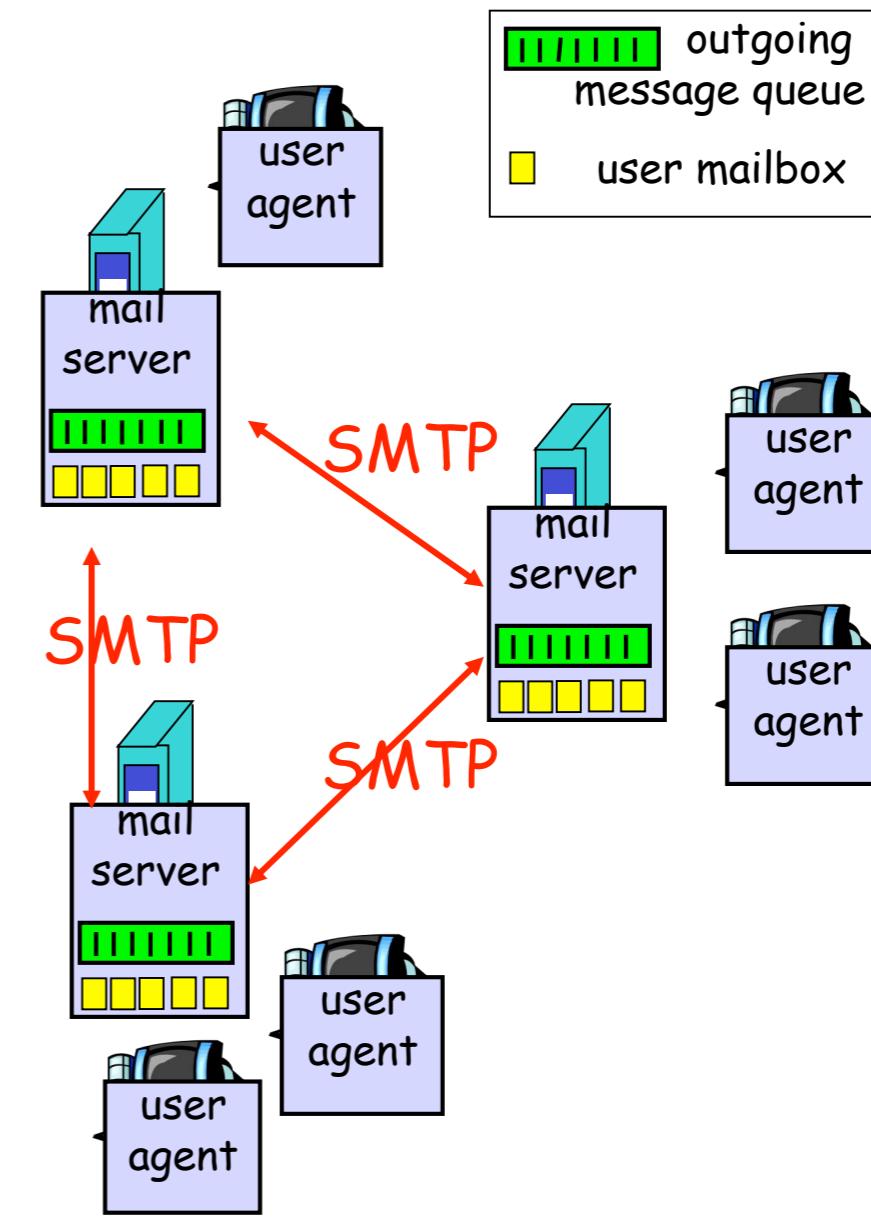
- Monotonie
 - nach dem Hinzufügen neuer Caches (Buckets) sollten keine Seiten (Items) zwischen alten Caches verschoben werden
- Balance
 - Alle Caches sollten gleichmäßig ausgelastet werden
- Spread (Verbreitung, Streuung)
 - Eine Seite sollte auf eine beschränkte Anzahl von Caches verteilt werden
- Load
 - Kein Cache sollte wesentlich mehr als die durchschnittliche Anzahl von Seiten enthalten

Distributed Hash Tables als Lösung



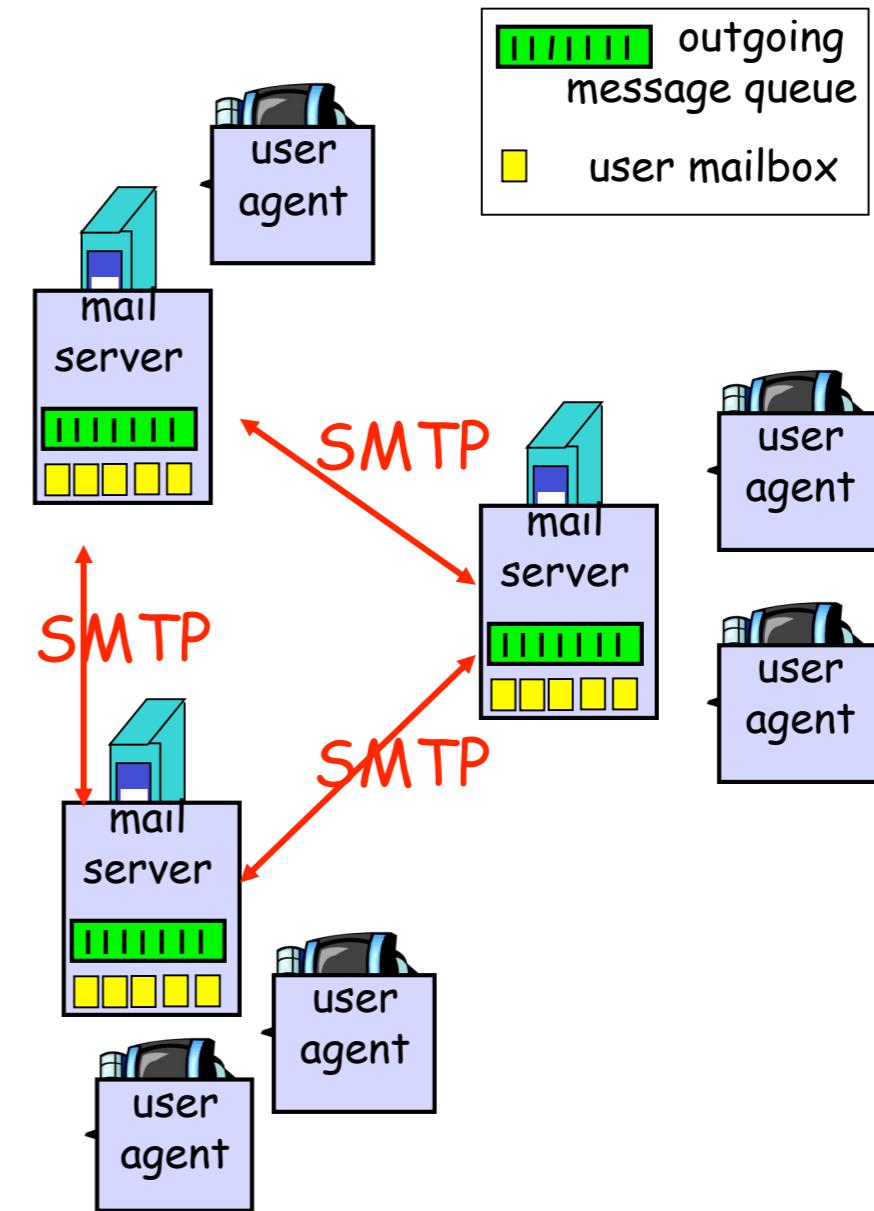
Electronic Mail

- Hauptkomponenten
 - user agents
 - mail servers
 - simple mail transfer protocol: SMTP
- User Agent
 - Mail Client
 - Erstellen, ändern und lesen von E-Mail-Nachrichten
 - z.B. Eudora, Outlook, pine, Mozilla Thunderbird
 - abgehende und ankommende Nachrichten werden auf dem Server gespeichert



Mail-Servers

- Mailbox speichert eingehende Nachrichten für den User
- Nachrichten-Warteschlange (queue) der zu versendenden Nachrichten
- SMTP-Protocol zwischen Mail-Servern um E-Mail-Nachrichten zu schicken

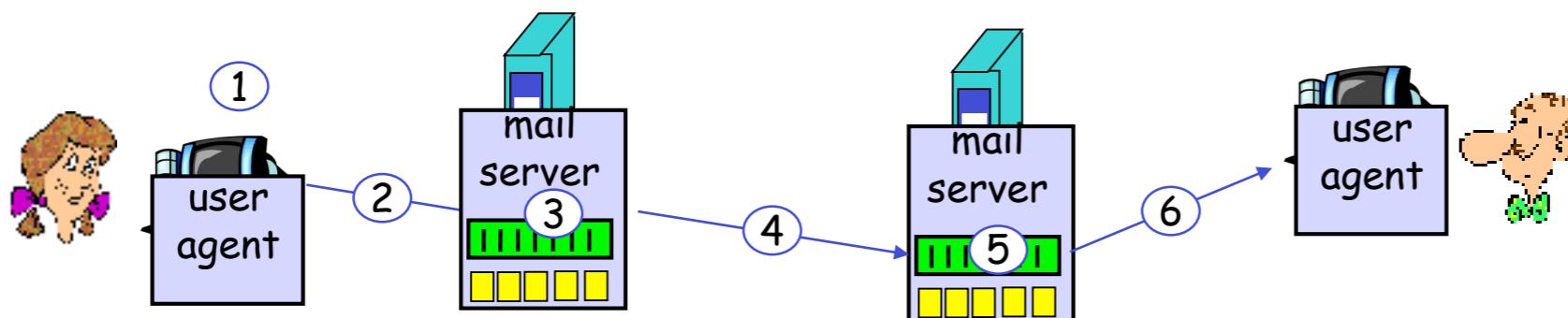


Electronic Mail: SMTP [RFC 2821]

- verwendet TCP um zuverlässig E-Mail-Nachrichten vom Client auf Port 25 zu verschicken
- Direkte Übertragung von Absender-Server zum Empfangs-Server
- 3 Phasen in der Übertragung
 - Handshake
 - Transfer der Nachricht
 - Abschluss
- Befehle und Antwort
 - Befehle als ASCII text
 - Antwort: Status-Code und Kurzbeschreibung
- Nachrichten sind in 7-bit ASCII

Beispiel: Alice sendet eine Nachricht an Bob

- 1) Alice verwendet UA um die Nachricht zu erzeugen mit Eintrag “to” bob@someschool.edu
- 2) Alice UA sendet die Nachricht zu ihren Mail-Server
 - Nachricht wird in der Nachrichtenwarteschlange platziert
- 3) Client-Seite des SMTP öffnet TCP-Verbindung mit Bobs Mail-Server
- 4) SMTP Client sendet Alice Nachricht über die TCP-Verbindung
- 5) Bobs Mail-Server schreibt die Nachricht in Bobs Mailbox
- 6) Bob ruft seinen User Agent auf, um die Nachricht zu lesen



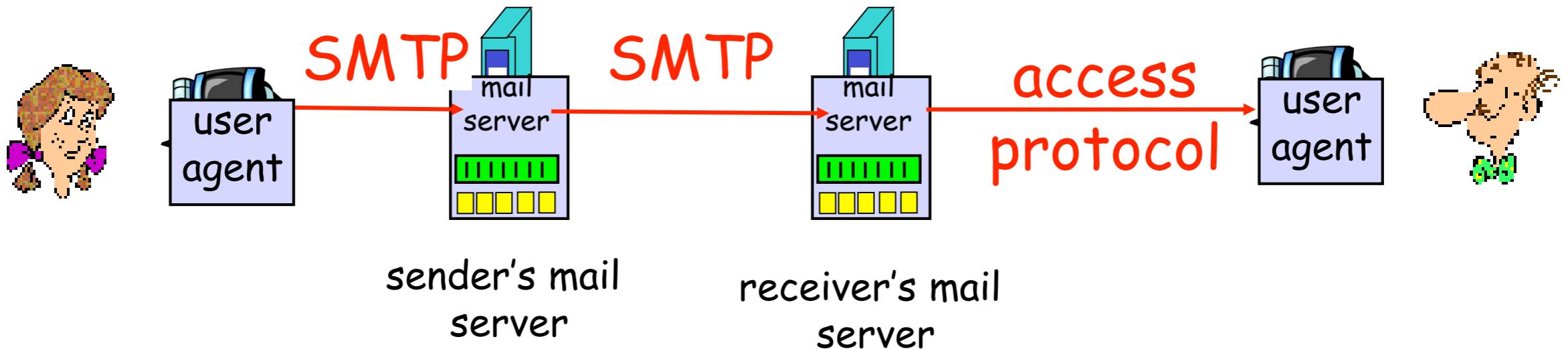
Beispiel SMTP Interaktion

```
S: 220 hamburger.edu
C: HELO crepes.fr
S: 250 Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection
```

SMTP - Abschließende Bemerkungen

- SMTP
 - verwendet persistente Verbindungen
 - verlangt Nachrichten (header & body) in 7-bit ASCII
 - SMTP-Server verwenden „CRLF.CRLF“ um das Ende einer Nachricht zu beschreiben
- Vergleich mit HTTP:
 - HTTP: pull
 - SMTP: push
 - beide haben ASCII Befehls- und Antwort-Interaktion und Status-Codes
- HTTP
 - jedes Objekt wird in eigener Nachricht verpackt
- SMTP
 - verschiedene Objekte werden in einer Multipart-Nachricht verschickt

Mail-Zugriffsprotokolle



- SMTP: Auslieferung und Speicher zum Server des Empfängers
- Mail-Zugriffsprotokoll: E-Mail-Abruf vom Server
 - POP: Post Office Protocol [RFC 1939]
 - Authentifizierung (zwischen Agent und Server) und Download
 - IMAP: Internet Mail Access Protocol [RFC 1730]
 - mehr Features und komplexer
 - Bearbeitung von gespeicherten Nachrichten **auf dem Server**
 - HTTP: gmail, Hotmail, Yahoo! Mail, web.de, etc.

POP3 und IMAP

- POP3 (Post-Office-Protocol)
 - User kann im “download and delete” Modus E-Mails einmalig herunterladen
 - User kann E-Mails noch einmal lesen, wenn er den Client wechselt:
 - “Download-and-keep”: Kopien der Nachricht auf verschiedenen Clients
 - POP3 ist zustandslos (stateless) von einer Sitzung zur nächsten
- IMAP (Internet Message Access Protocol)
 - hält alle Nachrichten an einem Ort: dem Server
 - erlaubt dem User die Nachrichten in Ordnern zu organisieren
 - IMAP speichert den Benutzer-Status zwischen Sitzungen
 - Namen der Ordner und Zuordnung zwischen Nachrichten-ID und Ordnernamen

Meilensteine P2P Systeme

- Napster 1999-2000
 - Filesharing, nur rudimentäres P2P
- Gnutella 2000
 - 1. echtes P2P-Netzwerk
- Edonkey 2000
 - Mehr Filesharing als P2P
- FreeNet 2000
 - Anonymisiertes P2P-Netzwerk
- FastTrack 2001
 - KaZaa, Morpheus, Grokster
- BitTorrent 2001
- Skype 2003
 - VoIP (voice over IP), Chat, Video

Milestones Theorie

- Distributed Hash-Tables (DHT) (1997)
 - Ziel: Lastbalancierung für Web-Server
- CAN (2001)
 - DHT-Netzwerk-Struktur
- Chord (2001)
 - Erstes effiziente P2P-Netzwerk
 - Logarithmische Suchzeit
- Pastry/Tapestry (2001)
 - Effizientes verteiltes P2P-Netzwerk unter Verwendung des Plaxton-Routing
- Und viele andere Ansätze
 - Viceroy, Distance-Halving, Koorde, Skip-Net, P-Grid, ...
- In den letzten fünf Jahren:
 - Network Coding for P2P
 - Game theory in P2P
 - Anonymity, Security

Was ist ein P2P-Netzwerk

- Was ist P2P **NICHT**?
 - Ein Client-Server network
- Etymologie: peer
 - lateinisch: par = gleich
 - Standesgleich
 - P2P, Peer-to-Peer: Beziehung zwischen gleichwertigen Partnern
- Definition
 - Ein Peer-to-Peer Network ist ein Kommunikationsnetzwerk im Internet
 - ohne zentrale Kontrolle
 - mit gleichwertigen, unzuverlässigen Partnern

Distributed Hash-Table (DHT)

■ Hash-Tabellen

- nicht praktikabel in P2P

■ Verteilte Hash-Tabellen

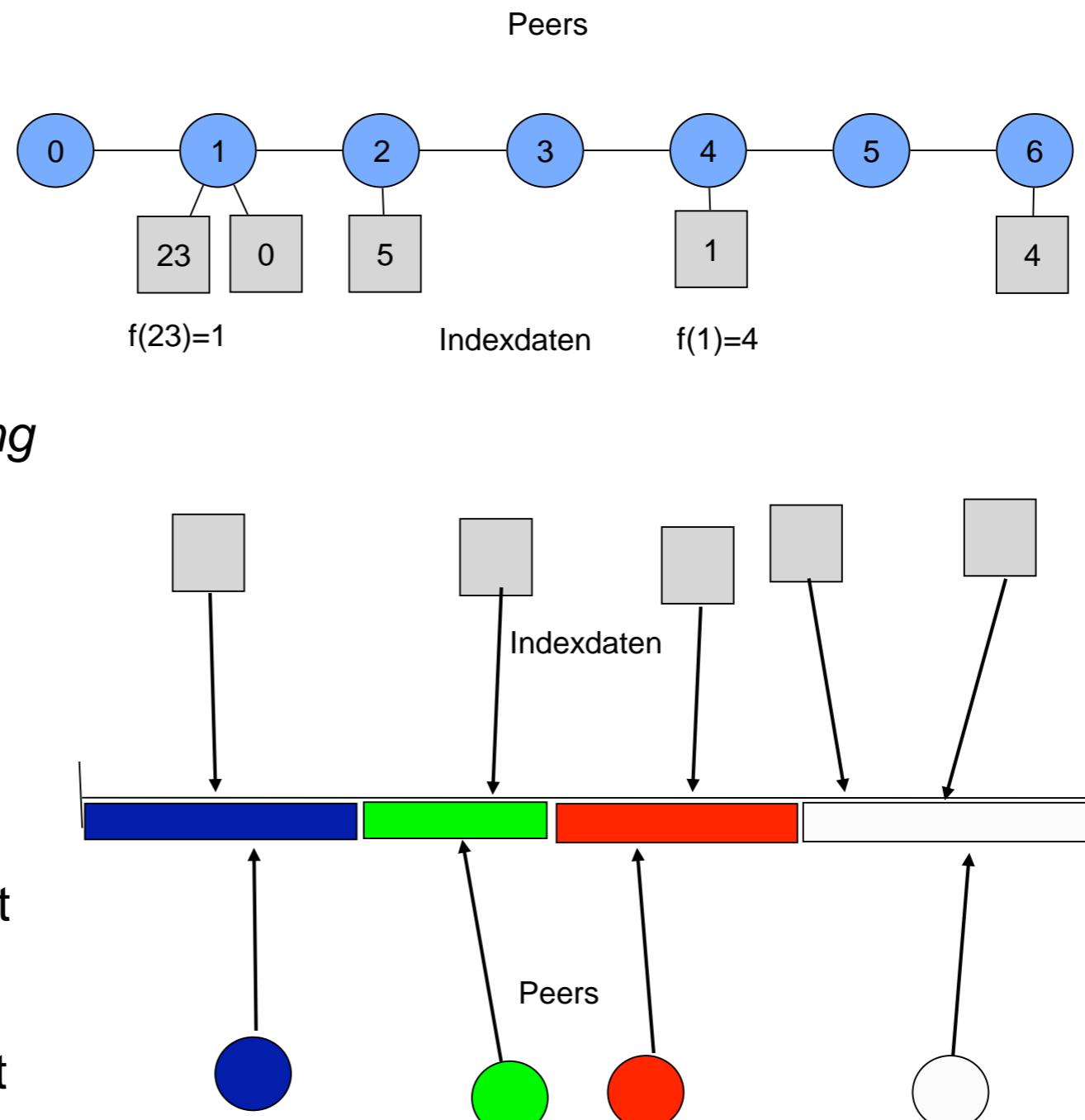
- *Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web*, Karger, Lehman, Leighton, Levine, Lewin, Panigrahy, STOC 1997

■ Daten

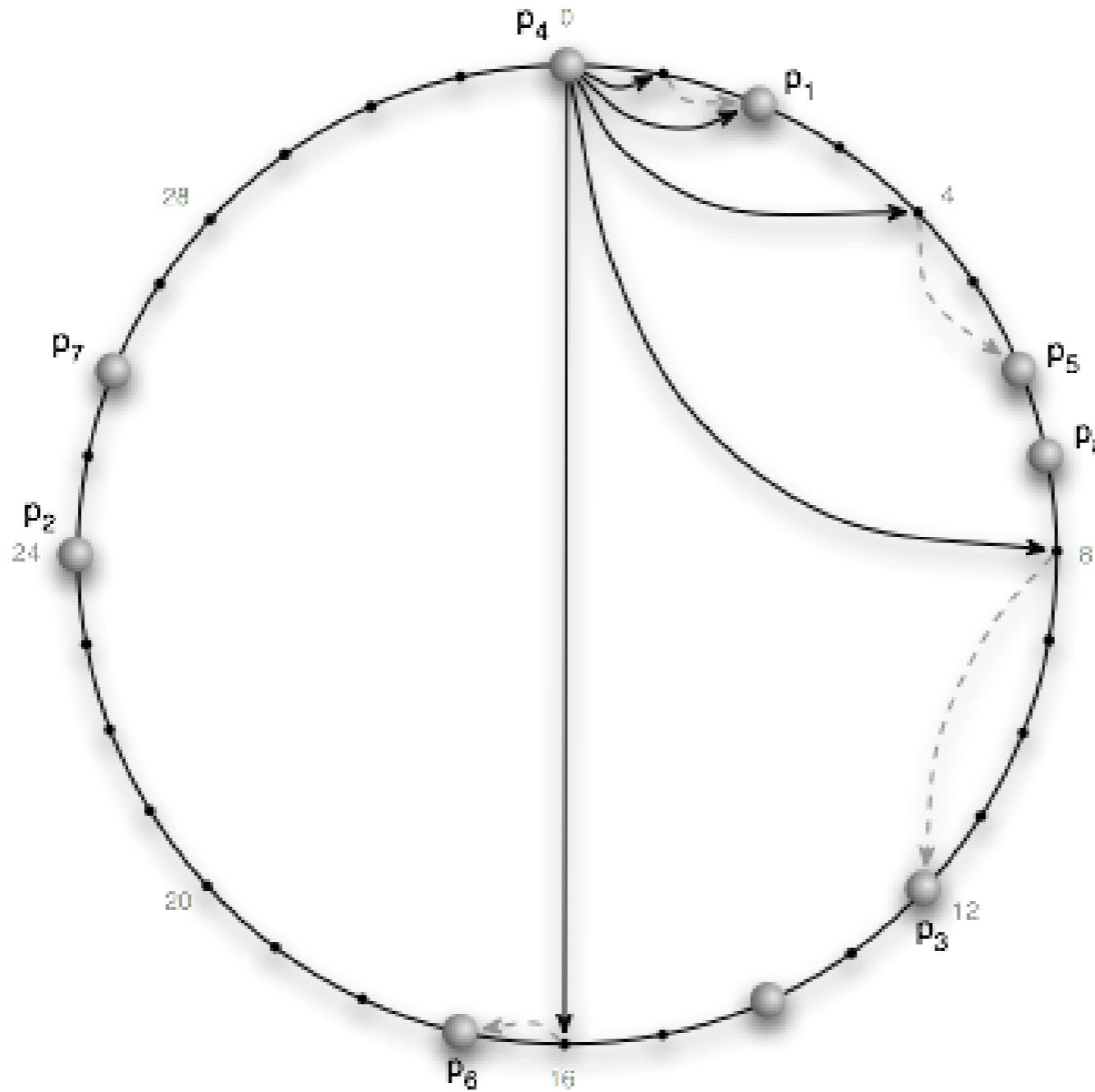
- werden *gehasht* und nach Bereich den Peers zugeordnet

■ Peers

- werden an eine Stelle *gehasht* und erhalten Bereiche des Wertebereichs der Hashfunktion zugeteilt

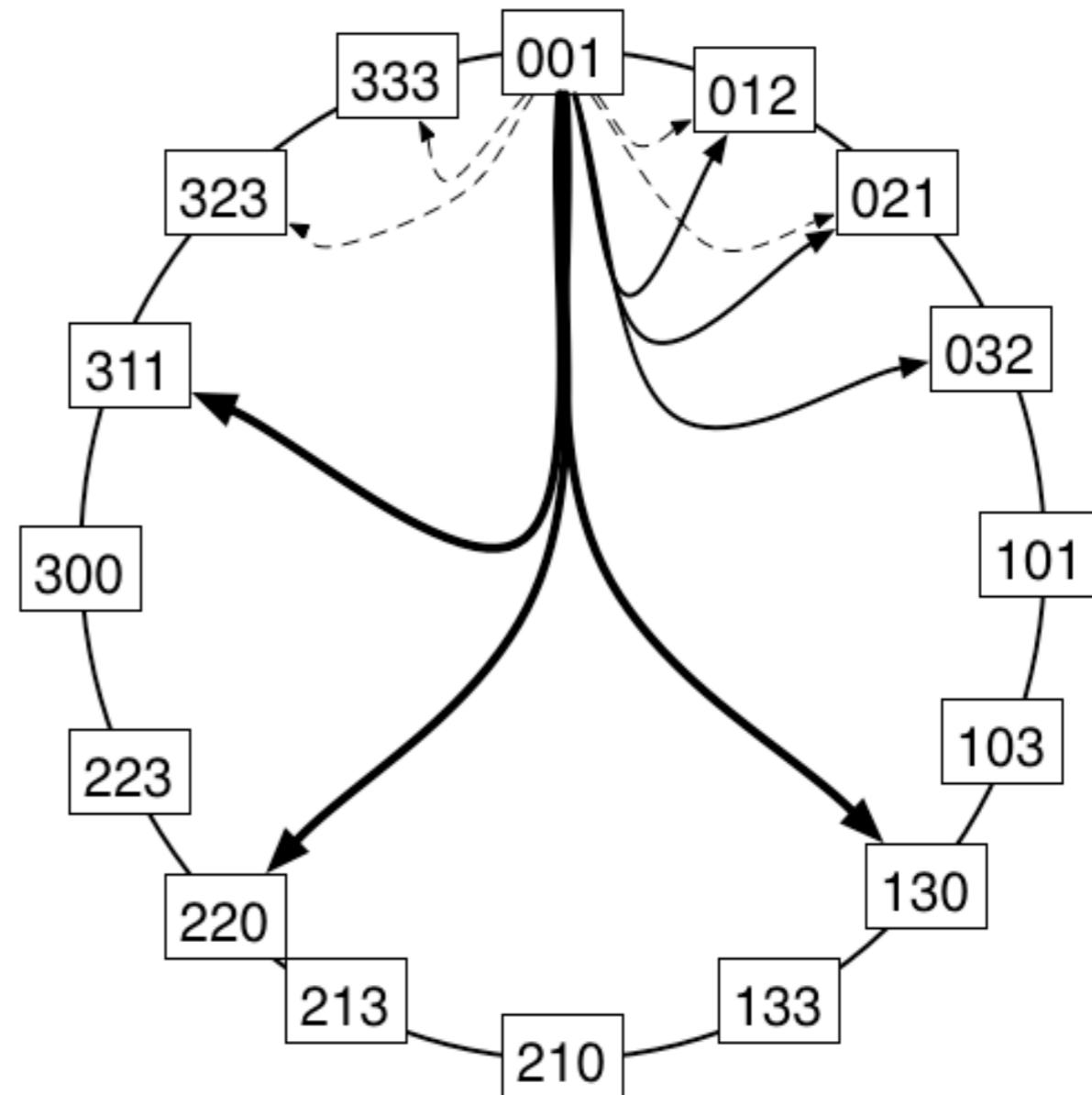


Zeiger-Struktur in Chord



Pastry

- Peter Druschel
 - jetzt Direktor des Max-Planck-Instituts für Informatik, Saarbrücken/Kaiserslautern
- Antony Rowstron
 - Microsoft Research, Cambridge, GB
- Pastry
 - *Scalable, decentralized object location and routing for large scale peer-to-peer-network*
 - Chord-ähnliches Netzwerk, welches das Routing von Plaxton, Rajamaran, Richa (1997) verwendet



BitTorrent

■ Bram Cohen

- BitTorrent ist ein P2P-Netzwerk für den Download von Dateien
- Dateien werden in Blöcke aufgeteilt
- verwendet implizit Multicast-Bäume für die Verteilung von Blöcken

■ Ziele

- schneller Download einer Datei unter Verwendung des Uploads vieler Peers
 - Upload ist der Flaschenhals
 - z.B. wegen asymmetrischen Aufbau von ISDN oder DSL
- Fairness
 - seeders against leeches
- Gleichzeitige Verwendung vieler Peers

Systeme II

5. Die Anwendungsschicht

Christian Schindelhauer

Technische Fakultät

Rechnernetze und Telematik

Albert-Ludwigs-Universität Freiburg

Systeme II

7. Sicherheit

Christian Schindelhauer
Technische Fakultät
Rechnernetze und Telematik
Albert-Ludwigs-Universität Freiburg
(Version 19.07.2017)

- Folien und Inhalte aus
 - Computer Networking: A Top Down Approach
5th edition.
Jim Kurose, Keith Ross
Addison-Wesley, April
2009.
 - Copyright liegt bei den Autoren Kurose und Ross

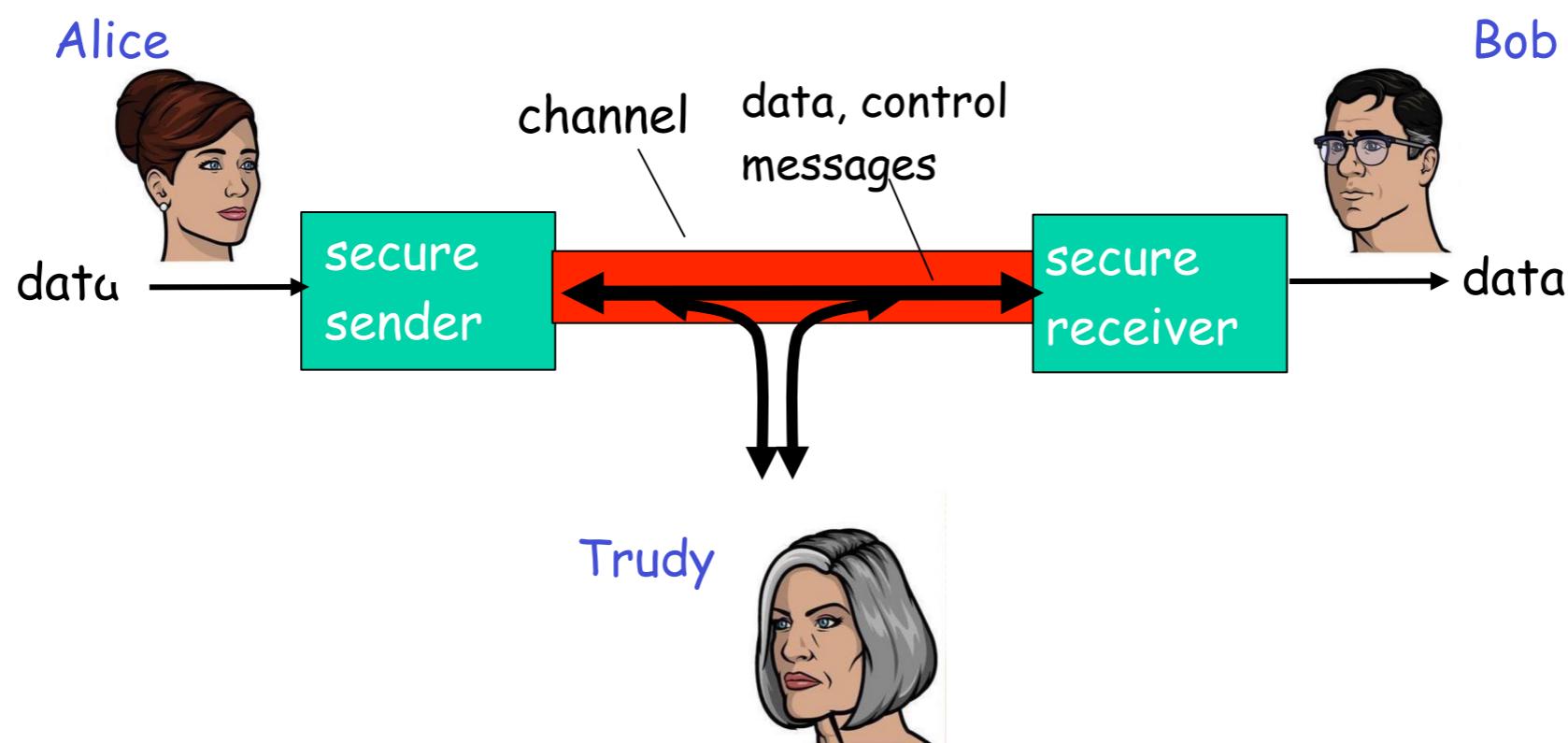
- Grundlagen von Netzwerksicherheit
 - Kryptographie und deren vielfältige Einsatzmöglichkeiten
 - Authentifizierung
 - Message Integrity
- Sicherheit in der Praxis
 - Firewalls und Intrusion Detection
 - Sicherheit in Anwendungs-, Transport-, Vermittlungs- und Sicherungsschicht

Was ist Netzwerk-Sicherheit

- Vertraulichkeit (Confidentiality)
 - Nur der Sender, gewünschter Empfänger sollte den Nachrichteninhalt „verstehen“
- Authentifizierung
 - Sender und Empfänger möchten sich ihrer Identität versichern
- Integrität (message integrity)
 - Sender und Empfänger wollen, dass eine Nachricht nicht unbemerkt verändert werden
 - bei der Übertragung oder später
- Zugriff und Verfügbarkeit
 - von Diensten

Freunde und Feinde: Alice, Bob und Trudy

- Standardnamen im Sicherheitsbereich
- Alice und Bob möchten „sicher“ kommunizieren
- Trude (In-Trude-r) möchte mithören, löschen, hinzufügen, verändern



Wer steckt hinter Alice und Bob

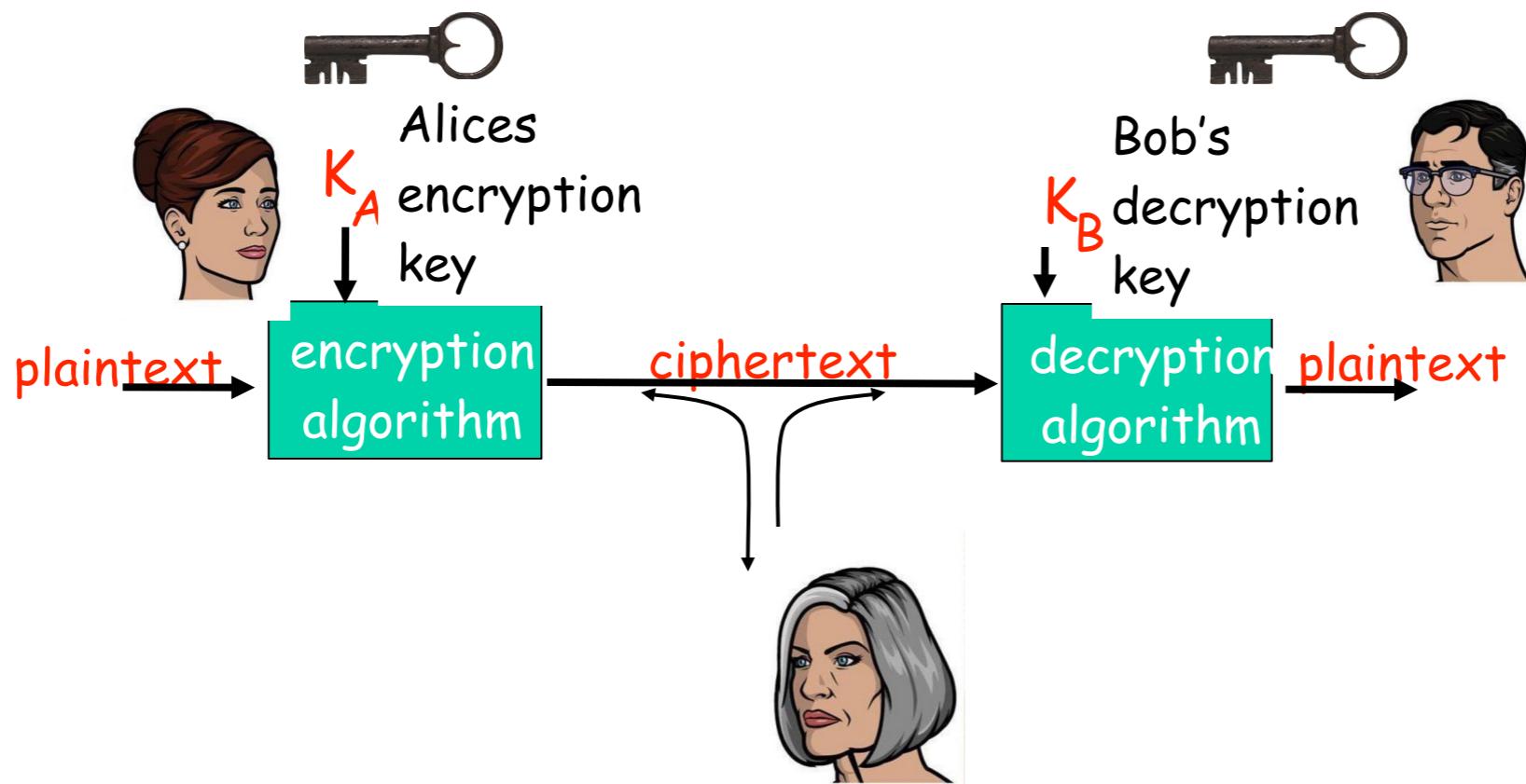
- Echte Menschen
- Web-Browser
- Online-Banking-Clients und Servers
- DNS-Servers
- Routers, die Routing-Tabellen austauschen
- etc.

Was kann ein böser Mensch so tun?

- Abhören (eavesdrop)
 - Nachrichten abfangen und lesen
- Einfügen von Nachrichten
 - Nachrichten werden in die bestehende Verbindung eingefügt
- Sich als jemand anders ausgeben (impersonation)
 - Quell-Adresse kann in einem Paket gefälscht werden
- Hijacking
 - Übernahme einer bestehenden Verbindung durch Ersetzen des Empfängers oder Senders
- Denial of Service
 - Dienst abschalten
 - durch Überlast oder direkten Angriff

Ein kurzer Rundgang durch die Kryptographie

- m : Originalnachricht (message)
- $K_A(m)$: mit Schlüssel K_A verschlüsselte Nachricht
- $m = K_B(K_A(m))$



Einfache Verschlüsselung

- Monoalphabetischer Schlüssel
 - ersetze jeden Buchstaben durch einen anderen
- Beispiel: Edgar Allen Poe „The Gold Bug“
 - 53305))6*;4826)4)4;806*;488¶60))85;1-(;:8-83(88)5*
 - ;46(;88*96*?;8)*(;485);5*2:*(;4956*2(5*-4)8¶8*;40692
 - 85);)68)4;1(9;48081;8:81;4885;4)485528806*81(9;48;
 - (88;4(?34;48)4;161::188;?;
- Jedes Symbol steht für einen Buchstaben:
 - 8 = e
 - ; = h
 - ...

Einfache Verschlüsselung

- Monoalphabetischer Schlüssel
 - ersetze jeden Buchstaben durch einen anderen

plaintext: abcdefghijklmnopqrstuvwxyz

ciphertext: mnbvcxzasdfghjklpoiuytrewq



E.g.: Plaintext: bob. i love you. alice
ciphertext: nkn. s gktc wky. mgsbc

- n monoalphabetische Schlüssel, M_1, M_2, \dots, M_n
- Zyklus-Muster
 - e.g., n=4, $M_1, M_3, M_4, M_3, M_2; M_1, M_3, M_4, M_3, M_2;$
- Für jeden neuen Buchstaben aus den monoalphabetischen Schlüsseln einer ausgewählt
 - „aus“: a from M_1 , u from M_3 , s from M_4
 - Schlüssel: n Schlüsselverfahren und der Zyklus

Bruch einer Kodierung

- Cipher-text only Attack
 - nur mit verschlüsselten Text
 - Zwei Ansätze:
 - Durchsuche alle Schlüssel und teste ob sie einen vernünftigen Text produzieren
 - Statistische Analyse des Schlüssels
- Known-Plaintext-Attack
 - mit der Originalnachricht und dem verschlüsselten Text
- Chosen Plaintext Attack
 - Trudy wählt den Text und lässt Alice ihn verschlüsseln
 - Trudy erhält den verschlüsselten Text

Methoden der Kryptographie

- Geheime Schlüssel sind die Sicherheitsgrundlage
 - Der Algorithmus ist bekannt
 - außer bei „security by obscurity“
- Public-Key-Cryptography
 - verwendet zwei Schlüssel
 - ein geheimer und ein öffentlicher Schlüssel
- Symmetrische Kryptographie
 - beide Seiten verwenden den selben geheimen Schlüssel
- Hash-Funktion
 - Ohne Schlüssel und ohne Geheimnis

Chiffrierungstypen

- Stromchiffrierer (stream cipher)
 - verschlüsselt bitweise
- Blockchiffre, Blockverschlüsselung (block ciphers)
 - Originaltext wird in gleichgroße Blöcke unterteilt
 - Jeder Block wird einzeln kodiert

Stream Ciphers

- Kombiniere jedes Bit eines Schlüsselstroms (key stream) mit dem Original bit
 - $m(i)$ = i-tes Bit der Nachricht
 - $ks(i)$ = i-tes Bit des Key Streams
 - $c(i)$ = i-tes bit des verschlüsselten Texts
- Verschlüsselung
 - $c(i) = ks(i) + m(i) \pmod{2}$
 $= ks(i) \oplus m(i)$
- Entschlüsselung
 - $m(i) = ks(i) \oplus c(i)$

RC4 Stream Cipher

- RC4 ist ein populärer Streamchiffrierer
 - ausführlich analysiert und als sicher angesehen
 - Schlüssellänge: von 1 bis 256 Bytes
 - wird in WEP für 802.11 verwendet
 - kann in SSL verwendet werden

Quell-Code RC4

```
k[]: gegebene Schlüssel-Zeichenfolge der Länge 5 bis 256 Byte
L := Länge des Schlüssels in Byte
s[]: Byte-Vektor der Länge 256
Für i = 0 bis 255
    s[i] := i
j := 0
Für i = 0 bis 255
    j := (j + s[i] + k[i mod L]) mod 256
vertausche s[i] mit s[j]
```

```
klar[]: gegebene Klartext-Zeichenfolge der Länge X
schl[]: Vektor zum Abspeichern des Schlüsseltextes
i := 0
j := 0
Für n = 0 bis X-1
    i := (i + 1) mod 256
    j := (j + s[i]) mod 256
    vertausche s[i] mit s[j]
    zufallszahl := s[(s[i] + s[j]) mod 256]
    schl[n] := zufallszahl XOR klar[n]
```

- Aus Wikipedia

- <http://de.wikipedia.org/wiki/Rc4>

Block-Chiffre

- Nachrichten werden in Blöcken von k bits verschlüsselt
 - z.B. 64-bit Blöcke
- Injektive Abbildung um den Quelltext in den k-bit verschlüsselten Text umzuwandeln
- Beispiel k=3:

<u>input</u>	<u>output</u>	<u>input</u>	<u>output</u>
000	110	100	011
001	111	101	010
010	101	110	000
011	100	111	001

Block-Chiffre

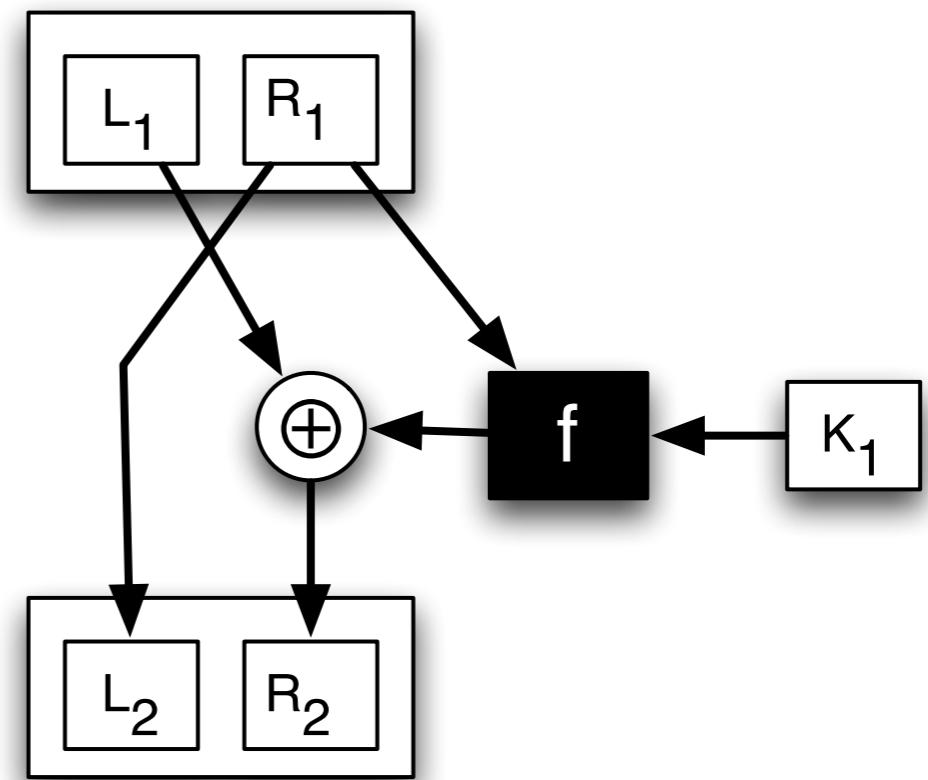
- Wie viele mögliche Abbildungen gibt es für k-Bit Block-Chiffre?

Block-Chiffre

- Wie viele mögliche Abbildungen gibt es für k-Bit Block-Chiffre?
 - Im allgemeinen: $2^k!$
 - riesig für $k=64$
 - und absolut sicher, wenn man sie zufällig auswählt
- Problem:
 - Die meisten dieser Abbildungen benötigen große Tabellen um sie zu berechnen
- Lösung
 - Statt einer Tabelle, verwendet man eine Funktion, die diese Tabelle simuliert
 - Dadurch verliert man möglicherweise wieder die Sicherheit

Feistel-Chiffre

- Aufteilung der Nachricht in zwei Hälften L_1, R_1
 - Schlüssel K_1, K_2, \dots
 - Mehrere Runden: resultierender Code: L_n, R_n
- Verschlüsselung
 - $L_i = R_{i-1}$
 - $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$
- Entschlüsselung
 - $R_{i-1} = L_i$
 - $L_{i-1} = R_i \oplus f(L_i, K_i)$
- f : beliebige, komplexe Funktion

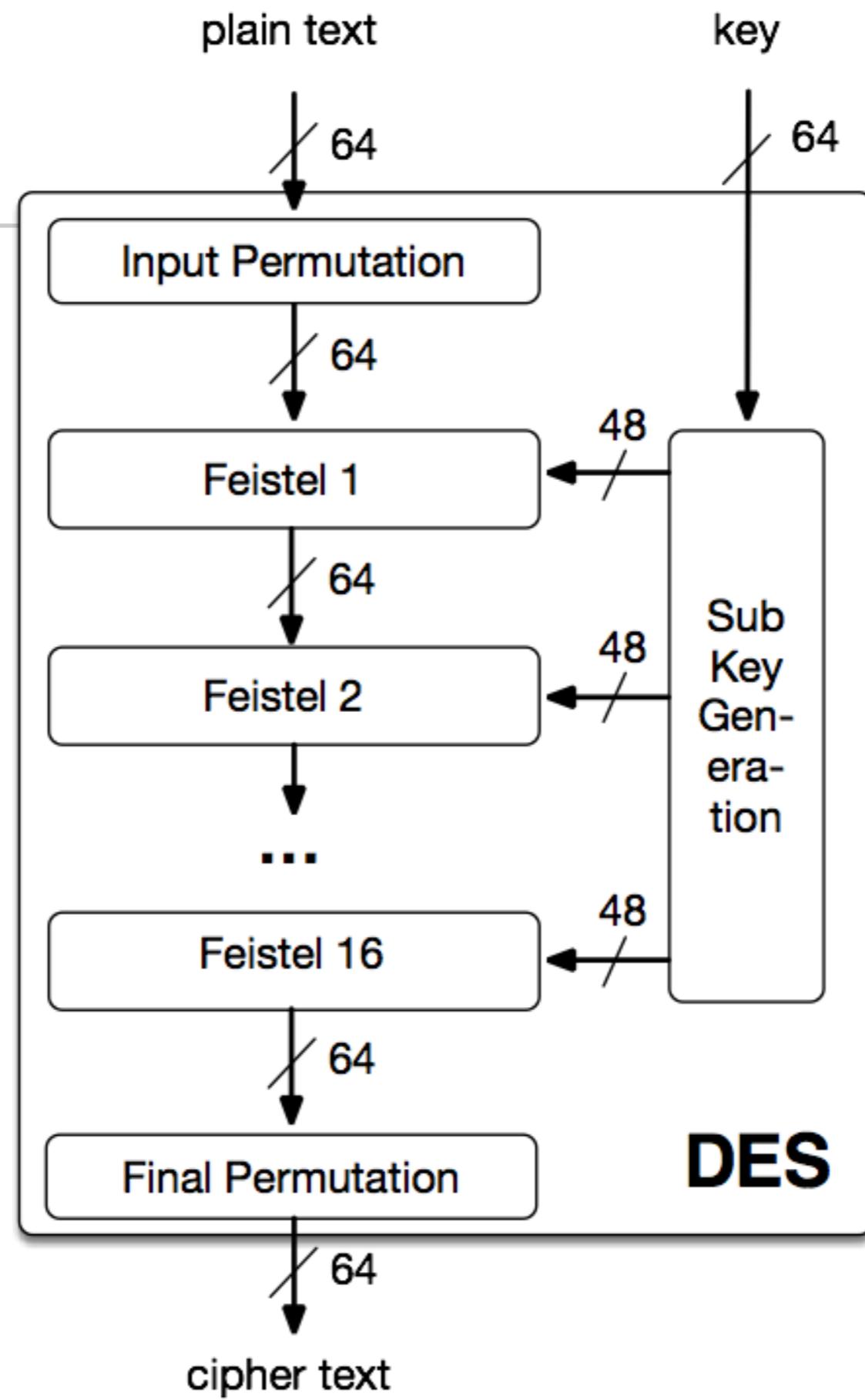


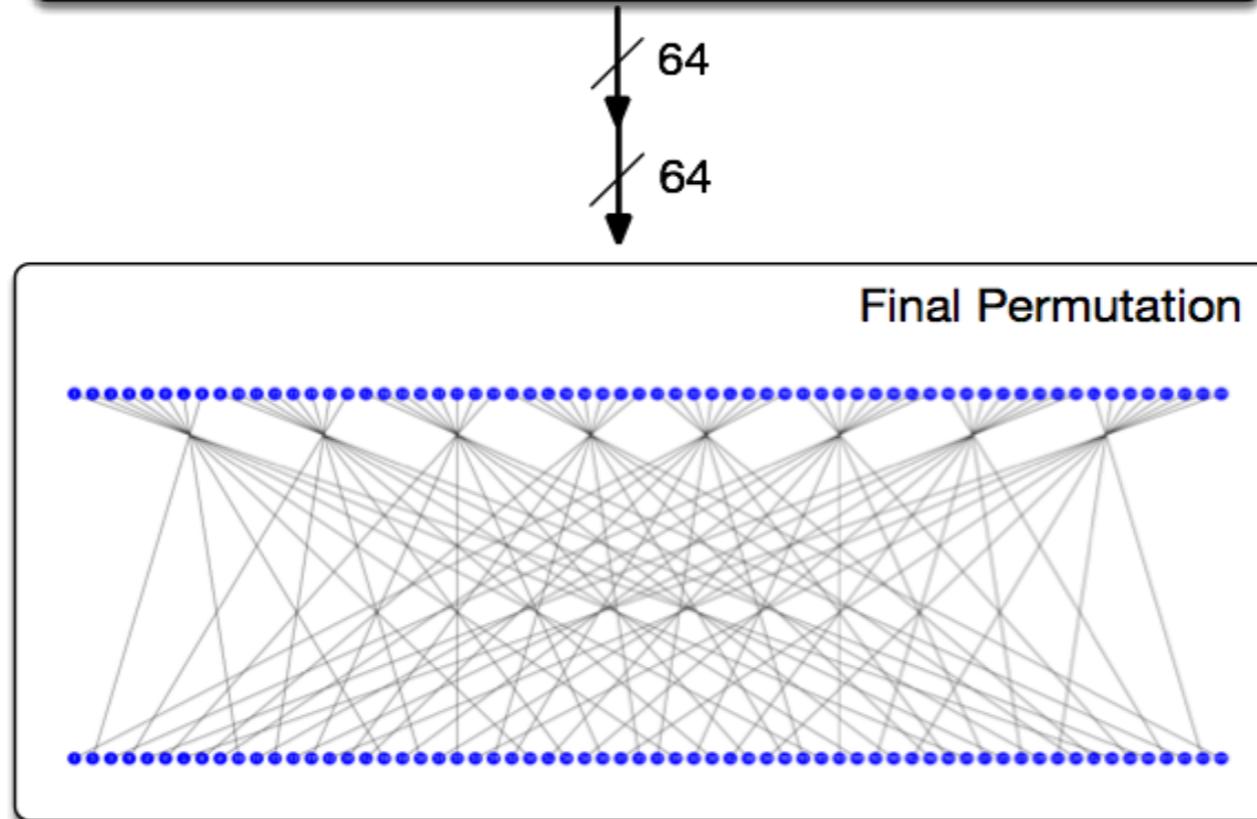
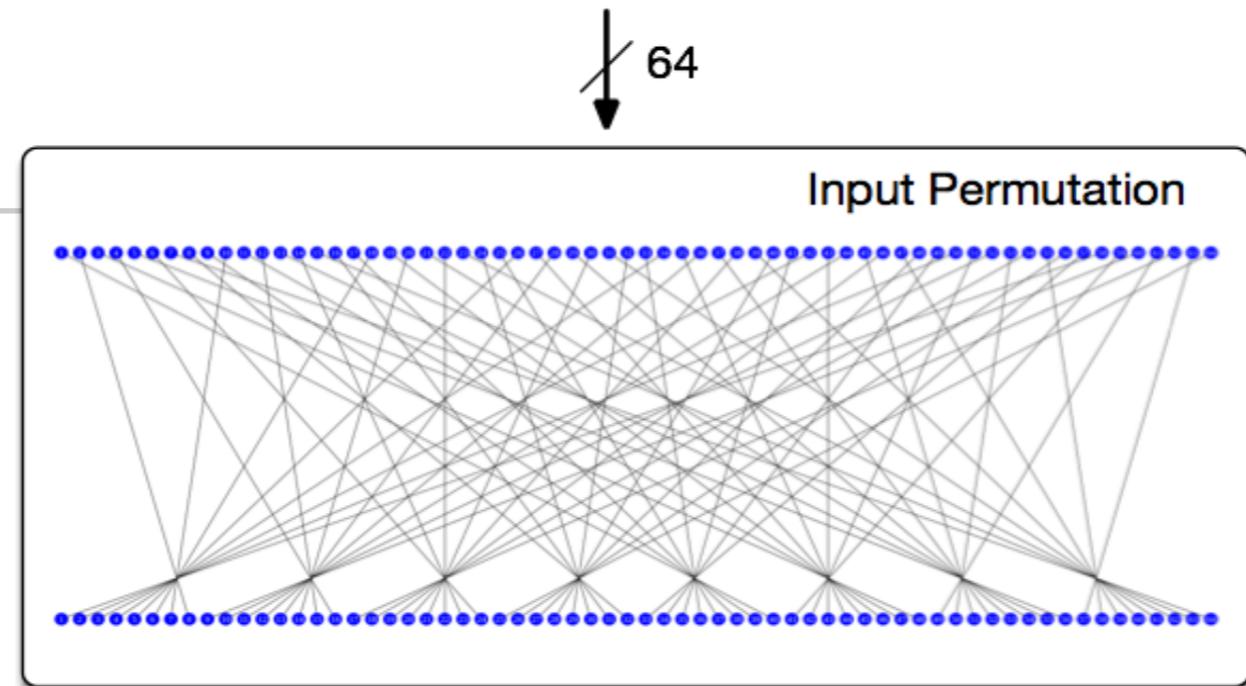
- Skipjack
 - 80-Bit symmetrischer Code
 - baut auf Feistel-Chiffre auf
 - wenig sicher
- RC5
 - Schlüssellänge 1-2048 Bits
 - Rivest Code 5 (1994)
 - Mehrere Runden der Feistel-Chiffre

Digital Encryption Standard

- Geschickt gewählte Kombination von
 - Xor-Operationen
 - Feistel-Chiffre
 - Permutationen
 - Table-Lookups
 - verwendet 56-Bit Schlüssel
- 1975 entwickelt von Wissenschaftlern von IBM
 - Mittlerweile nicht mehr sicher
 - leistungsfähigere Rechner
 - Erkenntnisse in Kryptologie
- Nachfolger: AES (2001)

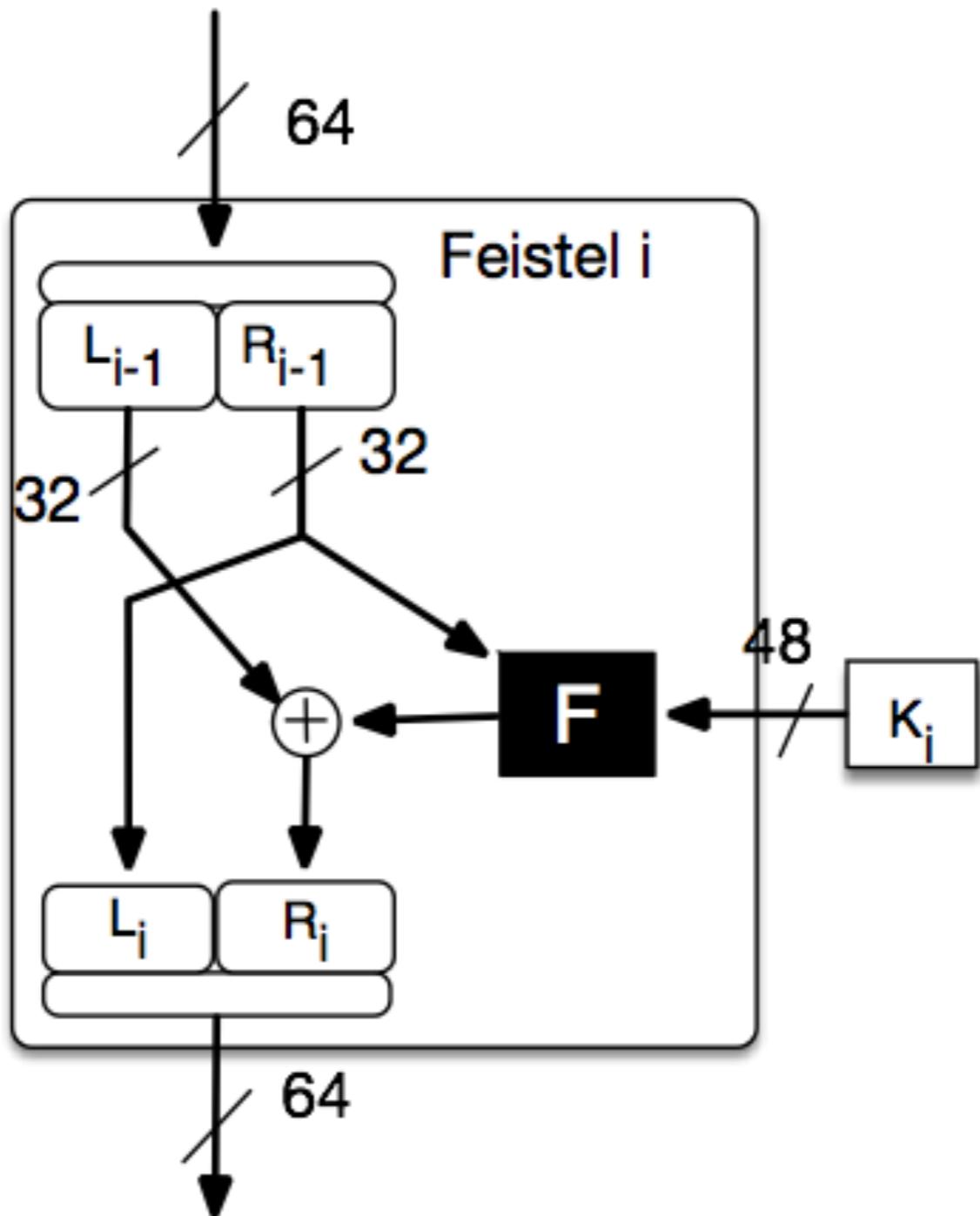
DES



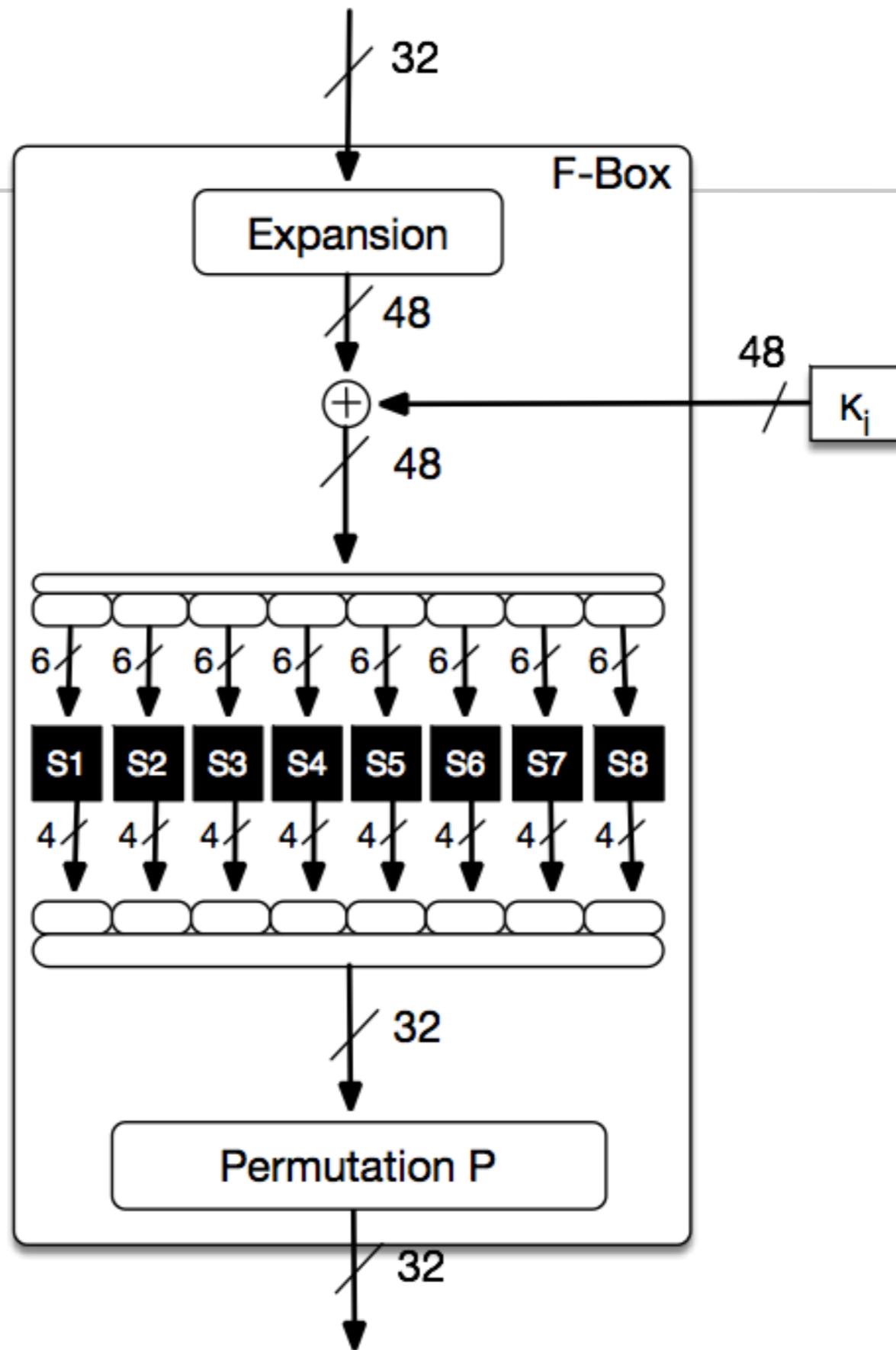


64
64
64

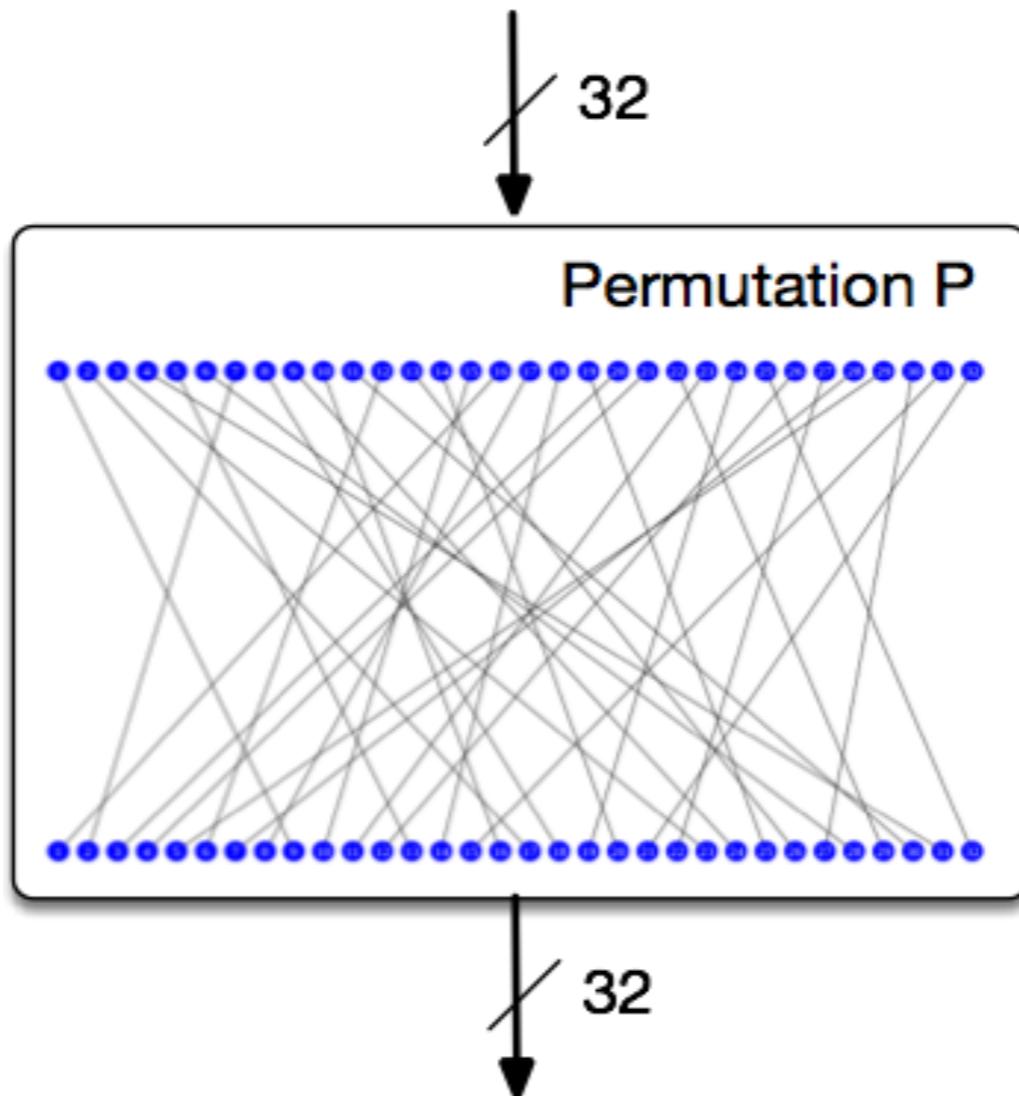
Feistel



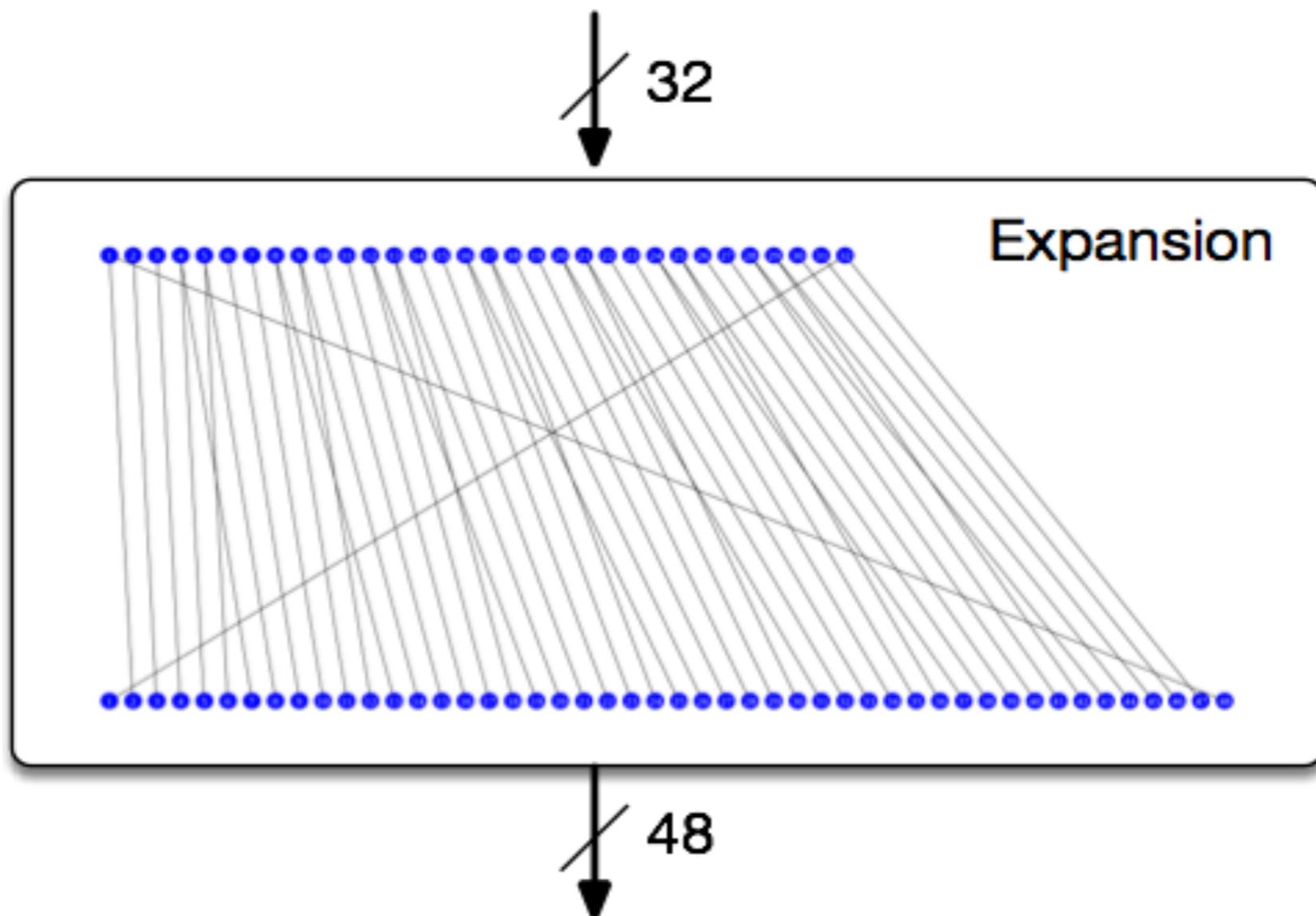
F-Box



Permutation



Expansion



S-Box

S-boxes																
S₁	x0000x	x0001x	x0010x	x0011x	x0100x	x0101x	x0110x	x0111x	x1000x	x1001x	x1010x	x1011x	x1100x	x1101x	x1110x	x1111x
0yyyy0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0yyyy1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
1yyyy0	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
1yyyy1	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S₂	x0000x	x0001x	x0010x	x0011x	x0100x	x0101x	x0110x	x0111x	x1000x	x1001x	x1010x	x1011x	x1100x	x1101x	x1110x	x1111x
0yyyy0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
0yyyy1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
1yyyy0	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
1yyyy1	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S₃	x0000x	x0001x	x0010x	x0011x	x0100x	x0101x	x0110x	x0111x	x1000x	x1001x	x1010x	x1011x	x1100x	x1101x	x1110x	x1111x
0yyyy0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
0yyyy1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
1yyyy0	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1yyyy1	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S₄	x0000x	x0001x	x0010x	x0011x	x0100x	x0101x	x0110x	x0111x	x1000x	x1001x	x1010x	x1011x	x1100x	x1101x	x1110x	x1111x
0yyyy0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
0yyyy1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
1yyyy0	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
1yyyy1	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S₅	x0000x	x0001x	x0010x	x0011x	x0100x	x0101x	x0110x	x0111x	x1000x	x1001x	x1010x	x1011x	x1100x	x1101x	x1110x	x1111x
0yyyy0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
0yyyy1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
1yyyy0	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
1yyyy1	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S₆	x0000x	x0001x	x0010x	x0011x	x0100x	x0101x	x0110x	x0111x	x1000x	x1001x	x1010x	x1011x	x1100x	x1101x	x1110x	x1111x
0yyyy0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
0yyyy1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
1yyyy0	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
1yyyy1	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S₇	x0000x	x0001x	x0010x	x0011x	x0100x	x0101x	x0110x	x0111x	x1000x	x1001x	x1010x	x1011x	x1100x	x1101x	x1110x	x1111x
0yyyy0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
0yyyy1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1yyyy0	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
1yyyy1	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S₈	x0000x	x0001x	x0010x	x0011x	x0100x	x0101x	x0110x	x0111x	x1000x	x1001x	x1010x	x1011x	x1100x	x1101x	x1110x	x1111x
0yyyy0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
0yyyy1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
1yyyy0	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
1yyyy1	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Advanced Encryption Standard

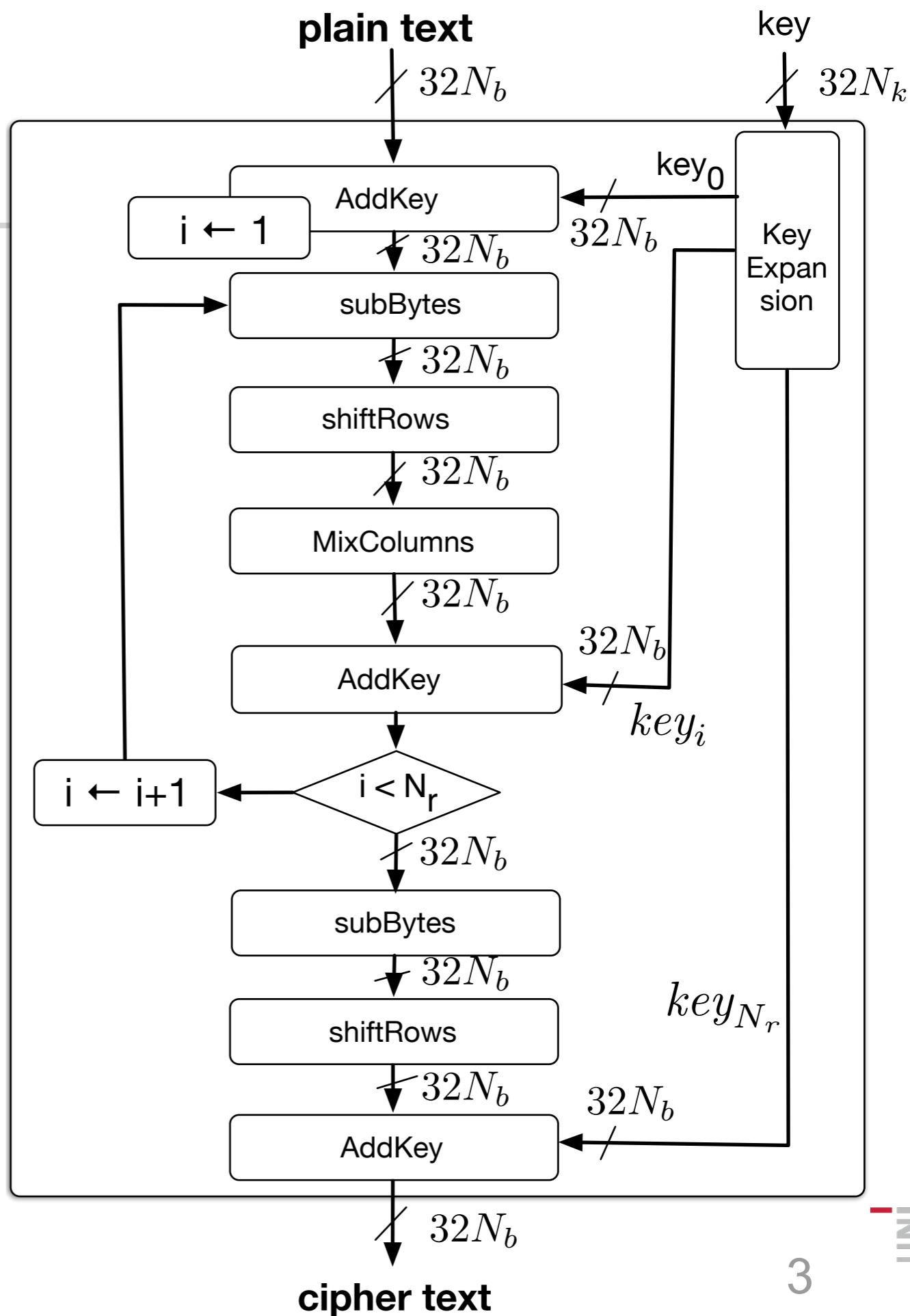
- Geschickt gewählte Kombination von
 - Xor-Operationen
 - Permutationen
 - Multiplikation in $GF[2^8]$
 - symmetrische 128,192 oder 256-Bit Schlüssel
- Joan Daemen und Vincent Rijmen
 - 2001 als AES unter vielen ausgewählt worden
 - bis heute als sicher erachtet

Zustände

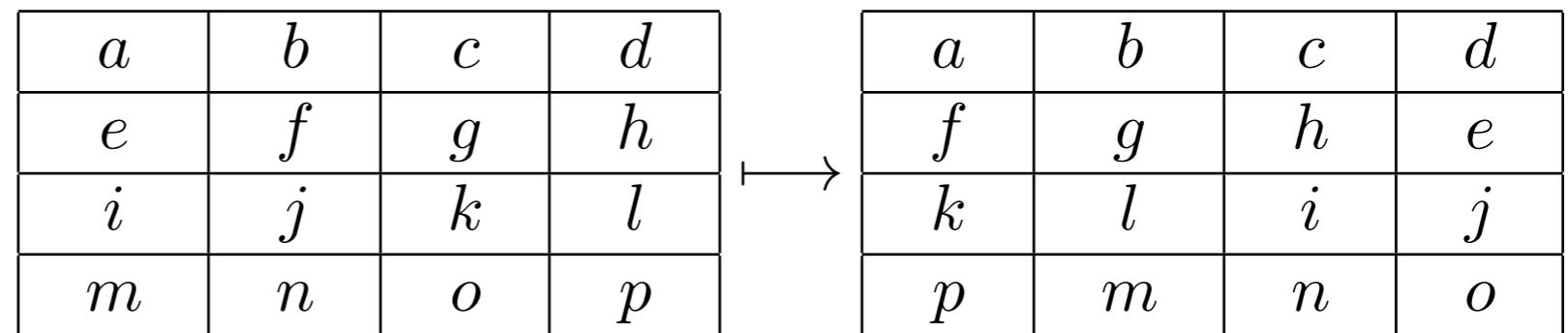
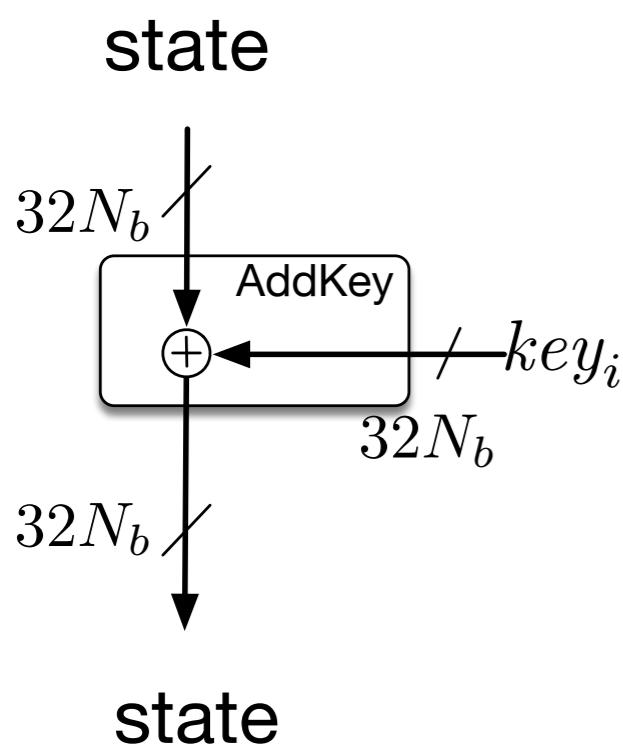
$$\begin{pmatrix} a_{0,0} & a_{0,1} & \dots & a_{0,N_b-1} \\ a_{1,0} & a_{1,1} & \dots & a_{1,N_b-1} \\ a_{2,0} & a_{2,1} & \dots & a_{2,N_b-1} \\ a_{3,0} & a_{3,1} & \dots & a_{3,N_b-1} \end{pmatrix}$$

N_r

N_k	N_b				
	4	5	6	7	8
4	10	11	12	13	14
5	11	11	12	13	14
6	12	12	12	13	14
7	13	13	13	13	14
8	14	14	14	14	14



AddKey & ShiftRows



SubBytes

$$S_{RD}(x) = g \circ f(x) = g(f(x))$$

$$f : \mathbb{F}_{2^8} \longrightarrow \mathbb{F}_{2^8}, \quad x \longmapsto \begin{cases} x^{-1} & \text{if } x \neq 0, \\ 0 & \text{if } x = 0. \end{cases}$$

$$g : \mathbb{F}_2^8 \longrightarrow \mathbb{F}_2^8, \quad x \longmapsto Ax + b,$$

$$A := \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \quad b := \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} a_{0,0} & a_{0,1} & \dots & a_{0,N_b-1} \\ a_{1,0} & a_{1,1} & \dots & a_{1,N_b-1} \\ a_{2,0} & a_{2,1} & \dots & a_{2,N_b-1} \\ a_{3,0} & a_{3,1} & \dots & a_{3,N_b-1} \end{pmatrix}$$

$$\mapsto \begin{pmatrix} S_{RD}(a_{0,0}) & S_{RD}(a_{0,1}) & \dots & S_{RD}(a_{0,N_b-1}) \\ S_{RD}(a_{1,0}) & S_{RD}(a_{1,1}) & \dots & S_{RD}(a_{1,N_b-1}) \\ S_{RD}(a_{2,0}) & S_{RD}(a_{2,1}) & \dots & S_{RD}(a_{2,N_b-1}) \\ S_{RD}(a_{3,0}) & S_{RD}(a_{3,1}) & \dots & S_{RD}(a_{3,N_b-1}) \end{pmatrix}$$

MixColumns

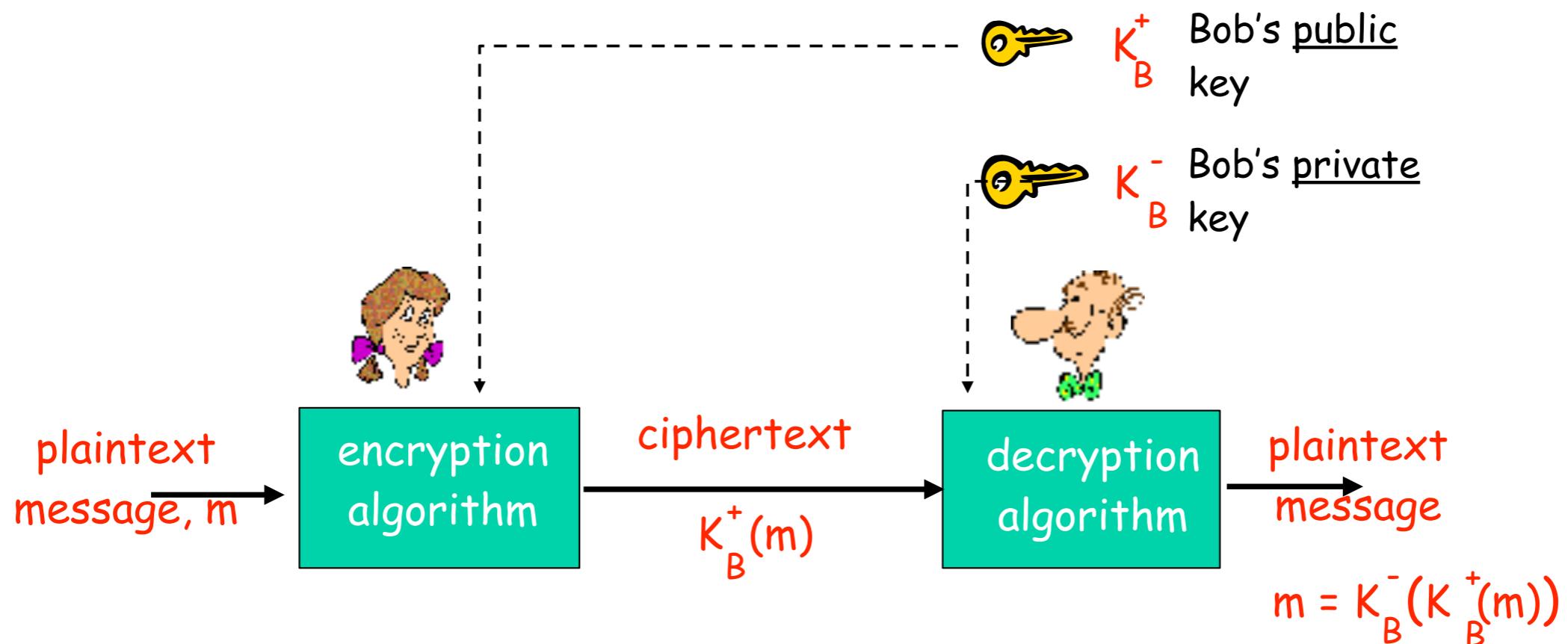
- Interprete a column of the status as a vector with four elements over GF[256]

$$\begin{pmatrix} b_{0,j} \\ b_{1,j} \\ b_{2,j} \\ b_{3,j} \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \cdot \begin{pmatrix} a_{0,j} \\ a_{1,j} \\ a_{2,j} \\ a_{3,j} \end{pmatrix}, \quad j = 0, 1, 2, 3.$$

- Inverse matrix:

$$\begin{pmatrix} 14 & 11 & 13 & 9 \\ 9 & 14 & 11 & 13 \\ 13 & 9 & 14 & 11 \\ 11 & 13 & 9 & 14 \end{pmatrix}$$

Public key cryptography



Asymmetrische Verschlüsselungsmethoden

- z.B. RSA, Ronald Rivest, Adi Shamir, Lenard Adleman, 1977
 - Diffie-Hellman, PGP
- Geheimer Schlüssel privat: kennt nur der Empfänger der Nachricht
- Öffentlichen Schlüssel offen: Ist allen Teilnehmern bekannt
- Wird erzeugt durch Funktion
 - keygen(privat) = offen
- Verschlüsselungsfunktion f und Entschlüsselungsfunktion g
 - sind auch allen bekannt
- Verschlüsselung
 - $f(\text{offen}, \text{text}) = \text{code}$
 - kann jeder berechnen
- Entschlüsselung
 - $g(\text{privat}, \text{code}) = \text{text}$
 - nur vom Empfänger

- R. Rivest, A. Shamir, L. Adleman
 - On Digital Signatures and Public Key Cryptosystems,
Communication of the ACM
- Verfahren beruht auf der Schwierigkeit der
Primfaktorzerlegung
- 1. Beispiel:
 - $15 = ? * ?$
 - $15 = 3 * 5$
- 2. Beispiel:
 - $3865818645841127319129567277348359557444790410289933586483552047443 = 1234567890123456789012345678900209 * 31313131313131313131313131300227$

- Bis heute ist kein effizientes Verfahren zur Primfaktorzerlegung bekannt
 - Aber das Produkt von Primzahlen kann effizient bestimmt werden
 - Primzahlen können effizient bestimmt werden
 - Primzahlen sehr häufig

■ Erzeugung der Schlüssel

- Wähle zufällig zwei Primzahlen p und q mit k bits ($k \geq 500$).
 - $n = p \cdot q$
 - e ist Zahl, die teilerfremd ist mit $(p - 1) \cdot (q - 1)$.
 - $d = e^{-1} = 1/e \bmod (p - 1)(q - 1)$
 - es gilt $d \cdot e \equiv 1 \bmod (p - 1)(q - 1)$
- Public Key $P = (e, n)$
 - Secret Key $S = (d, n)$

- Kodierung
 - Teile Nachricht in Blöcke der Größe 2^{2k} auf
 - Interpretiere Block M als Zahl $0 \leq M < 2^{2k}$
 - Chiffre: $P(M) = M^e \text{ mod } n$
- Dekodierung
 - $S(C) = C^d \text{ mod } n$
- Korrektheit gilt nach dem kleinen Satz von Fermat
 - Für Primzahl p und von p teilerfremde Zahl a gilt:
$$a^p \equiv a \pmod{p}$$

RSA Beispiel

- Bob wählt $p=5$, $q=7$
 - $n=35$, $\varphi(n)=24$
 - $e = 5$
 - $d= 29$
 - $e d = 1 \bmod 24$
- Verschlüsselung von 8-Bit-Nachrichten

Bit pattern	m	m	$c=m^e \bmod n$
00001000	12	248 832	17

- Entschlüsselung

$$\begin{array}{ll} c & c^d = 17^{29} \\ 17 & 481968572106750915091411825223071697 \end{array} \quad \begin{array}{l} m=c^d \bmod n \\ 12 \end{array}$$

RSA Beispiel

- Berechnung von $17^{29} \bmod 35$
- $29 = 11101_2$
 - $17^{29} = 17 \cdot (17^{14})^2 \bmod 35$
 - $17^{14} = (17^7)^2 \bmod 35$
 - $17^7 = 17 \cdot (17^3)^2 \bmod 35$
 - $17^3 = 17 \cdot (17)^2 \bmod 35$
- Einsetzen:
 - $17^3 = 4913 = 13 \bmod 35$
 - $17^7 = 17 \cdot 13^2 = 2873 = 3 \bmod 35$
 - $17^{14} = 3^2 = 9 \bmod 35$
 - $17^{29} = 17 \cdot 9^2 = 1377 = 12 \bmod 35$

Nachrichtenintegrität

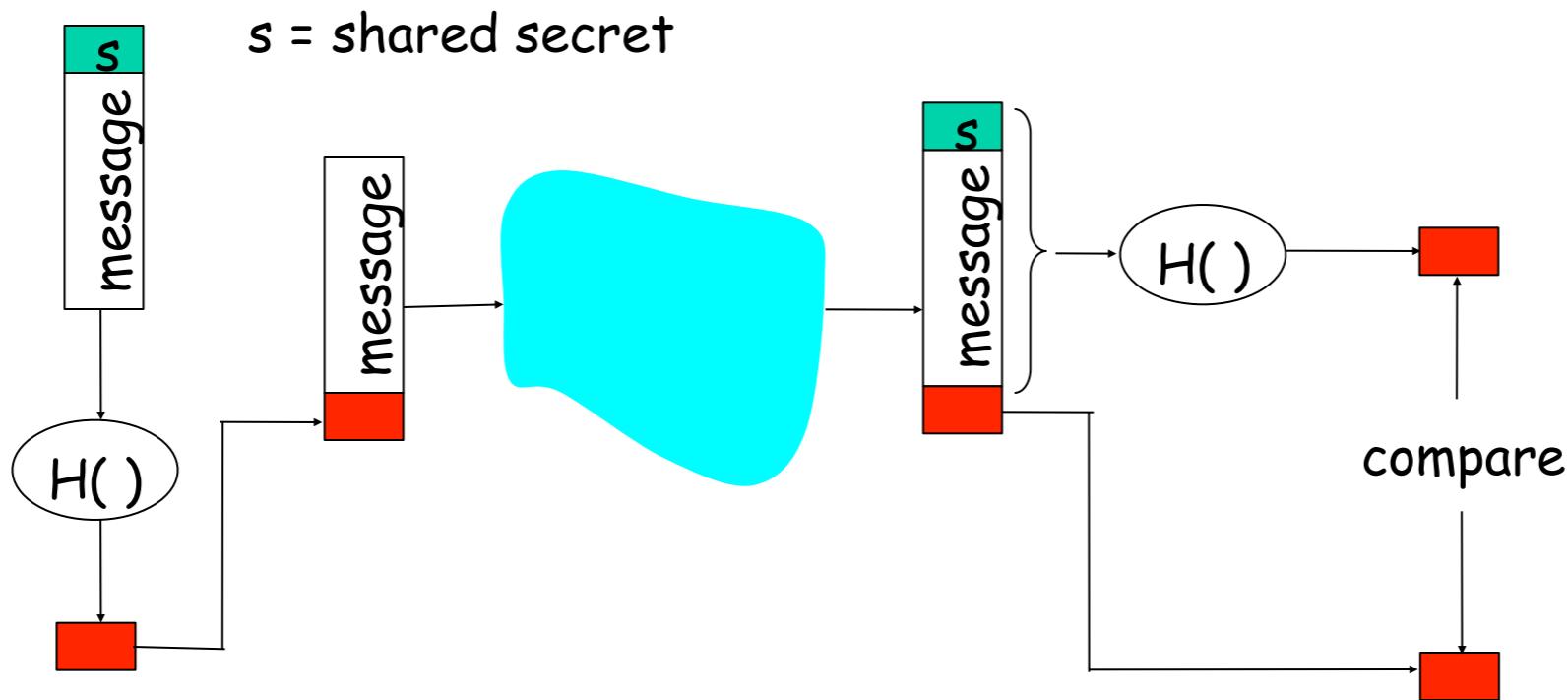
- Erlaubt den Kommunikationspartnern die Korrektheit und Authentizität der Nachricht zu überprüfen
 - Inhalt ist unverändert
 - Urheber ist korrekt
 - Nachricht ist keine Wiederholung
 - Reihenfolge der Nachrichten ist korrekt
- Message Digests

Kryptographische Hash-Funktion

- z.B. SHA-1, SHA-2, MD5
- Ein kryptographische Hash-Funktion h bildet einen Text auf einen Code fester Länge ab, so dass
 - $h(\text{text}) = \text{code}$
 - es unmöglich ist einen anderen Text zu finden mit:
 - $h(\text{text}') = h(\text{text})$ und $\text{text} \neq \text{text}'$
- Mögliche Lösung:
 - Verwendung einer symmetrischen Kodierung

- MD5 ist sehr verbreitet (RFC 1321)
 - berechnet 128-bit Nachricht
 - unsicher
- SHA-1 auch gebräuchlich
 - US standard [NIST, FIPS PUB 180-1]
 - 160-bit Message Digest
 - nicht mehr als sicher angesehen
- SHA-2
 - SHA-256/224
 - SHA-512/384
 - bis jetzt (2012) als sicher angesehen
- SHA-3
 - 2011 veröffentlicht

Message Authentication Code (MAC)



- Authentifiziert Absender
- Überprüft Nachrichtenintegrität
- Keine Verschlüsselung
- “keyed hash”
- Notation: $\text{MD}_m = H(s \parallel m)$; sende $m \parallel \text{MD}_m$

HMAC (Keyed-Hash Message Authentication Code)

- Populärer MAC-Standard
- Sicher gegen Anhänger von Nachrichten

$$HMAC_K(N) = H\left((K \oplus opad) \parallel H\left((K \oplus ipad) \parallel N\right)\right)$$

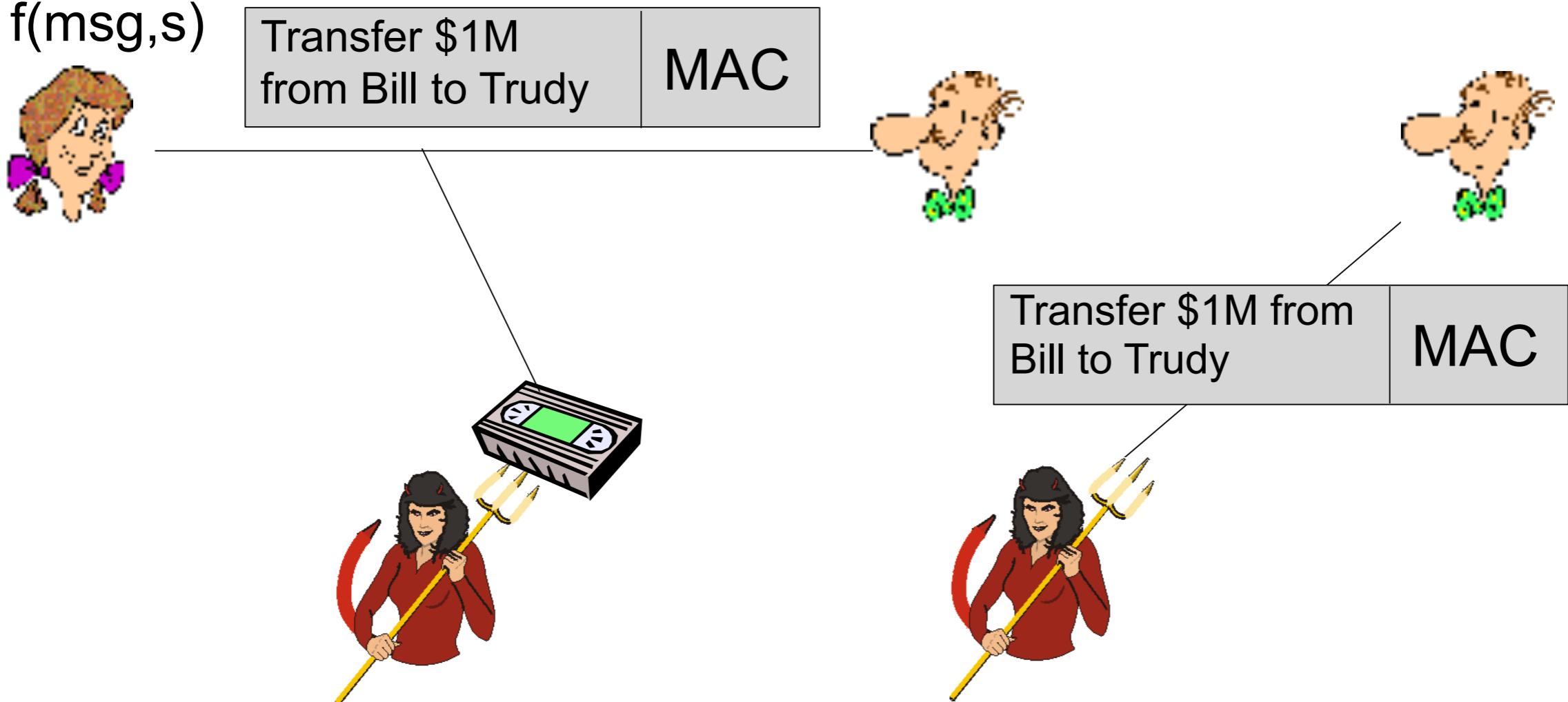
- Nachricht N
- geheimer Schlüssel K
- Konstante opad und ipad
- Erhöht Sicherheit gegen angreifbare Hash-Codes
 - wird in TLS und IPsec verwendet

Authentifizierung der Endpunkte

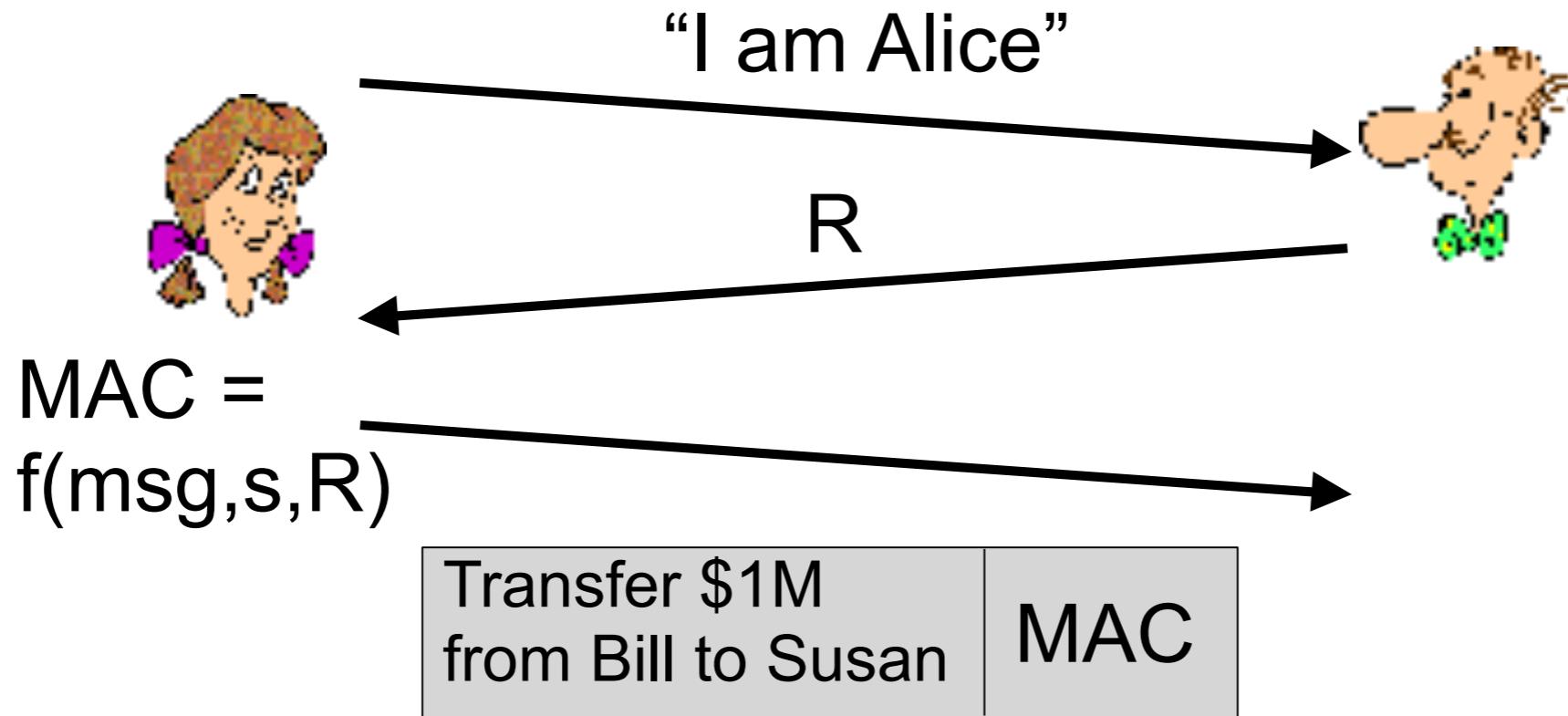
- Versicherung, dass der Kommunikationspartner korrekt ist
- Angenommen Alice und Bob haben ein gemeinsames Geheimnis, dann gibt MAC eine Authentifizierung der Endpunkte
 - (end-point authentication)
 - Wir wissen, dass Alice die Nachricht erzeugt hat
 - Aber hat sie sie auch abgesendet?

Playback-Attacke

MAC =
 $f(\text{msg}, s)$



Verteidigung gegen die Playback-Attacke: nonce (use only once)

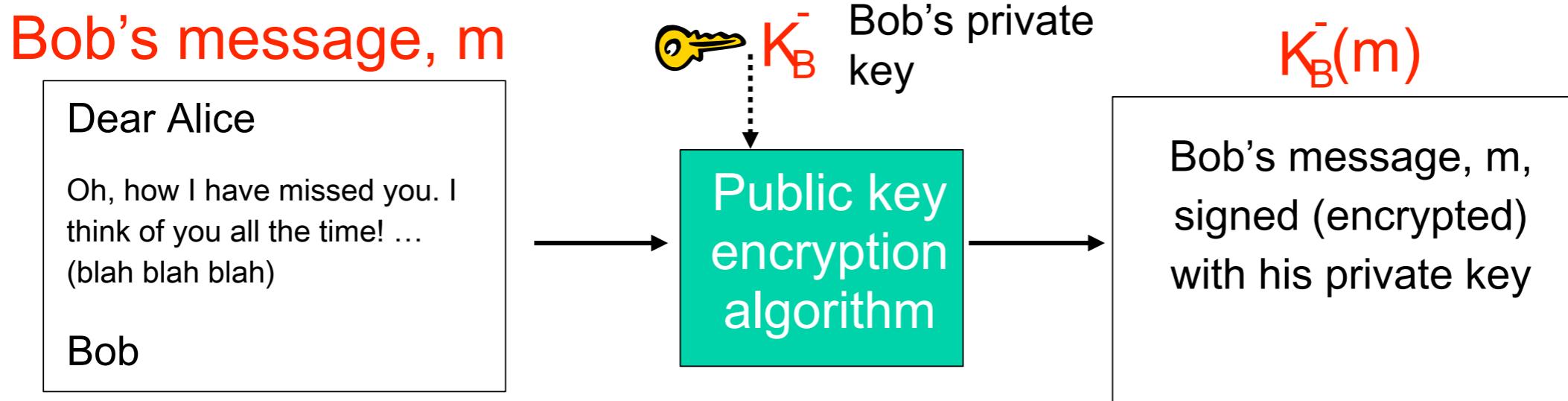


Digitale Unterschriften

- Kryptographischer Algorithmus analog zu handgeschriebenen Unterschriften
 - nur sicherer
- Absender (Bob) unterschreibt digital das Dokument
 - bestätigt seine Urheberschaft
- Ziel ist ähnlich wie MAC
 - aber mit Hilfe von Public-Key-Kryptographie
 - verifizierbar, nicht fälschbar:
 - Empfänger (Alice) kann anderen beweisen, dass Bob und sonst niemand das Dokument unterschrieben hat

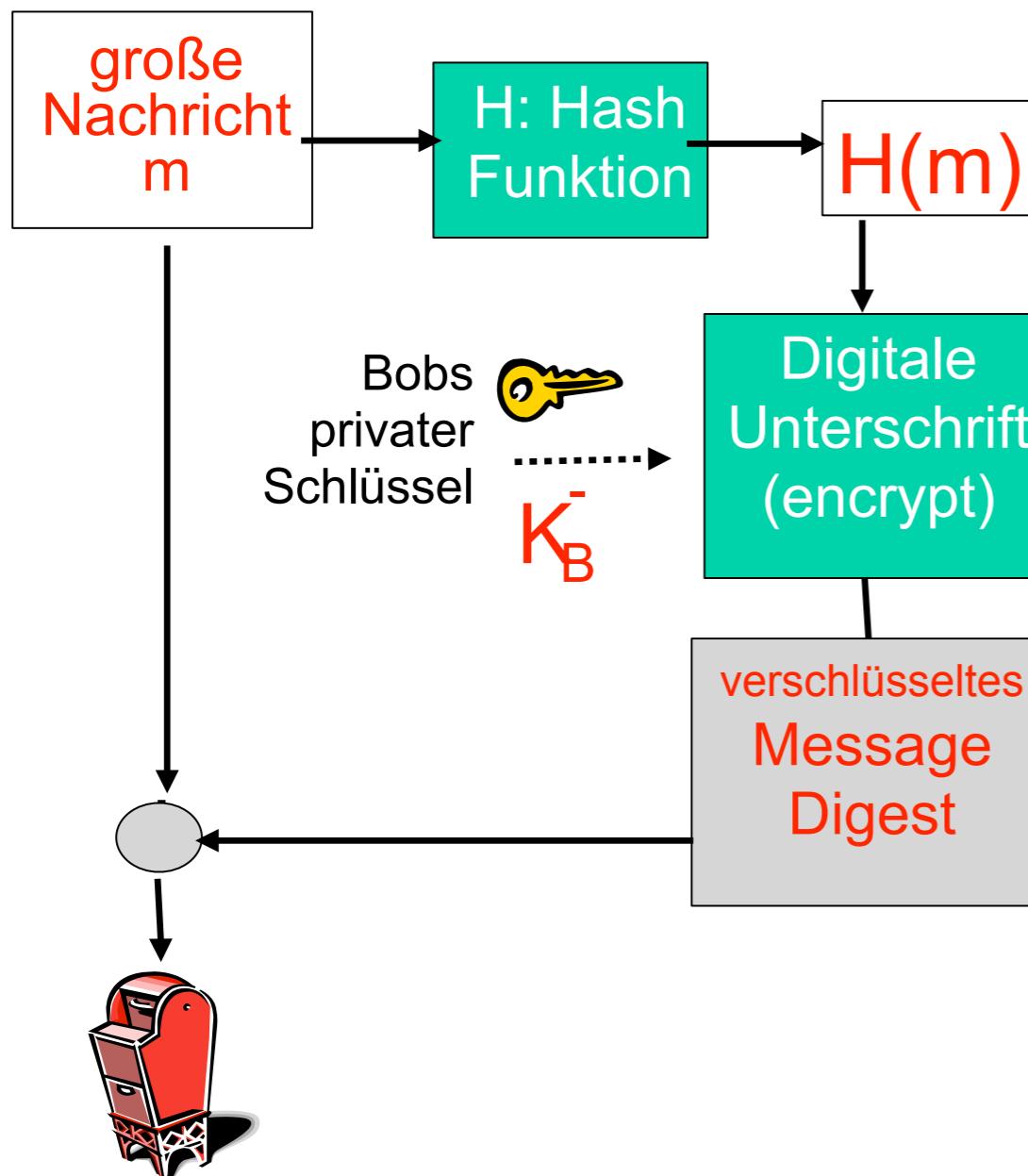
- Digitale Signaturen
 - Unterzeichner besitzt einen geheimen Schlüssel
 - Dokument wird mit geheimen Schlüssel unterschrieben
 - und kann mit einem öffentlichen Schlüssel verifiziert werden
 - Öffentlicher Schlüssel ist allen bekannt
- Beispiel eines Signaturschemas
 - m: Nachricht
 - Unterzeichner
 - berechnet $h(\text{text})$ mit kryptographischer Hashfunktion
 - und veröffentlicht m und
signatur = $g(\text{privat}, h(\text{text}))$, für die Entschlüsselungsfunktion g
 - Kontrolleur
 - berechnet $h(\text{text})$
 - und überprüft $f(\text{offen}, \text{signatur}) = h(\text{text})$, für die asymmetrische Verschlüsselungsfunktion g

Digitale Unterschrift

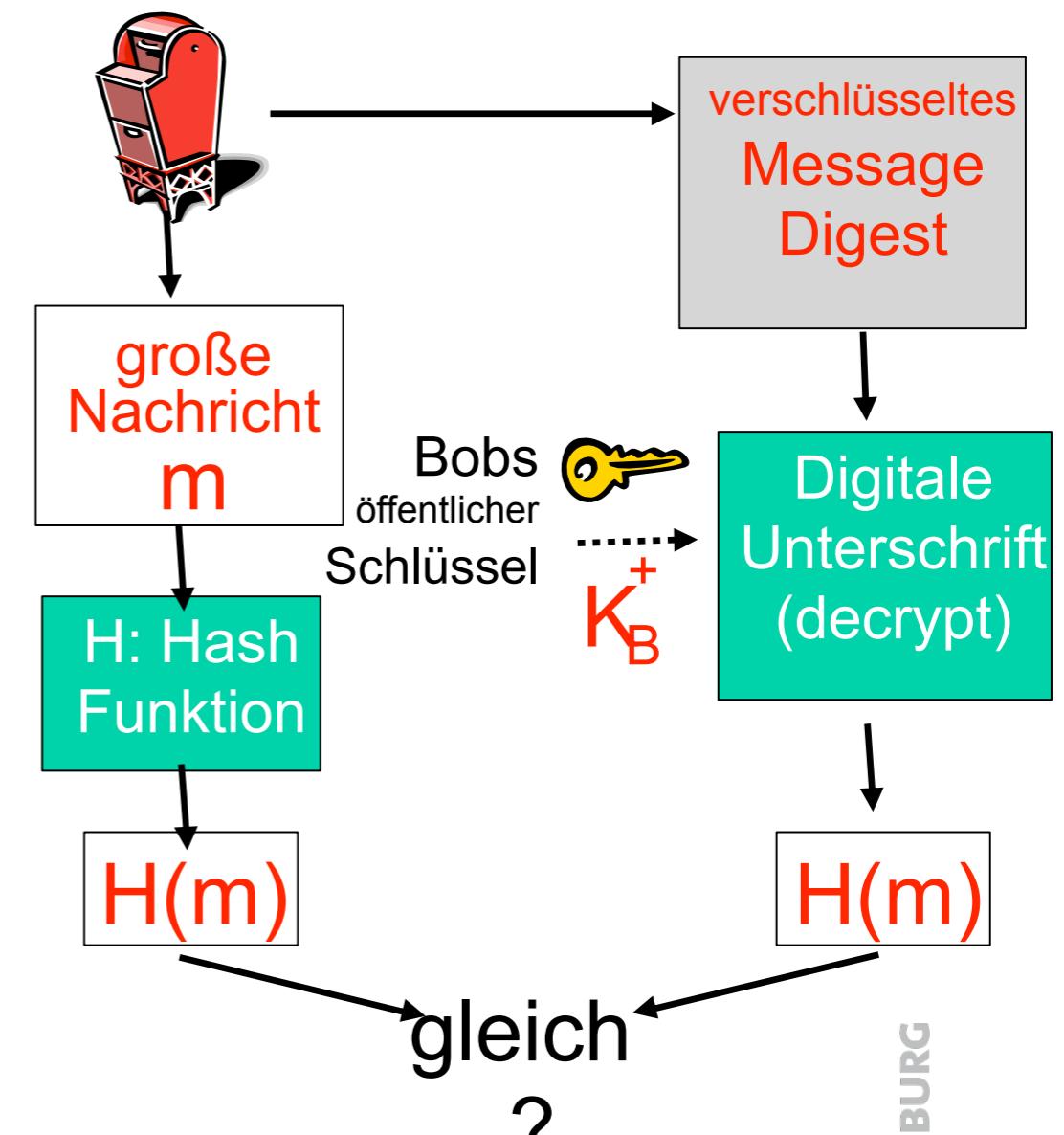


Digitale signature = signiertes Message Digest

Bob sendet eine digital unterschriebene Nachricht



Alice überprüft die Unterschrift und die Korrektheit der Nachricht



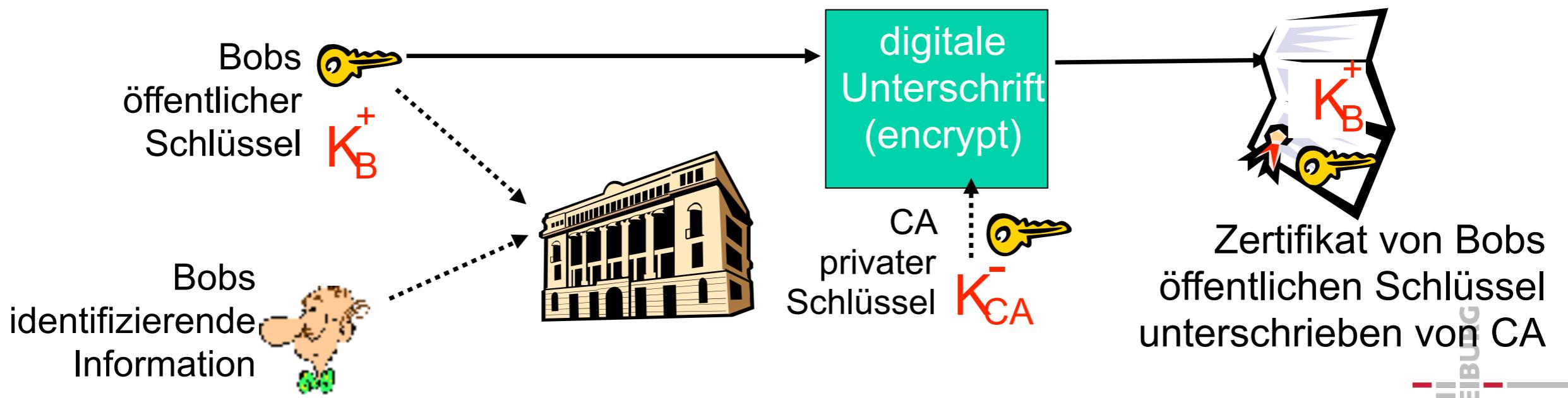
- Angenommen Alice erhält
 - die Nachricht m
 - mit digitaler Unterschrift $K_B^-(m)$
- Alice überprüft m
 - mit den öffentlichen Schlüssel von Bob
 - Ist $K_B^+(K_B^-(m)) = m$?
- Falls $K_B^+(K_B^-(m)) = m$
 - dann hat jemand Bobs geheimen Schlüssel
- Alice verifiziert daher, dass
 - Bob hat m unterschrieben
 - Niemand anders hat m unterschrieben
 - Bob hat m und nicht $m' \neq m$ unterschrieben
- Unleugbarkeit
 - Alice kann mit m und der Unterschrift vor Gericht gehen und beweisen, dass Bob m unterschrieben hat

Public-key Zertifizierung

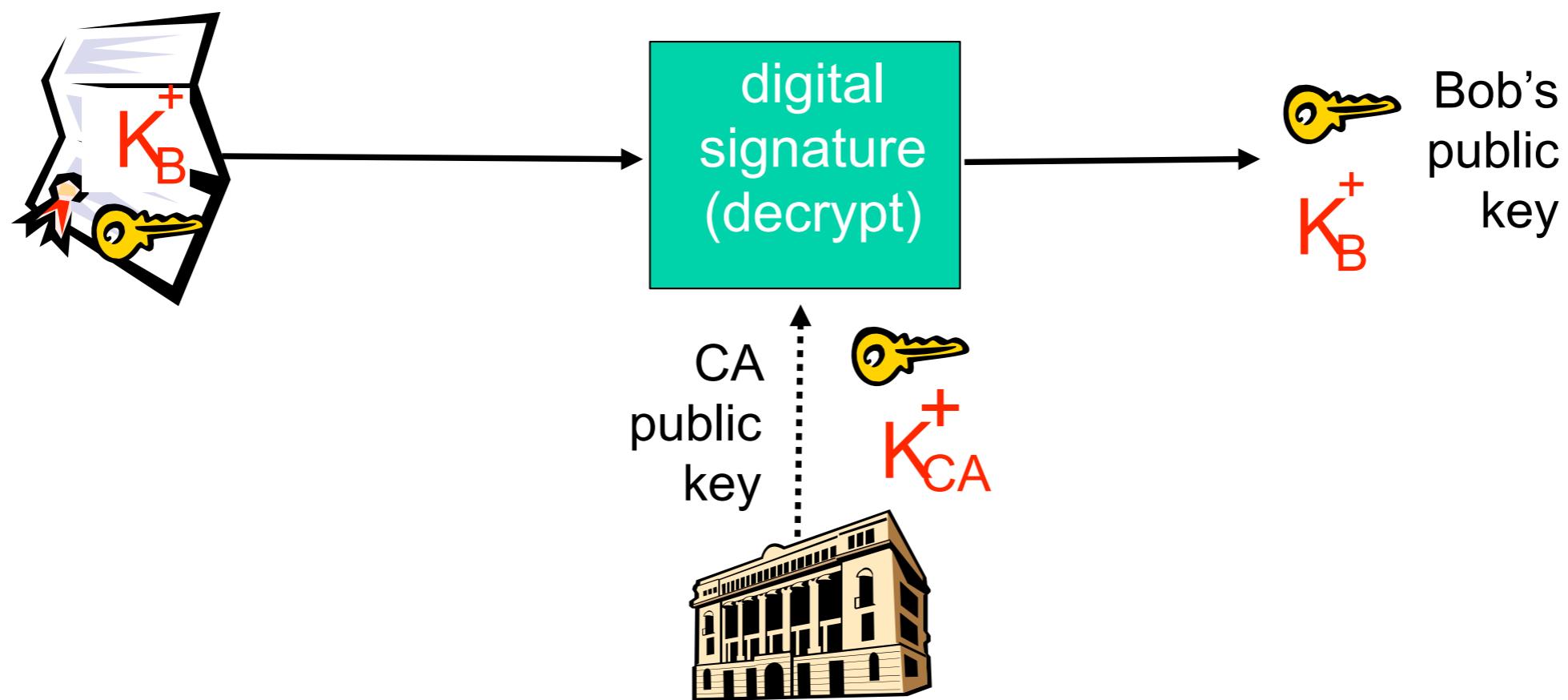
- Motivation: Trudy spielt Bob einen Pizza-Streich
- Trudy bestellt per e-mail order:
 - „Liebe Pizzeria, schick mir bitte vier Pepperoni-Pizza.
vielen Dank Bob“
- Trudy unterschreibt mit ihrem privaten Schlüssel
- Trudy sendet die Bestellung zur Pizzeria
- Trudy sendet der Pizzeria ihren öffentlichen Schlüssel
 - behauptet aber er gehöre Bob
- Die Pizzeria überprüft die Unterschrift
 - Bob mag gar keine Pepperoni

Zertifizierungsstelle Certification Authorities (CA)

- Zertifizierungsstelle (Certification authority – CA): verknüpft öffentlichen Schlüssel mit der Entität (Person, Service, Router) E
- E registriert seinen öffentlichen Schlüssel mit CA
 - E „beweist seine Identität“ der Zertifizierungsstelle
 - CA erzeugt eine Zertifizierungsverknüpfung von E mit seinem öffentlichen Schlüssel
 - Zertifikat mit E's öffentlichen Schlüssel wird von der CA digital unterschrieben:
 - „Das ist der öffentliche Schlüssel von E“



- Wenn Alice Bobs öffentlichen Schlüssel möchte
 - erhält Bobs Zertifikat
 - wendet CA's öffentlichen Schlüssel auf Bobs Zertifikat an
 - Alice erhält Bobs öffentlichen Schlüssel



Zertifikate

- Hauptstandard X.509 (RFC 2459)
- Zertifikat enthält
 - Name des Ausstellers (Issuer name)
 - Name der Entität, Adresse, Domain-Name, etc.
 - Öffentlicher Schlüssel der Entität
 - Digitale Unterschrift (unterschrieben mit dem geheimen Schlüssel des Ausstellers)
- Public-Key Infrastruktur (PKI)
 - Zertifikate und Zertifizierungsstellen

SSL: Secure Sockets Layer

- Weit verbreitetes Sicherheitsprotokoll
 - Unterstützt durch alle Browser und Web-Server
 - https
 - Jährlich Transaktionen im Wert von Zigmilliarden Euro über SSL
- 1993 entworfen von Netscape
- Aktueller Name
 - TLS: transport layer security, RFC 2246
- Gewährleistet
 - Vertraulichkeit (Confidentiality)
 - Nachrichtenintegrität (Integrity)
 - Authentifizierung

SSL: Secure Sockets Layer

- Ursprüngliche Motivation
 - Web E-Commerce Transaktionen
 - Verschlüsselung (Credit-Karte)
- Web-server Authentifizierung
 - Optional Client Authentifizierung
- Kleinstmöglicher Aufwand für Einsteiger
- In allen TCP Anwendungen verfügbar
 - Secure socket interface

- DES – Data Encryption Standard: Block
- 3DES – Triple strength: Block
- RC2 – Rivest Cipher 2: Block
- RC4 – Rivest Cipher 4: Stream

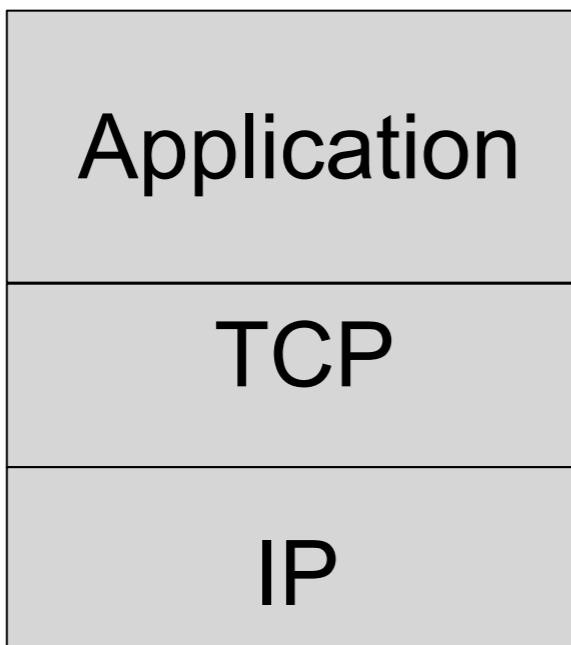
- Auch Public-Key-Verschlüsselung
 - RSA

SSL Cipher Suite

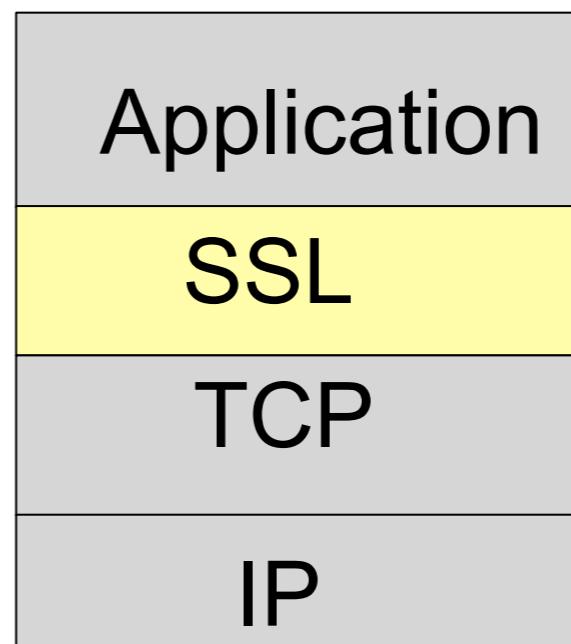
- **Cipher Suite**
 - Public-key Algorithmus
 - Symmetrische Verschlüsselungsalgorithmus
 - MAC Algorithmus
- **SSL unterstützt mehrere Kodierungsverfahren**
- **Verbindungsvereinbarung (Negotiation)**
 - Client und Server einigen sich auf ein Kodierungsverfahren
- **Client bietet eine Auswahl an**
 - Server wählt davon eines

SSL and TCP/IP

- SSL stellt eine Programm-Interface für Anwendungen zur Verfügung
- C and Java SSL Bibliotheken/Klassen verfügbar



Normale Anwendung



Anwendung
mit SSL

- Ziel
 - Server Authentifizierung
 - Verbindungsvereinbarung:
 - Einigung auf gemeinsames kryptographische Verfahren
 - Schlüsselaustausch
 - Client Authentifizierung (optional)

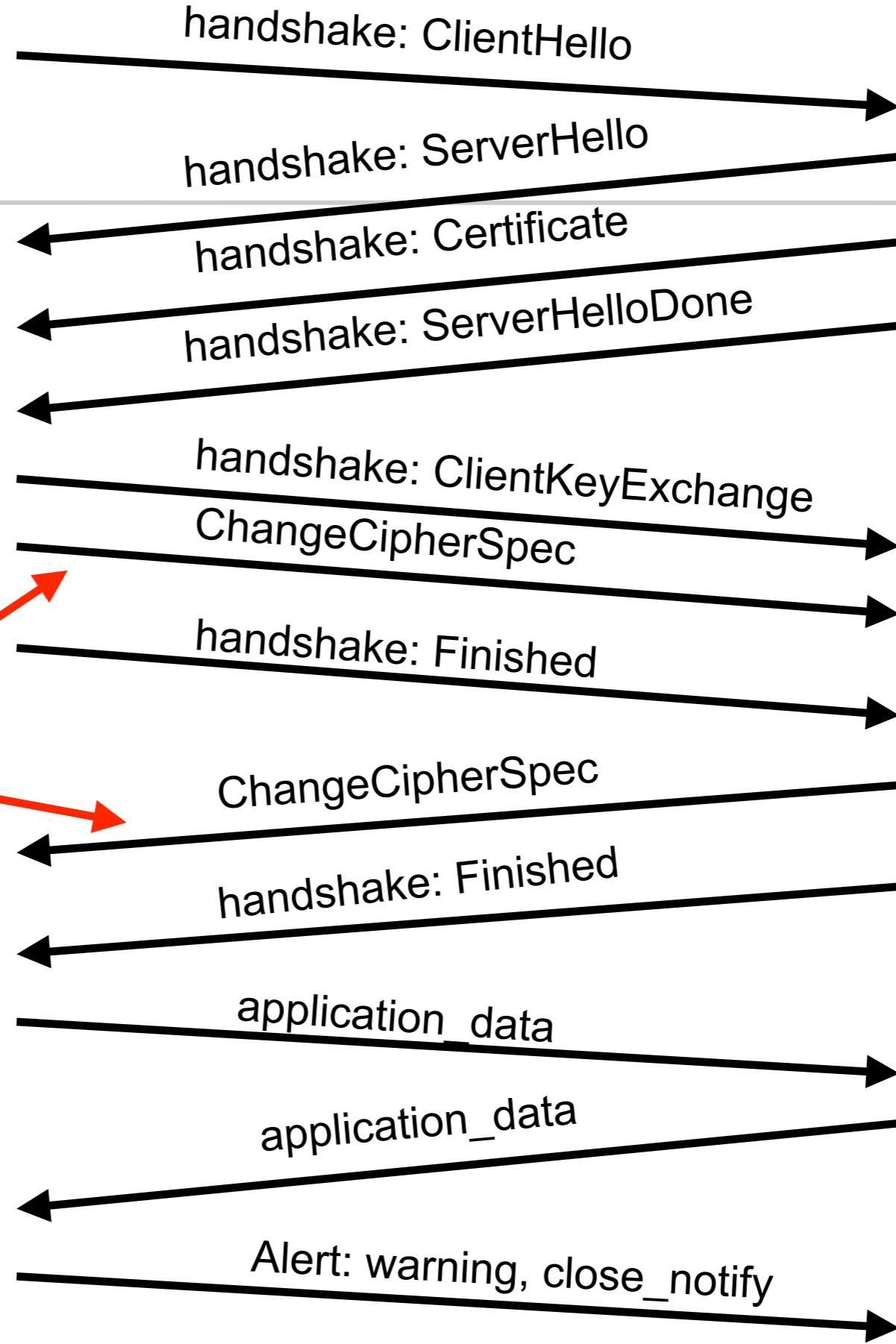
SSL: Handshake (2)

- Client sendet
 - Liste unterstützter Krypto-Algorithmen
 - Client nonce (salt)
- Server
 - wählt Algorithmen von der Liste
 - sendet zurück: Wahl + Zertifikat + Server Nonce
- Client
 - verifiziert Zertifikat
 - extrahiert Servers öffentlichen Schlüssel
 - erzeugt pre_master_secret verschlüsselt mit Servers öffentlichen Schlüssel
 - sendet pre_master_secret zum Server
- Client und Server
 - berechnen unabhängig die Verschlüsselungs- und MAC-Schlüssel aus pre_master_secret und Nonces
- Client sendet ein MAC von allen Handshake-Nachrichten
- Server sendet ein MAC von allen Handshake-Nachrichten



Ab hier ist
alles verschlüsselt

TCP Fin folgt



Schlüsselherstellung

- Client Nonce, Server Nonce und pre-master secret werden in Pseudozufallsgenerator gegeben
 - Ausgabe: Master Secret
- Master Secret und neue Nonces werden in anderen Pseudozufallsgenerator mit Ausgabe: “key block”
- Key block:
 - Client MAC key
 - Server MAC key
 - Client encryption key
 - Server encryption key
 - Client initialization vector (IV)
 - Server initialization vector (IV)

- Spielt eine Rolle in den Schichten
 - Bitübertragungsschicht
 - Sicherungsschicht
 - Vermittlungsschicht
 - Transportschicht
 - Anwendungsschicht
- Was ist eine Bedrohung (oder ein Angriff)?
- Welche Methoden gibt es?
 - Kryptographie
- Wie wehrt man Angriffe ab?
 - Beispiel: Firewalls

Was ist eine Bedrohung?

- **Definition:**
 - Eine Bedrohung eines Rechnernetzwerks ist jedes mögliche Ereignis oder eine Folge von Aktionen, die zu einer Verletzung von Sicherheitszielen führen kann
 - Die Realisierung einer Bedrohung ist ein Angriff
- **Beispiel:**
 - Ein Hacker erhält Zugang zu einem geschlossenen Netzwerk
 - Veröffentlichung von durchlaufenden E-Mails
 - Fremder Zugriff zu einem Online-Bankkonto
 - Ein Hacker bringt ein System zum Absturz
 - Jemand agiert unautorisiert im Namen anderer (Identity Theft)

Sicherheitsziele

- **Vertraulichkeit:**
 - Übertragene oder gespeicherte Daten können nur vom vorbestimmten Publikum gelesen oder geschrieben werden
 - Vertraulichkeit der Identität der Teilnehmer: Anonymität
- **Datenintegrität**
 - Veränderungen von Daten sollten entdeckt werden
 - Der Autor von Daten sollte erkennbar sein
- **Verantwortlichkeit**
 - Jedem Kommunikationsereignis muss ein Verursacher zugeordnet werden können
- **Verfügbarkeit**
 - Dienste sollten verfügbar sein und korrekt arbeiten
- **Zugriffskontrolle**
 - Dienste und Informationen sollten nur autorisierten Benutzern zugänglich sein

Angriffe

- Maskierung (Masquerade)
 - Jemand gibt sich als ein anderer aus
- Abhören (Eavesdropping)
 - Jemand liest Informationen, die nicht für ihn bestimmt sind
- Zugriffsverletzung (Authorization Violation)
 - Jemand benutzt einen Dienst oder eine Resource, die nicht für ihn bestimmt ist
- Verlust oder Veränderung (übertragener) Information
 - Daten werden verändert oder zerstört
- Verleugnung der Kommunikation
 - Jemand behauptet (fälschlicherweise) nicht der Verursacher von Kommunikation zu sein
- Fälschen von Information
 - Jemand erzeugt (verändert) Nachrichten im Namen anderer
- Sabotage
 - Jede Aktion, die die Verfügbarkeit oder das korrekte Funktionieren der Dienste oder des Systems reduziert

Bedrohungen und Sicherheitsziele

Sicherheits- ziele	Bedrohungen						
	Mas- kierung	Abhören	Zugriffs- ver- letzung	Verlust oder Verän- derung (über- tragener) information	Verleug- nung der Kommuni- kation	Fäl- schen von Infor- mation	Sabotage (z.B. Überlast)
Vertraulichkeit	x	x	x				
Datenintegrität	x		x	x		x	
Verantwort- lichkeit	x		x		x	x	
Verfügbarkeit	x		x	x			x
Zugriffs- kontrolle	x		x			x	

- **Sicherheitsdienst**
 - Ein abstrakter Dienst, der eine Sicherheitseigenschaft zu erreichen sucht
 - Kann mit (oder ohne) Hilfe kryptografischer Algorithmen und Protokolle realisiert werden, z.B.
 - Verschlüsselung von Daten auf einer Festplatte
 - CD im Safe
- **Kryptografischer Algorithmus**
 - Mathematische Transformationen
 - werden in kryptografischen Protokollen verwendet
- **Kryptografisches Protokoll**
 - Folge von Schritten und auszutauschenden Nachrichten um ein Sicherheitsziel zu erreichen

- Authentisierung
 - Digitale Unterschrift: Das Datum ist nachweislich vom Verursacher
- Integrität
 - Sichert ab, dass ein Datum nicht unbemerkt verändert wird
- Vertraulichkeit
 - Das Datum kann nur vom Empfänger verstanden werden
- Zugriffskontrolle
 - kontrolliert, dass nur Berechtigte Zugang zu Diensten und Information besitzen
- Unleugbarkeit
 - beweist, dass die Nachricht unleugbar vom Verursacher ist

Systeme II

7. Sicherheit

Christian Schindelhauer
Technische Fakultät
Rechnernetze und Telematik
Albert-Ludwigs-Universität Freiburg