

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

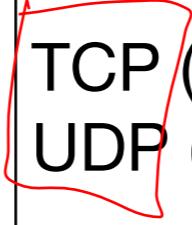
Übungen

- 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

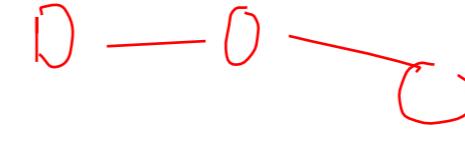
- Erscheinen jeden Mittwoch in ILIAS
 - Abgabe als PDF bis Montag 23.59 Uhr (GMT+1) der Folgewoche
 - Abgabe über ILIAS
 - Namenskonvention beachten:
 - <BlattNr>-<Gruppennummer>-<Matrikelnummer>.pdf
 - 01-G1-726818.pdf
- Grundlage für schriftliche Klausur
- Besprechung am Tag nach der Abgabe
 - Korrektur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

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.)

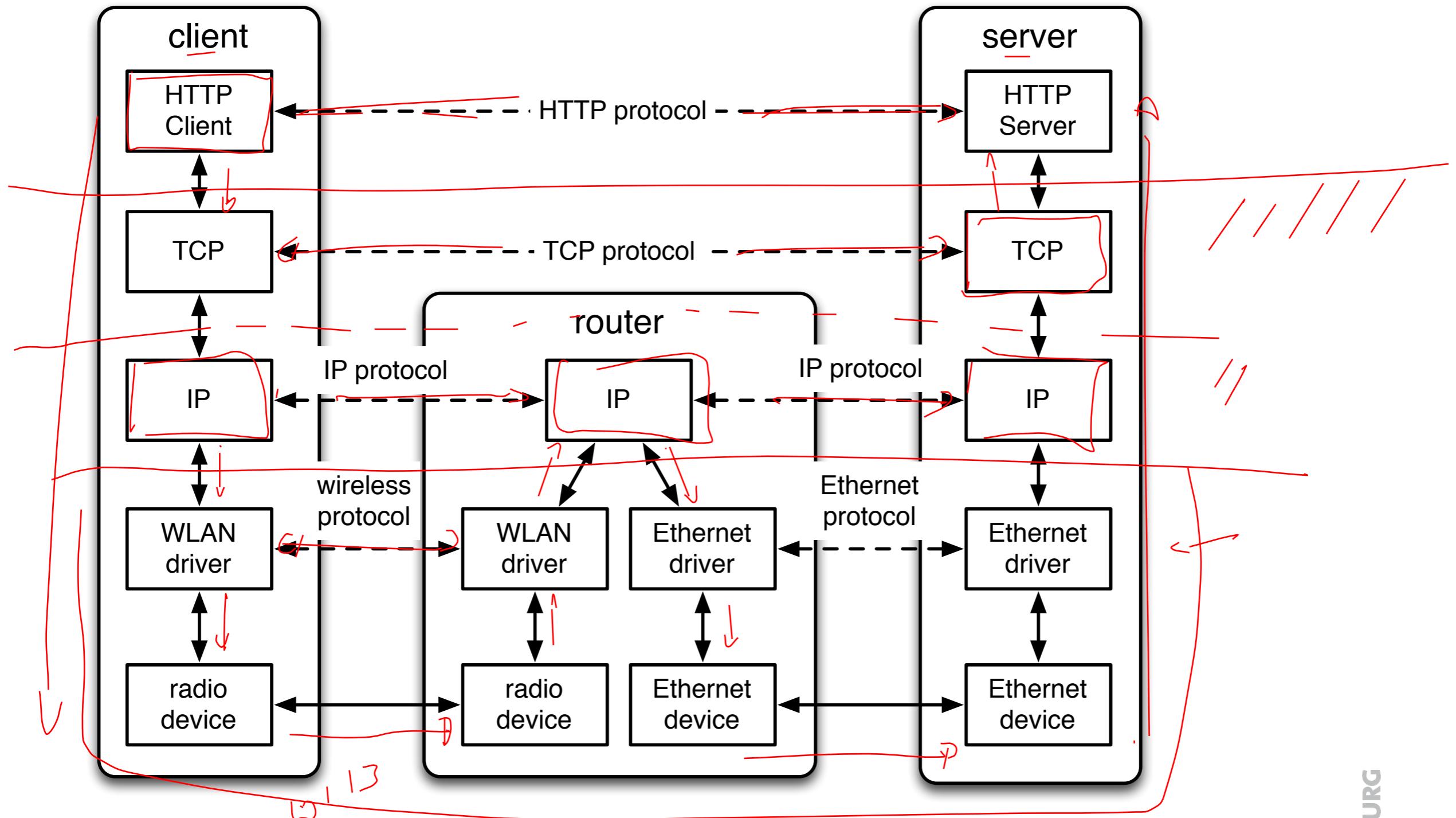
Internet-Schichtenmodell

- 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, ...

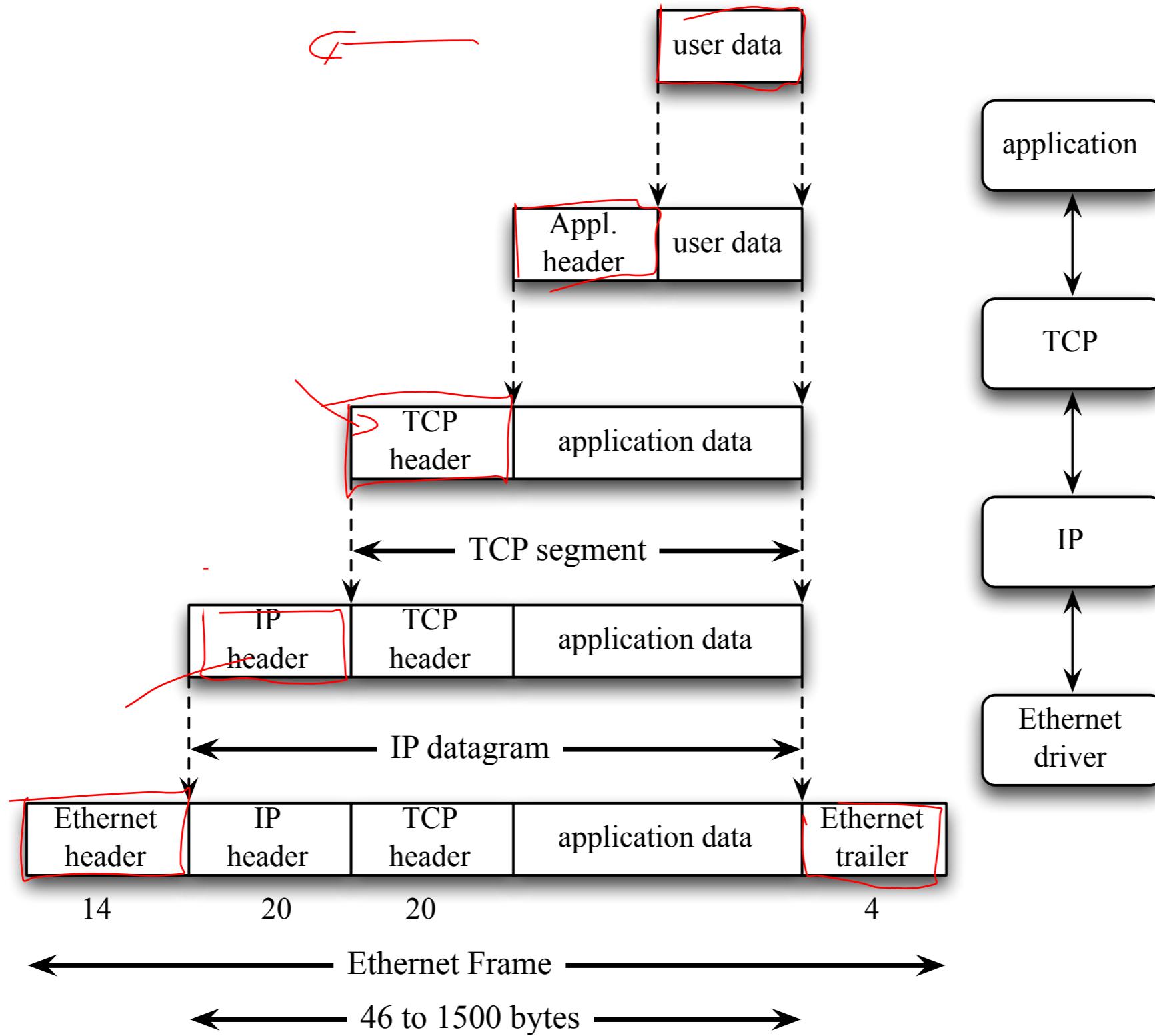


LJ

Beispiel zum Zusammenspiel der Schichten

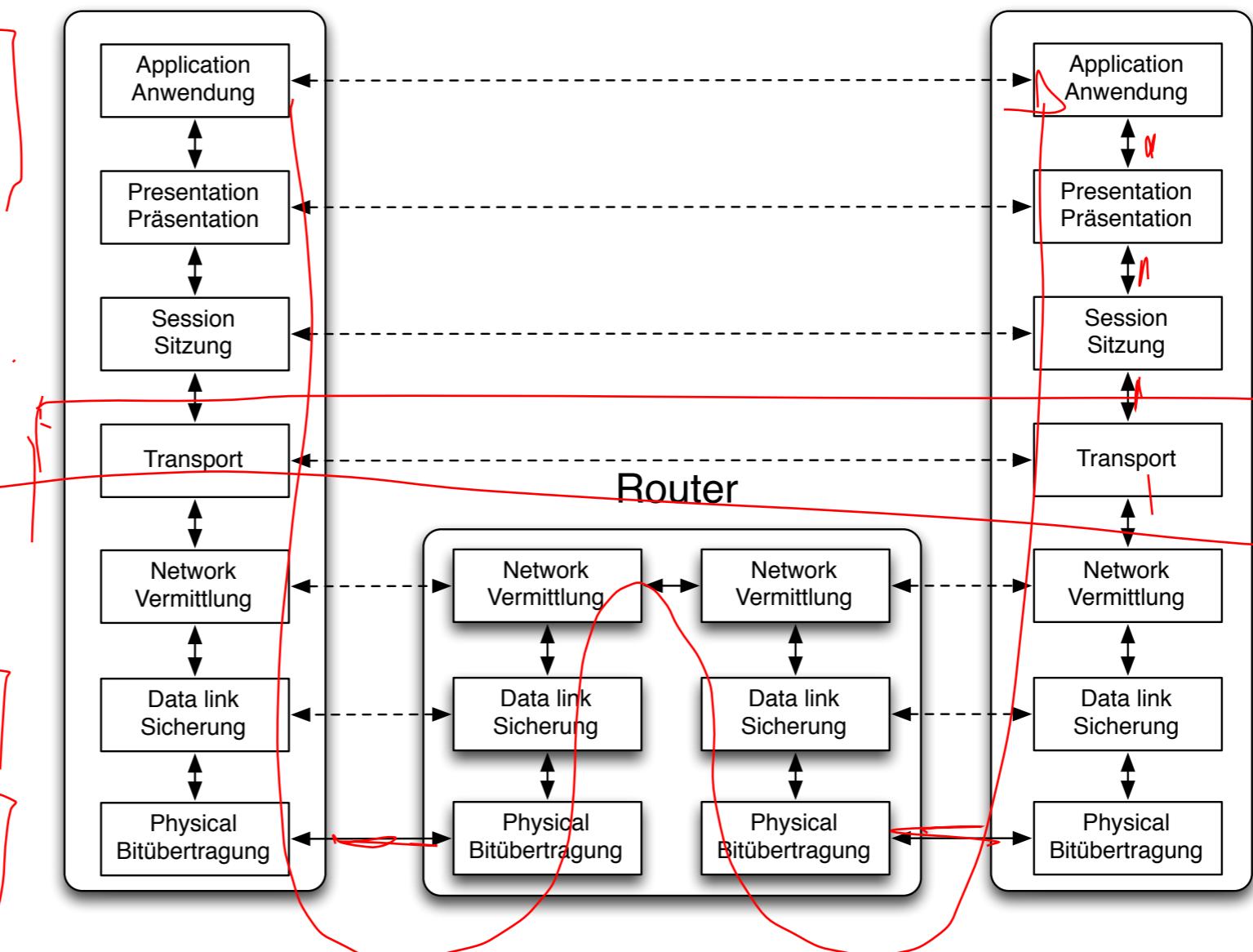


Datenkapselung



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)

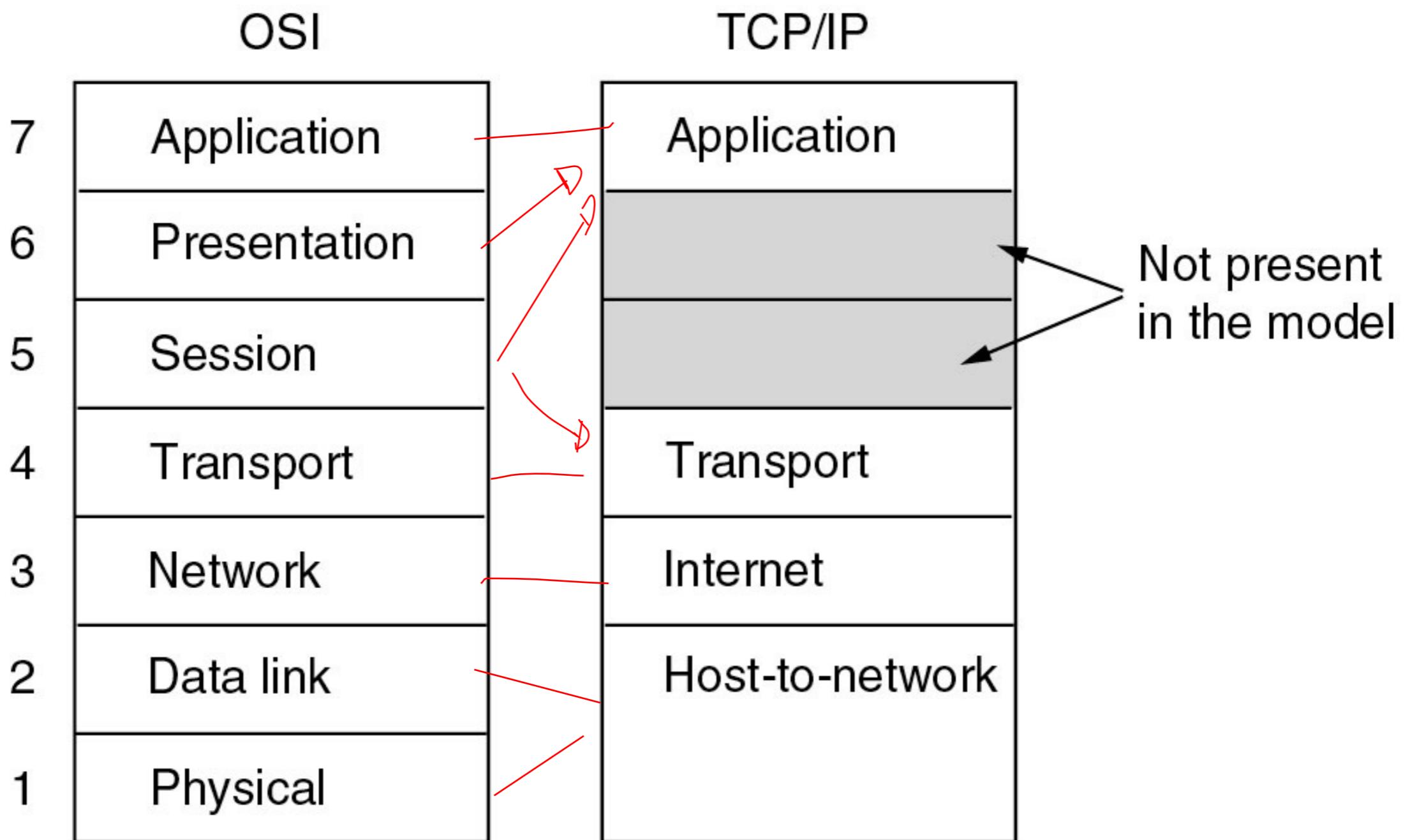
6. Präsentationsschicht

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

7. Anwendungsschicht

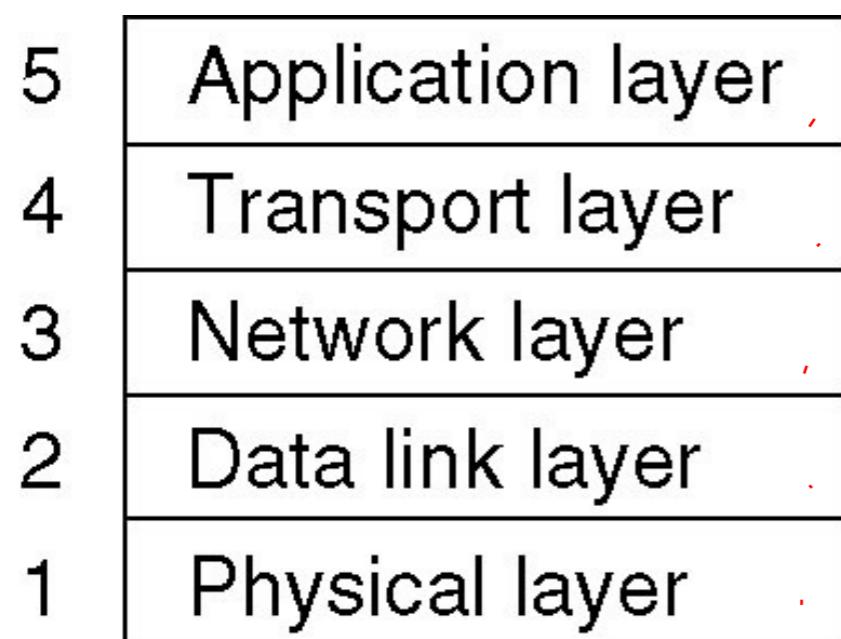
- Große Vielfalt aller möglichen Funktionen, z.B.
 - Virtuelle Terminals, Filetransfer, E-mail, Online-Video, Twitter, Radio-Streams, Internet-Telefonie, Online-Games ...

OSI versus TCP/IP



Hybrides Modell

Tanenbaum



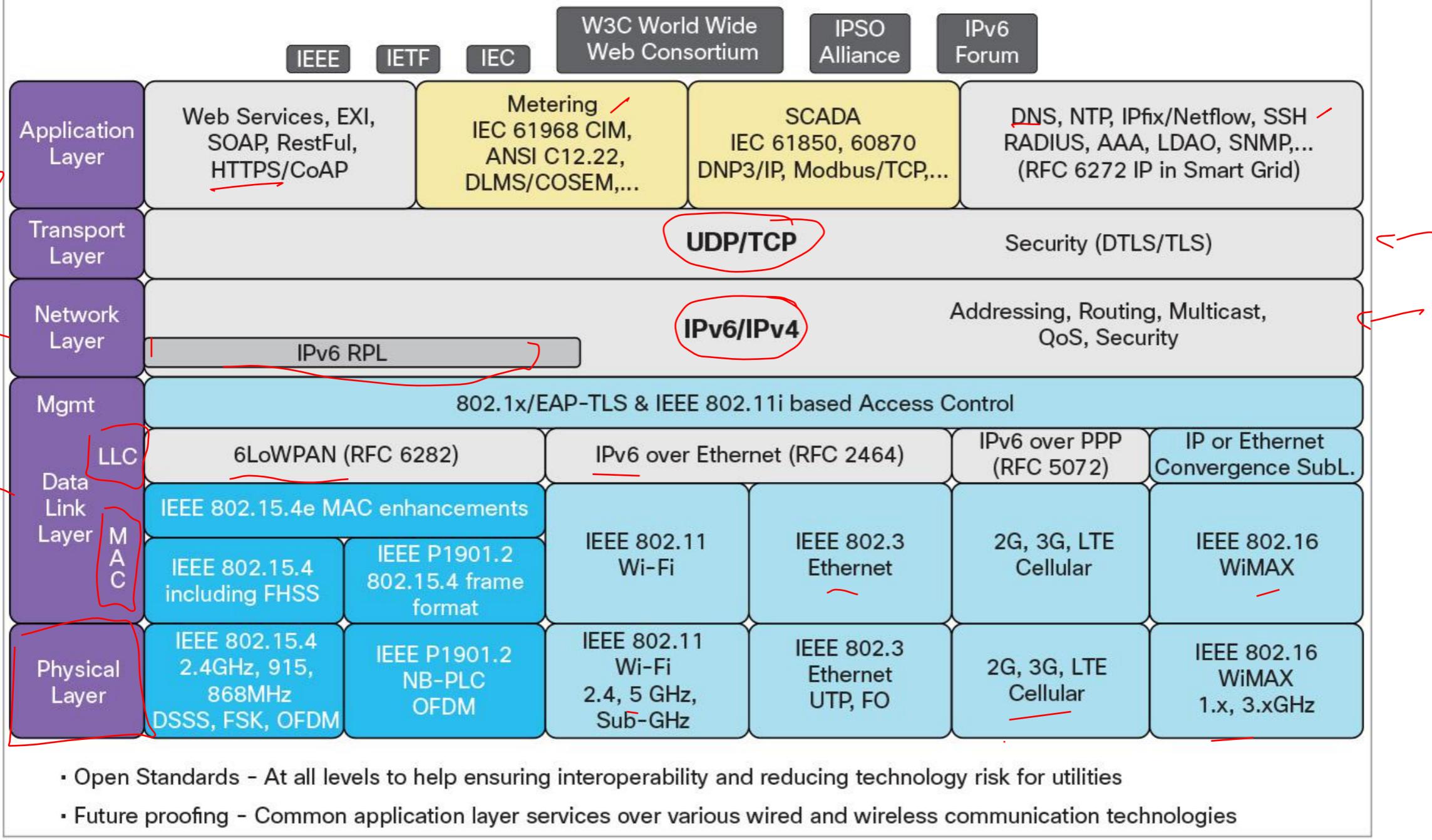
MAC-Layer

TCP/IP

(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

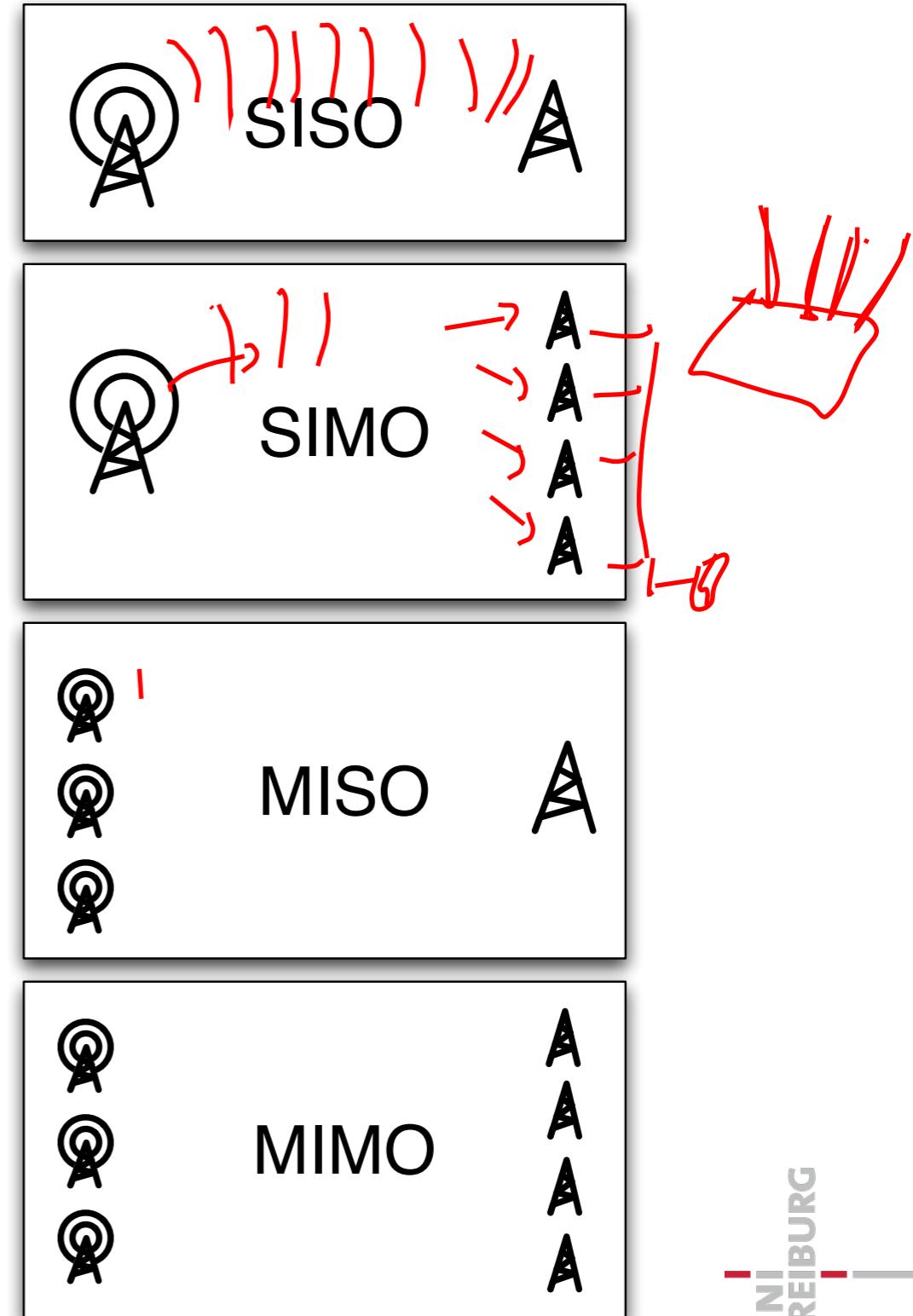
Rechnernetze und Telematik

Albert-Ludwigs-Universität Freiburg

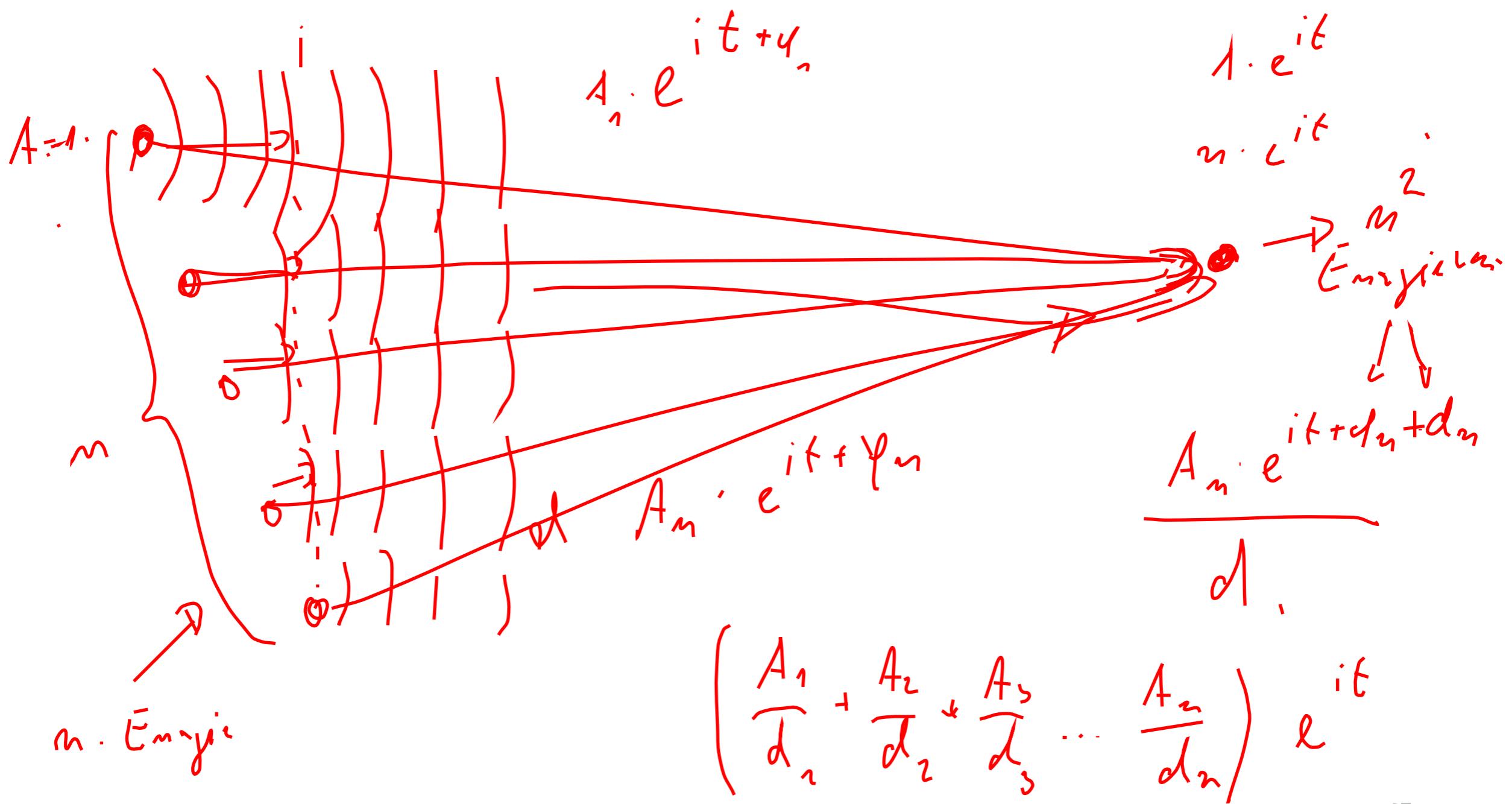
Version 26.04.2017

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



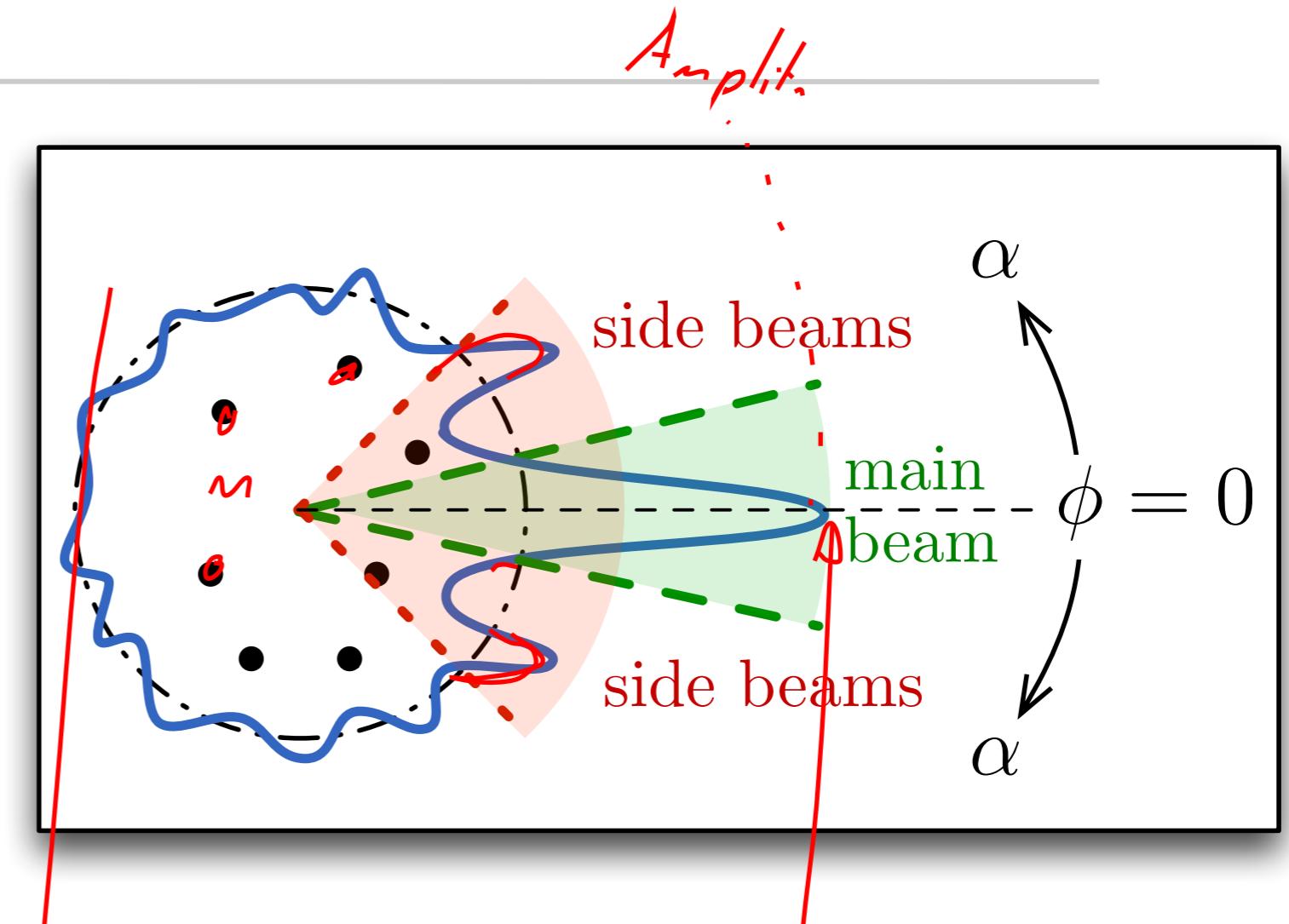
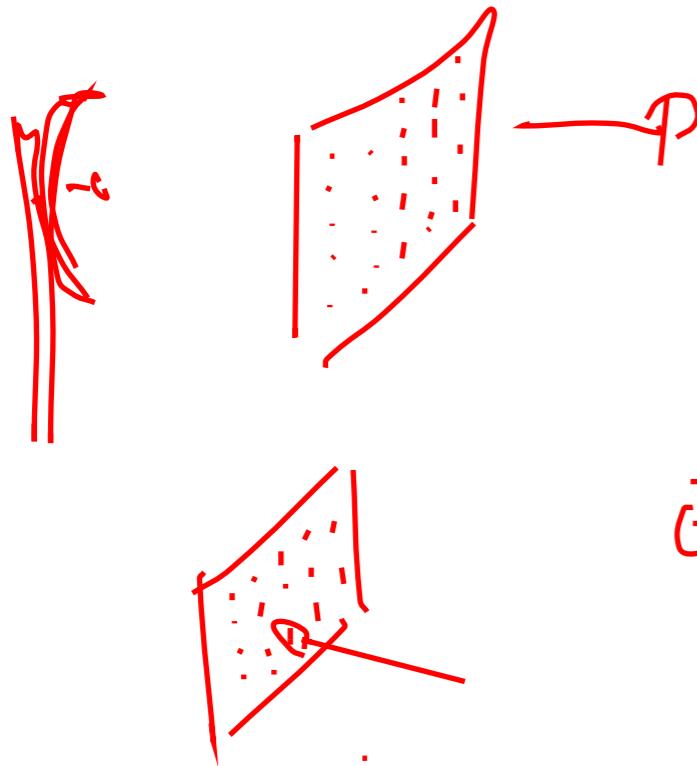
Superposition



$$\left(\frac{A_1}{d_1} + \frac{A_2}{d_2} + \frac{A_3}{d_3} + \dots + \frac{A_n}{d_n} \right) e^{it}$$

Beamforming

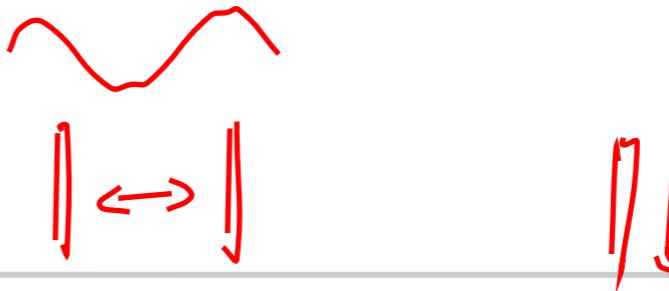
- Durch geschickte Phasenverschiebung kann ein gerichteter Sendestrahl gesendet werden
 - oder symmetrisch auch empfangen werden



Amplitude
Energy ist ~~m^{-1}~~ groß

Amplitude: $m^{-1} \text{ groß}$
Energy: m^2

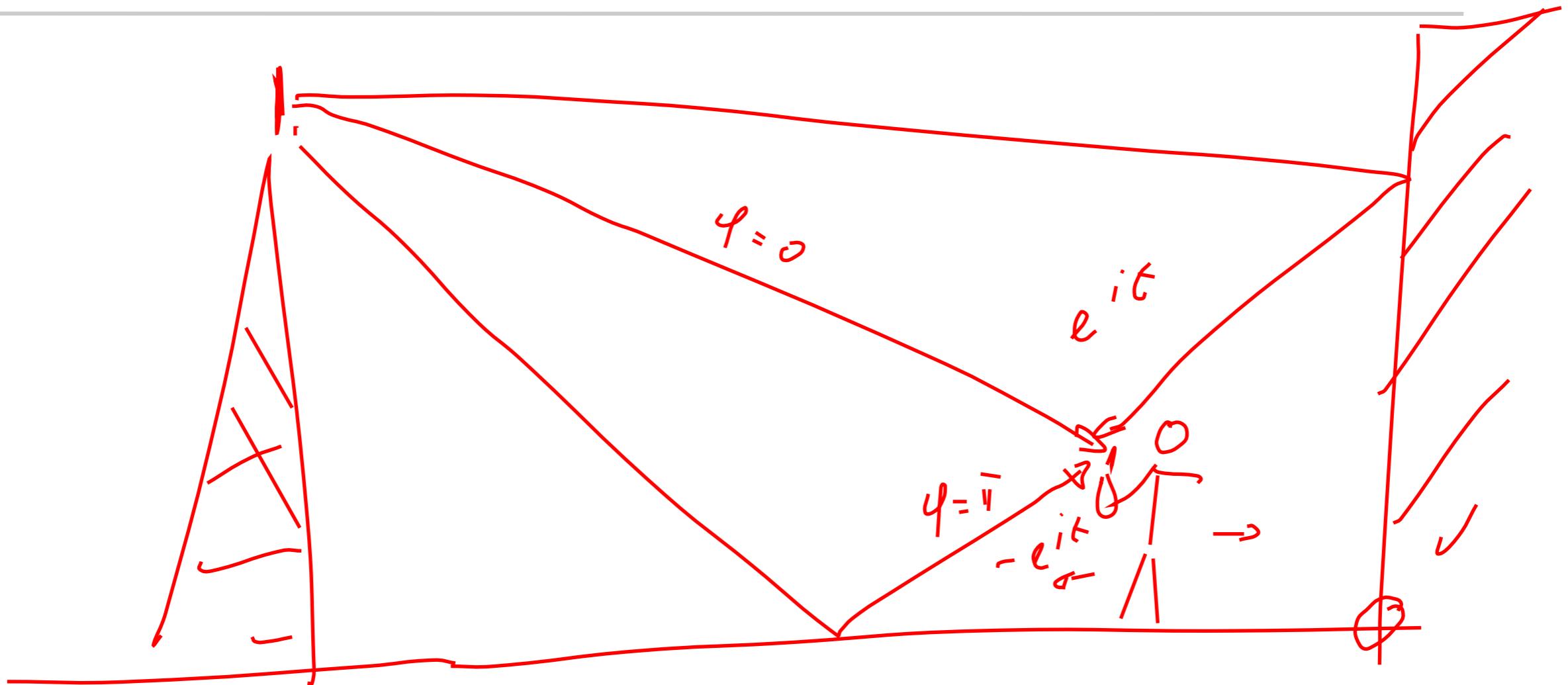
Power Gain



- 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 = $D \sim 1/d$

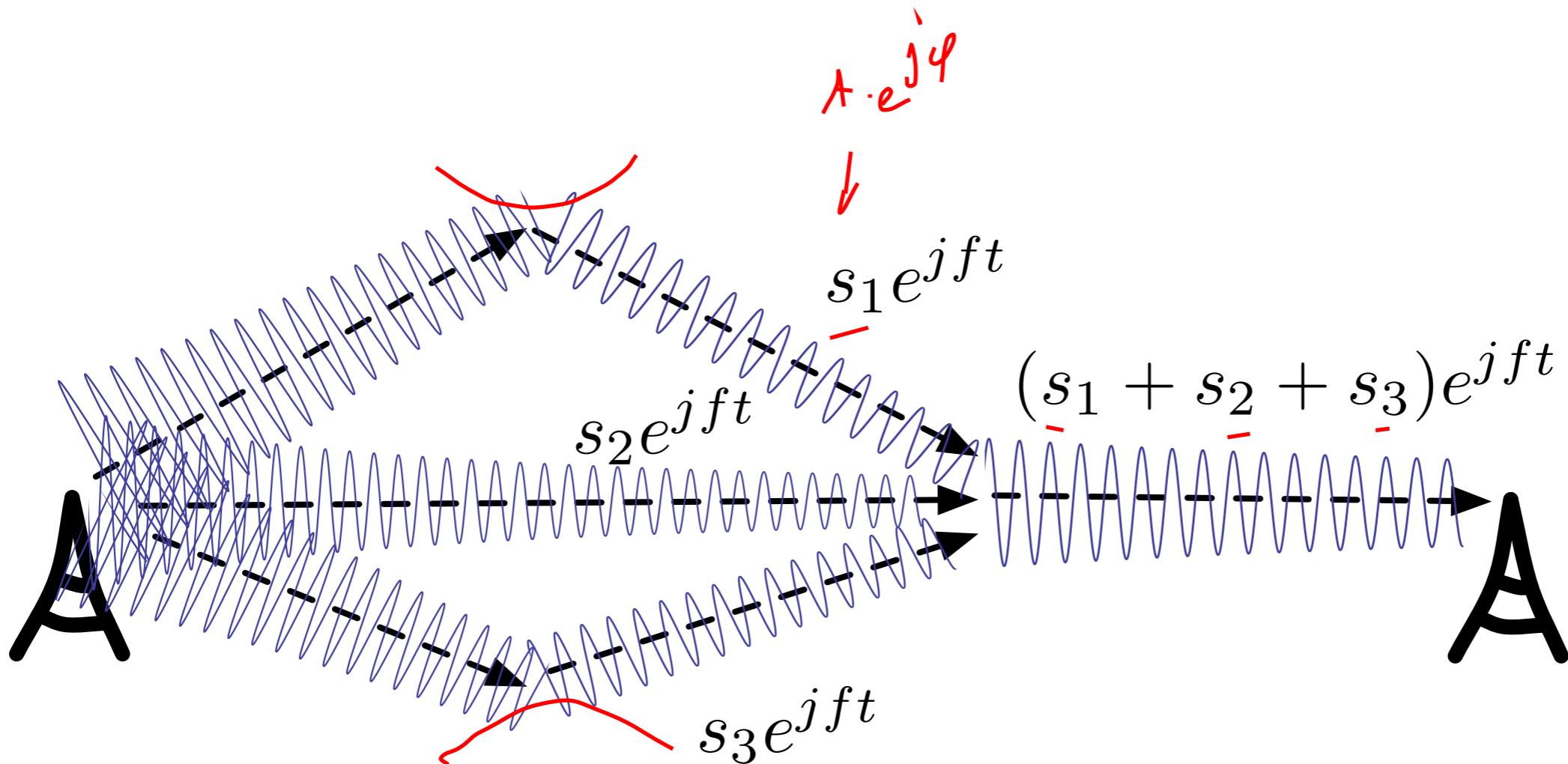
addit.

- 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



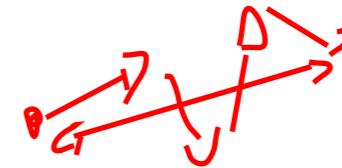
Multipath Channel

- Superposition von Reflektionen

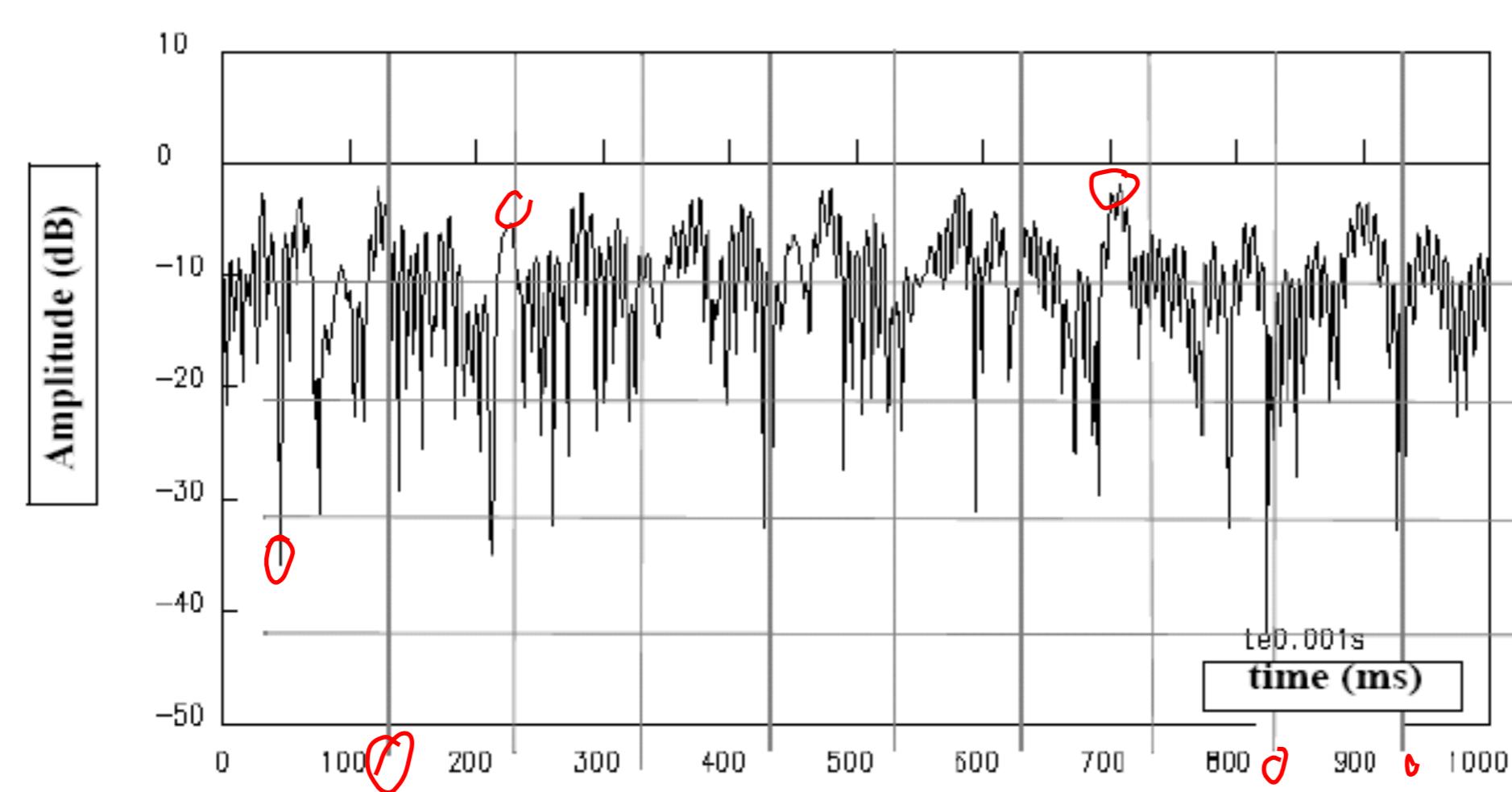


$$h = \underline{s_1 + s_2 + s_3}$$

Rayleigh fading



- Superposition führt zu drastischen Einbrüchen



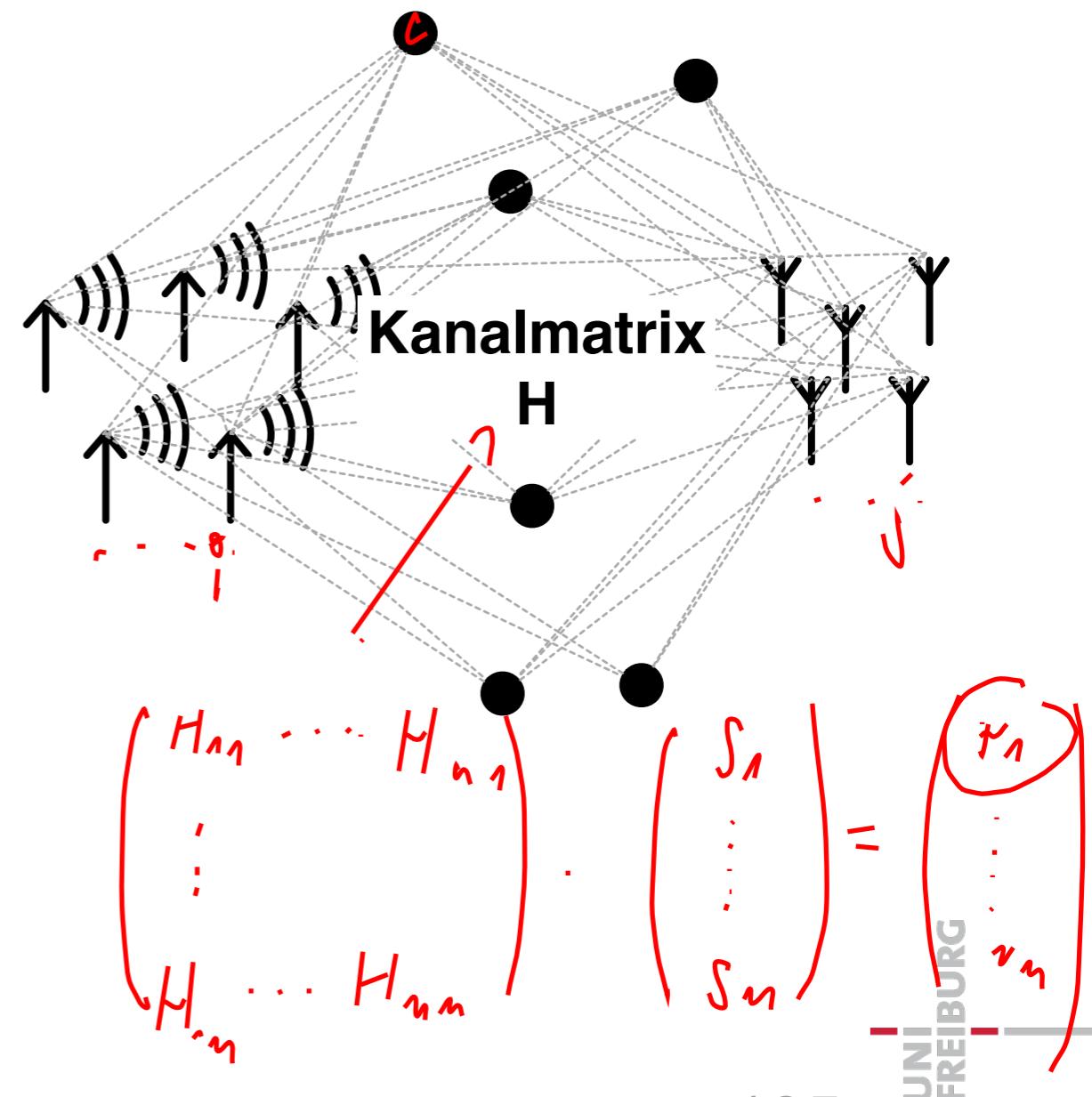
Introduction to Wireless MIMO – Theory and Applications
Jacob Sharony IEEE LI 2006

Diversity Gain

$$H_{ij} \in \mathbb{C}$$

$|H_{ij}|$ = Amplitude
 $\angle H_{ij}$ = Phase

- Wenn in der Umgebung viele Reflektoren (scatterers) vorhanden sind,
 - dann ergibt sich für die Beschreibung der Sender-/Empfänger-Beziehung eine Kanalmatrix H
- $H_{ij} =$
 - 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



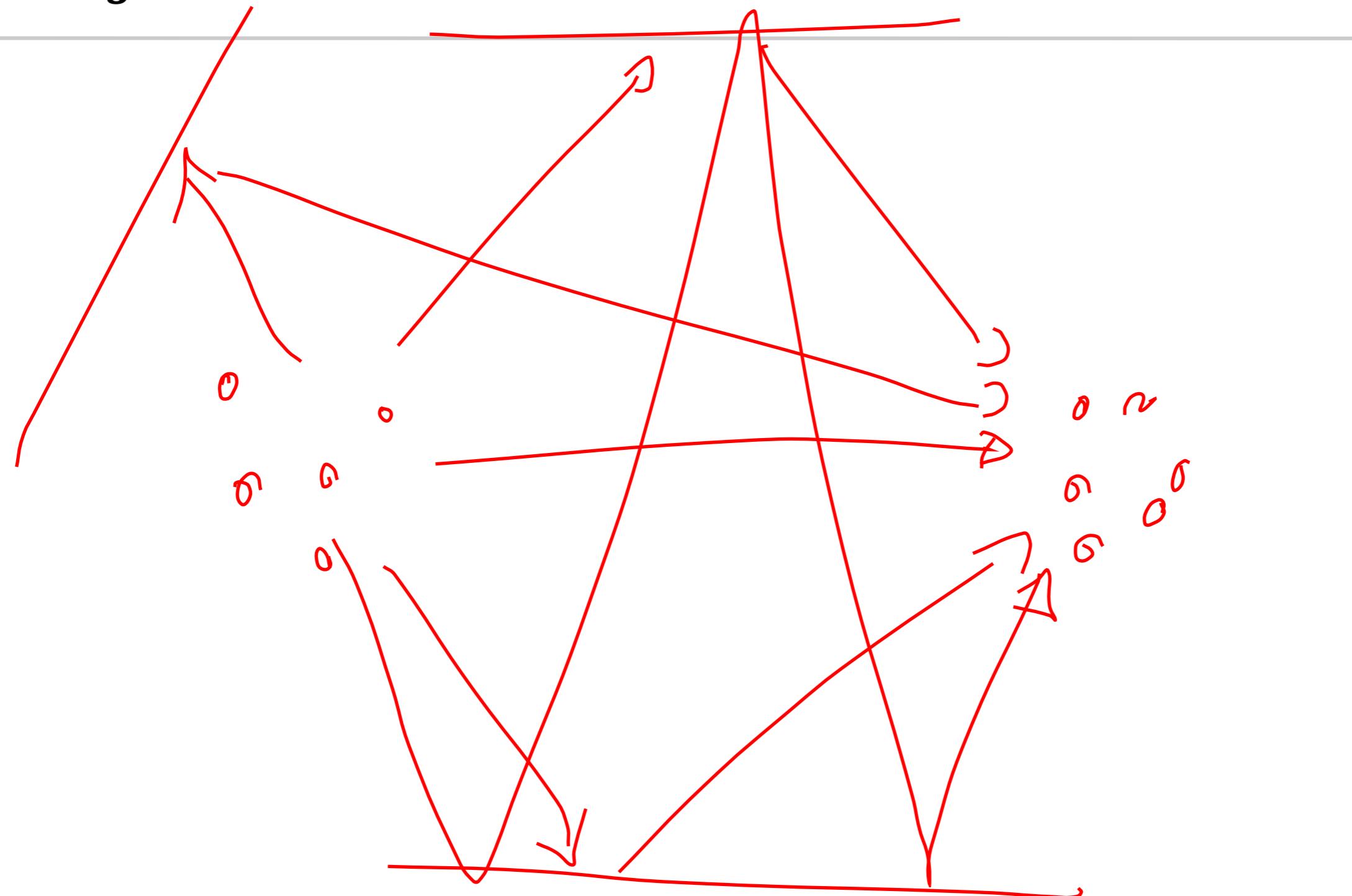
$$H \cdot S^{\downarrow} = r \stackrel{\rightarrow}{=} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix}$$

$$S^{\downarrow} = H^{-1} \cdot r$$

$H \cdot S^{\downarrow}$

$$H \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

$$\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = H^{-1} \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$



Systeme II

2. Die physikalische Schicht

Christian Schindelhauer

Technische Fakultät

Rechnernetze und Telematik

Albert-Ludwigs-Universität Freiburg

S

0 1 0 , 1 0 ' 0

R

UNI
FREIBURG

Systeme II

3. Die Datensicherungsschicht

Christian Schindelhauer

Technische Fakultät

Rechnernetze und Telematik

Albert-Ludwigs-Universität Freiburg

Version 31.05.2017

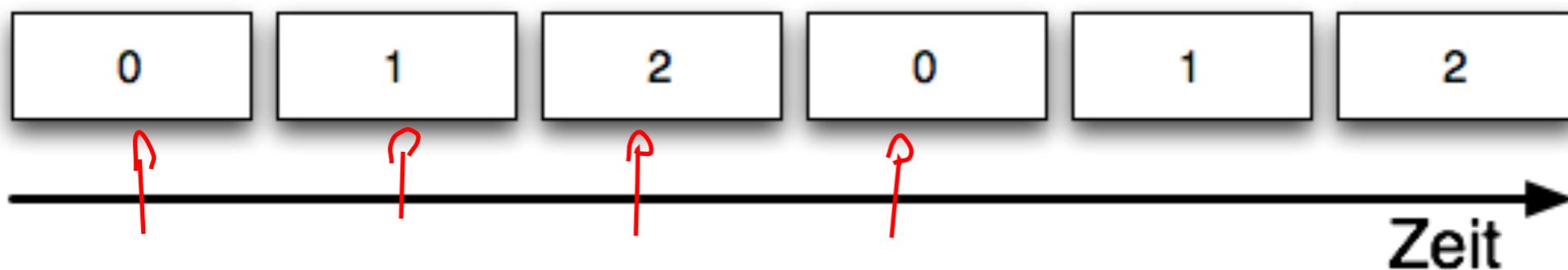
Der Mediumzugriff in der Sicherungsschicht

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

Statisches Multiplexen

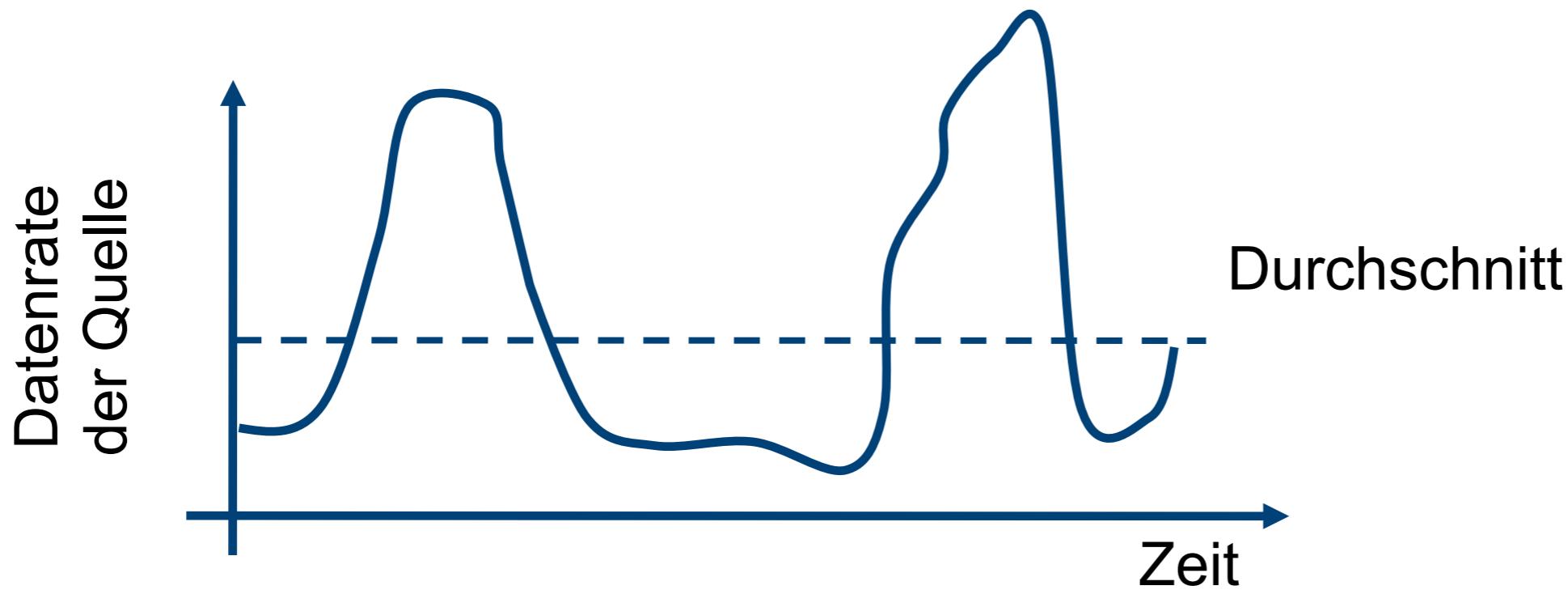
T DMA

- 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



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

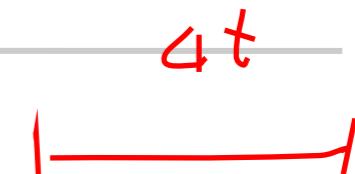
- für alle Stationen
- Keine weitere Verbindungen möglich

■ 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

$$N \rightarrow \infty$$

$X : \# \text{Pakete}$



$P : \text{Wahrsch.}$

$$P[X = k]$$

~~$E[X]$~~ $P[X = 1] = p$

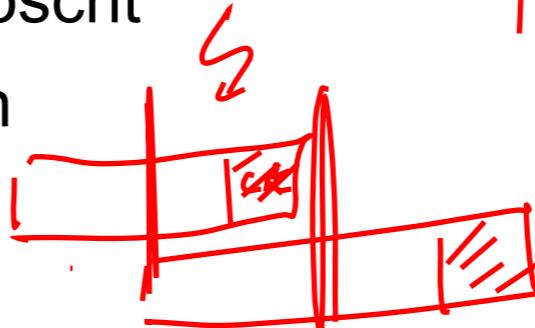
$$P[X = 0] = 1 - p$$

$$E[X] = \sum_{k=0}^{\infty} k \cdot P[X = k]$$

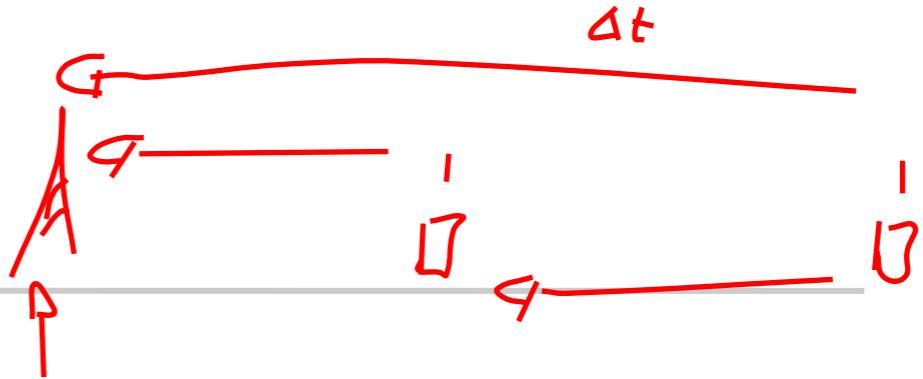
$$= p$$

$$P[X = 2] = \frac{1}{6}$$

$$\sum k \cdot P[X = k]$$



Annahmen



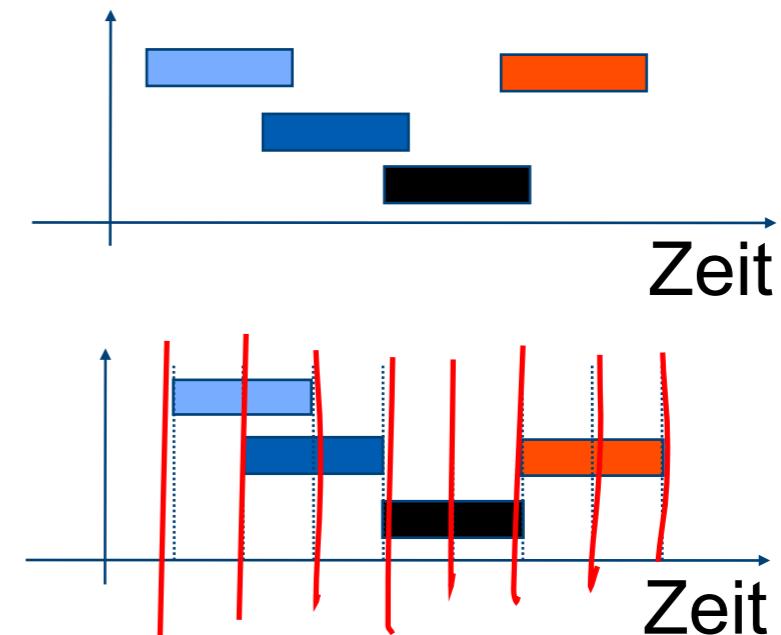
■ Zeitmodelle

- 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)

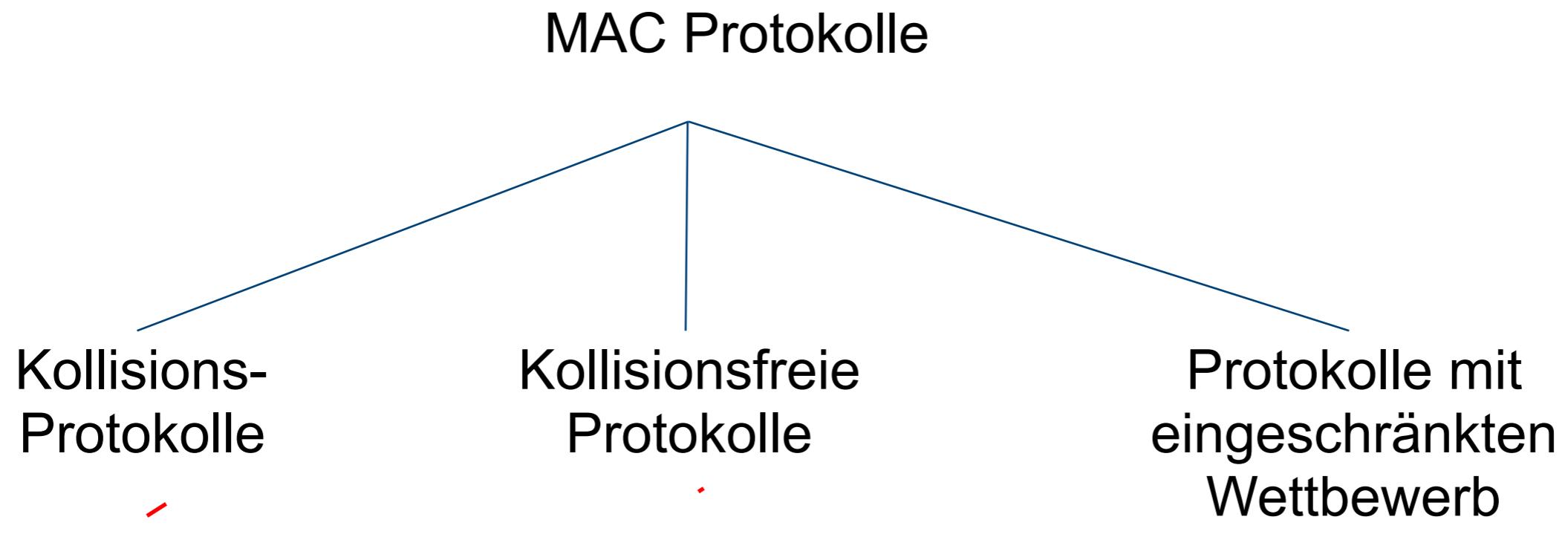
- Stationen können erkennen ob der Kanal momentan von anderen Stationen verwendet wird
 - Nicht notwendigerweise zuverlässig

Bewertung des Verhaltens

- Methoden zur Bewertung der Effizienz einer Kanalzuweisung
- ① Durchsatz (throughput)
 - Anzahl Pakete pro Zeiteinheit
 - Besonders bei großer Last wichtig
- ② Verzögerung (delay)
 - Zeit für den Transport eines Pakets
 - Muss bei geringer Last gut sein
- ③ Gerechtigkeit (fairness)  
 - Gleichbehandlung aller Stationen
 - Fairer Anteil am Durchsatz und bei Delay

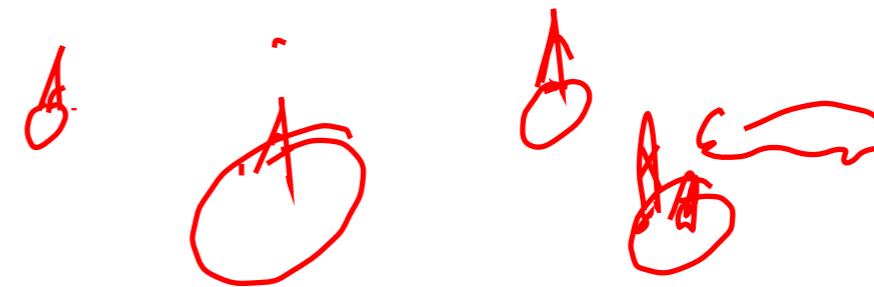
Mögliche MAC-Protokolle

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

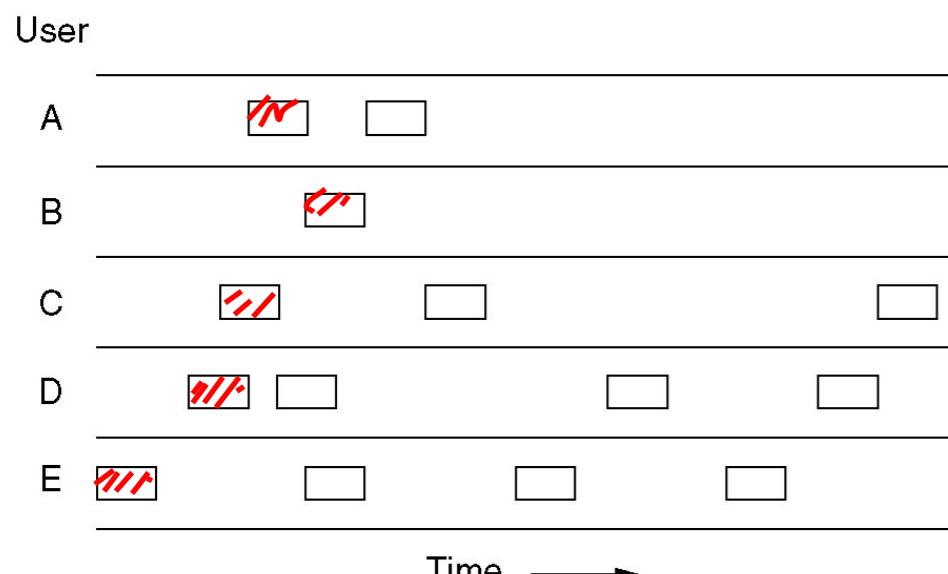
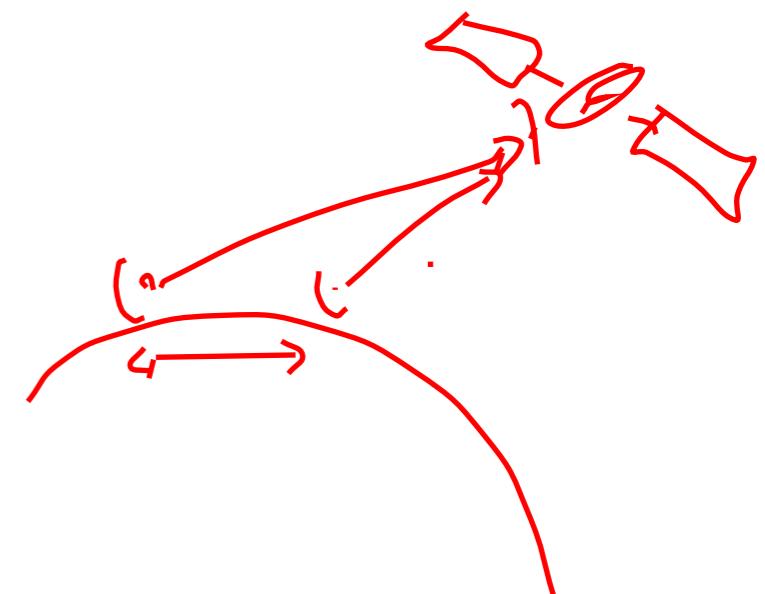


System mit Kollisionen: **Contention System**

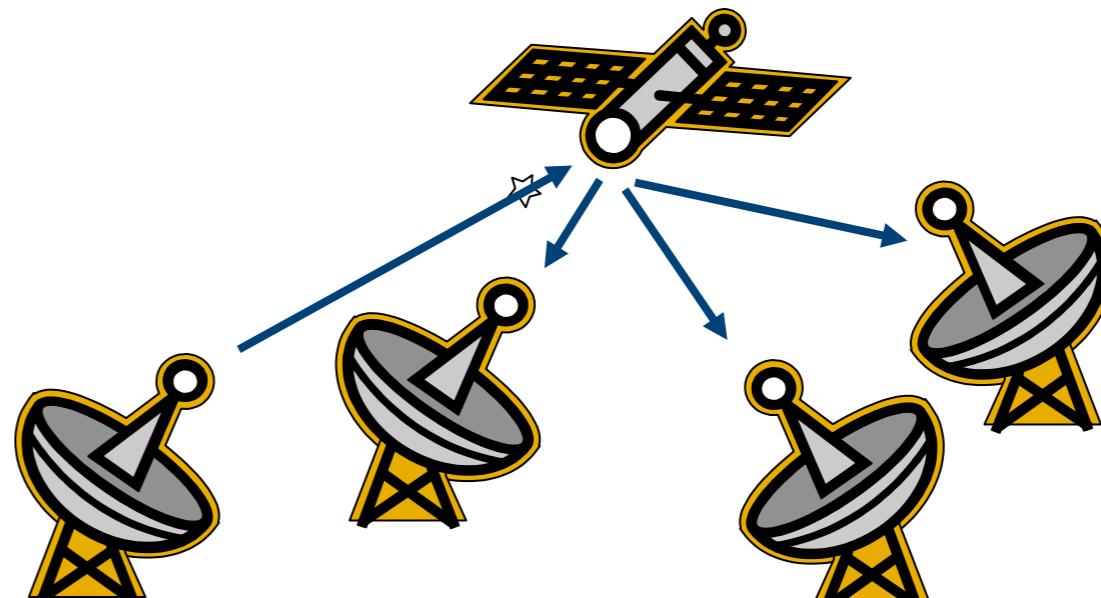
ALOHA



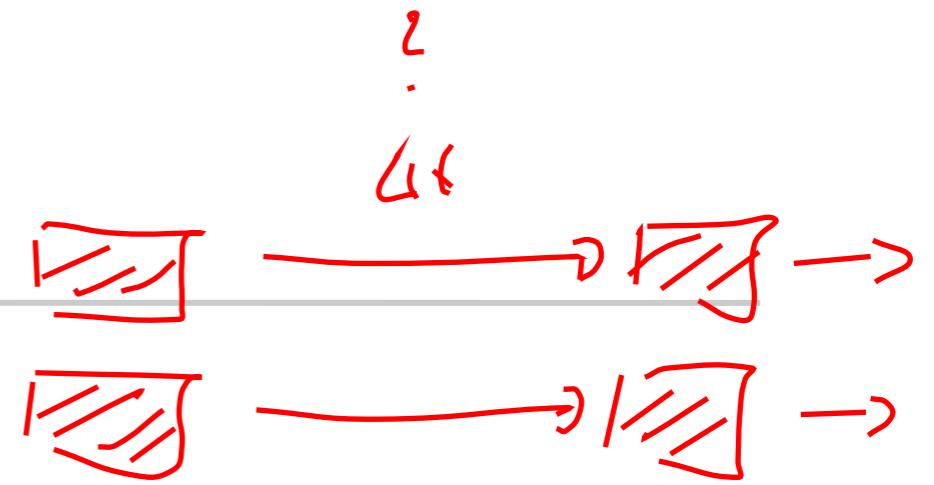
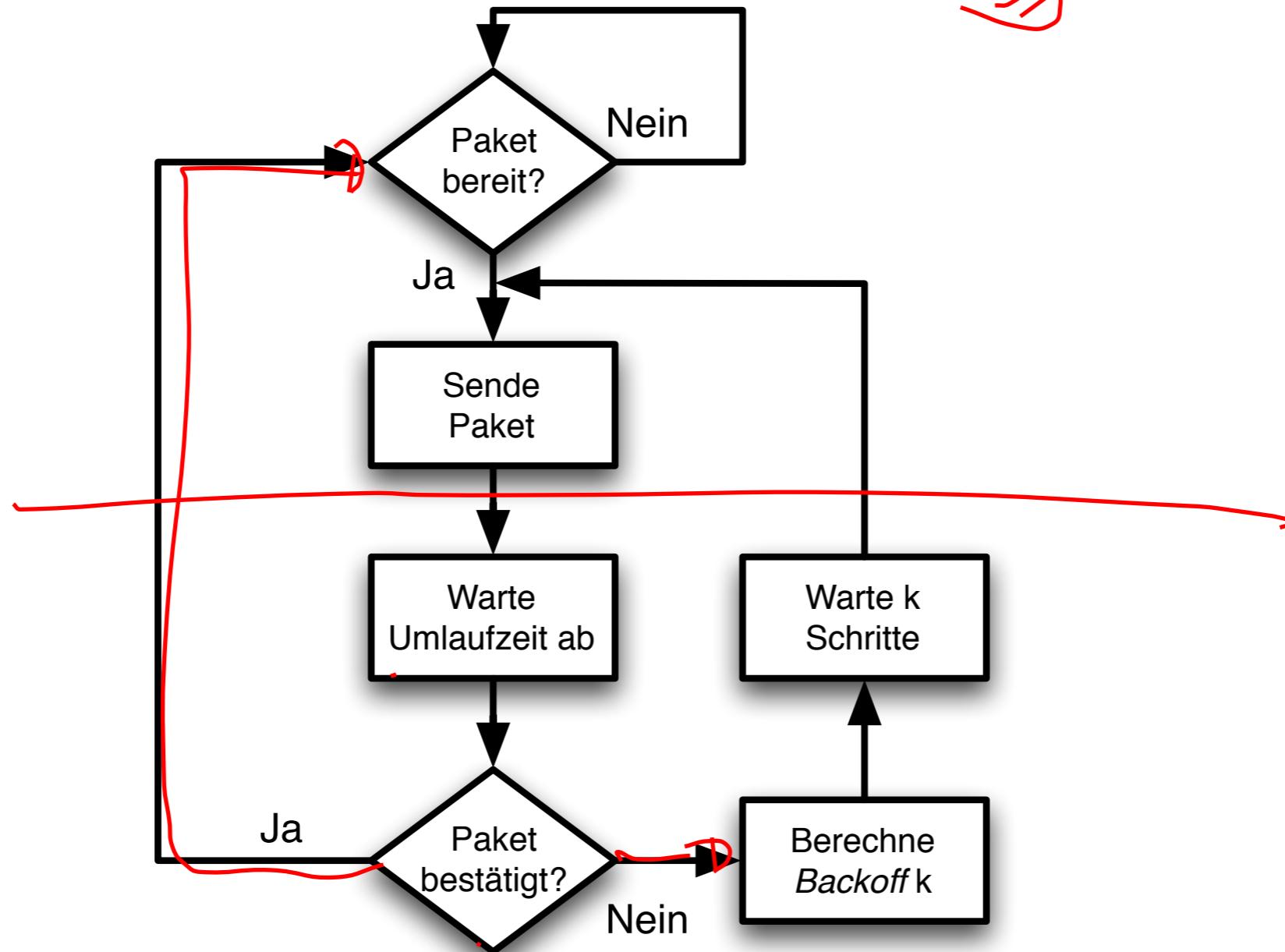
- Algorithmus
 - Sobald ein Paket vorhanden ist, wird es gesendet
- Ursprung
 - ~~1985~~⁷⁰ by Abramson et al., University of Hawaii
 - Ziel: Verwendung in ~~Satelliten~~-Verbindung



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

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

- 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

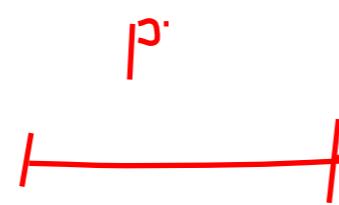
$$P[k \text{ Versuche}] = \frac{G^k}{k!} e^{-G}$$

$$\begin{aligned} E[X] &= \sum_{k=0}^{\infty} k \cdot \frac{G^k}{k!} e^{-G} \\ &= G \end{aligned}$$

- Um eine erfolgreiche Übertragung zu erhalten, darf keine Kollision mit einem anderen Paket erfolgen

- Wie lautet die Wahrscheinlichkeit für eine solche Übertragung?

n Sends



$P[k \text{ Pakete im Intervall}]$

$$= \binom{n}{k} \cdot (1-p)^{n-k} \cdot p^k$$

$$k=0 : p^0 (1-p)^n \binom{n}{0}$$

$$k=1 : p(1-p)^{n-1} \binom{n}{1}$$

$$k=2 : p^2 (1-p)^{n-2} \binom{n}{2}$$

$\frac{6}{\lambda}$

$$E[X_n] = \sum_{k=0}^n k \cdot P[X=k] = E[\underbrace{X_1 + X_1 + \dots + X_1}_{n \text{ Paketen}}] = E[n \cdot X_1] = n \cdot E[X_1] = p \cdot n$$

$$p = \frac{\lambda}{n}$$

$$\lim_{n \rightarrow \infty} \binom{n}{k} \cdot \left(\frac{1}{n}\right)^k \cdot \left(1 - \frac{\lambda}{n}\right)^{n-k}$$

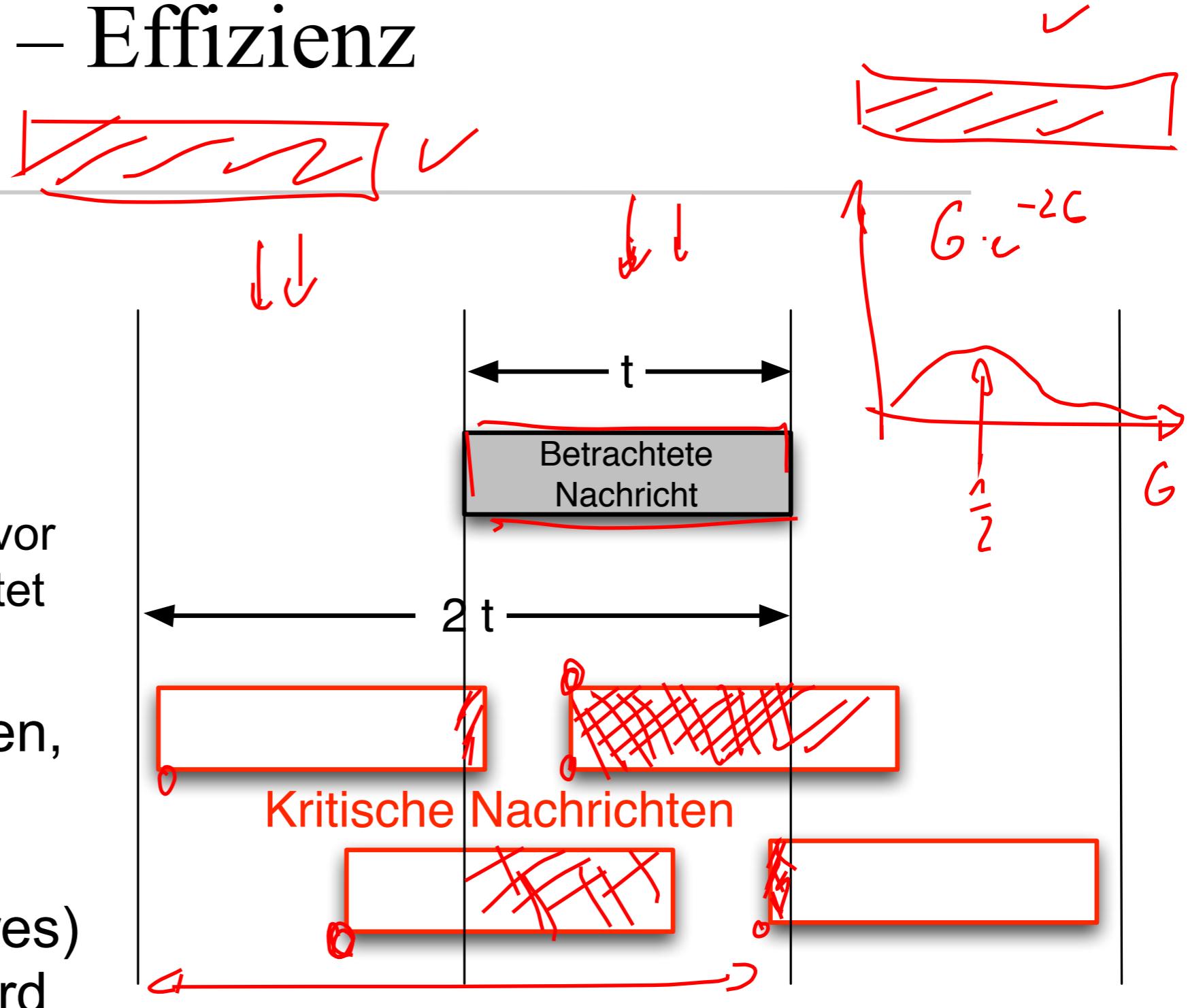
$$= \frac{\lambda^k}{k!} \frac{n(n-1)(n-2)\dots(n-k+1)}{n \cdot n \cdot n \dots n} \left(1 - \frac{\lambda}{n}\right)^{\frac{n-k}{\lambda}} \cdot \lambda^{\frac{k}{\lambda}}$$

$$= \frac{1}{k!} \cdot \lambda^k \cdot e^{-\lambda} \stackrel{!}{=} \text{Poisson}$$

$$\left(1 - \frac{\lambda}{n}\right)^{\frac{n}{\lambda}} = \left(1 - \frac{1}{x}\right)^x \rightarrow \frac{1}{e}$$

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:
 - $S(G) = \underline{G e^{-2G}}$
 - Optimal für $G=1/2$, $S=\underline{1/e}$



$$P[X=0]$$

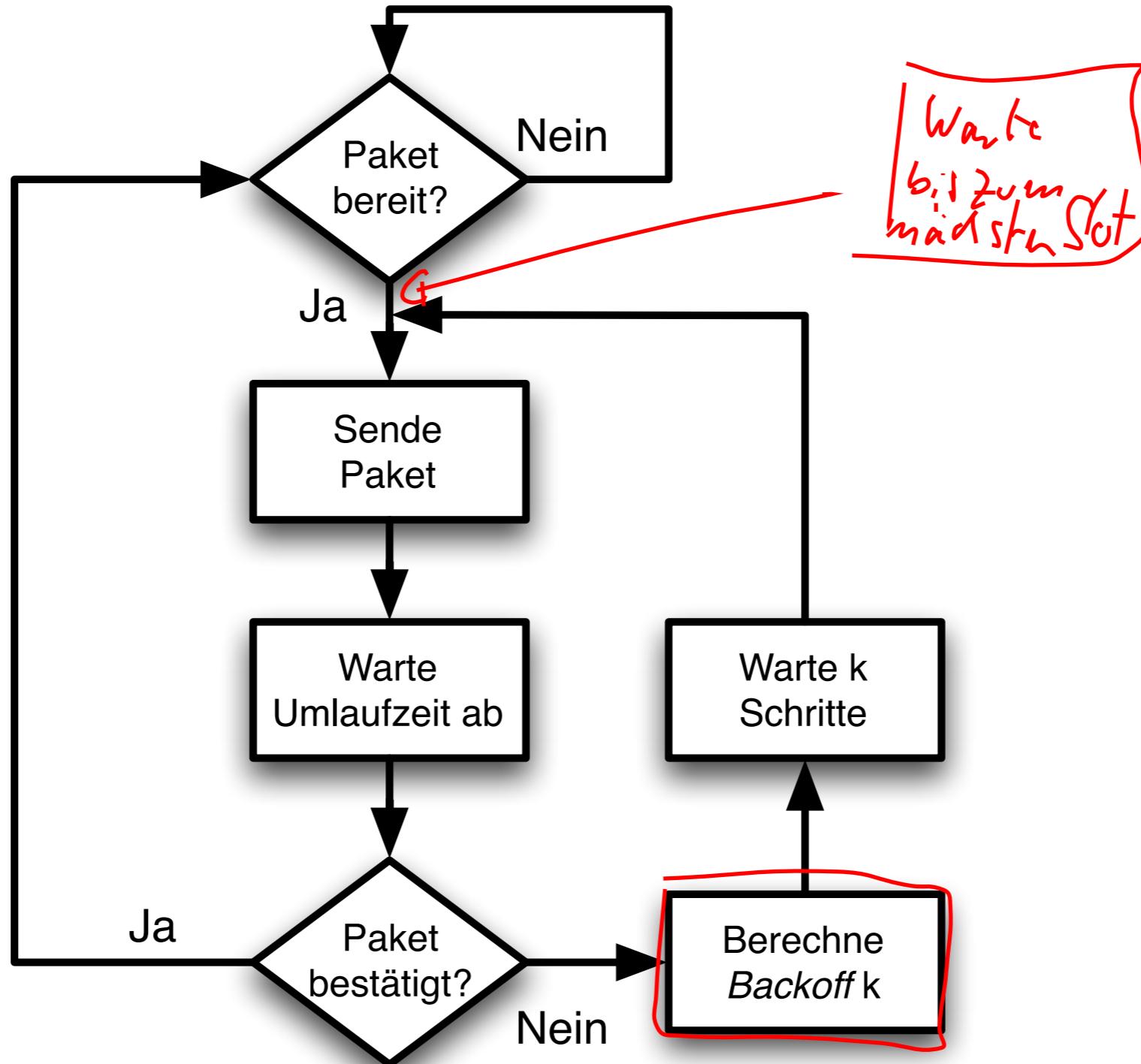
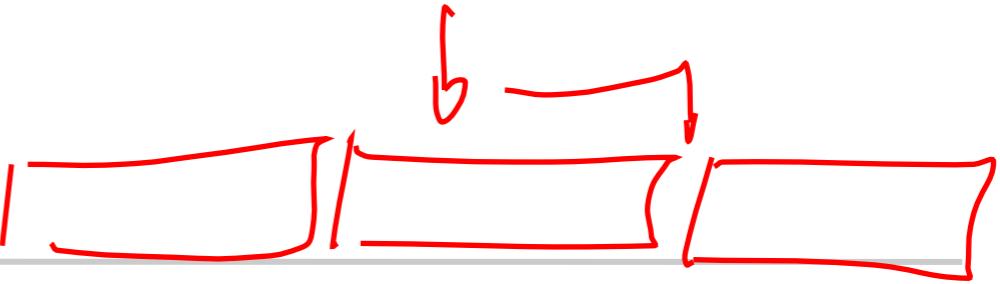
$$\frac{G^0}{0!} \cdot e^{-G}$$

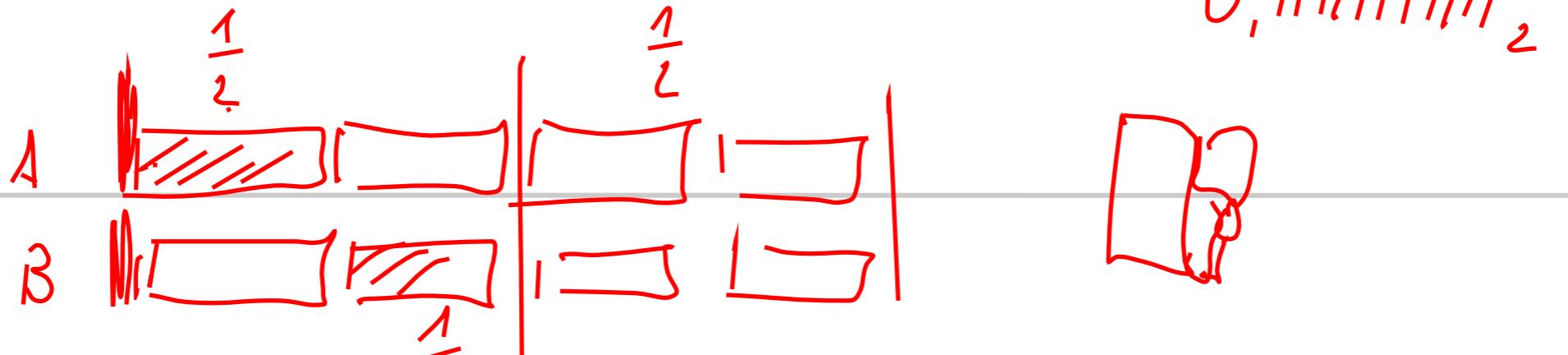
$$P[X=1]$$

$$\cdot \frac{G^1}{1!} \cdot e^{-G}$$

$$= \underline{G \cdot e^{-LG}}$$

Slotted ALOHA





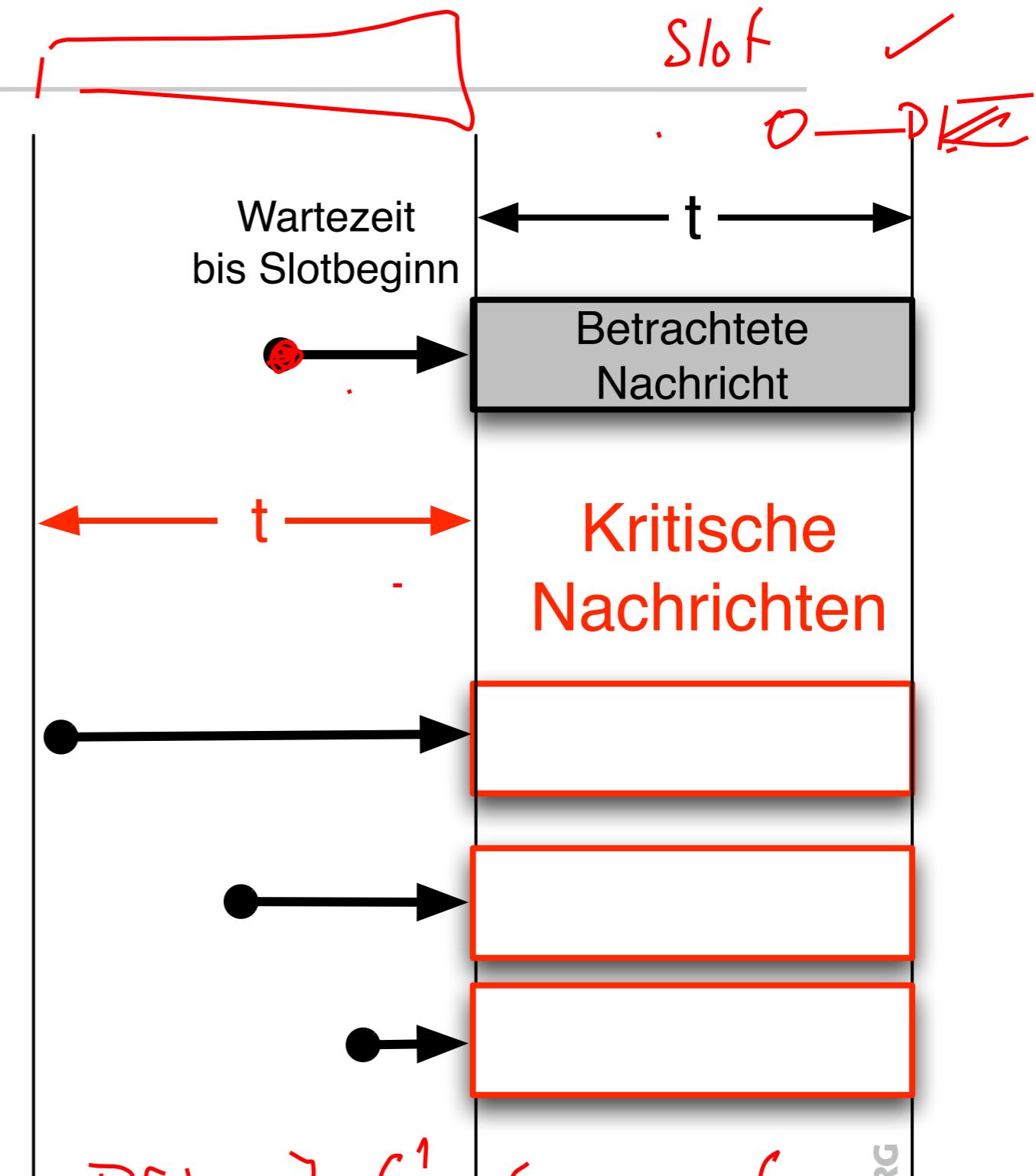
$$\left(\frac{1}{2} \cdot \frac{1}{2}\right) \cdot 2 = \frac{1}{2}$$

$$1 \cdot \frac{1}{2} + 2 \cdot \frac{1}{4} + 3 \cdot \frac{1}{8} + \dots = 2$$

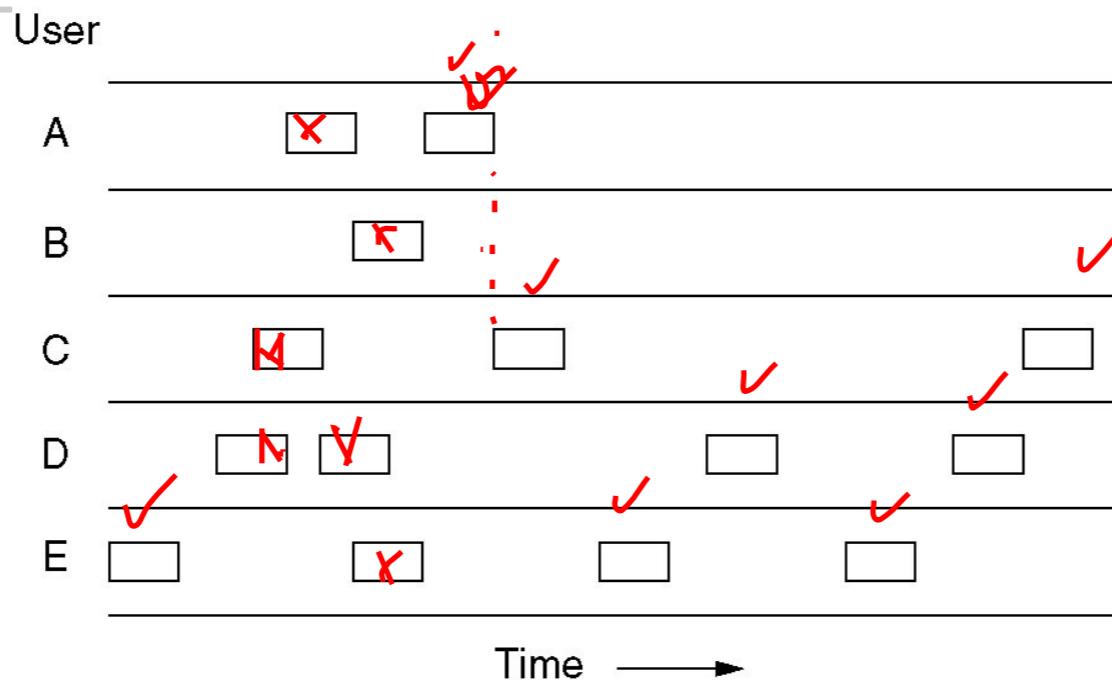
$$\begin{aligned}
 &= \frac{1}{2} + \frac{1}{4} - \frac{1}{4} + \frac{1}{8} - \frac{1}{8} + \frac{1}{16} - \dots = \frac{1}{4} \\
 &\quad \left. \begin{array}{l} \\ \\ \\ \\ \end{array} \right\} 2 \\
 &\quad \left. \begin{array}{l} \\ \\ \\ \\ \end{array} \right\} 1
 \end{aligned}$$

Slotted ALOHA – Effizienz

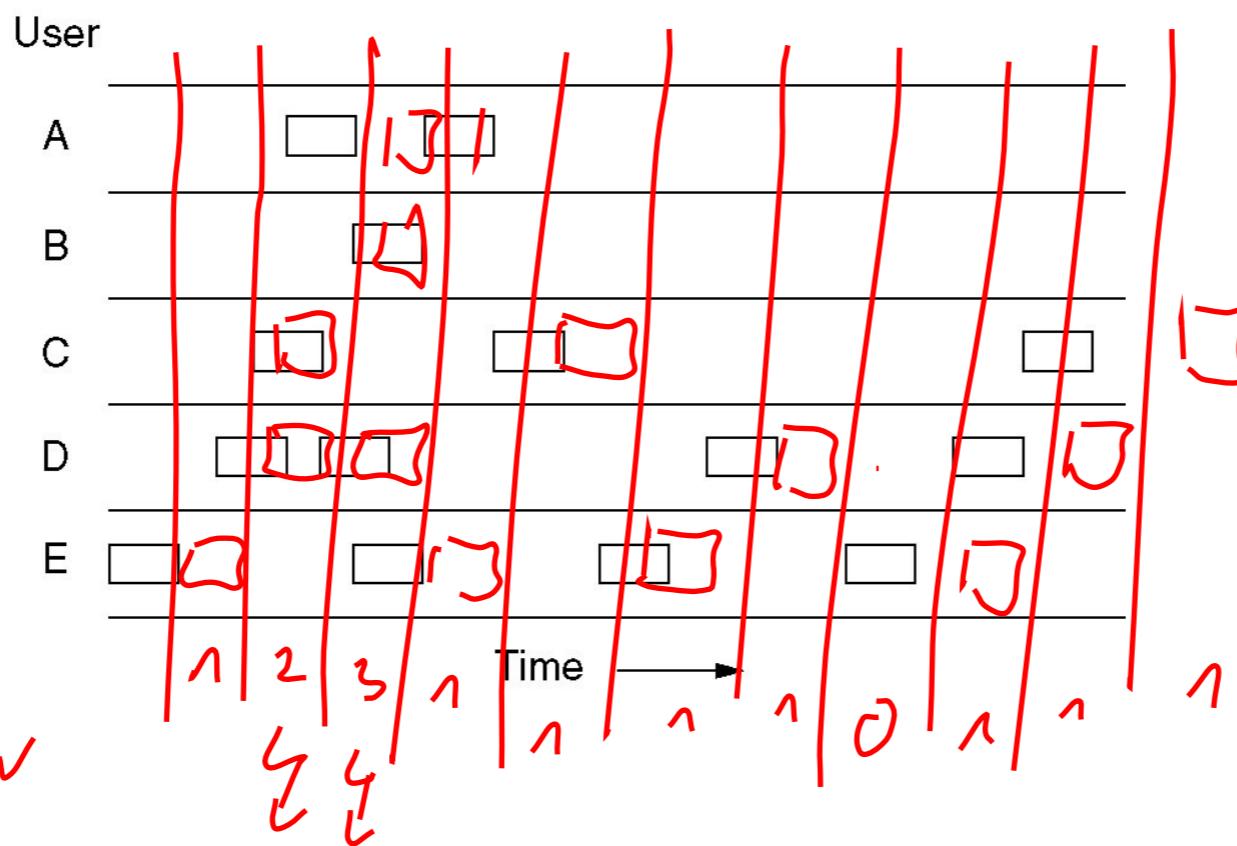
- Ein Paket X wird gestört, wenn
 - ein Paket kurz vor X startet
 - wenn ein Paket kurz vor dem Ende von X startet
- Das Paket wird erfolgreich übertragen, wenn in einem Zeitraum von zwei Paketen kein (anderes) Paket übertragen wird



$$P[X=1] = \frac{G^1}{1!} e^{-G} = G \cdot e^{-G}$$



Aloha



Slotted

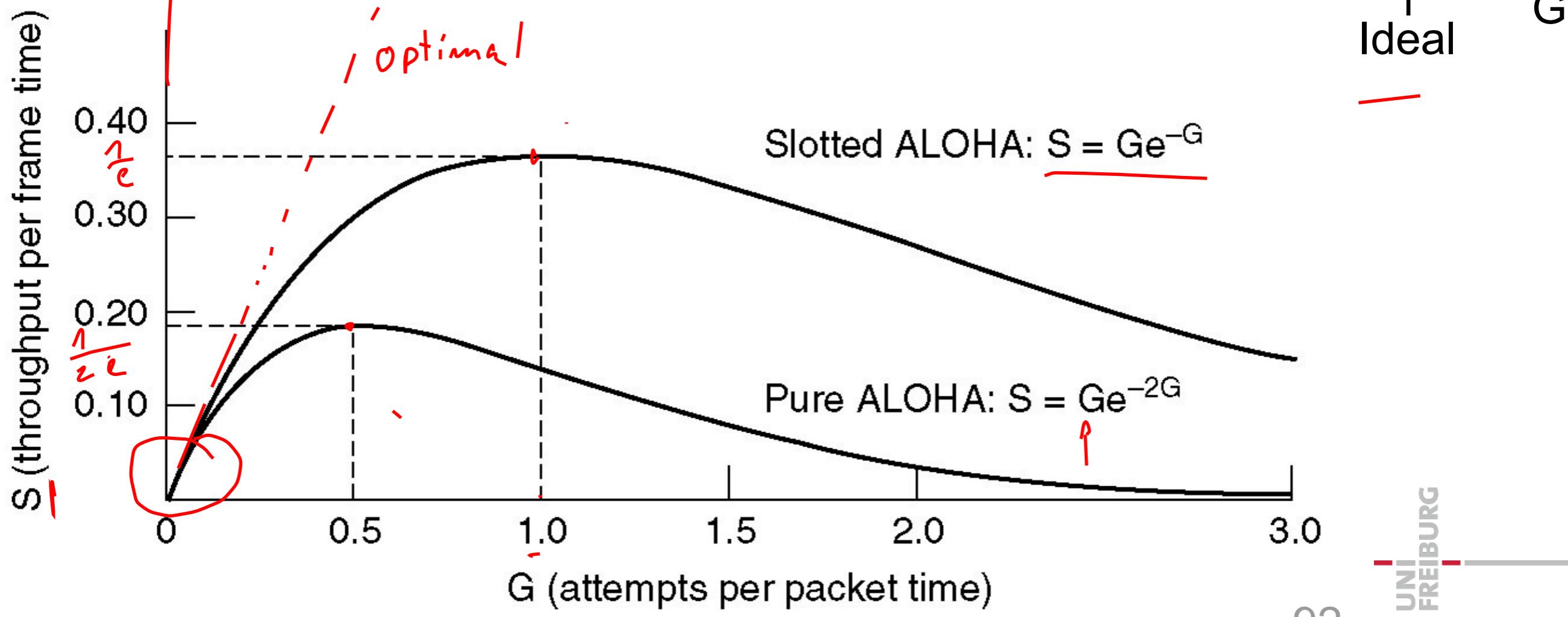
Aloha

Slotted ALOHA

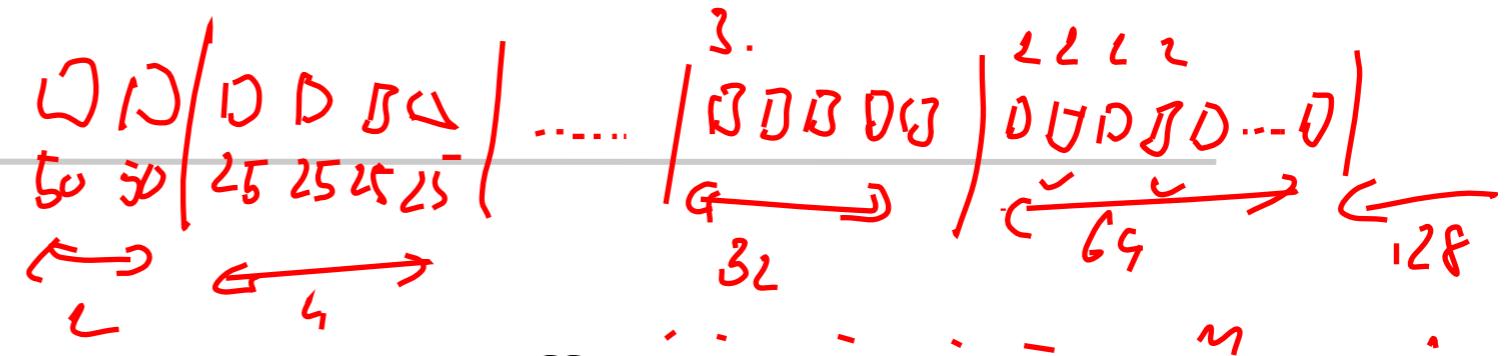
- ALOHAs Problem:
 - Lange Verwundbarkeit eines Pakets
- Reduktion durch Verwendung von Zeitscheiben (Slots)
 - Synchronisation wird vorausgesetzt
- Ergebnis:
 - Verwundbarkeit wird halbiert
 - Durchsatz:
 - $S(G) = G e^{-G}$
 - Optimal für $G=1$, $S=1/e$

Durchsatz in Abhängigkeit der Last

- Für (slotted) ALOHA ist eine geschlossene Darstellung in Abhängigkeit von G möglich
- Kein gutes Protokoll
 - Durchsatz bricht zusammen, wenn die Last zunimmt



Bestimmung der Warte-Zeit



- Nach der Kollision:

- Algorithmus binary exponential backoff

- $k := 2$

- Solange Kollision beim letzten Senden

- Wähle t gleichwahrscheinlich zufällig aus $\{0, \dots, k-1\}$
- Warte t Zeit-Slots
- Sende Nachricht (Abbruch bei Collision Detection)
- $k := 2^k$

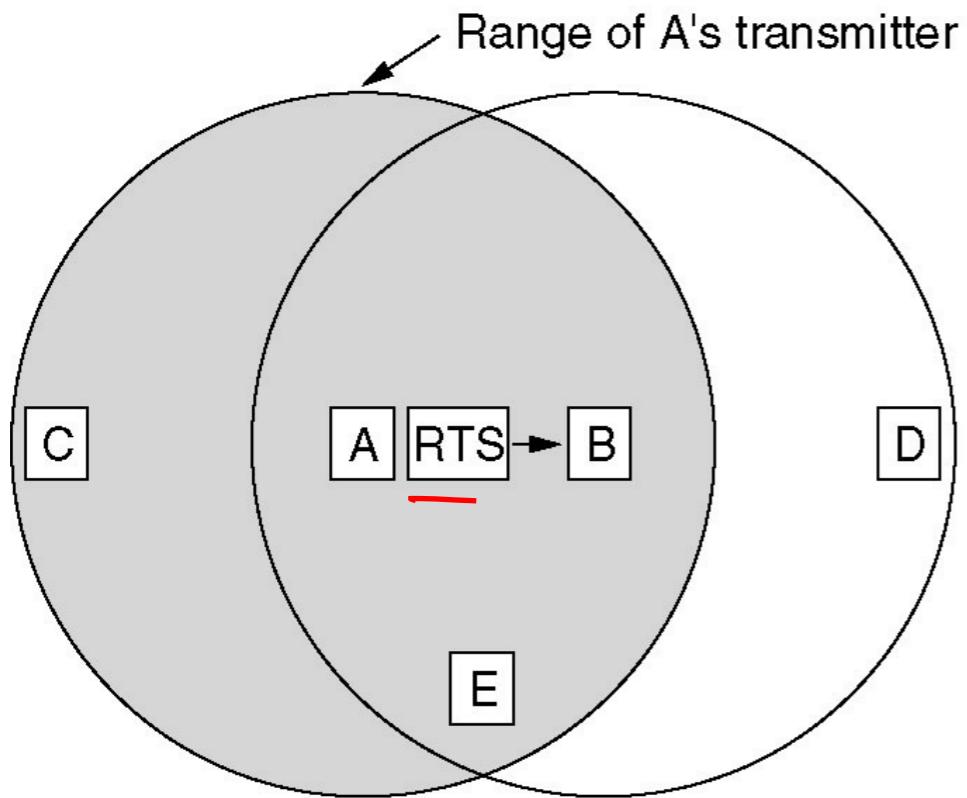
4 S�fs

- Algorithmus

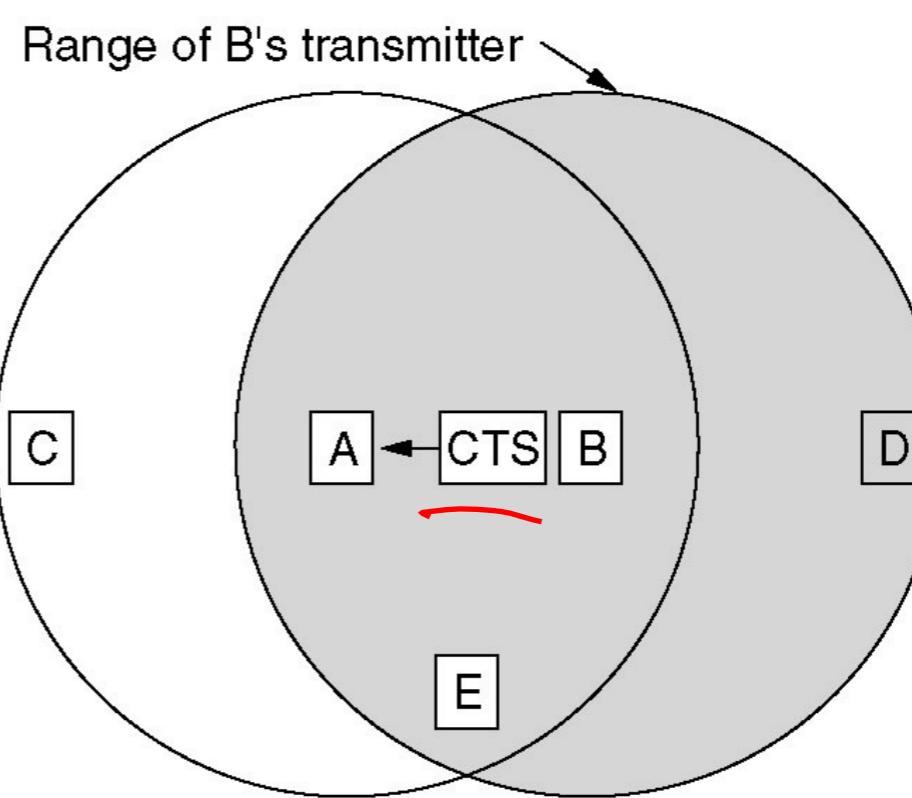
- passt Wartezeit dynamisch an die Anzahl beteiligter Stationen an
- sorgt für gleichmäßige Auslastung des Kanals
- ist fair (auf lange Sicht)

Multiple Access with Collision Avoidance

MACA



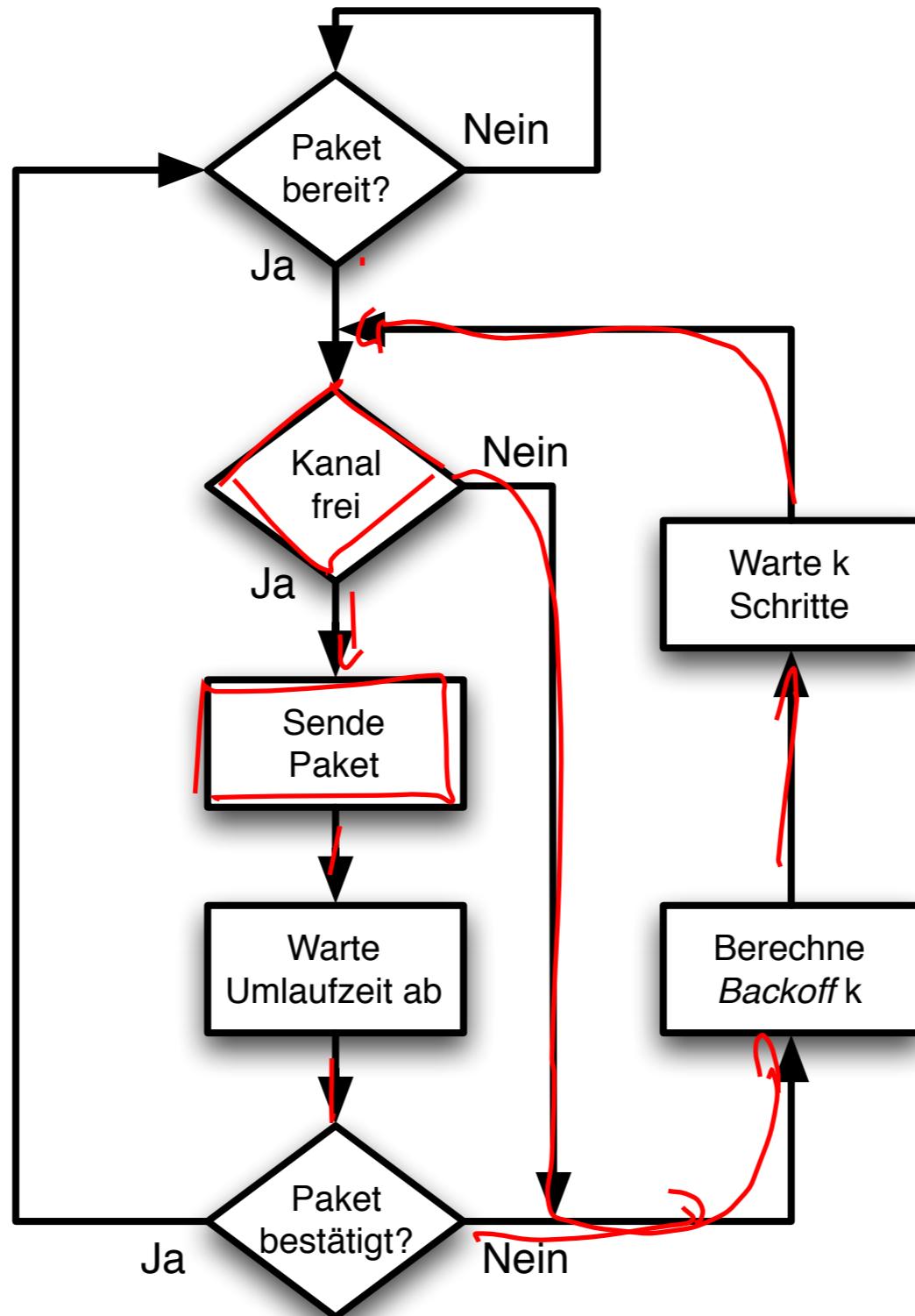
(a)



(b)

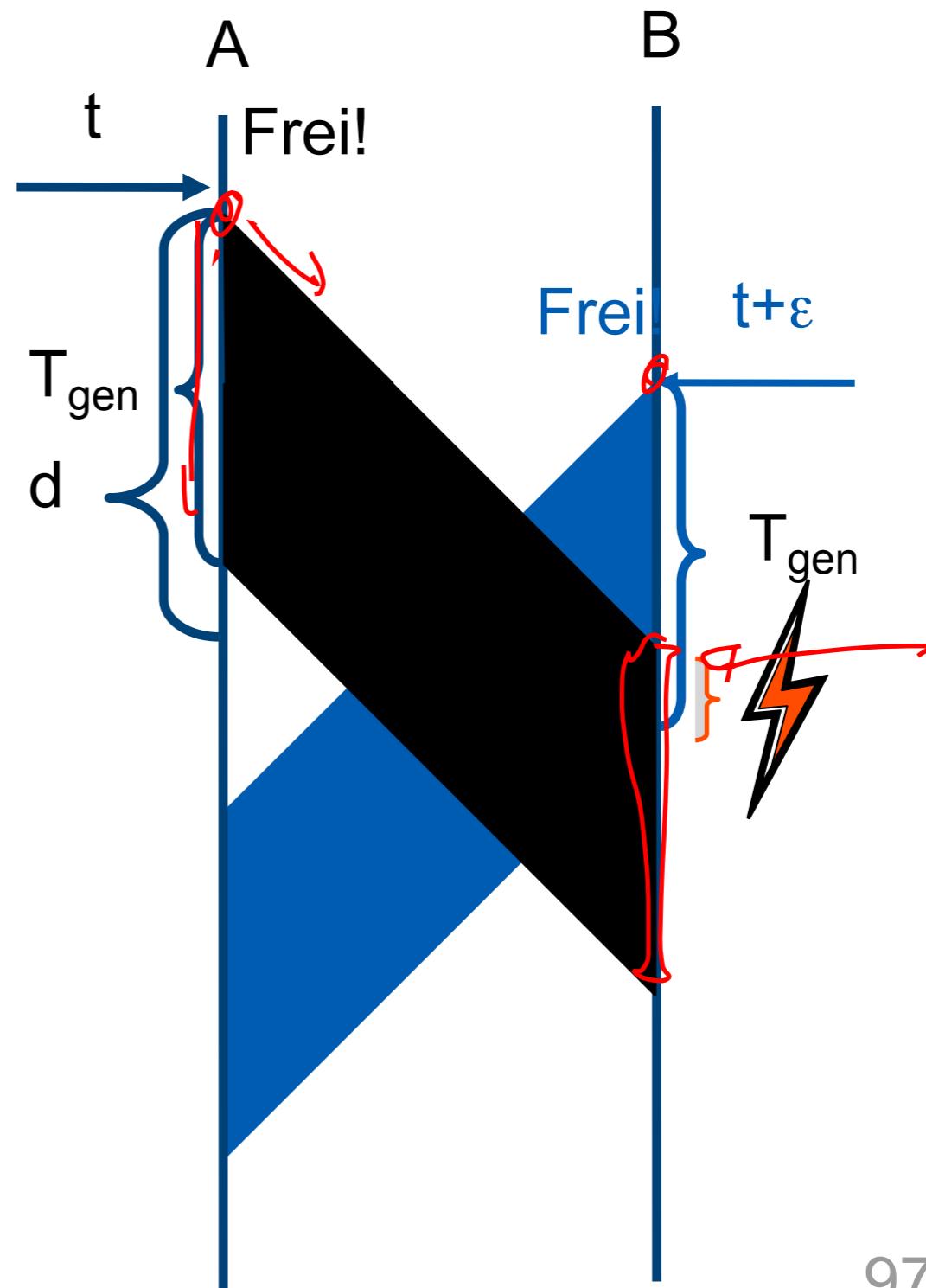
- (a) A sendet Request to Send (RTS) an B.
- (b) B antwortet mit Clear to Send (CTS) an A.

CSMA und Übertragungszeit



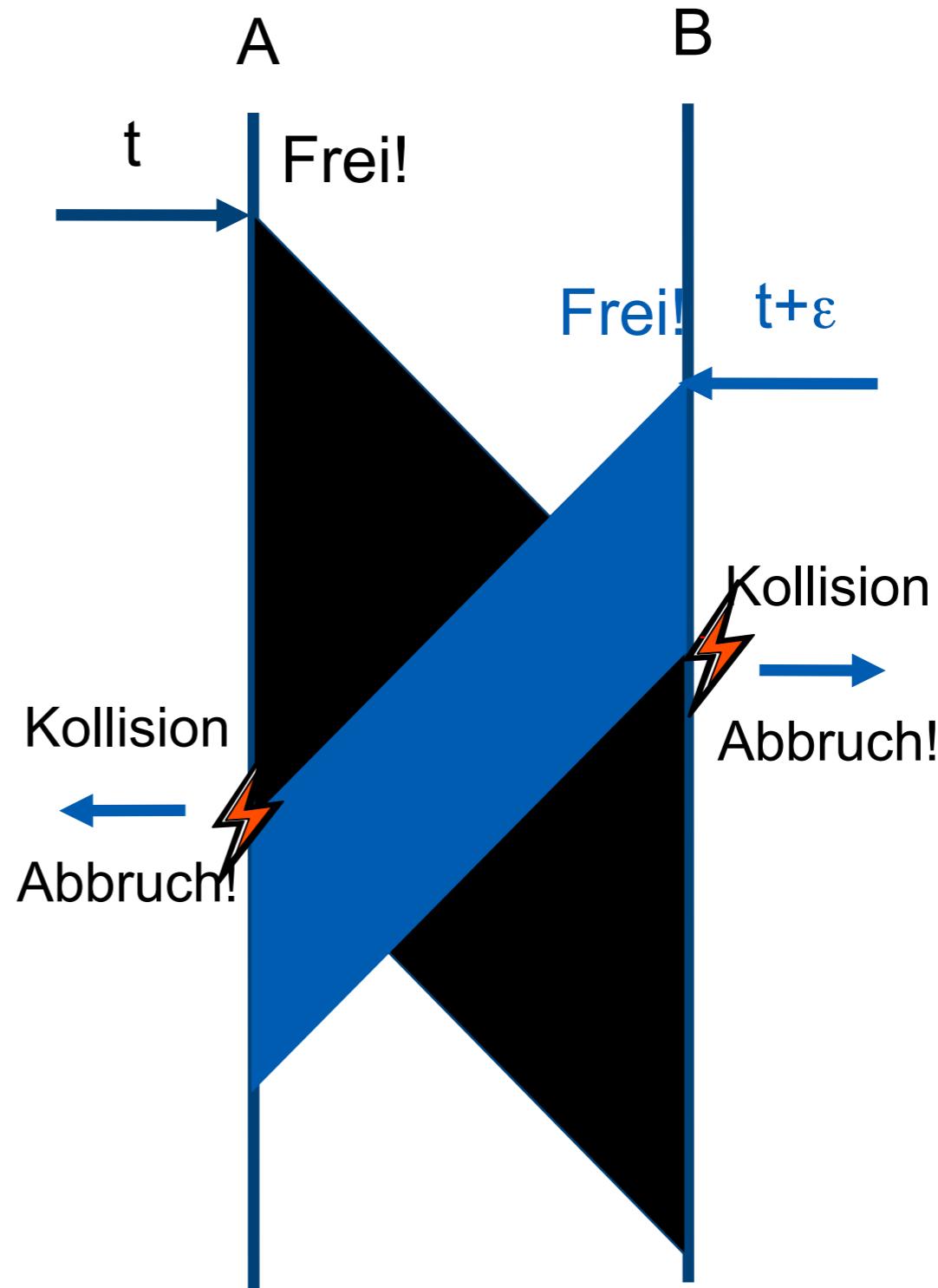
CSMA und Übertragungszeit

- Carrier Sense Multiple Access:
 - Erst senden wenn der Kanal frei ist
- CSMA-Problem:
 - Übertragungszeit d (propagation delay)
- Zwei Stationen
 - starten Senden zu den Zeitpunkten t und $t+\varepsilon$ mit $\varepsilon < d$
 - sehen jeweils einen freien Kanal
- Zweite Station
 - verursacht dann eine Kollision



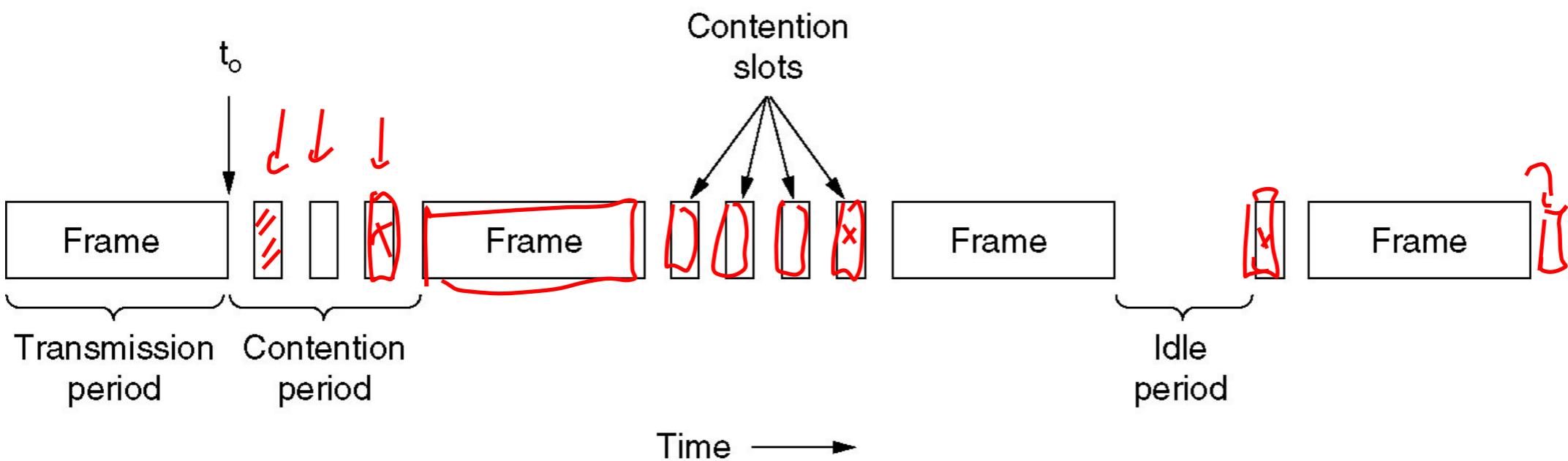
Kollisionserkennung – CSMA/CD

- Falls Kollisionserkennung (collision detection) möglich ist,
 - dann beendet der spätere Sender seine Übertragung
 - Zeitverschwendungen wird reduziert, da mindestens eine Nachricht (die erste) übertragen wird
- Fähigkeit der Kollisionserkennung hängt von der Bitübertragungsschicht ab
- CSMA/CD – Carrier Sense Multiple Access/Collision Detection
- Collision Detection
 - setzt gleichzeitiges Abhören des Kanals nach Kollisionen voraus
 - Ist das was auf dem Kanal geschieht, identisch zu der eigenen Nachricht?



Phasen in CSMA/CD

- Leer-Phase (IDLE)
 - Keine Station sendet einen Frame
 - Wettbewerbsphase (Contention Period)
 - Kollisionen entstehen, Übertragungen werden abgebrochen
 - Übertragungsphase (Transmission Period)
 - Keine Kollision, effektiver Teil des Protokolls
- ! Es gibt nur Wettbewerbs-, Übertragungsphasen und Leer-Phasen

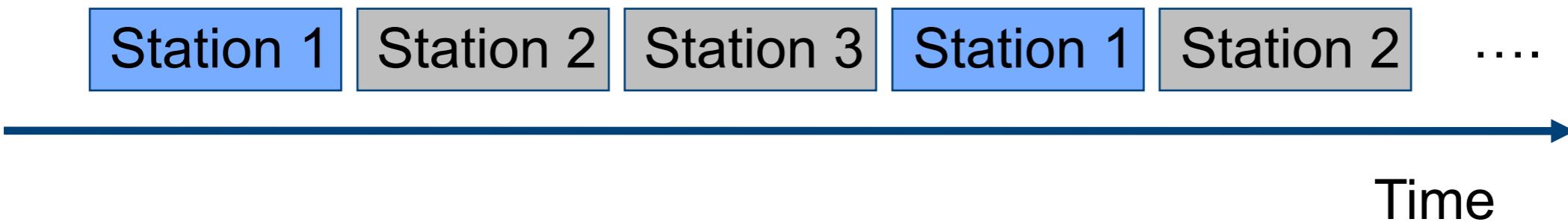


Der Mediumzugriff in der Sicherungsschicht

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

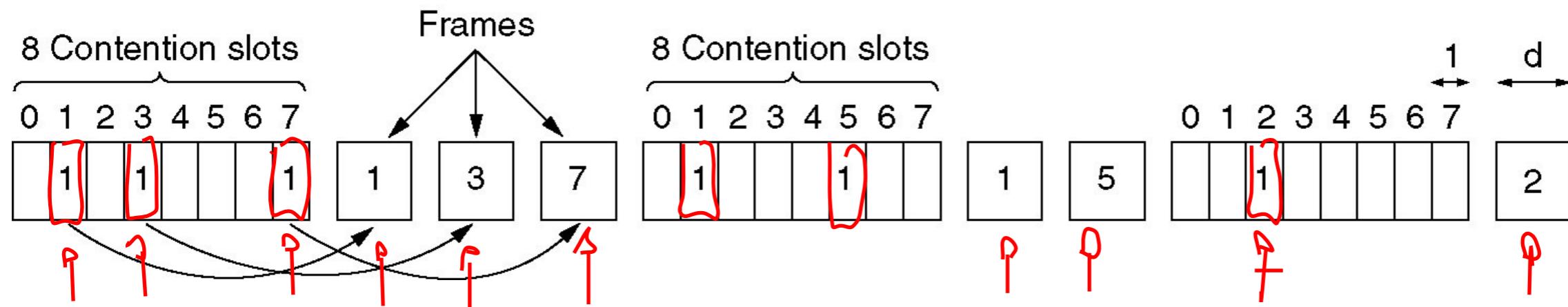
Wettbewerbsfreie Protokolle

- Einfaches Beispiel: Statisches Zeit-Multiplexen (TDMA)
 - Jeder Station wird ein fester Zeit-Slot in einem sich wiederholenden Zeitschema zugewiesen
- Nachteile bekannt und diskutiert
- Gibt es dynamische kollisionsfreie Protokoll?



Bit-map Protokoll

- Probleme von TDMA
 - Wenn eine Station nichts zu senden hat, dann wird der Kanal nicht genutzt
- Reservierungssystem: Bit-map protocol
 - Kurze statische Reservierung-Slots zur Ankündigung
 - Müssen von jeder Station empfangen werden



Bitmap-Protokolle

- Verhalten bei geringer Last
 - Falls keine Pakete verschickt werden, wird der (leere) Wettbewerbs-Slot wiederholt
 - Eine Station muss auf seinen Wettbewerbs-Slot warten
 - Erzeugt gewisse Verzögerung (delay)
- Verhalten bei hoher Last
 - Datenpakete dominieren die Kanalbelegung
 - Datenpakete sind länger als die Contention-Slots
 - Overhead ist vernachlässigbar
 - Guter und stabiler Durchsatz
- Bitmap ist ein Carrier-Sense Protokoll!

Der Mediumzugriff in der Sicherungsschicht

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

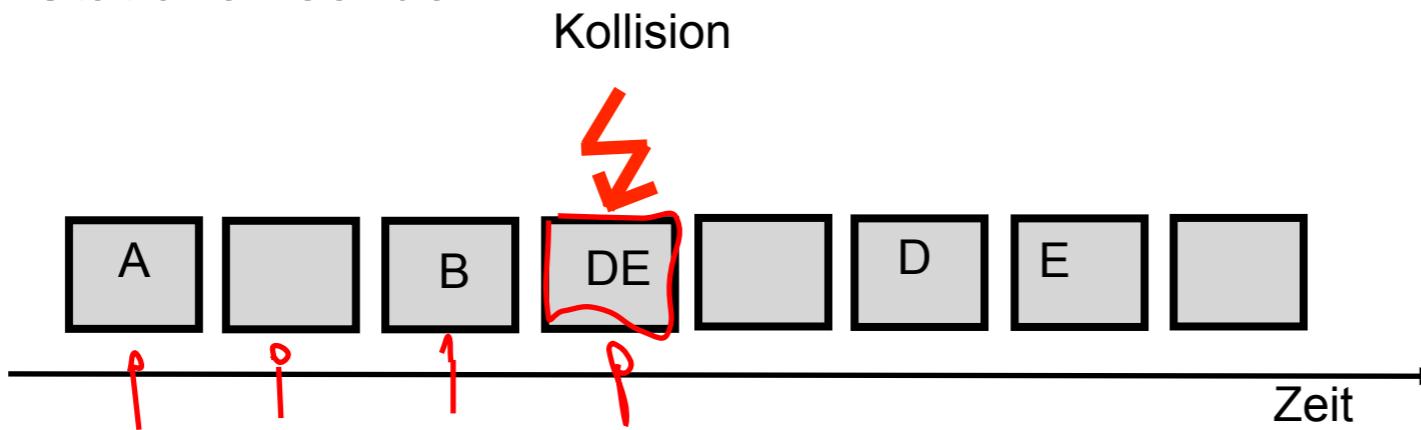
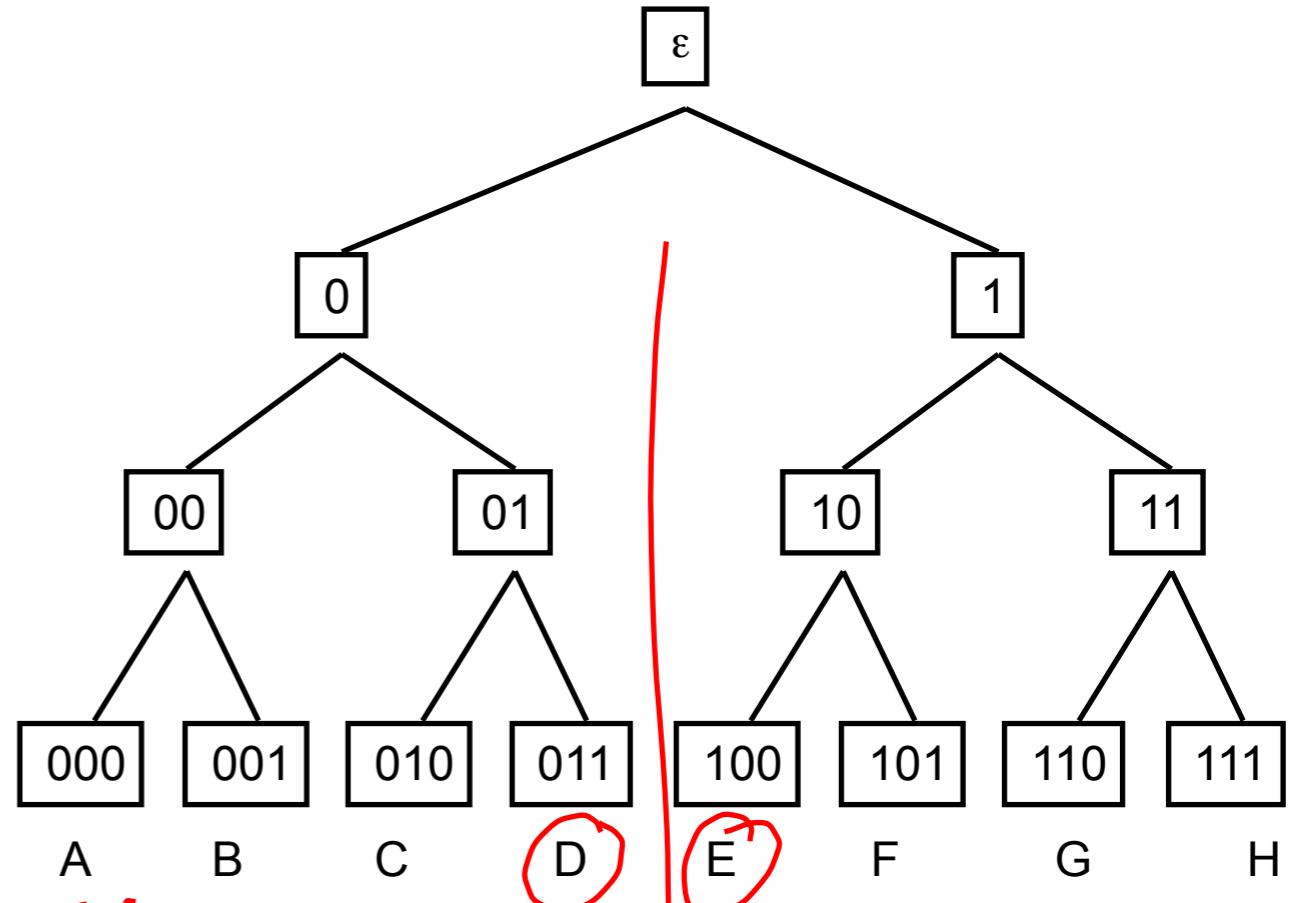
Protokolle mit beschränktem Wettbewerb

- Ziel
 - geringe Verzögerung bei kleiner Last
 - wie Kollisionsprotokolle
 - hoher Durchsatz bei großer Last
 - wie kollisionsfreie Protokolle
- Idee
 - Anpassung des Wettbewerb-Slots (contention slot) an die Anzahl der teilnehmenden Stationen
 - Mehrere Stationen müssen sich dann diese Slots teilen

Adaptives Baumprotokoll

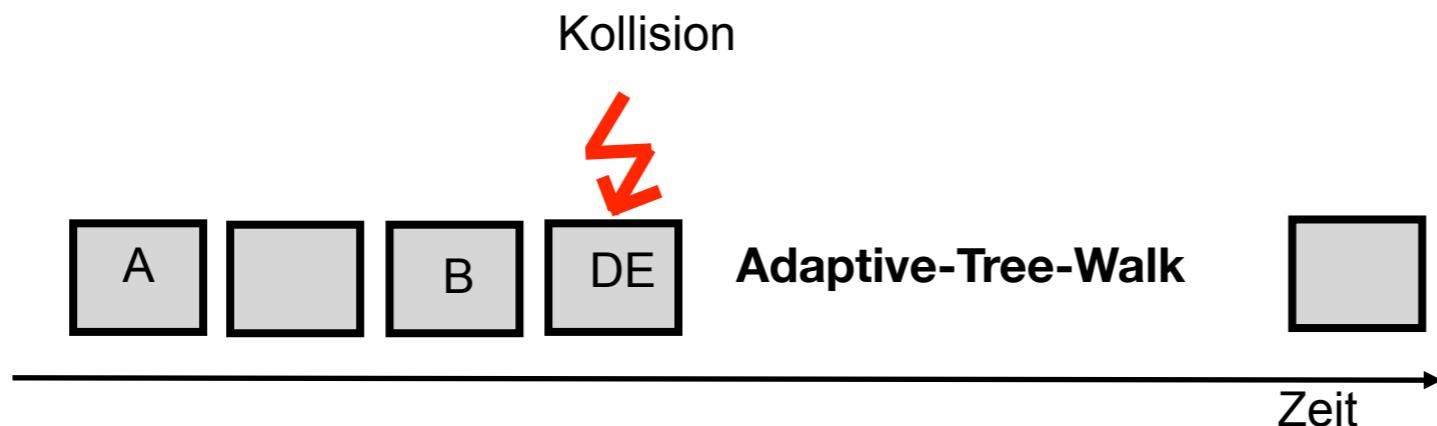
Voraussetzung

- 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

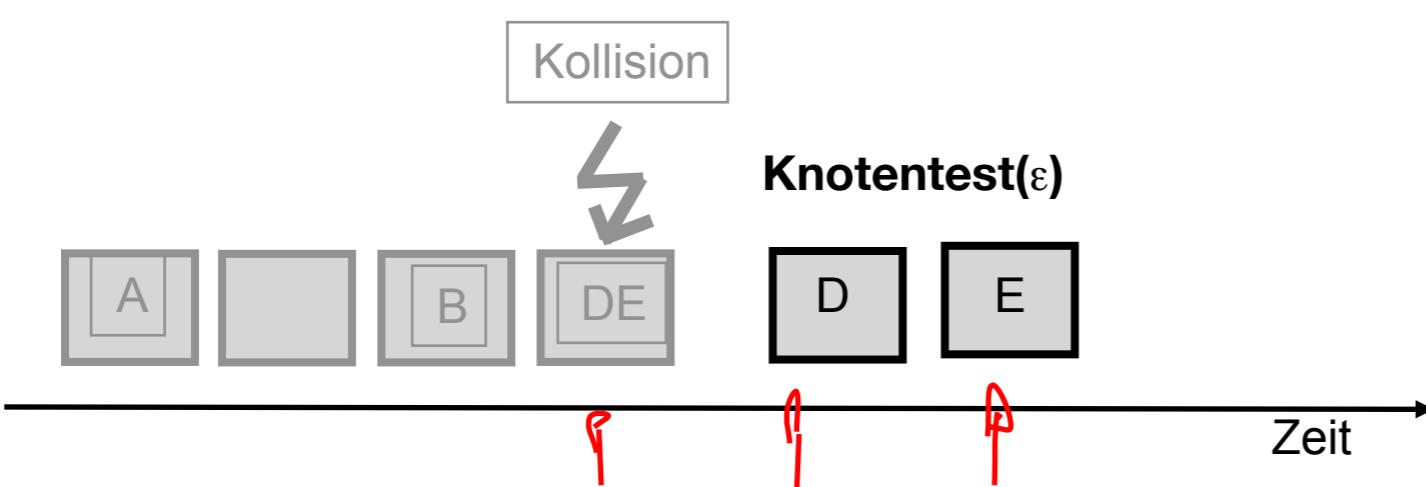
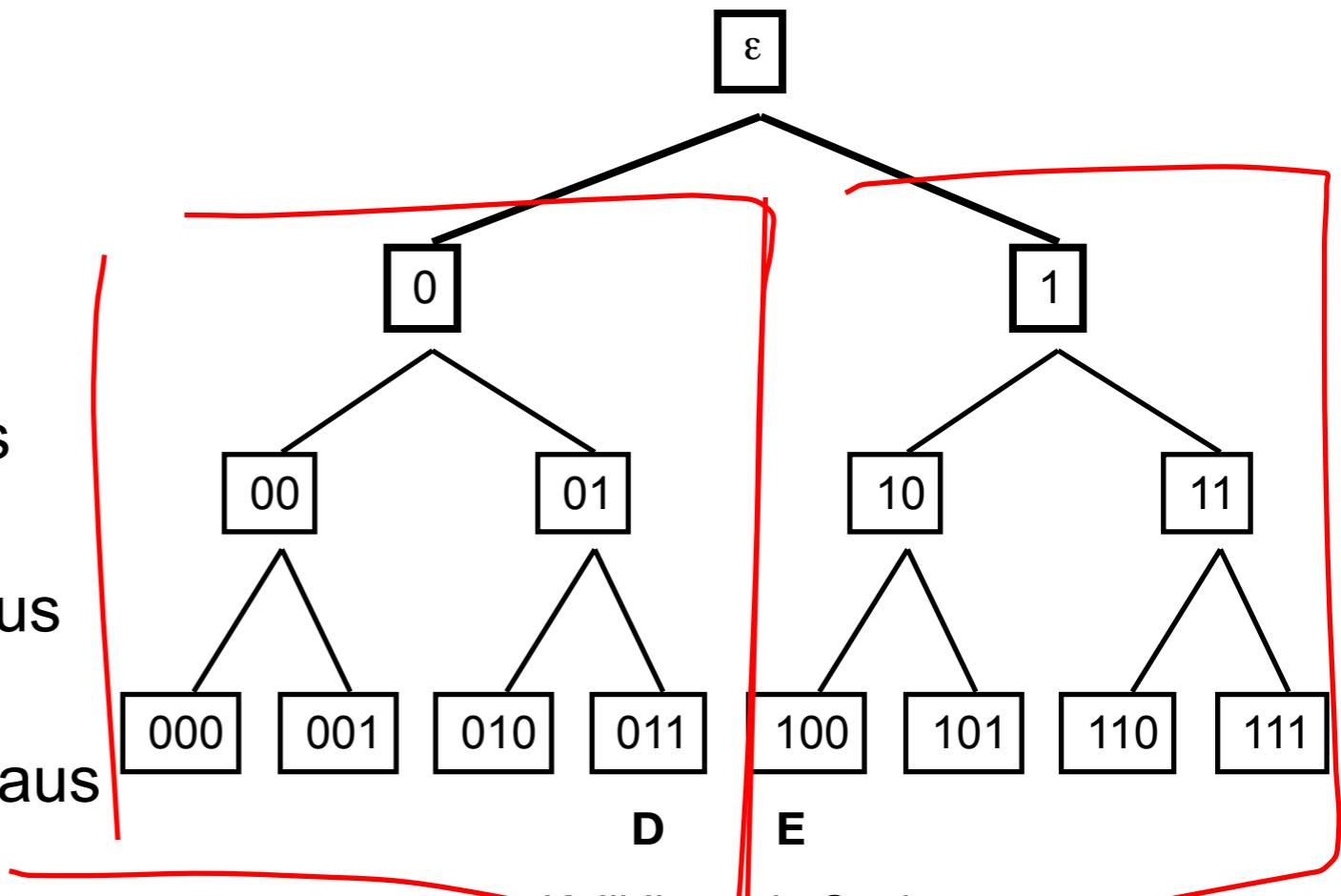
Knoten-Test

Algorithmus Knoten-Test

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

Knoten-Test(u)

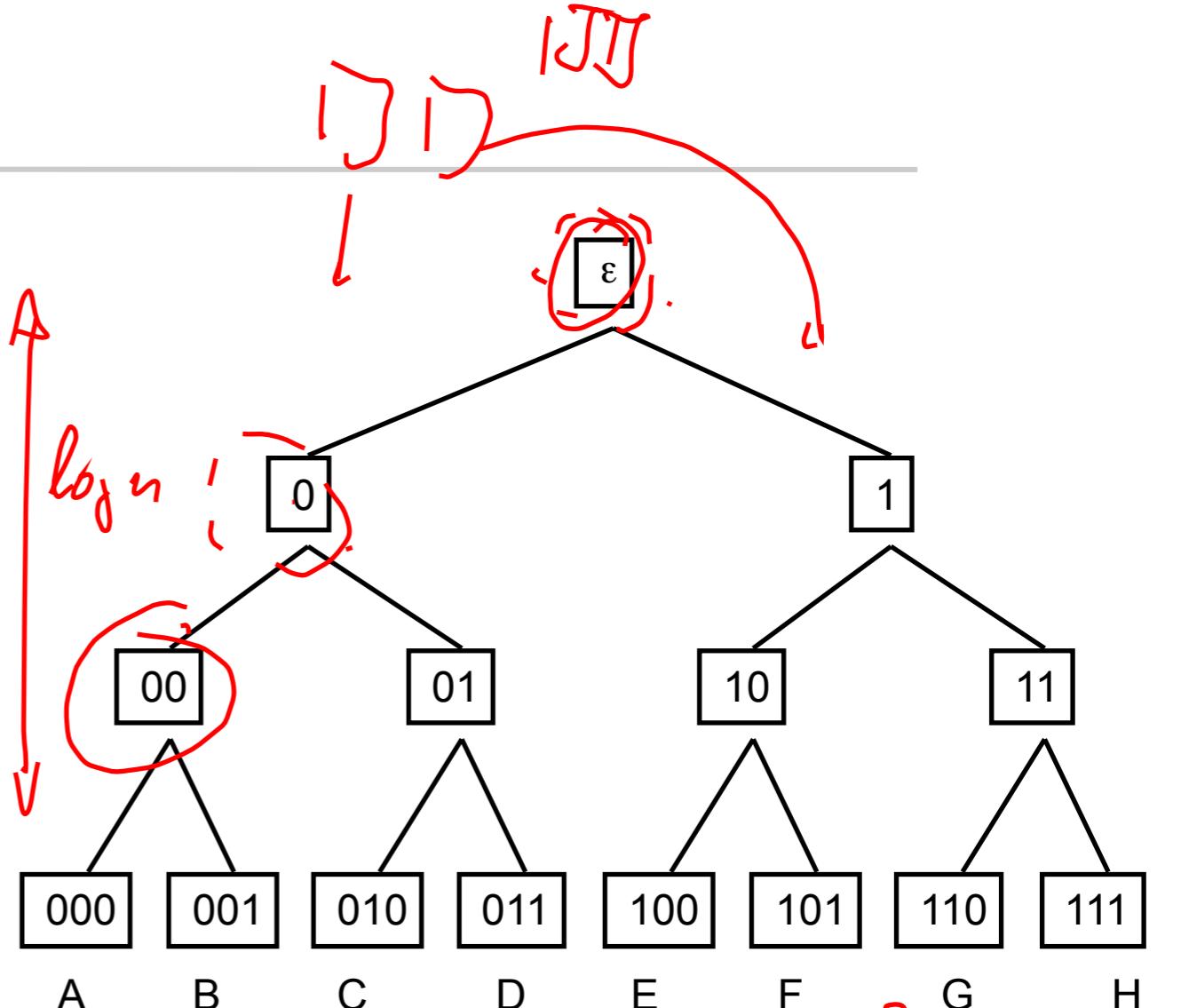
- Betrachte zwei Slots pro Knoten des Baums
- Im ersten Slot senden alle Knoten aus S , die mit ID u_0 anfangen
- Im zweiten Slot senden alle Knoten aus S , die mit ID u_1 anfangen



Adaptives Baumprotokoll

Kern-Algorithmus

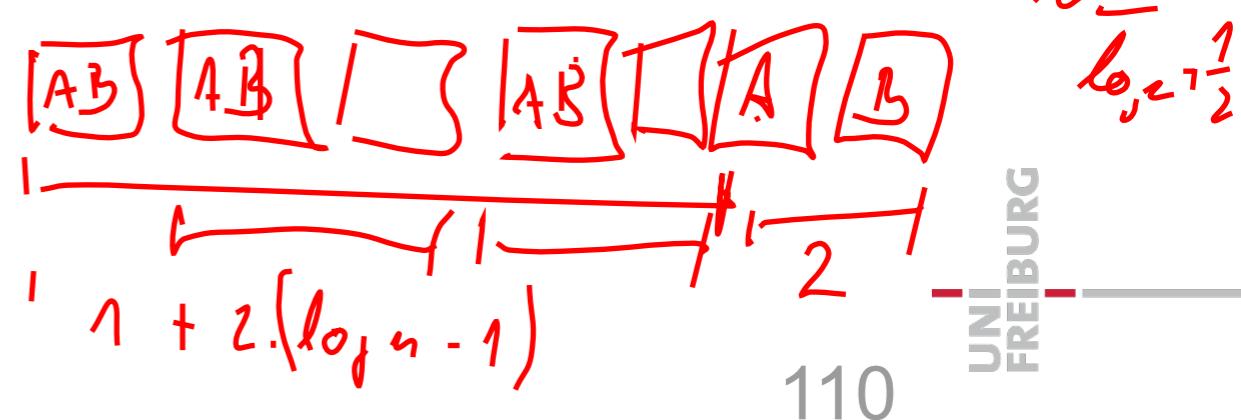
- Algorithmus Knoten-Test
 - für Knoten u des Baums und
 - kollidierende Menge S von Stationen
- Knoten-Test(u)
 - Betrachte zwei Slots pro Knoten des Baums
 - Im ersten Slot senden alle Knoten aus S , die mit ID u_0 anfangen
 - Im zweiten Slot senden alle Knoten aus S , die mit ID u_1 anfangen
- Adaptive 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



n Teilnehmer

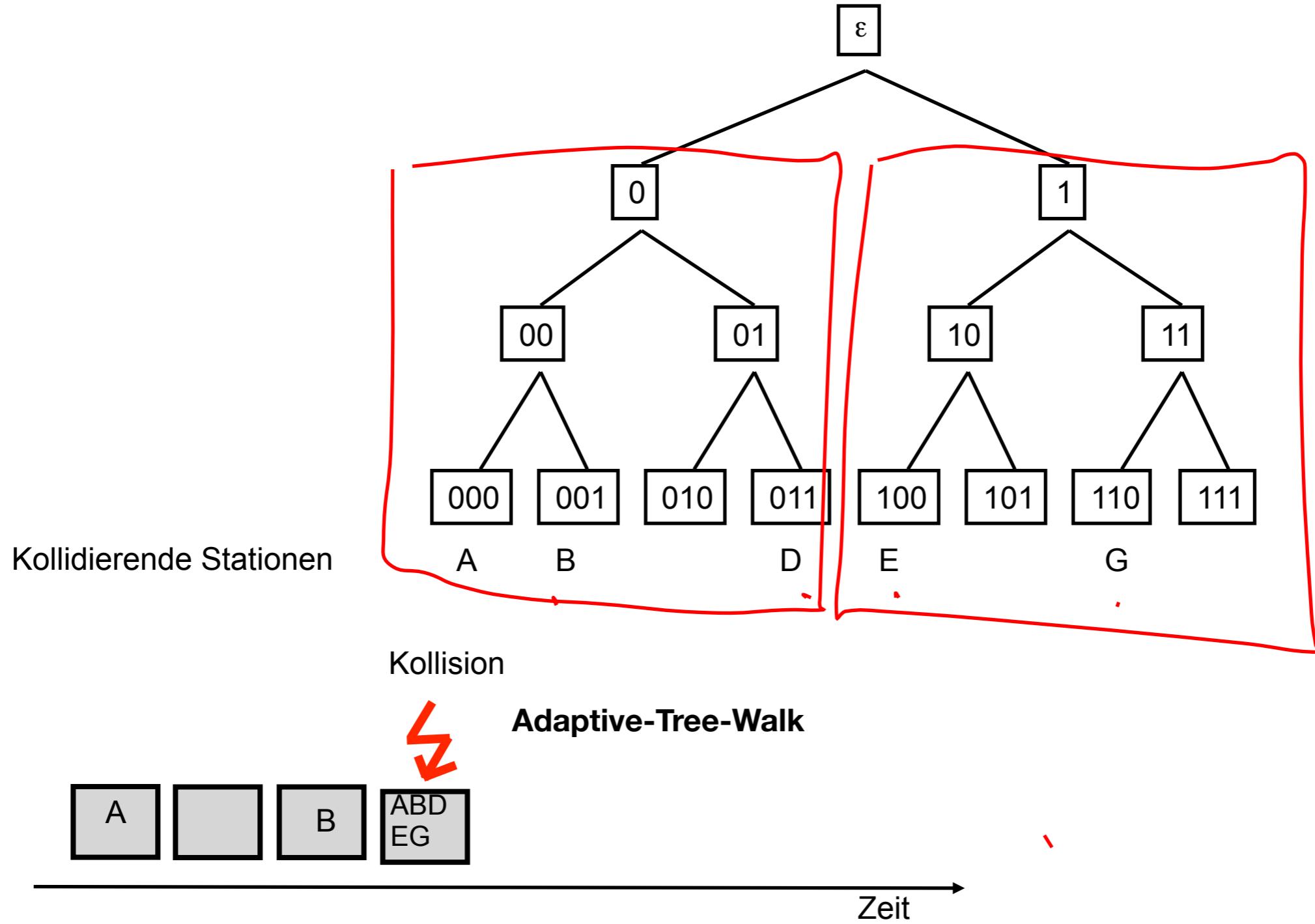
Stationen

$2 \log n + 1$

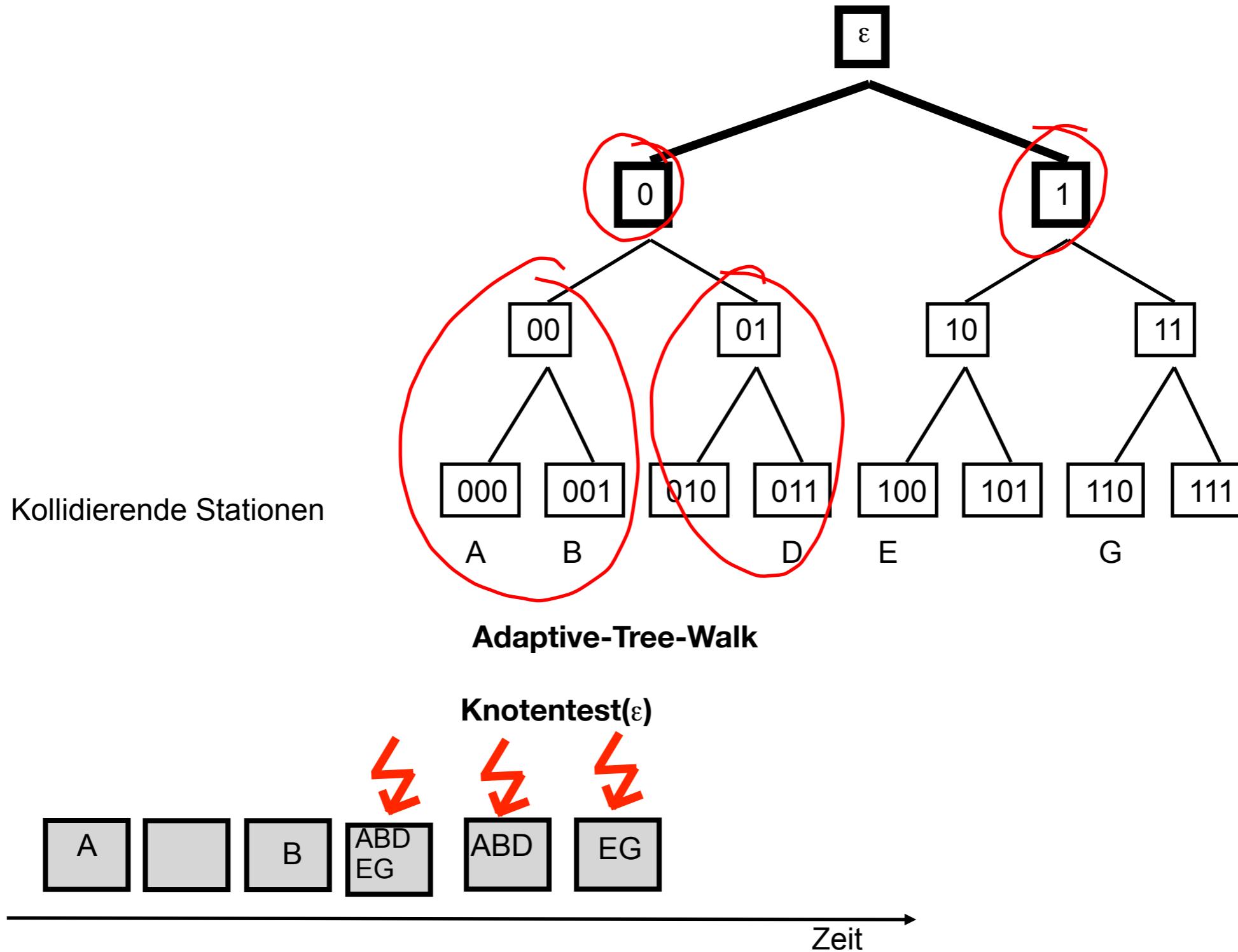


Adaptives Baumprotokoll

Beispiel (1)

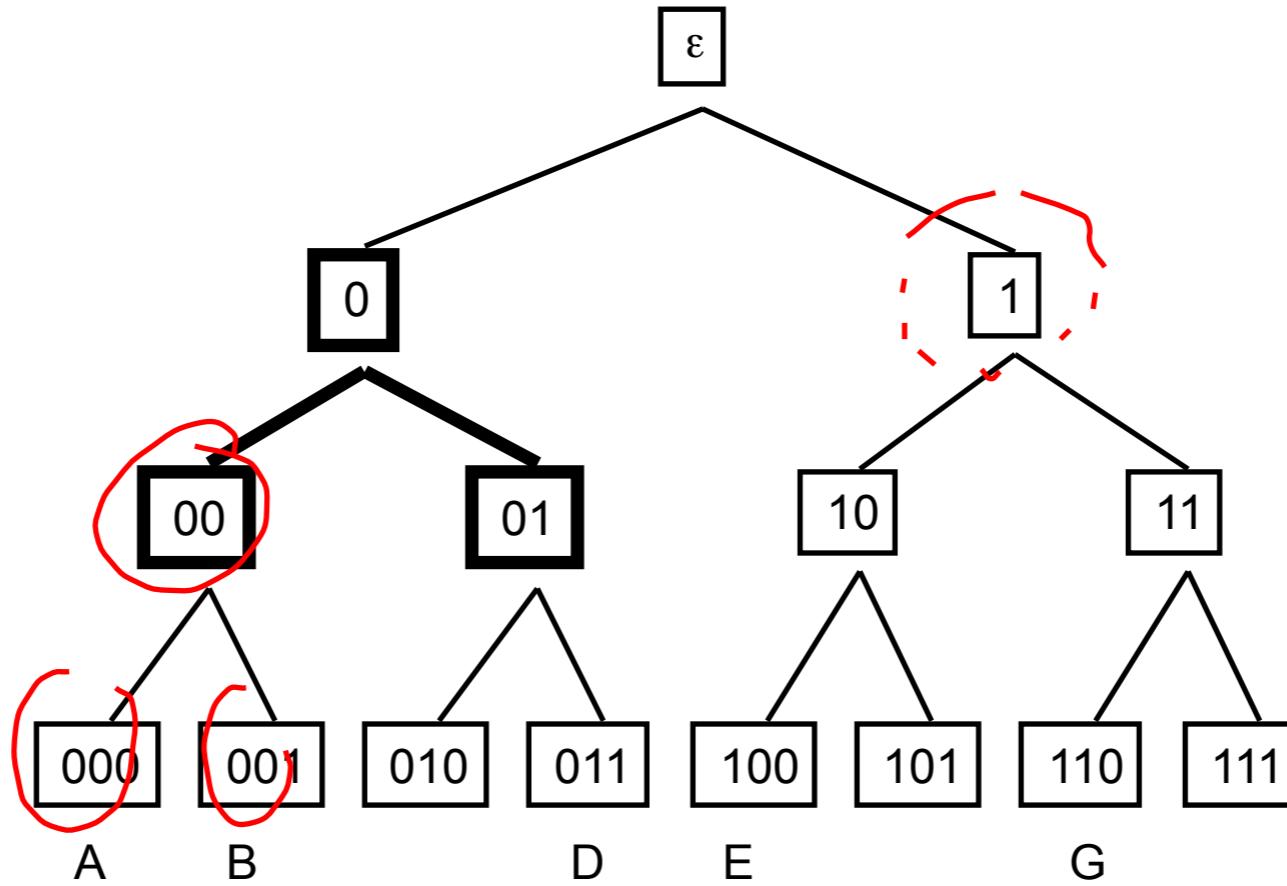


Adaptives Baumprotokoll Beispiel (2)

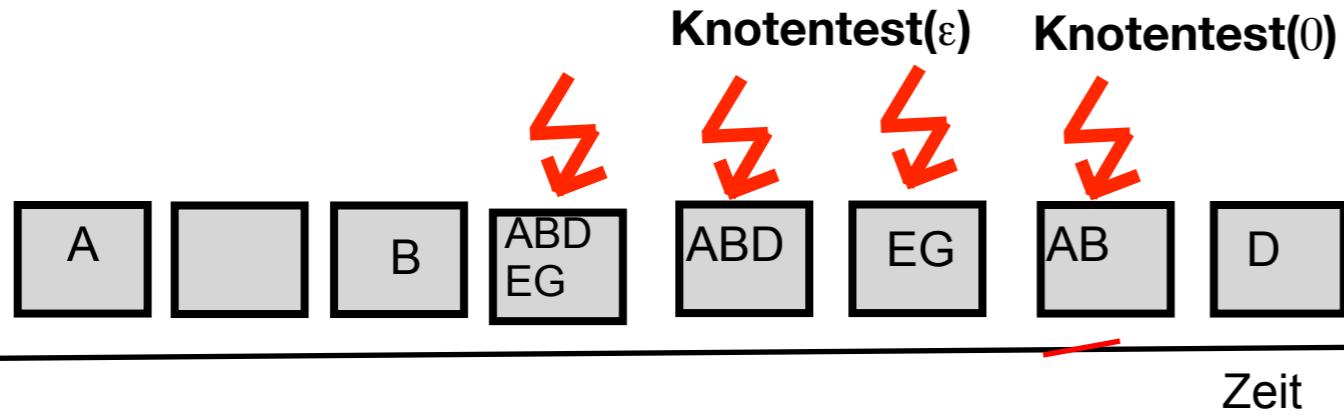


Adaptives Baumprotokoll Beispiel (3)

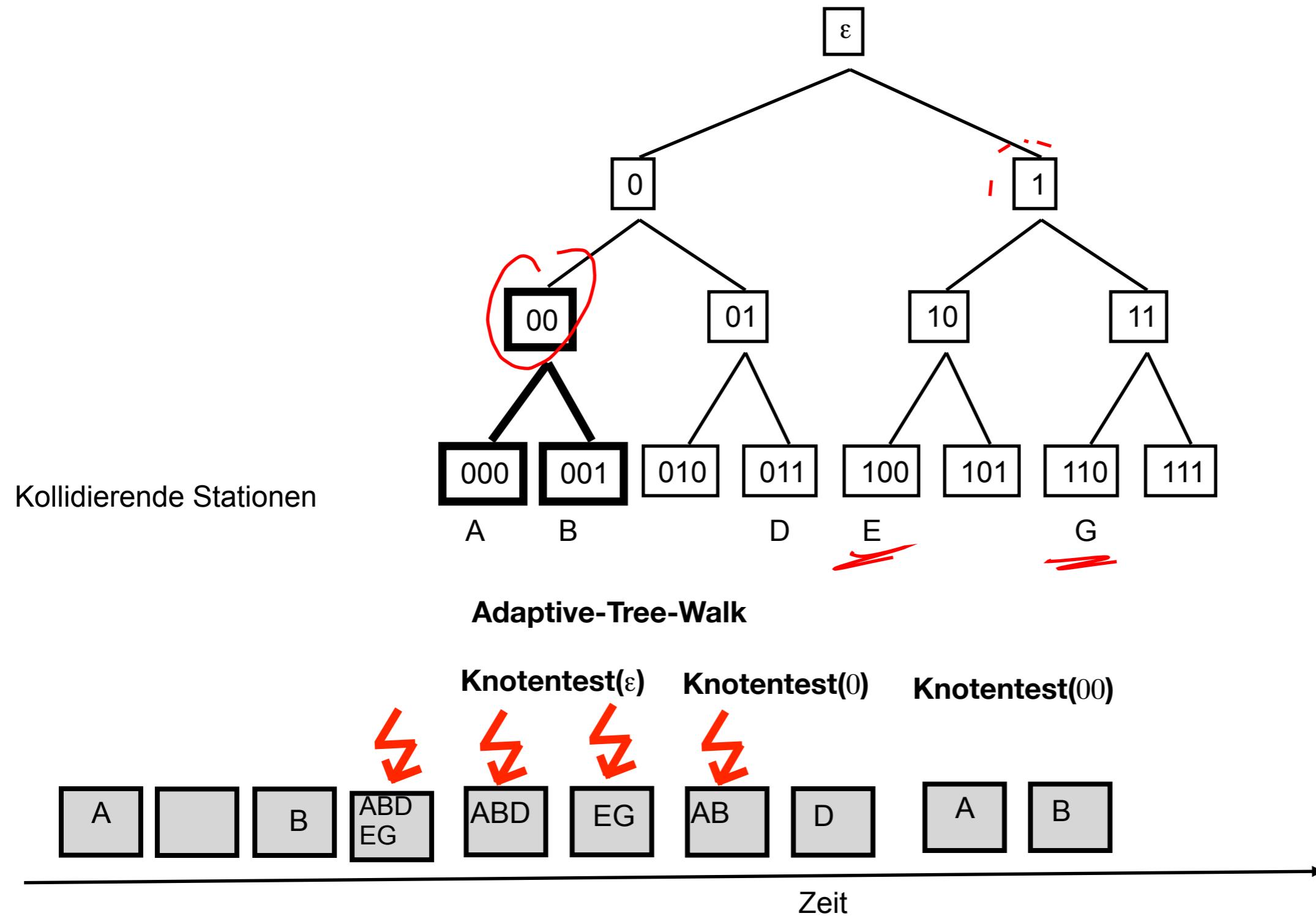
Kollidierende Stationen



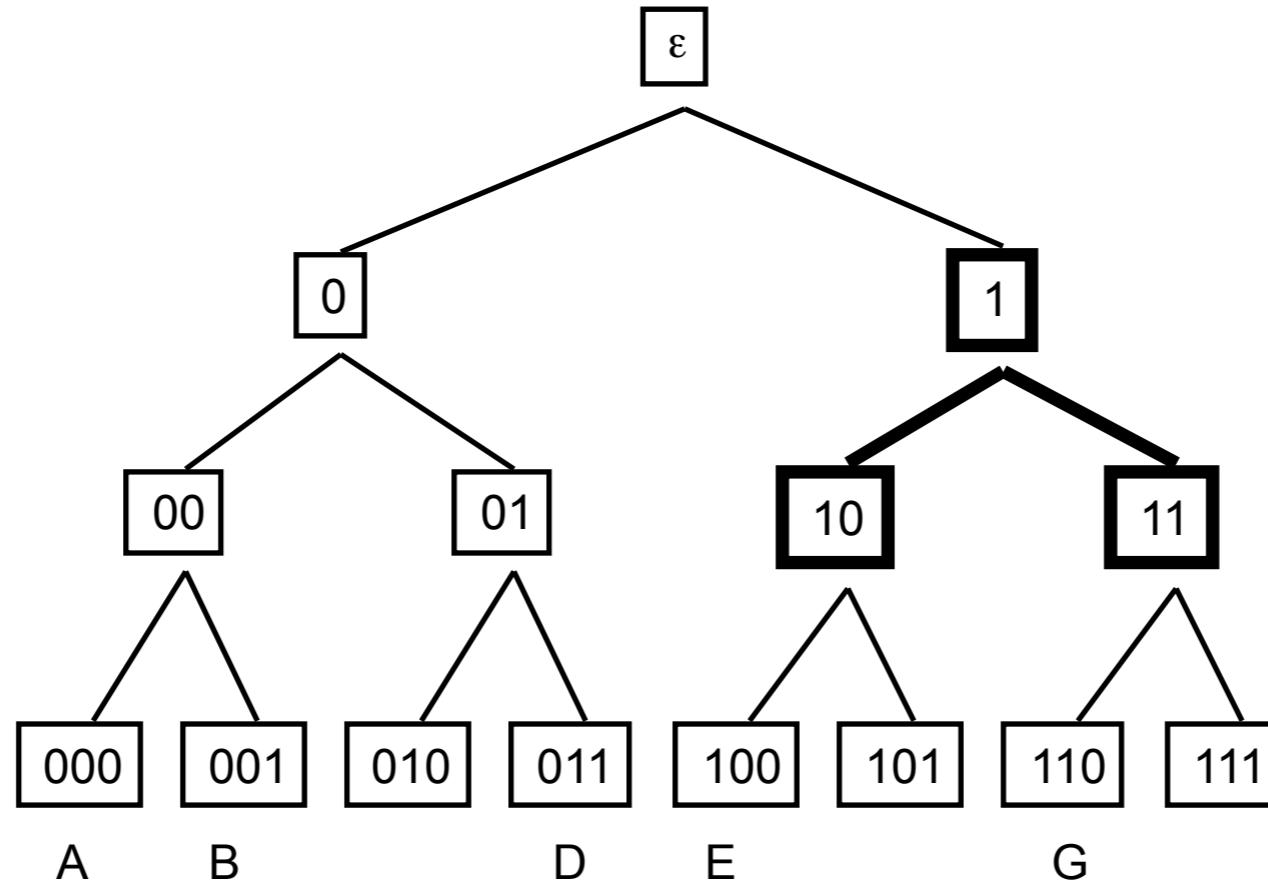
Adaptive-Tree-Walk



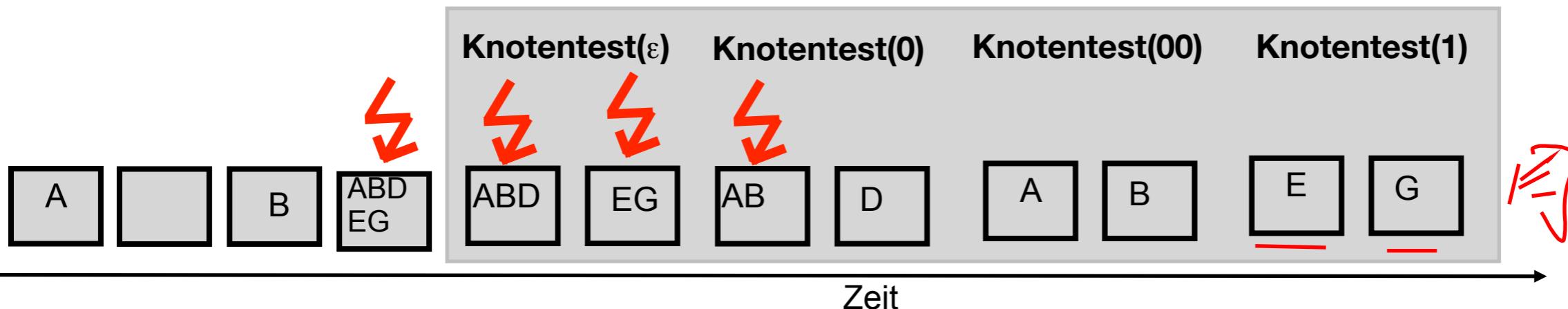
Adaptives Baumprotokoll Beispiel (4)



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

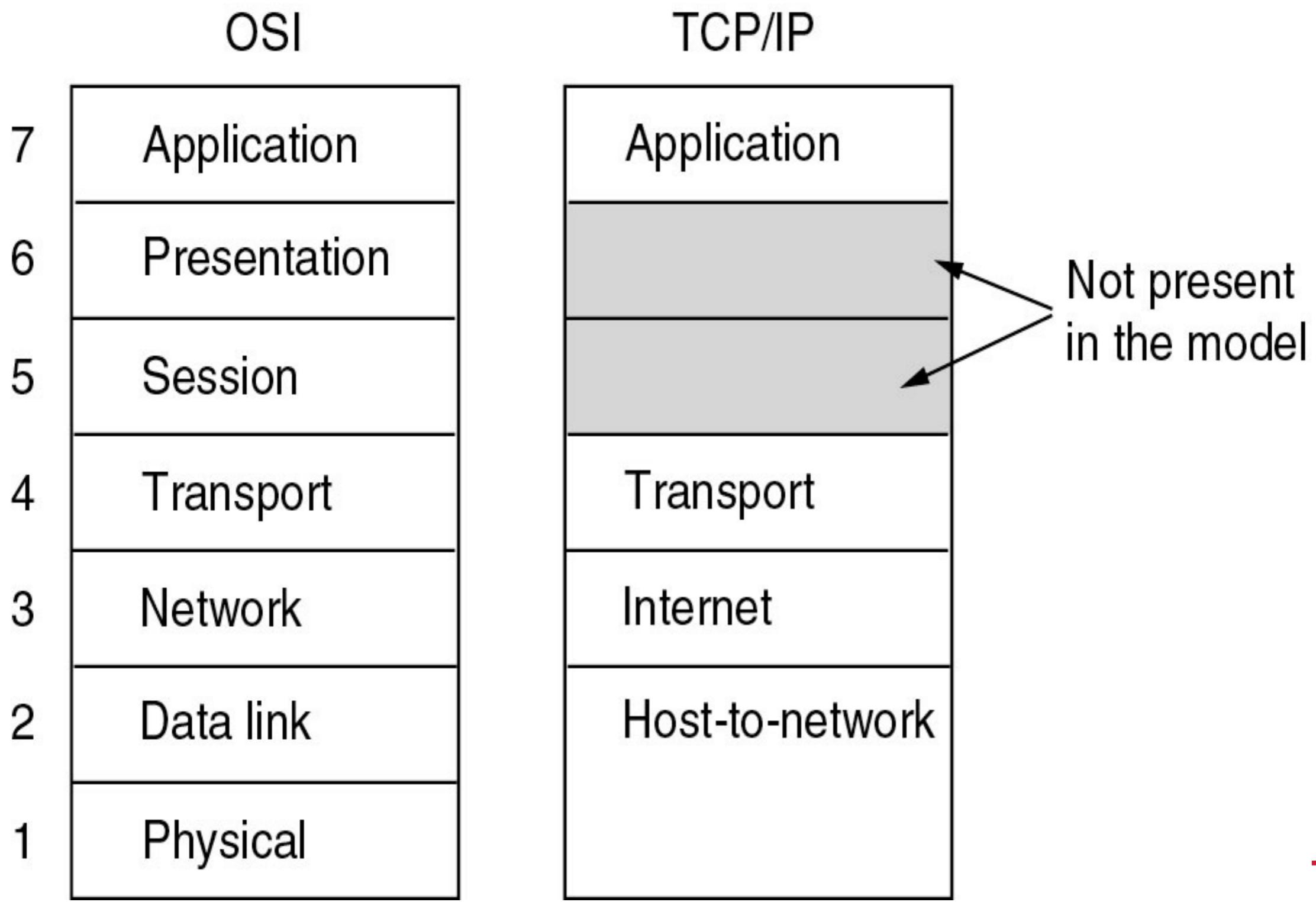
- Flusskontrolle zur Auslastung des Kanals

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

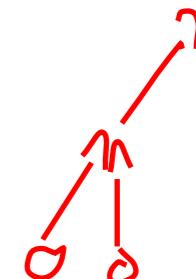
- Lokale Netzwerke können nicht nur über Hubs, Switches oder Bridges verknüpft werden
 - Hubs: Kollisionen nehmen überhand
 - Switches:
 - Routen-Information durch Beobachtung der Daten ineffizient
 - Broadcast aller Nachrichten schafft Probleme
 - Es gibt über 100 Mio. lokale Netzwerke im Internet...
- 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

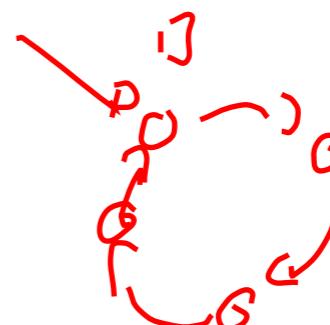
- enthält für Ziel (Destination) die Adresse des nächsten Rechners (Gateway)
- Destination kann einen Rechner oder ganze Sub-nets beschreiben
- Zusätzlich wird ein Default-Gateway angegeben

■ 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

- IP-Paket (datagram) enthält unter anderen
 - TTL (Time-to-Live): Anzahl der Hops (IPv6 hop-count)
 - Start-IP-Adresse
 - Ziel-IP-Adresse
- Behandlung eines Pakets
 - Verringere TTL (Time to Live) um 1
 - Falls TTL ≠ 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



C 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

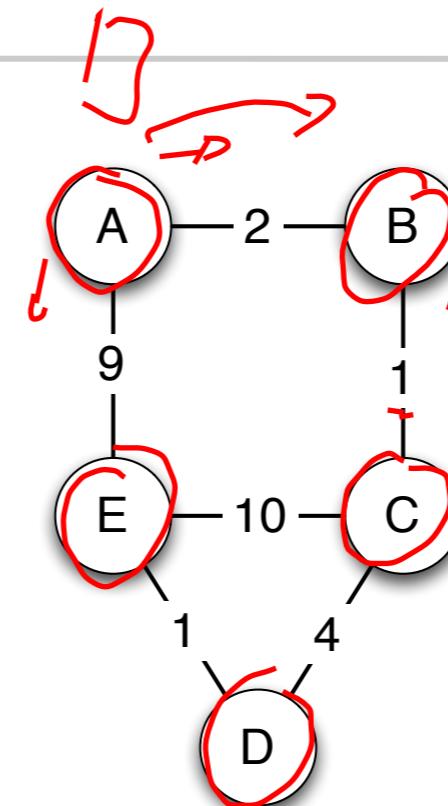
- Tabellen werden durch Routing-Algorithmus erstellt
- Zentraler Algorithmus, z.B. Link State
 - Einer/jeder kennt alle Information, muss diese erfahren
- Dezentraler Algorithmus, z.B. Distance Vector
 - arbeitet lokal in jedem Router
 - verbreitet lokale Information im Netzwerk

Dijkstra

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

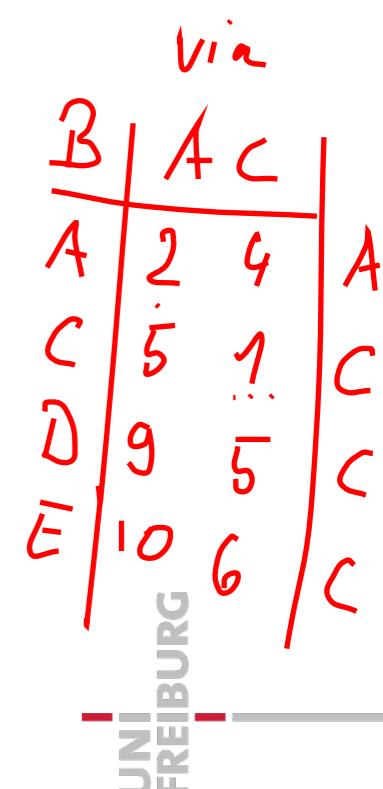
von C	über			Routing Tabellen Eintrag
	B	D	E	
nach A	3	11	18	B
B	1	9	16	B
D	6	4	11	D
E	7	5	10	D

A ↘

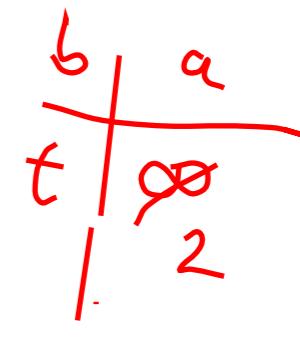
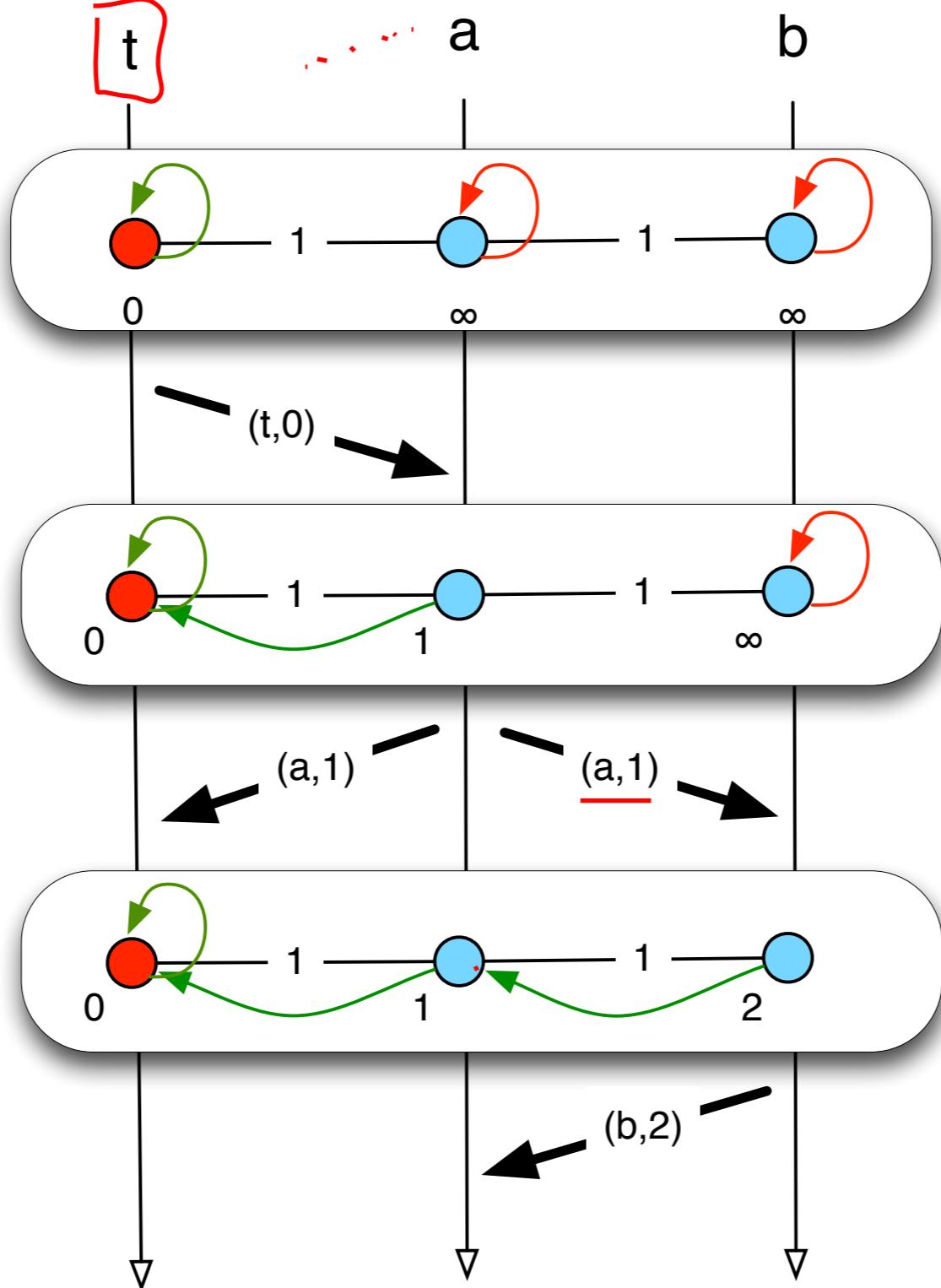
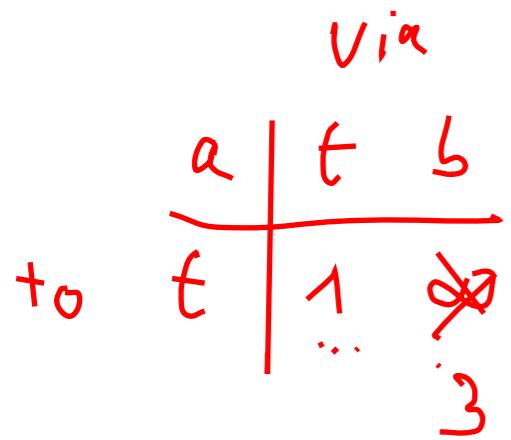
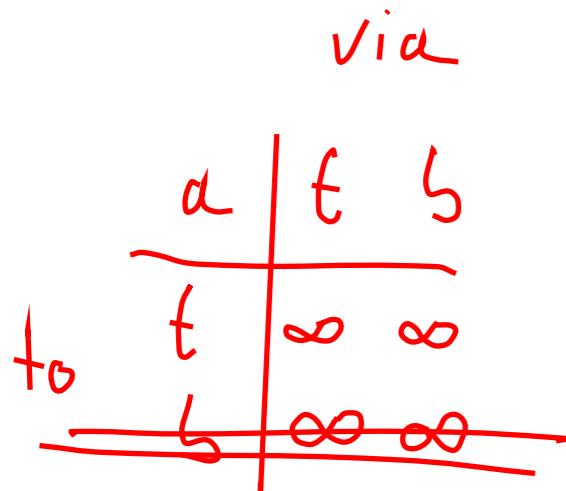
A ∞

A ∞

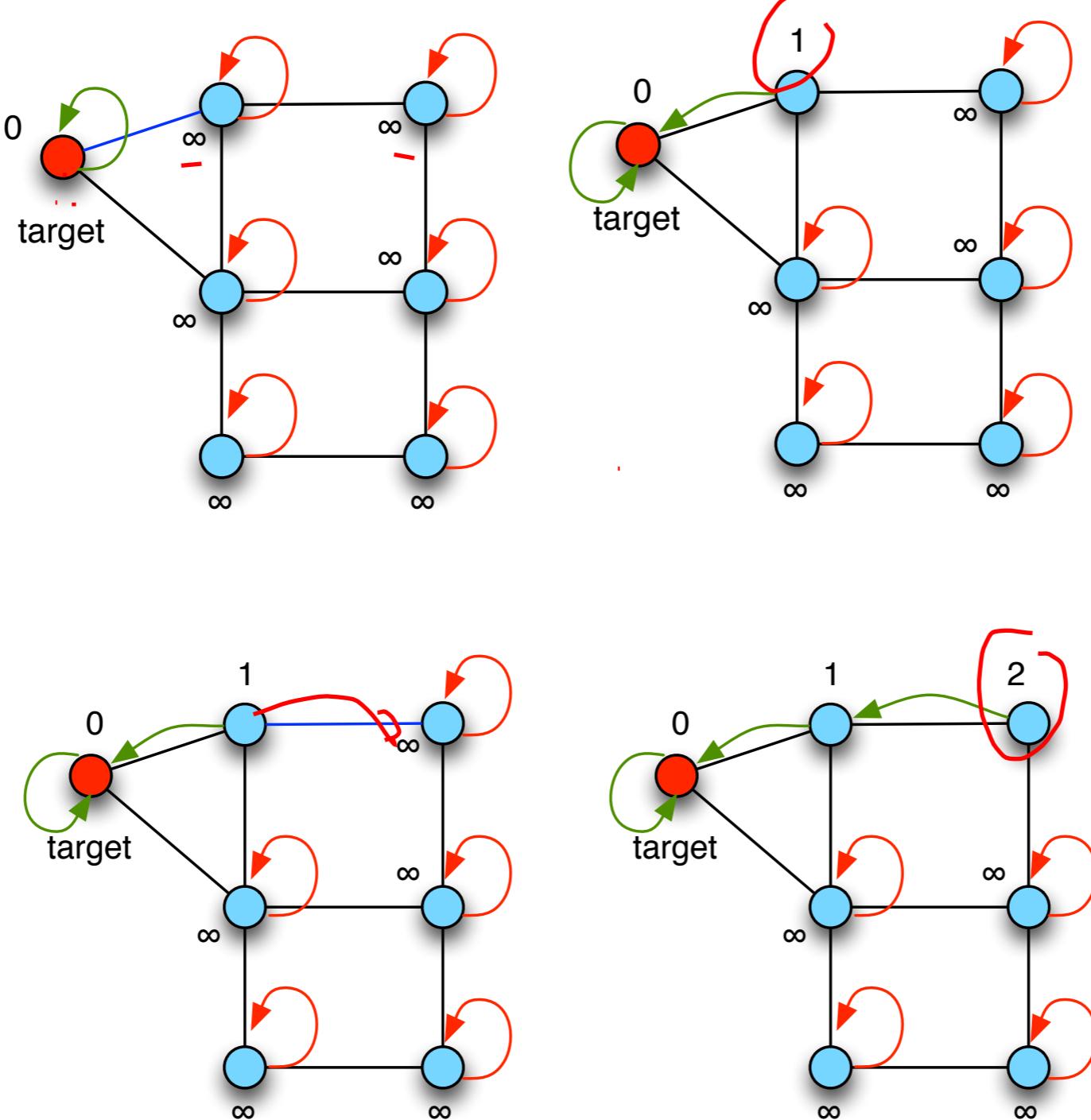
A ∞



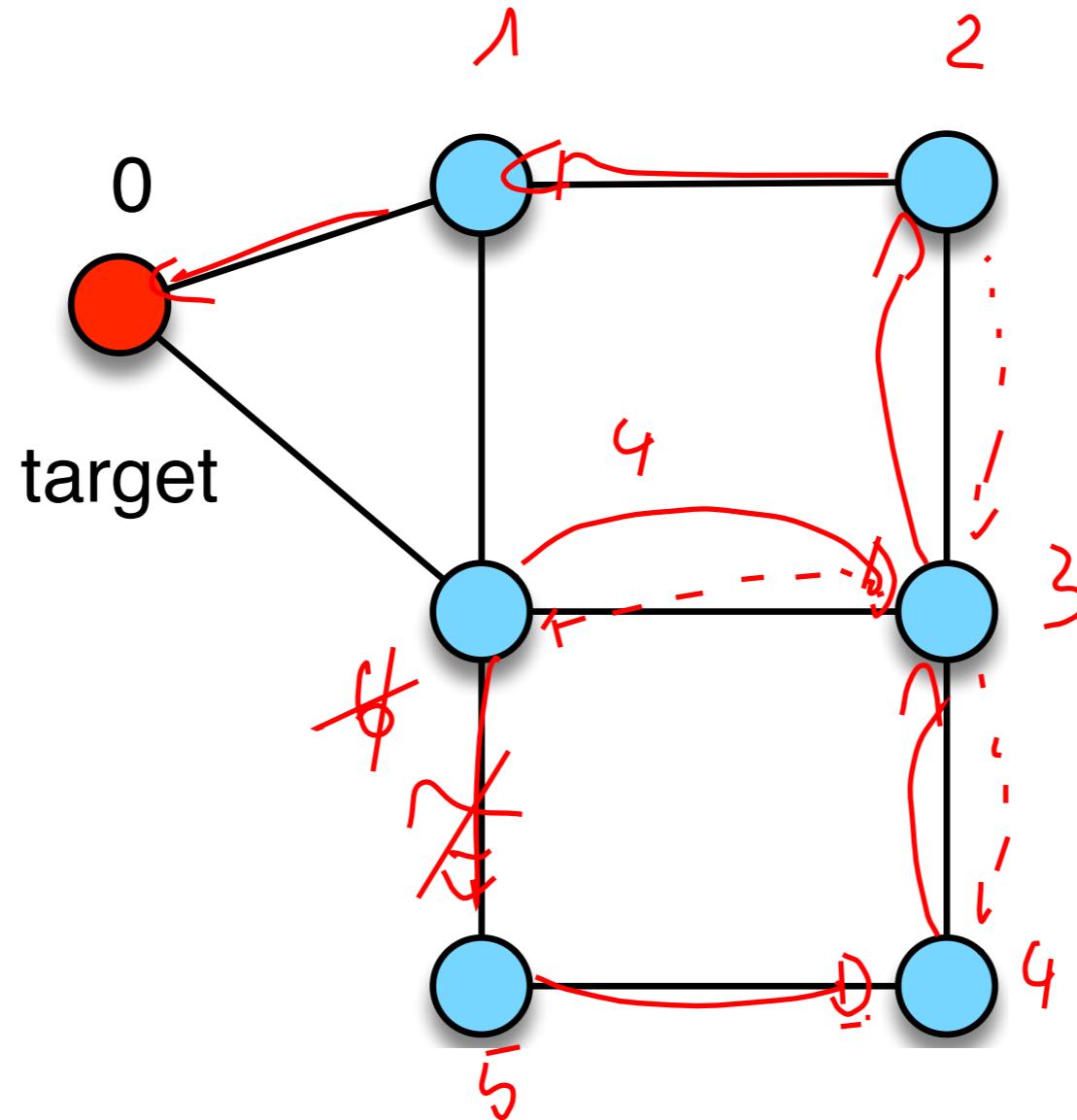
Beispiel für Distance-Vector für Ziel t



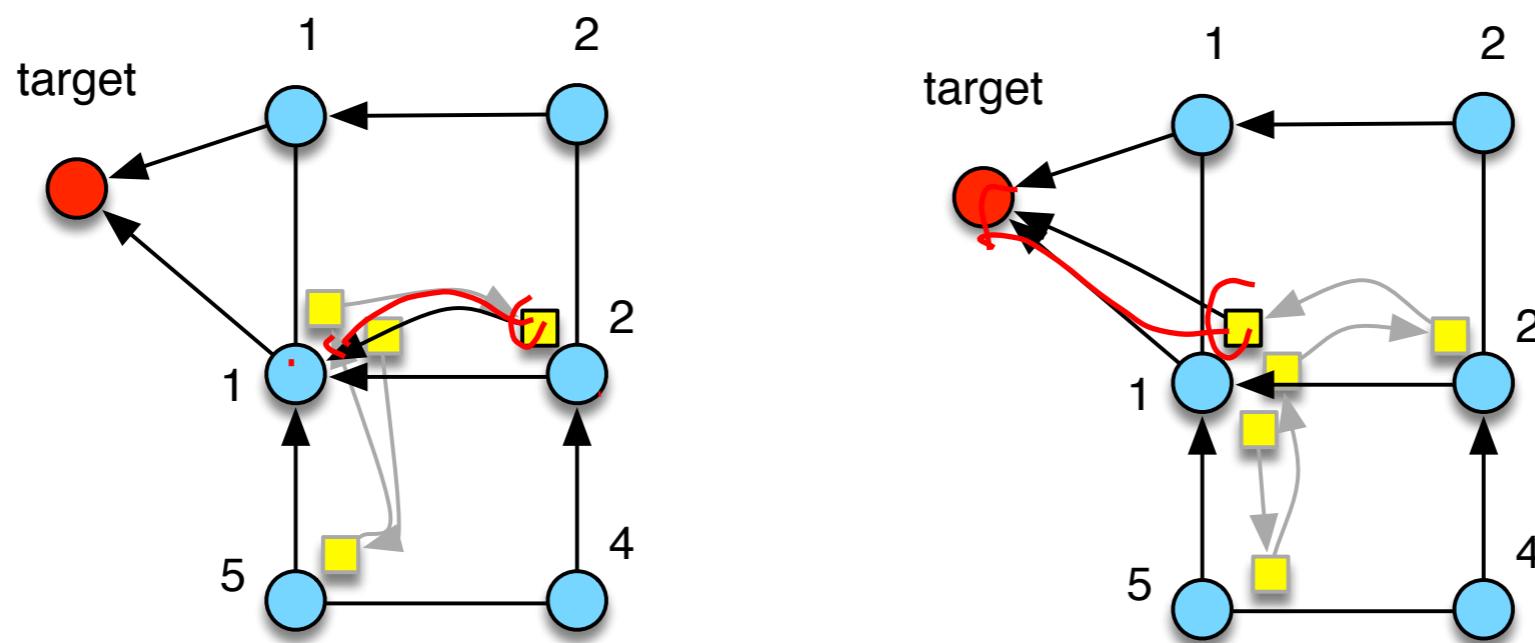
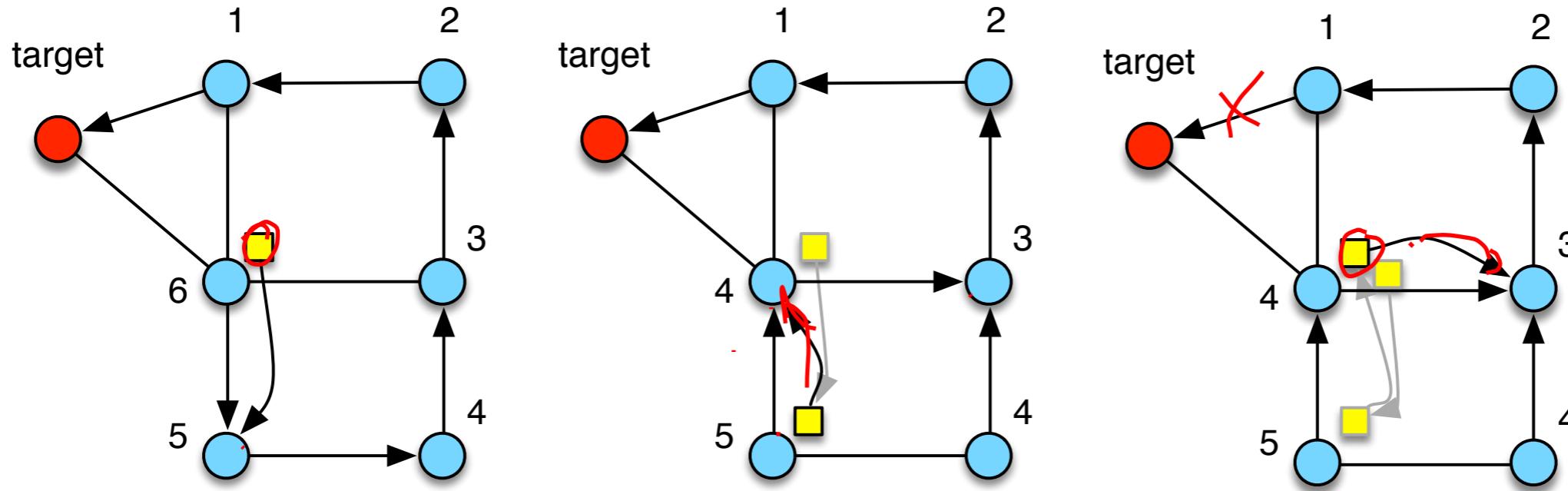
Distance-Vector für ein Ziel



Distance-Vector für ein Ziel

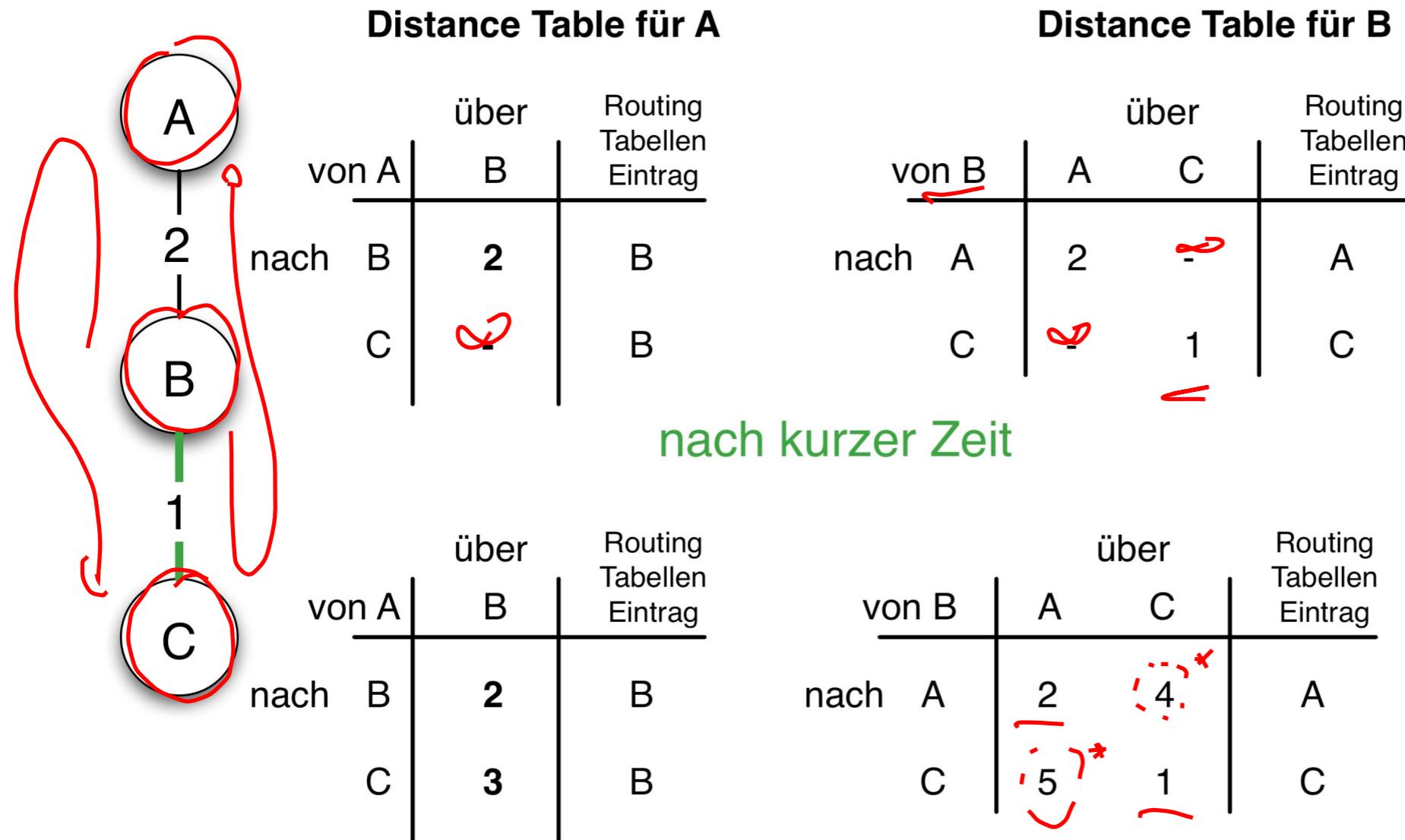


Irrlicher im Routing



Das “Count to Infinity” - Problem

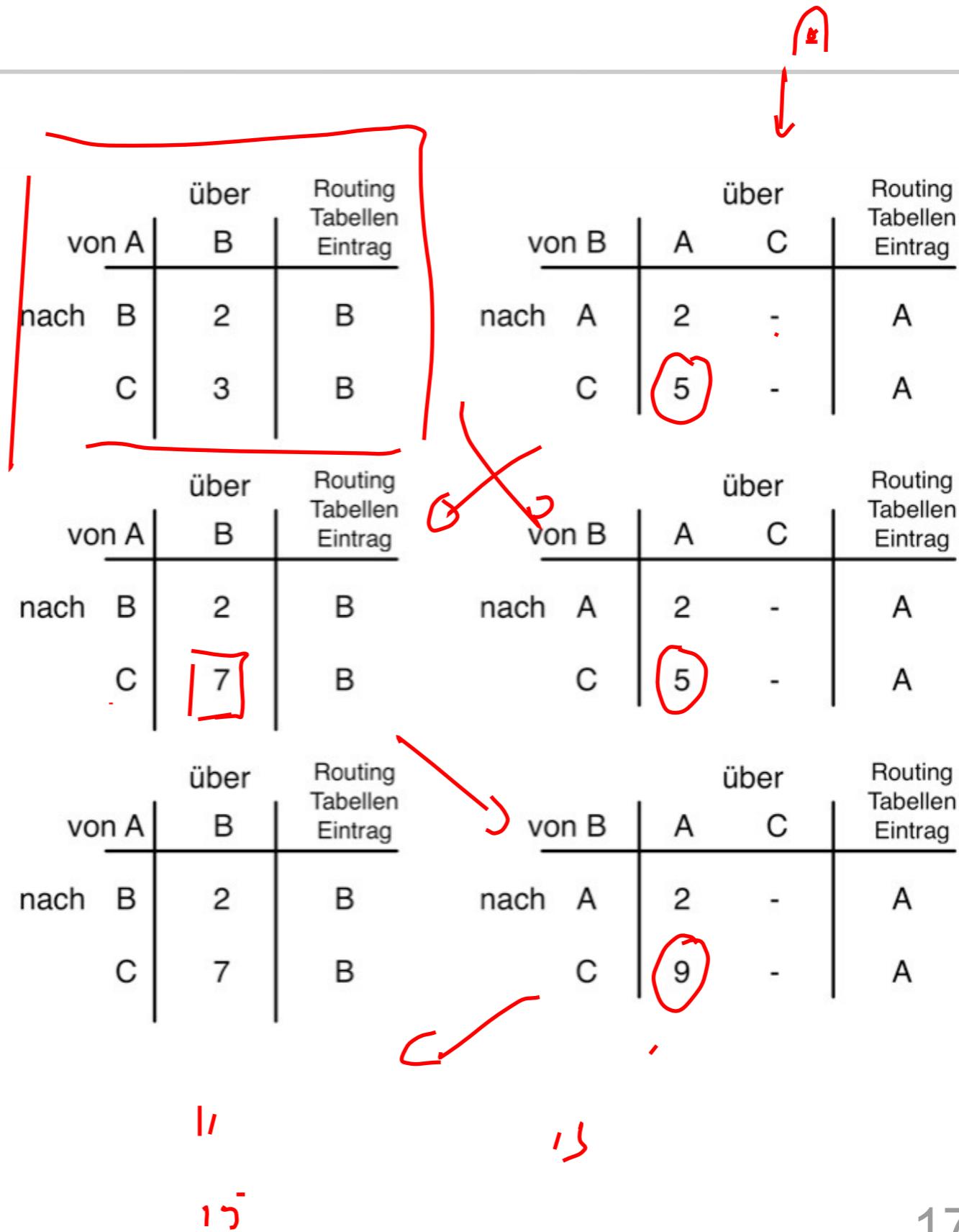
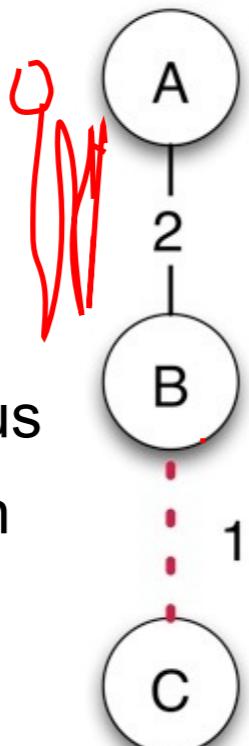
- Gute Nachrichten verbreiten sich schnell
 - Neue Verbindung wird schnell veröffentlicht



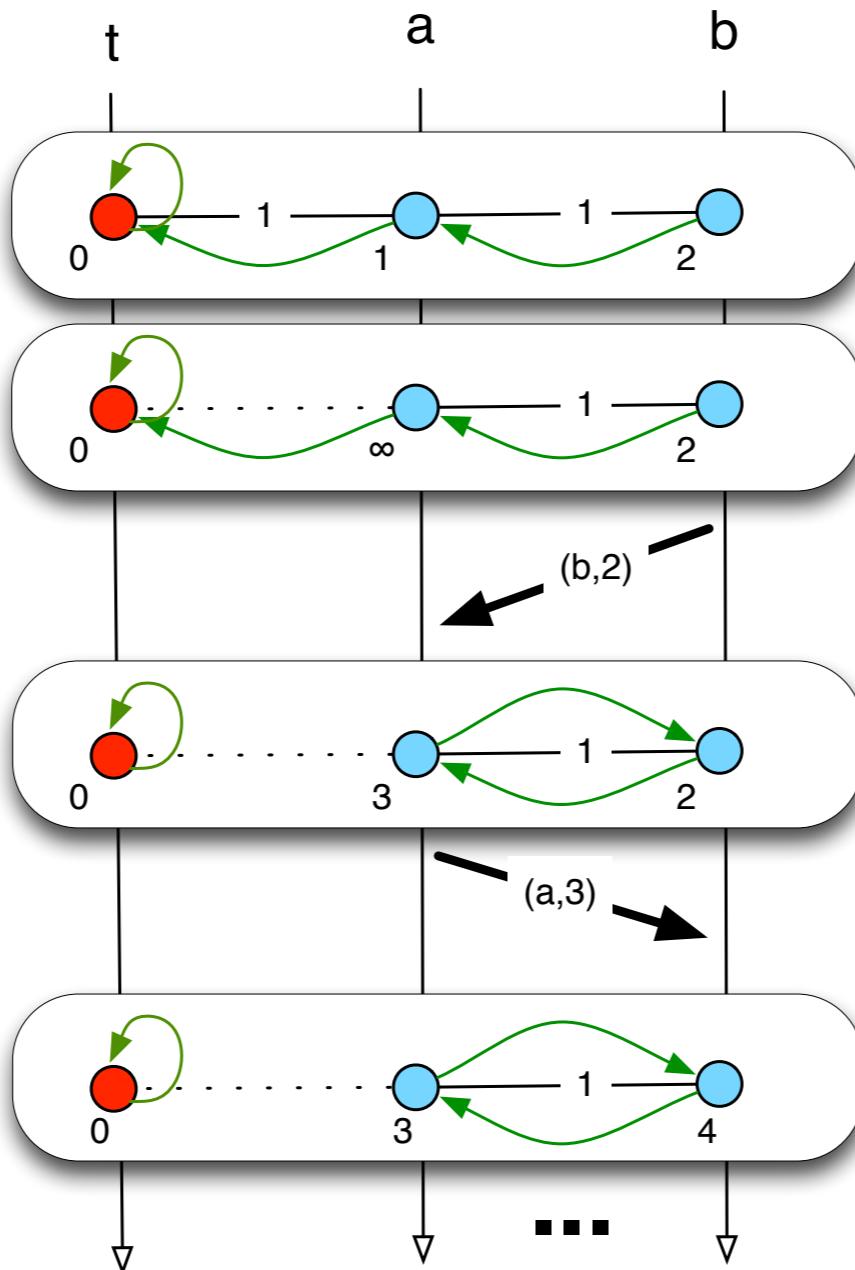
Das “Count to Infinity” - Problem

- Schlechte Nachrichten verbreiten sich langsam

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



Das “Count to Infinity” - Problem für Ziel t



Link-State Protocol

■ Link State Router

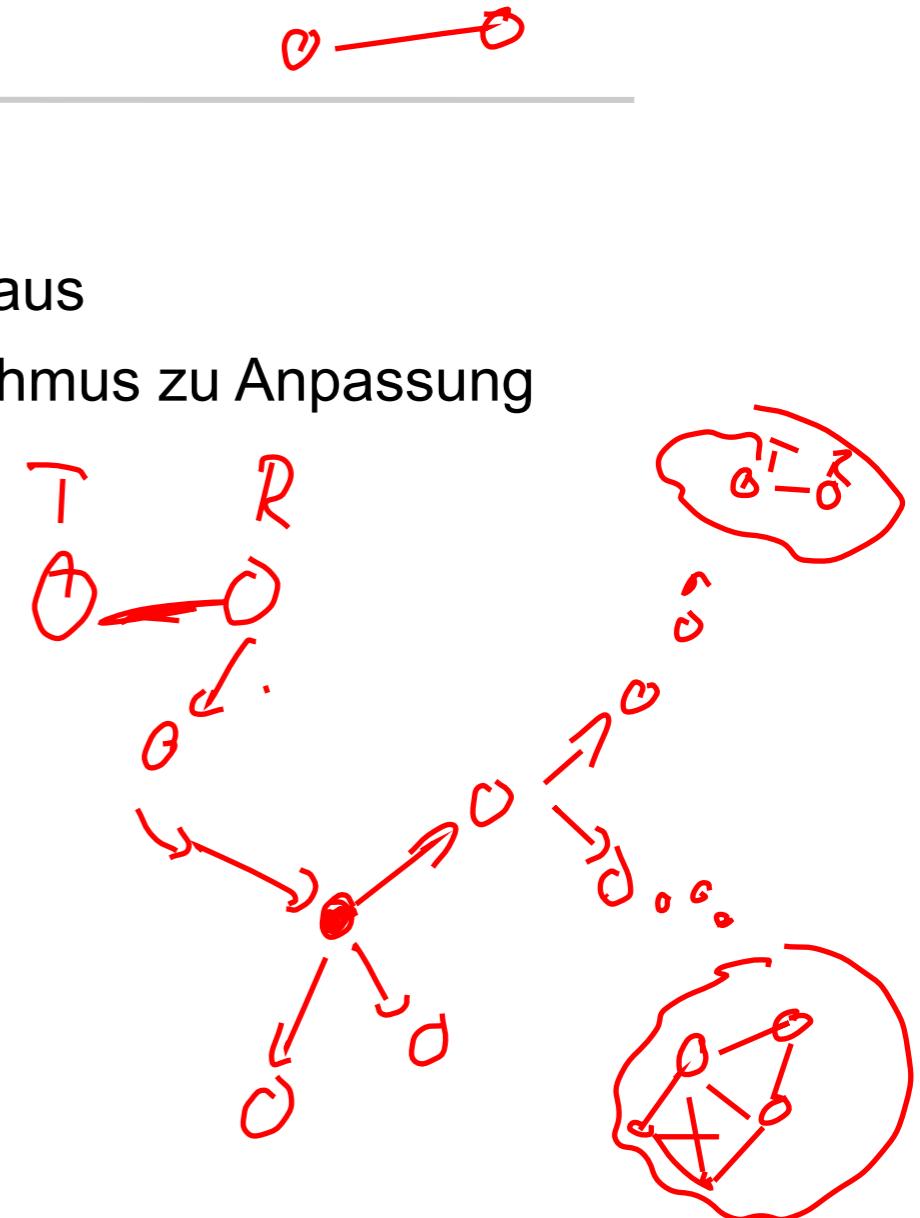
- tauschen Information mittels Link State Packets (LSP) aus
- Jeder verwendet einen eigenen Kürzeste-Wege-Algorithmus zu Anpassung der Routing-Tabelle

■ 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

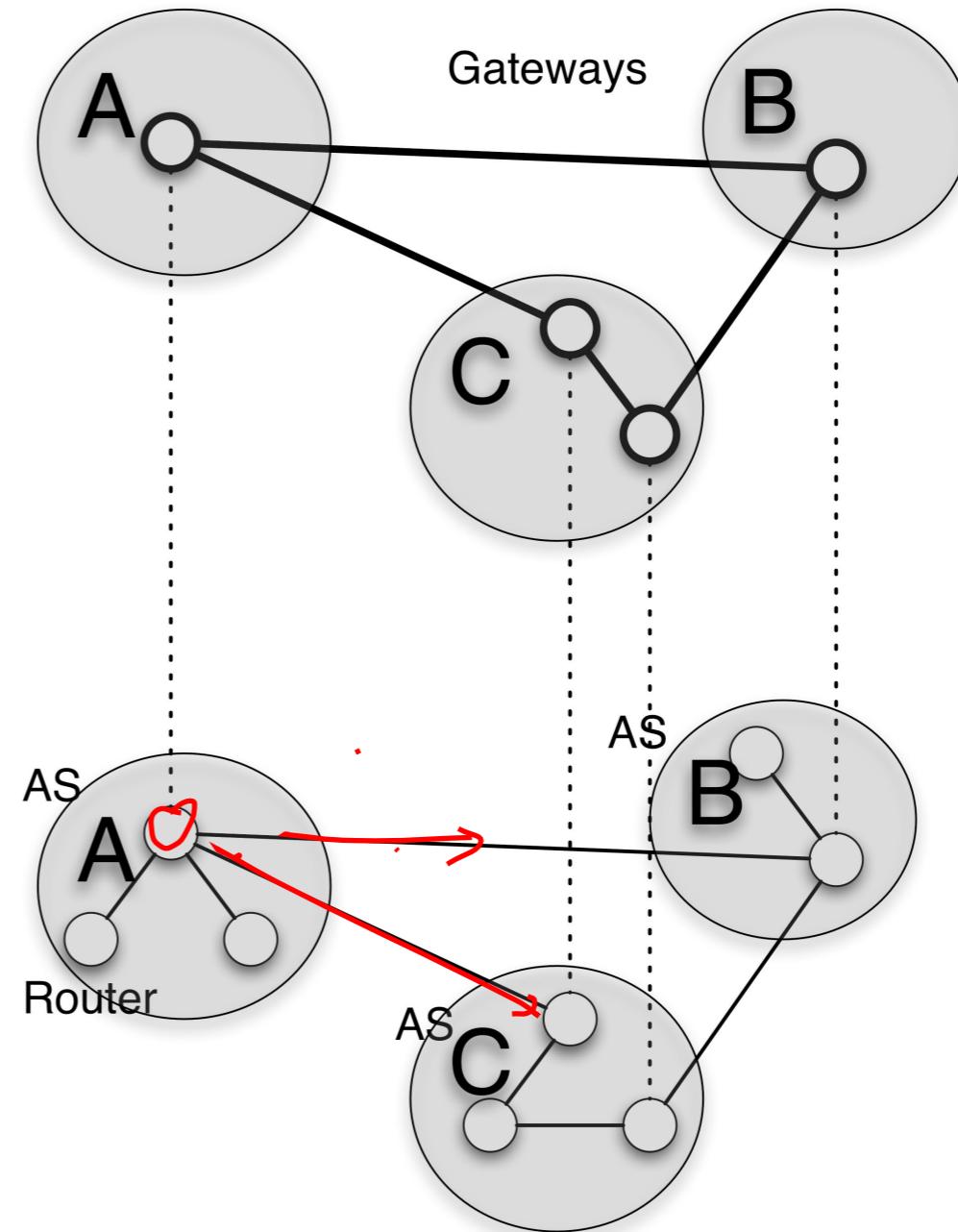


Die Grenzen des flachen Routing

- 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 \leftarrow temporär
 - 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



TCP Tahoe

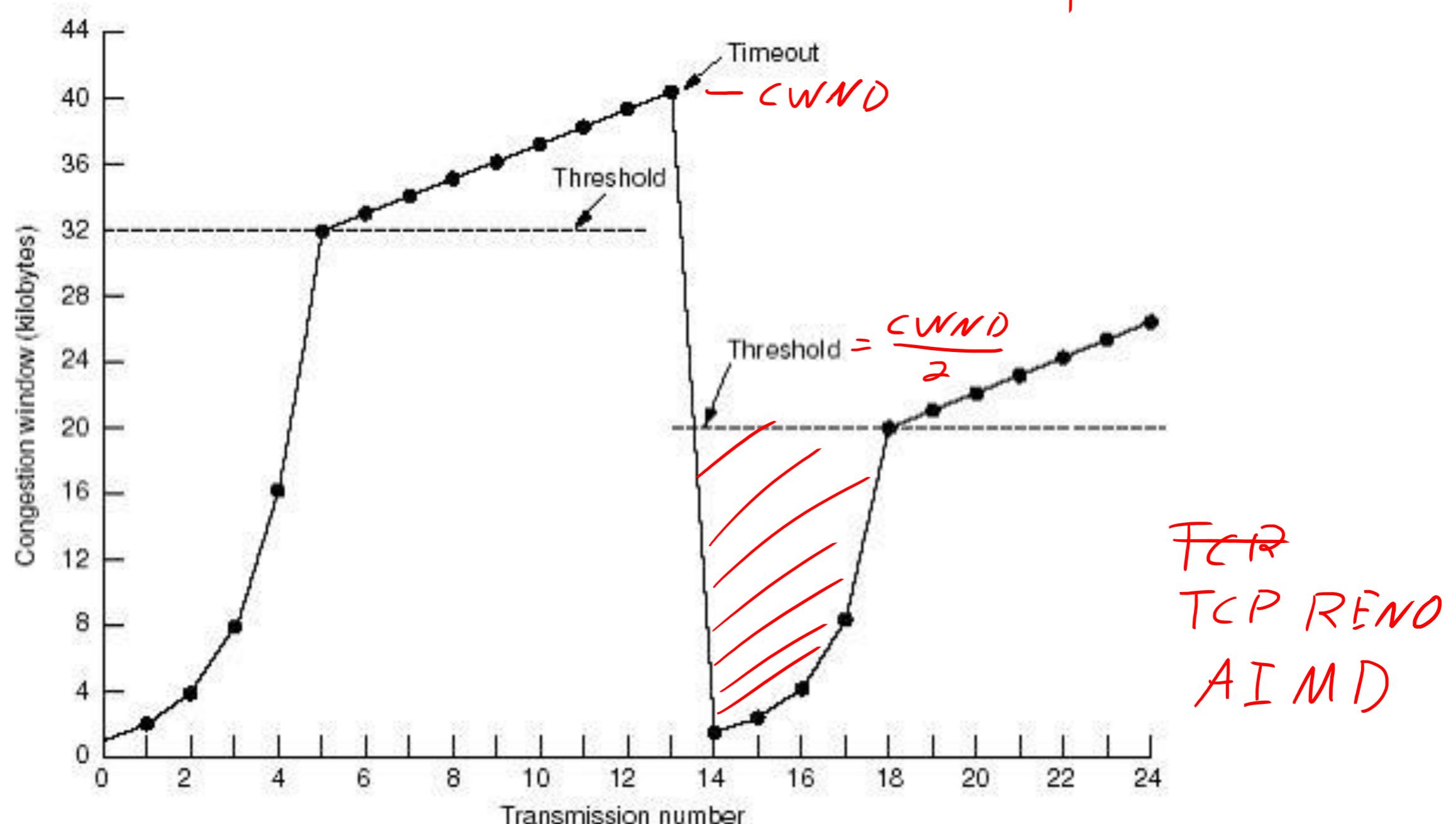


Fig3

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:
 - $cwnd \leftarrow ssthresh$

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

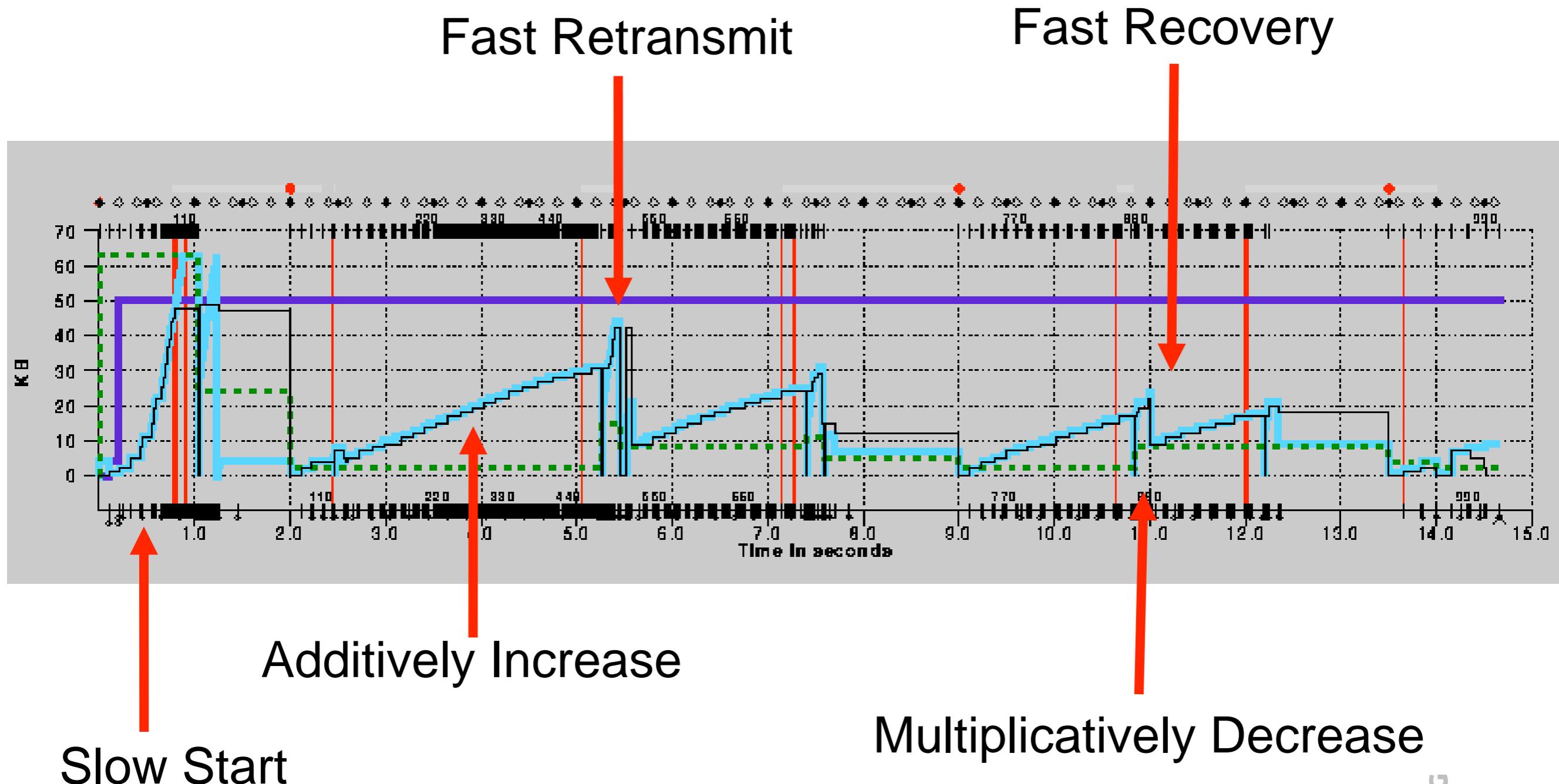
Stauvermeidungsprinzip: AIMD

- 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



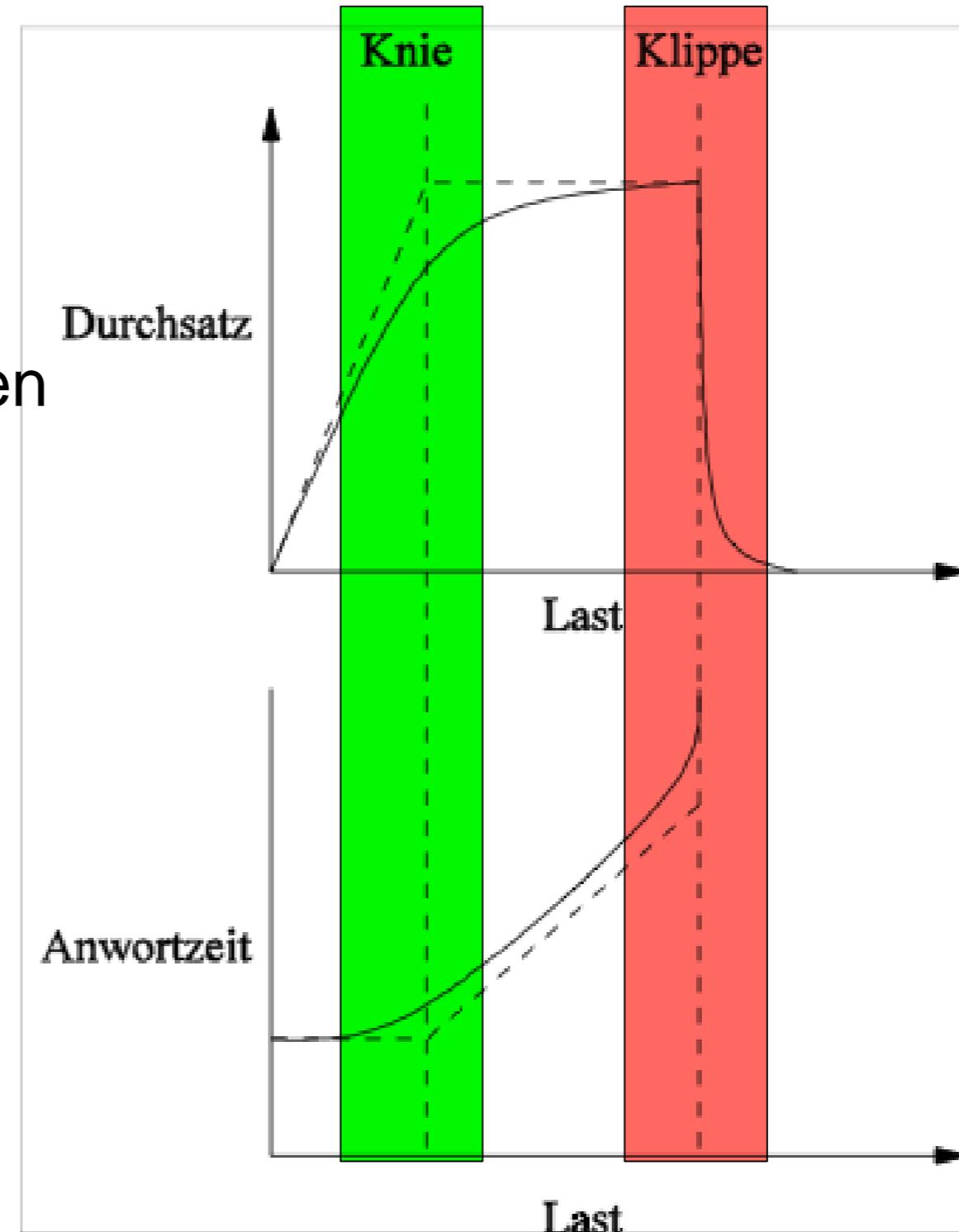
Durchsatz und Antwortzeit

■ 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 $x=(x_1, \dots, x_n)$:

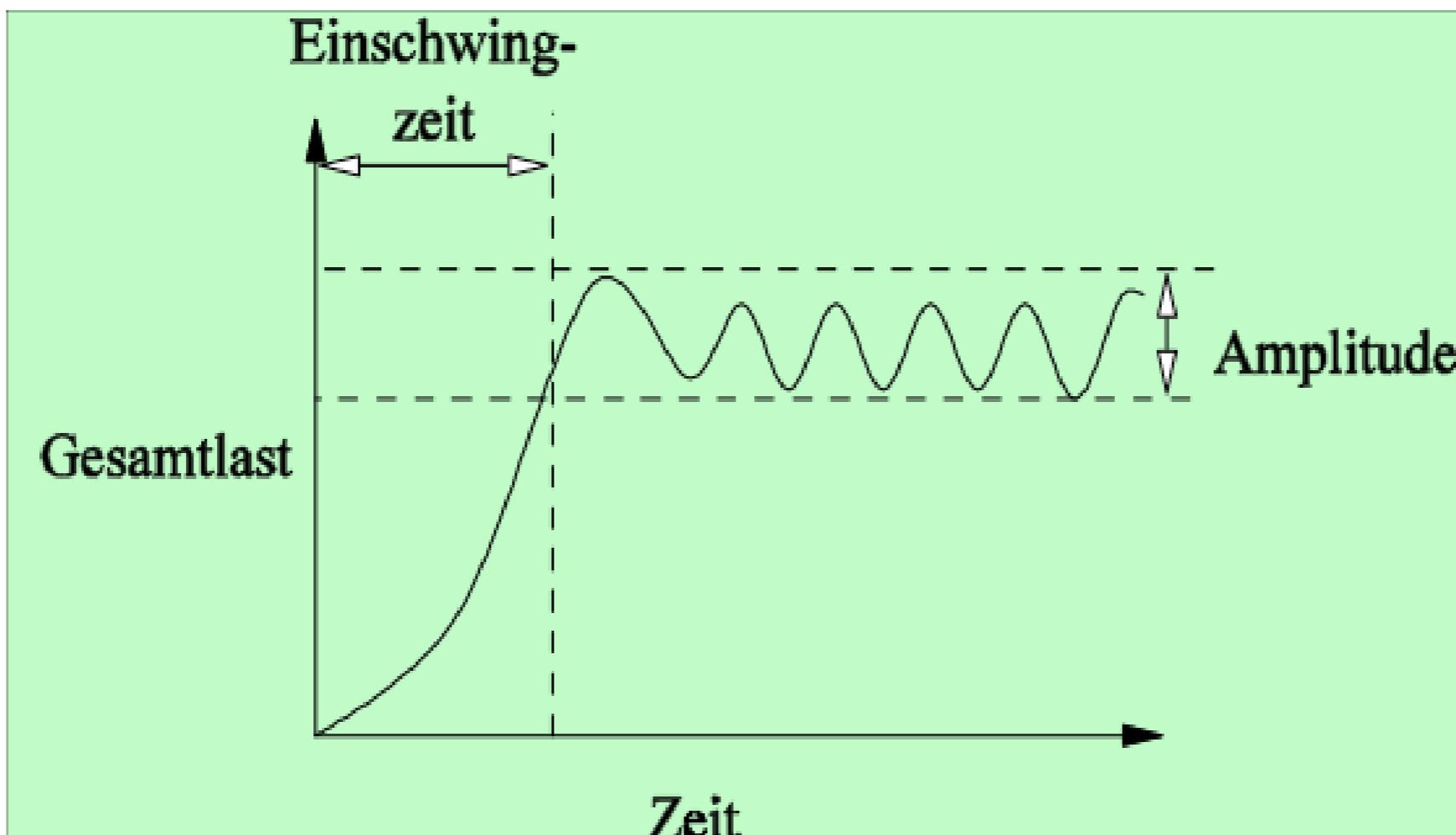
$$F(x) = \frac{(\sum_{i=1}^n x_i)^2}{n \sum_{i=1}^n (x_i)^2}.$$

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

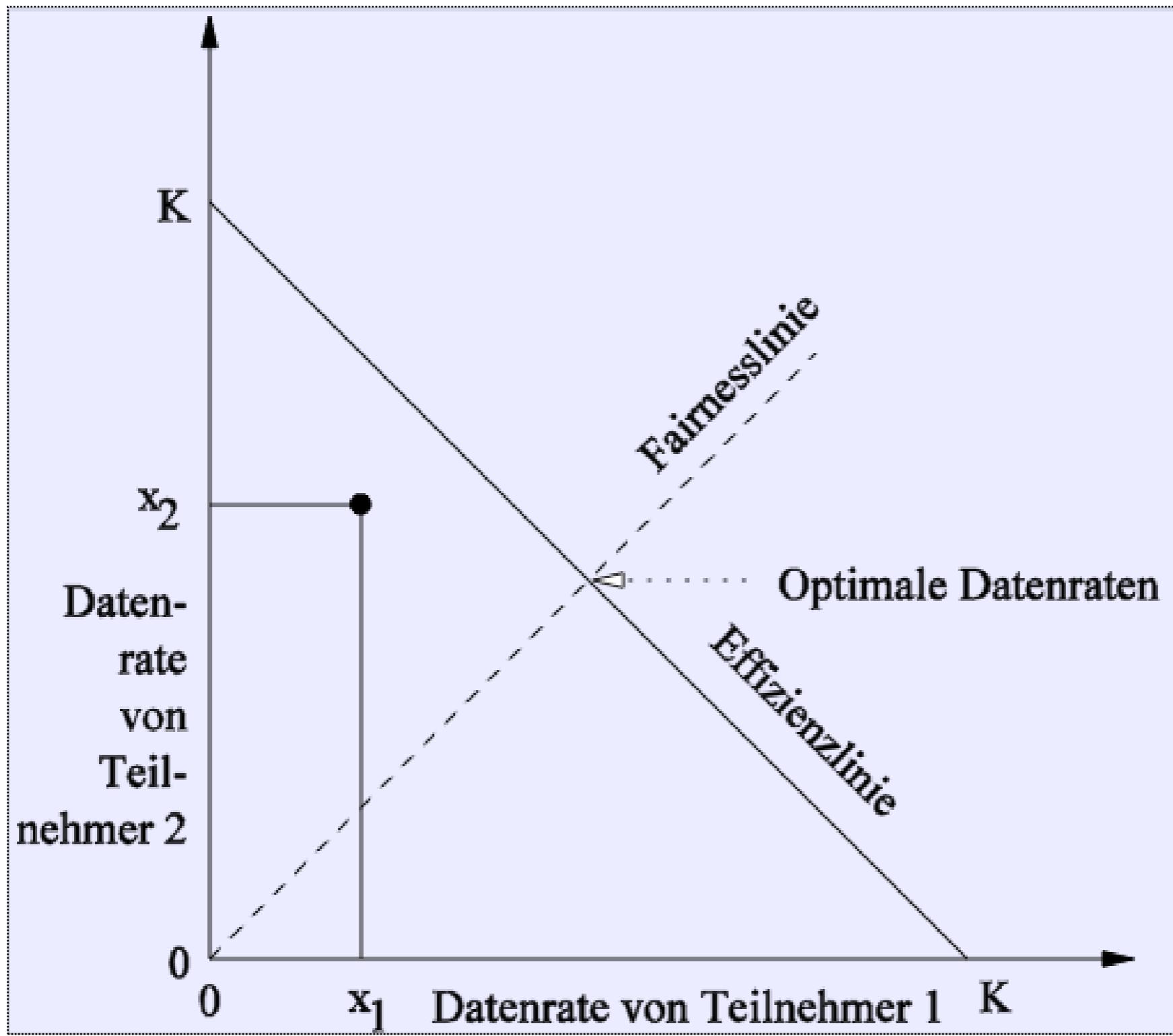
•

Konvergenz

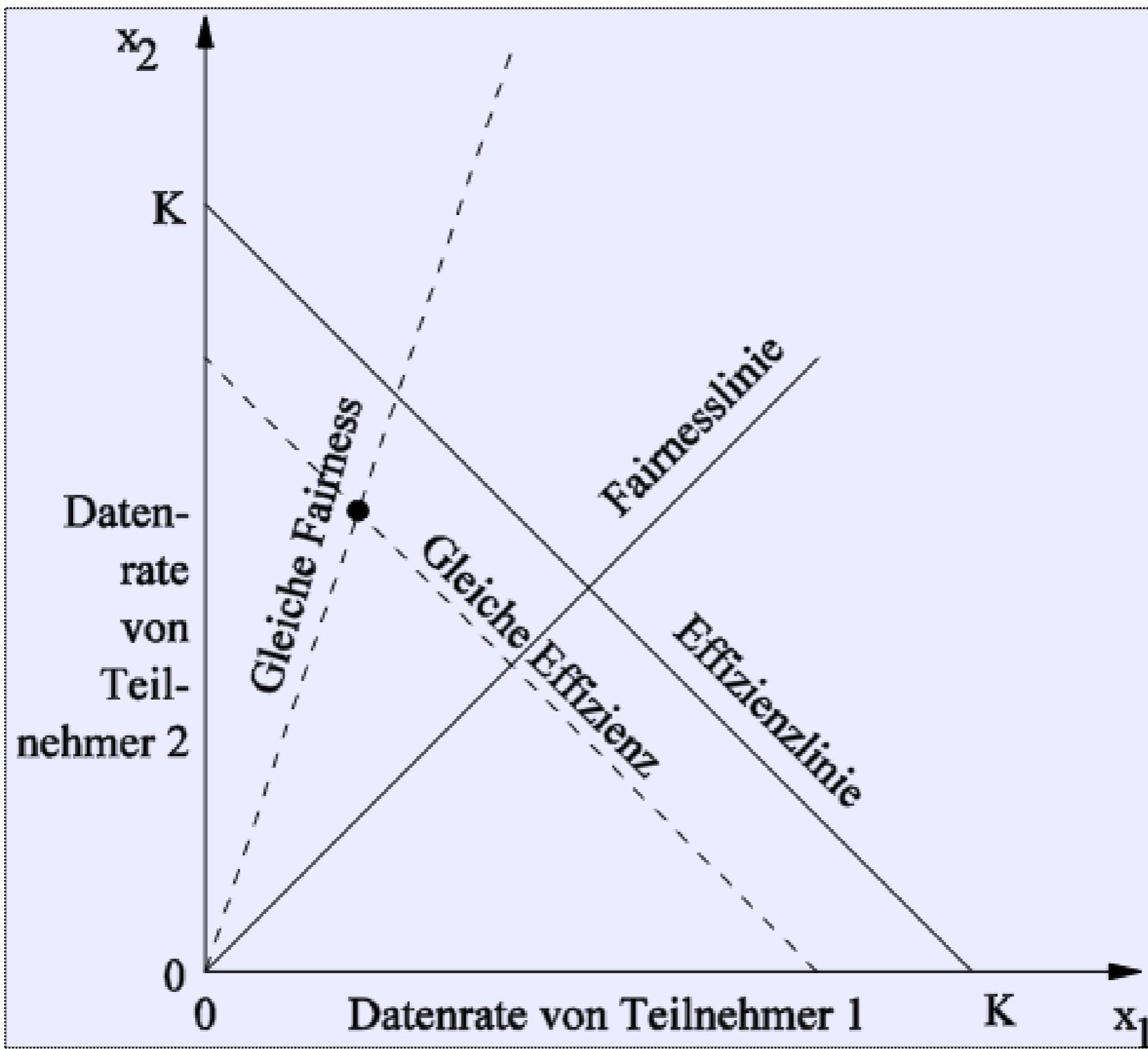
- Konvergenz unmöglich
- Bestenfalls Oszillation um Optimalwert
 - Oszillationsamplitude A
 - Einschwingzeit T



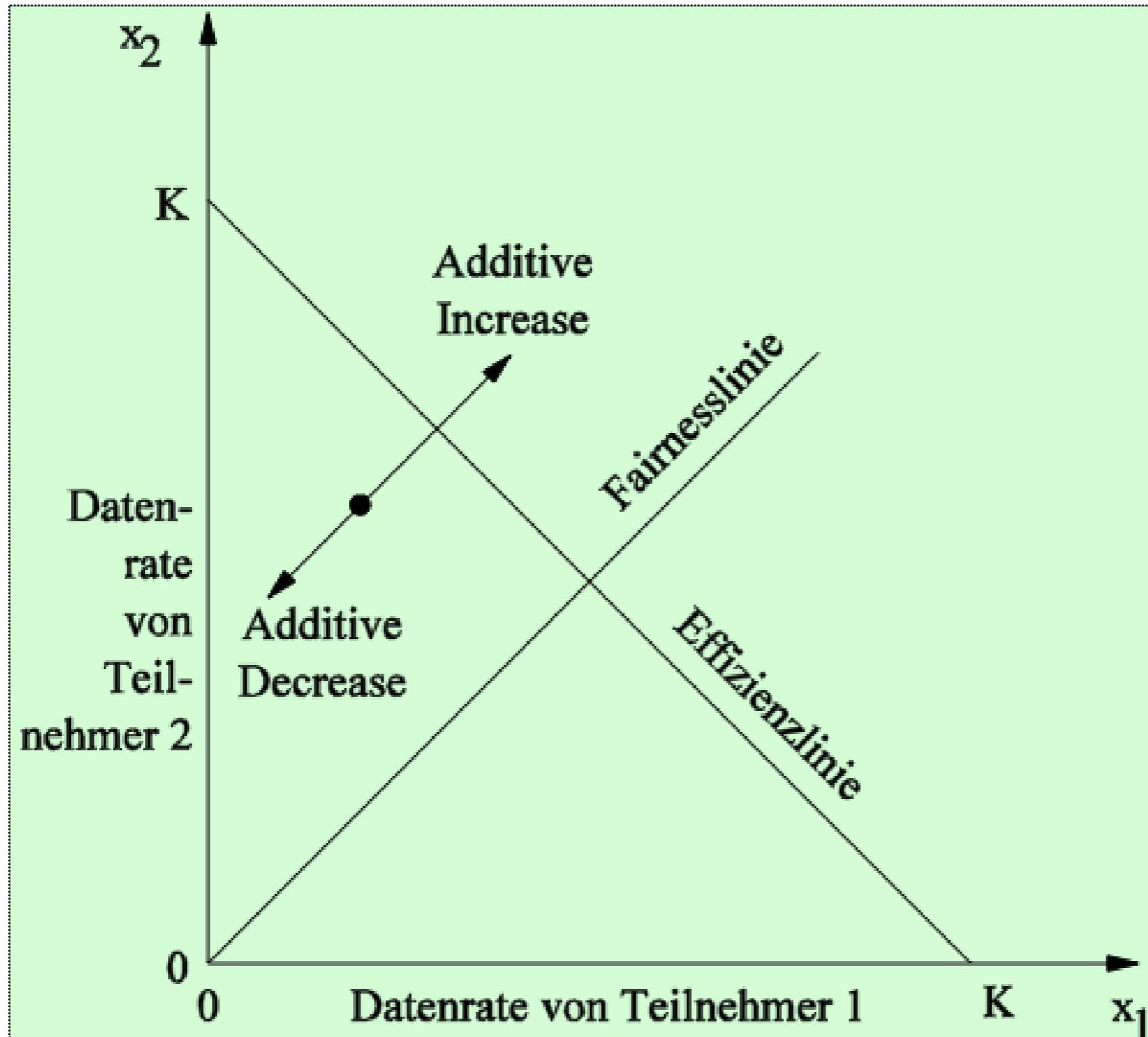
Vektordarstellung (I)



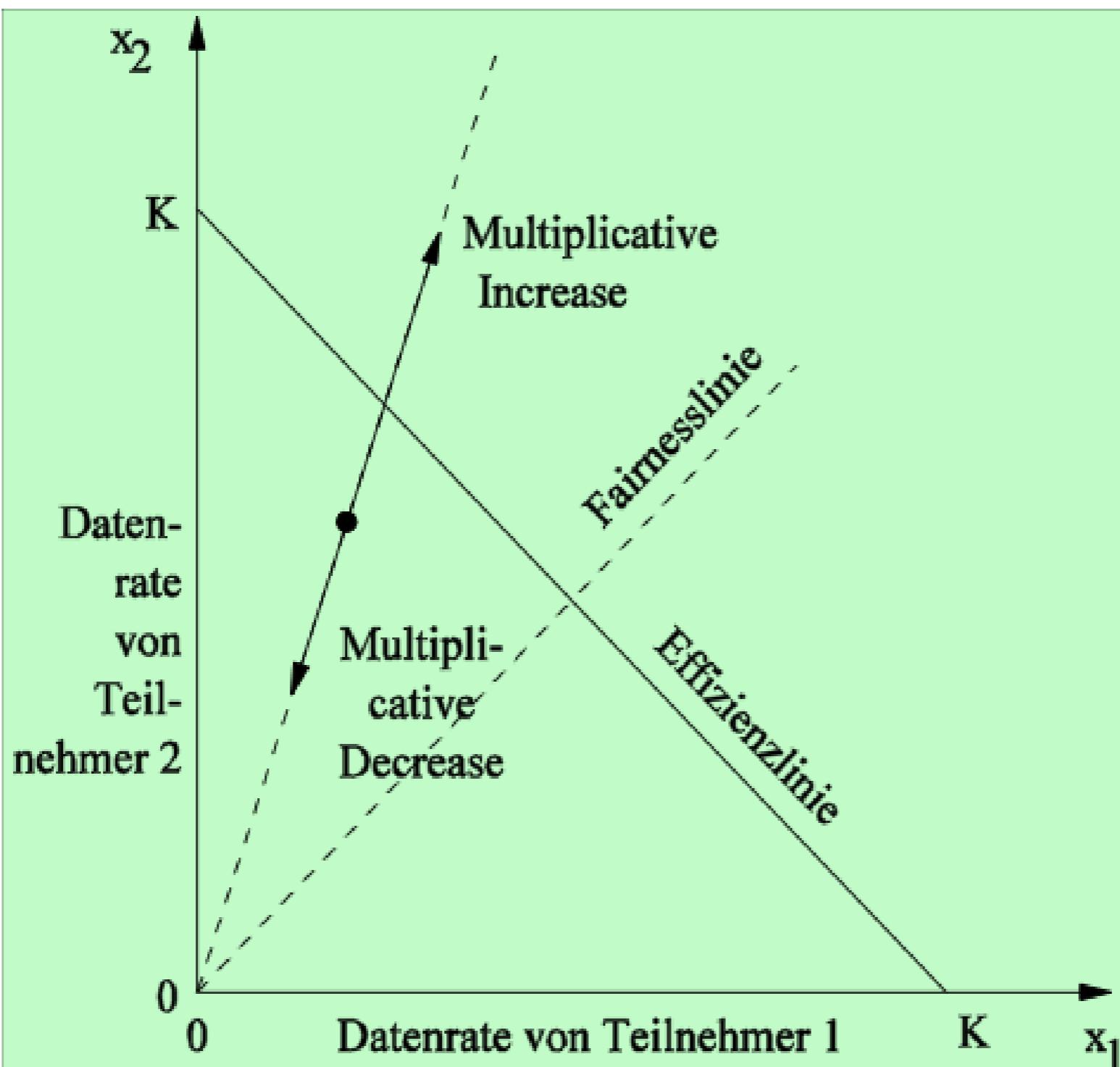
Vektordarstellung (II)



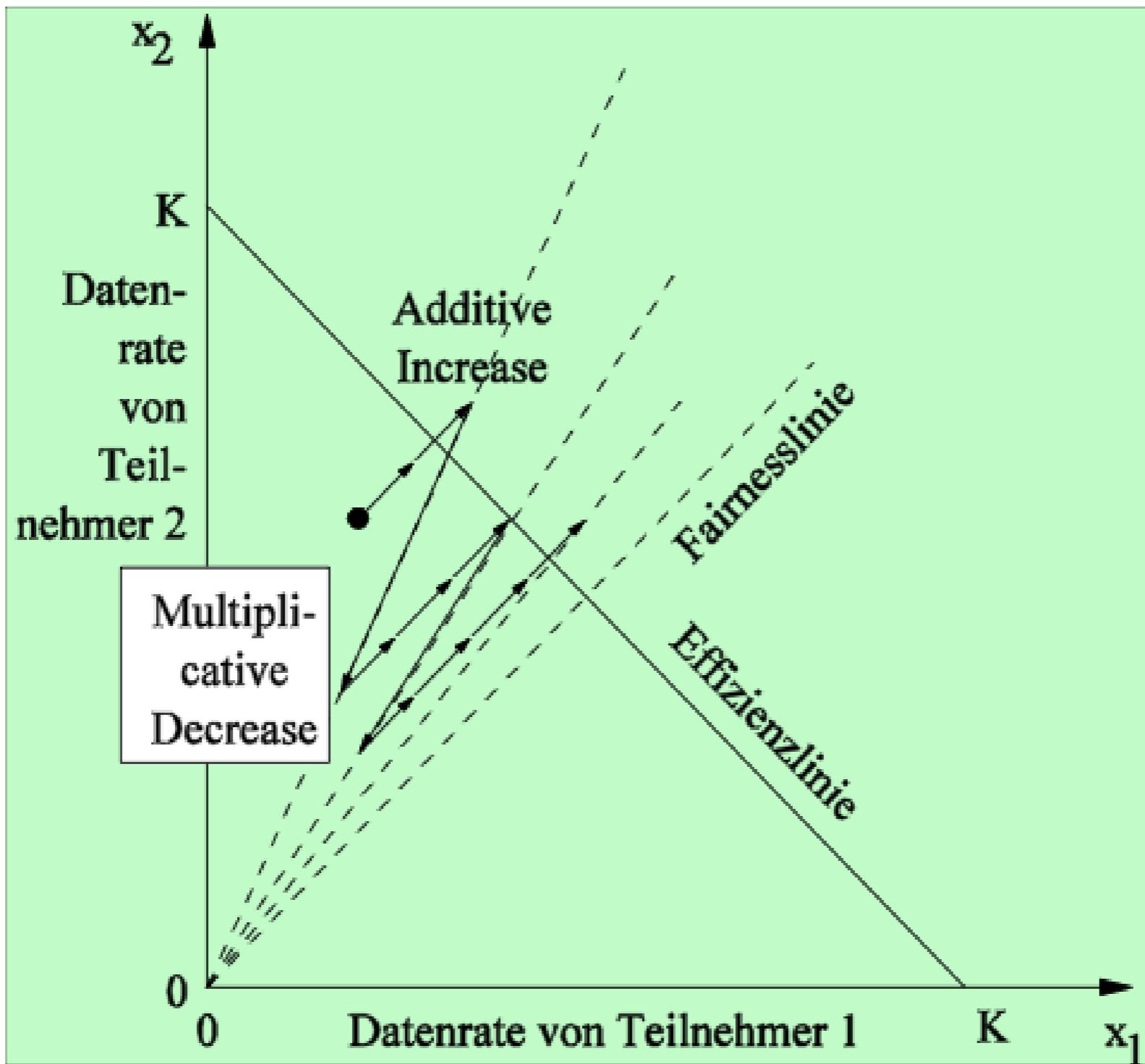
AIAD Additive Increase/ Additive Decrease

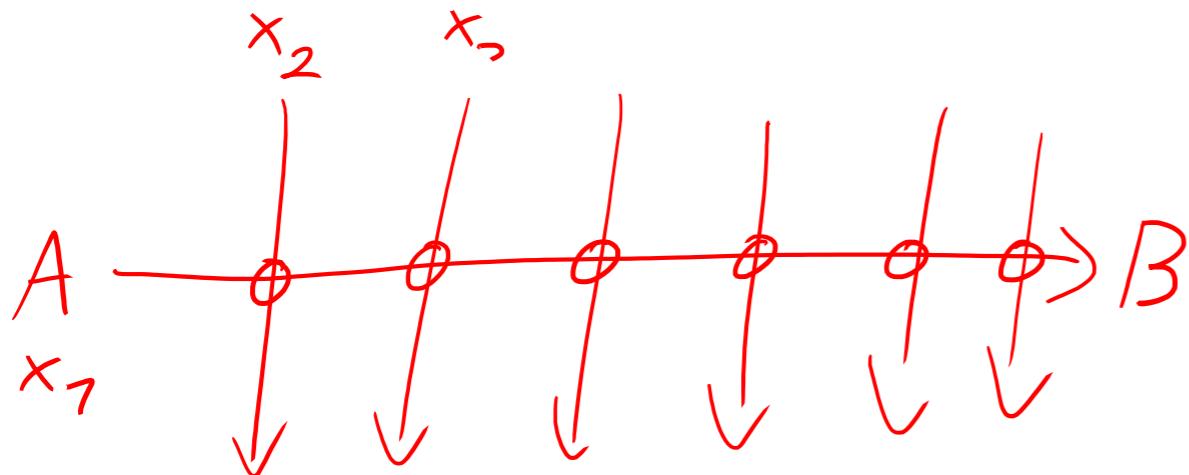


MIMD: Multiplicative Incr./ Multiplicative Decrease



AIMD: Additively Increase/ Multiplicatively Decrease





$$k = 10$$

x_1	x_2
1	8
2	9
1	4
2	5
7	6

x_1	x_3
1	6
2	7
1	8
2	9
7	4

x_1	x_2
1	8
7	9
4	5
5	6
6	7

A bemerkte Problem

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

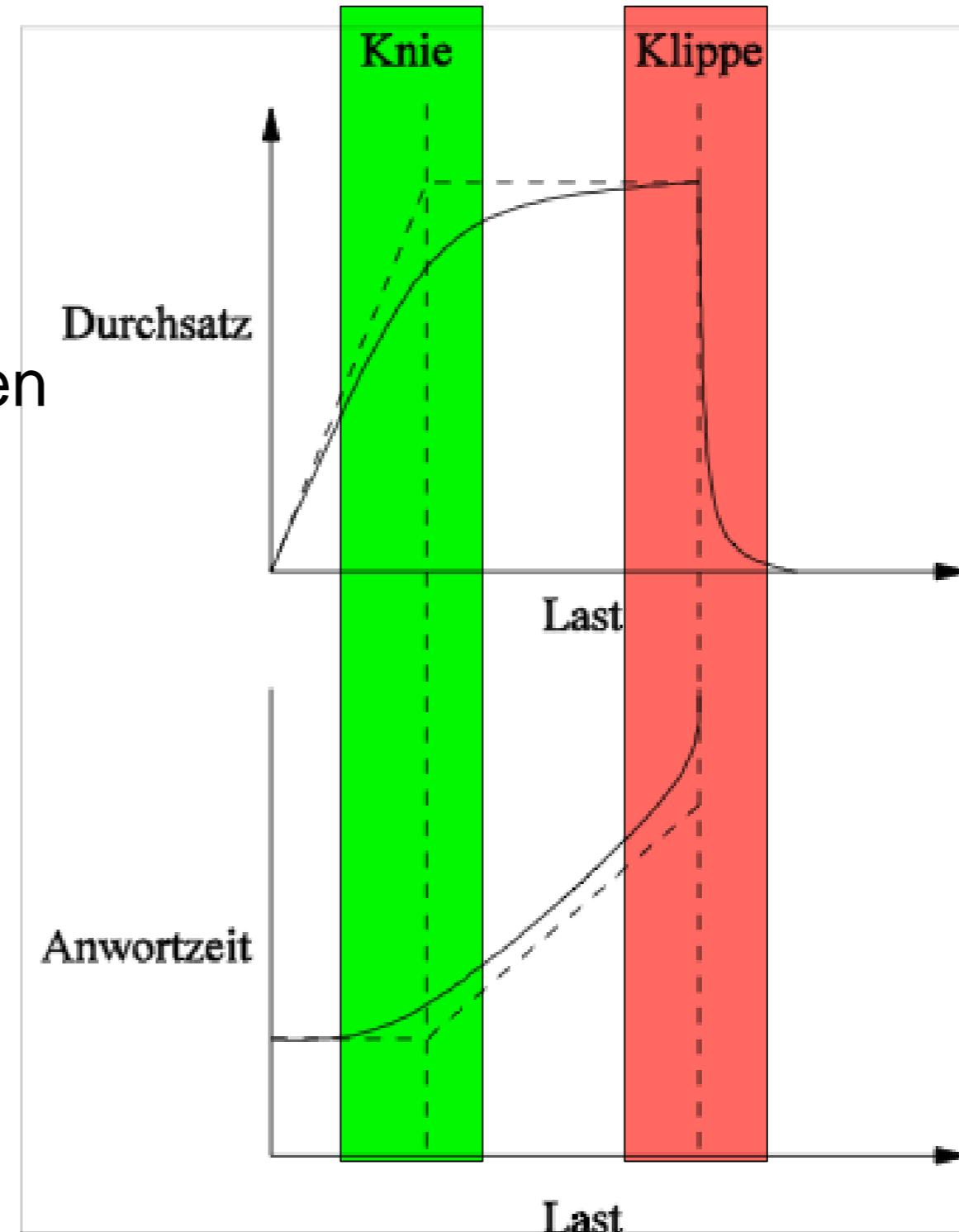
Durchsatz und Antwortzeit

- **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: $Expected = CWND/BaseRTT$

- $Diff = (Expected - Actual) BaseRTT$
- 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. $Actual \ll Expected$)

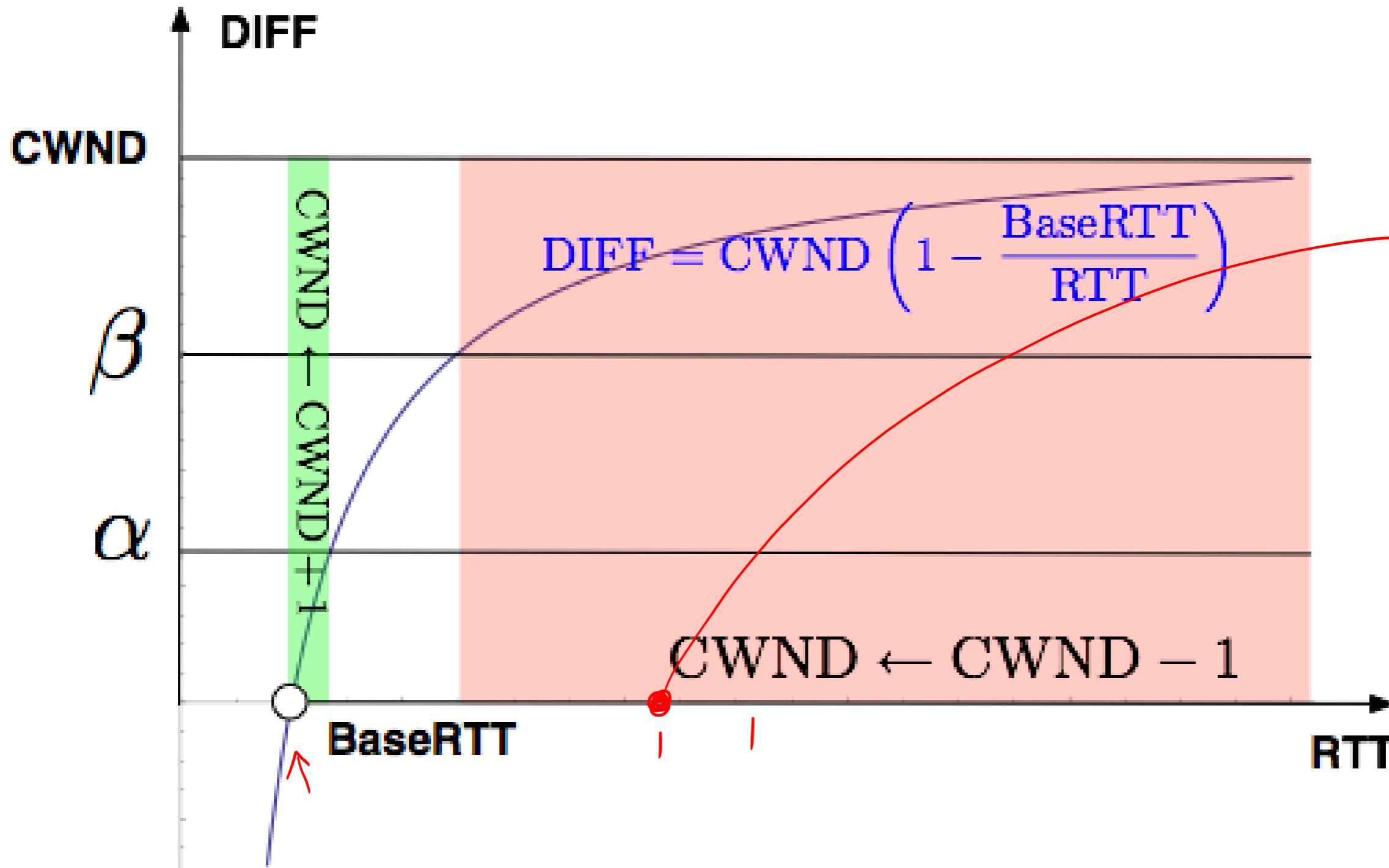
- Last ist zu hoch
- $CWND \leftarrow CWND - 1$

■ Sonst keine Aktion: $CWND \leftarrow CWND$

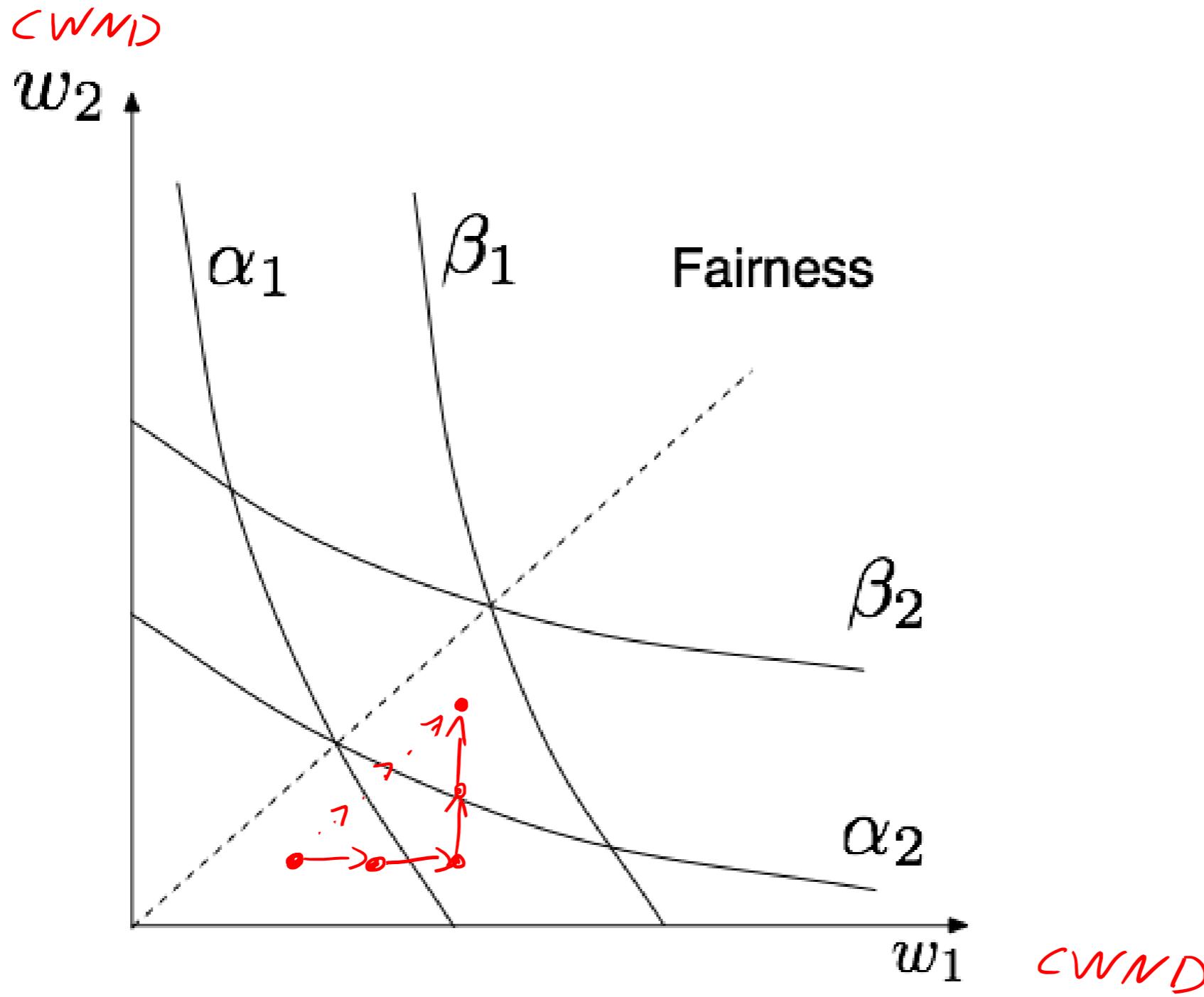


$$\begin{aligned}
 Diff &= \left(\frac{CWND}{BaseRTT} - \frac{CWND}{RTT} \right) BaseRTT \\
 &= CWND \cdot \left(1 - \frac{BaseRTT}{RTT} \right)
 \end{aligned}$$

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

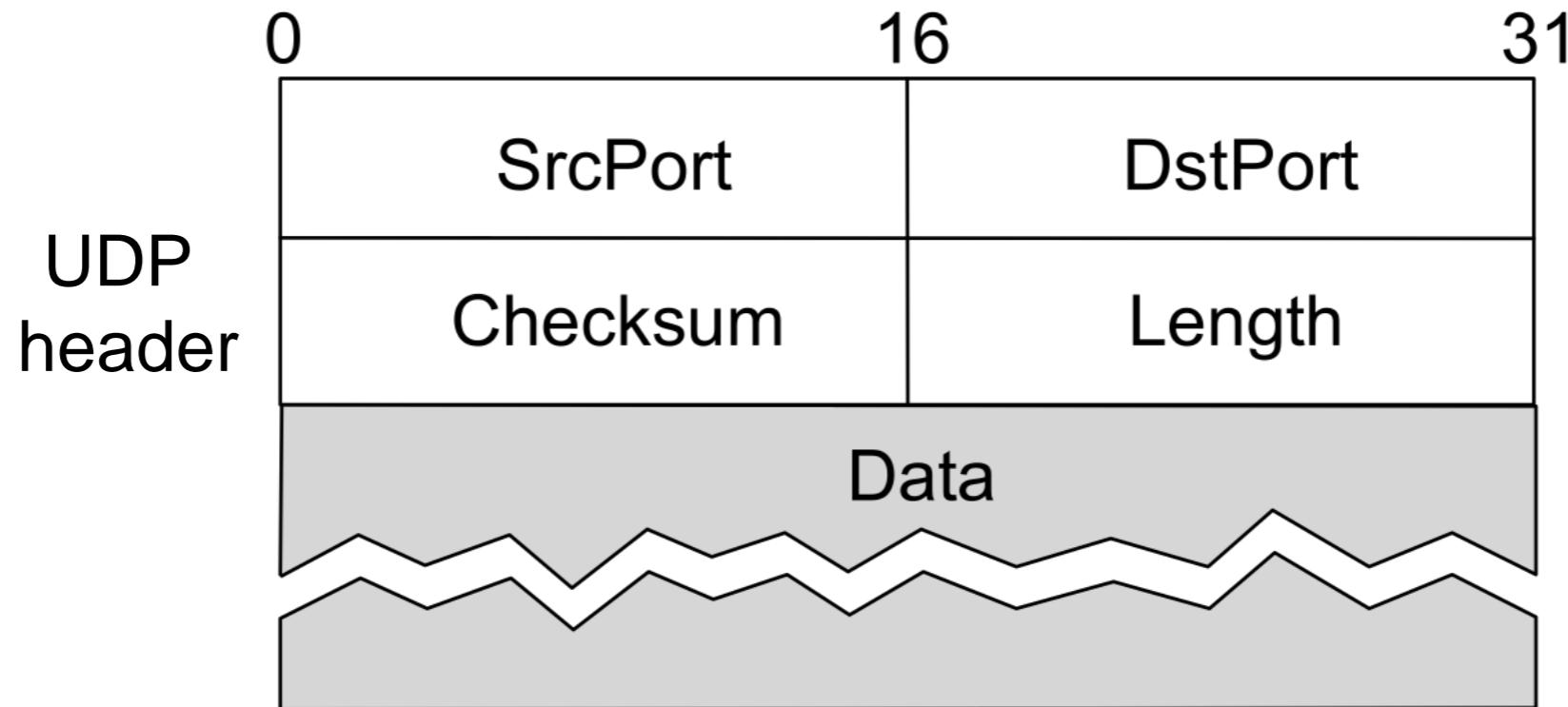
- Reaktion hängt von der Last anderer Transportprotokolle ab
 - z.B. UDP hat keine Congestion Control
- Andere Protokolle können jeder Zeit eingesetzt werden
- UDP und andere Protokoll können TCP Verbindungen unterdrücken

■ 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

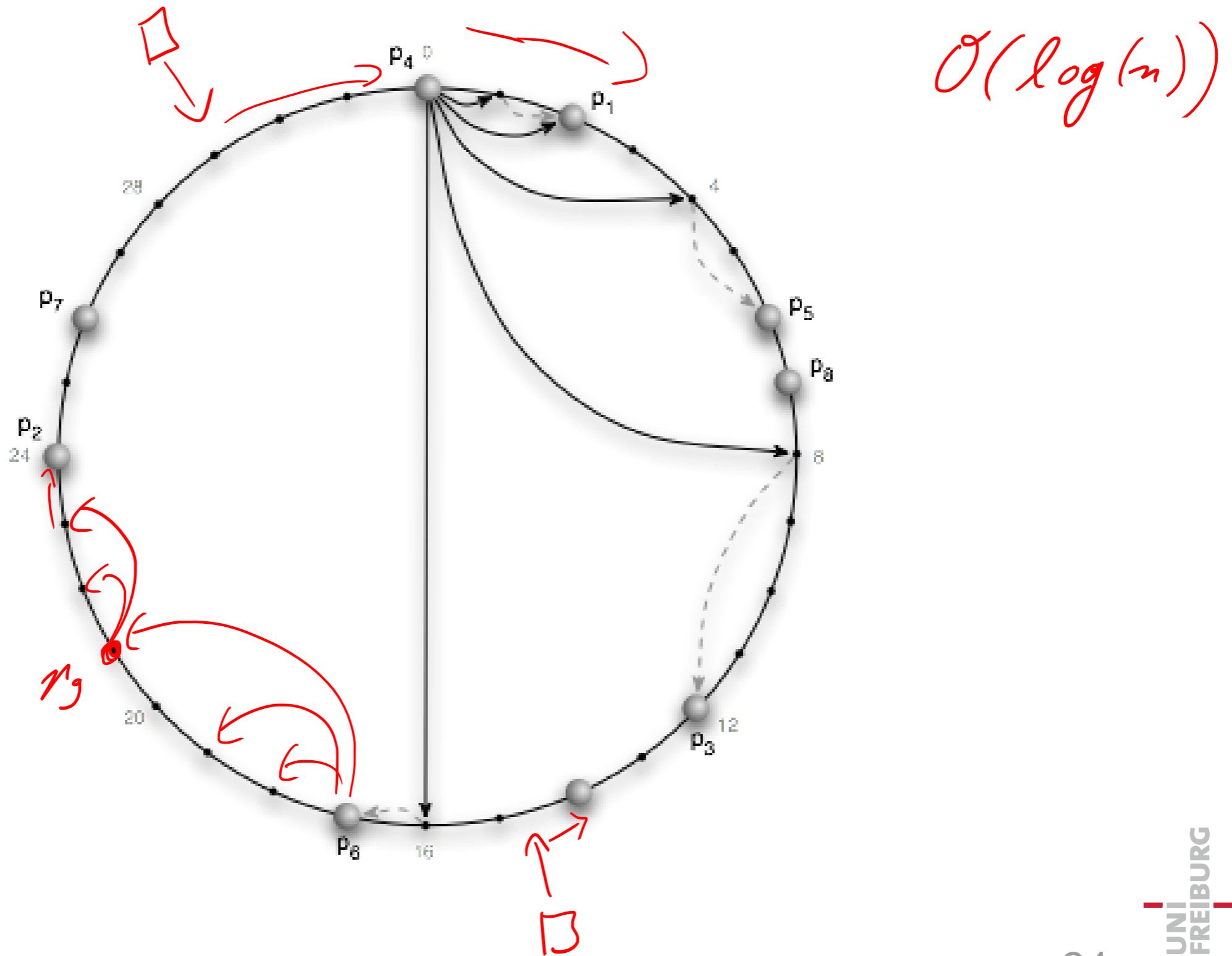
Christian Schindelhauer

Technische Fakultät

Rechnernetze und Telematik

Albert-Ludwigs-Universität Freiburg

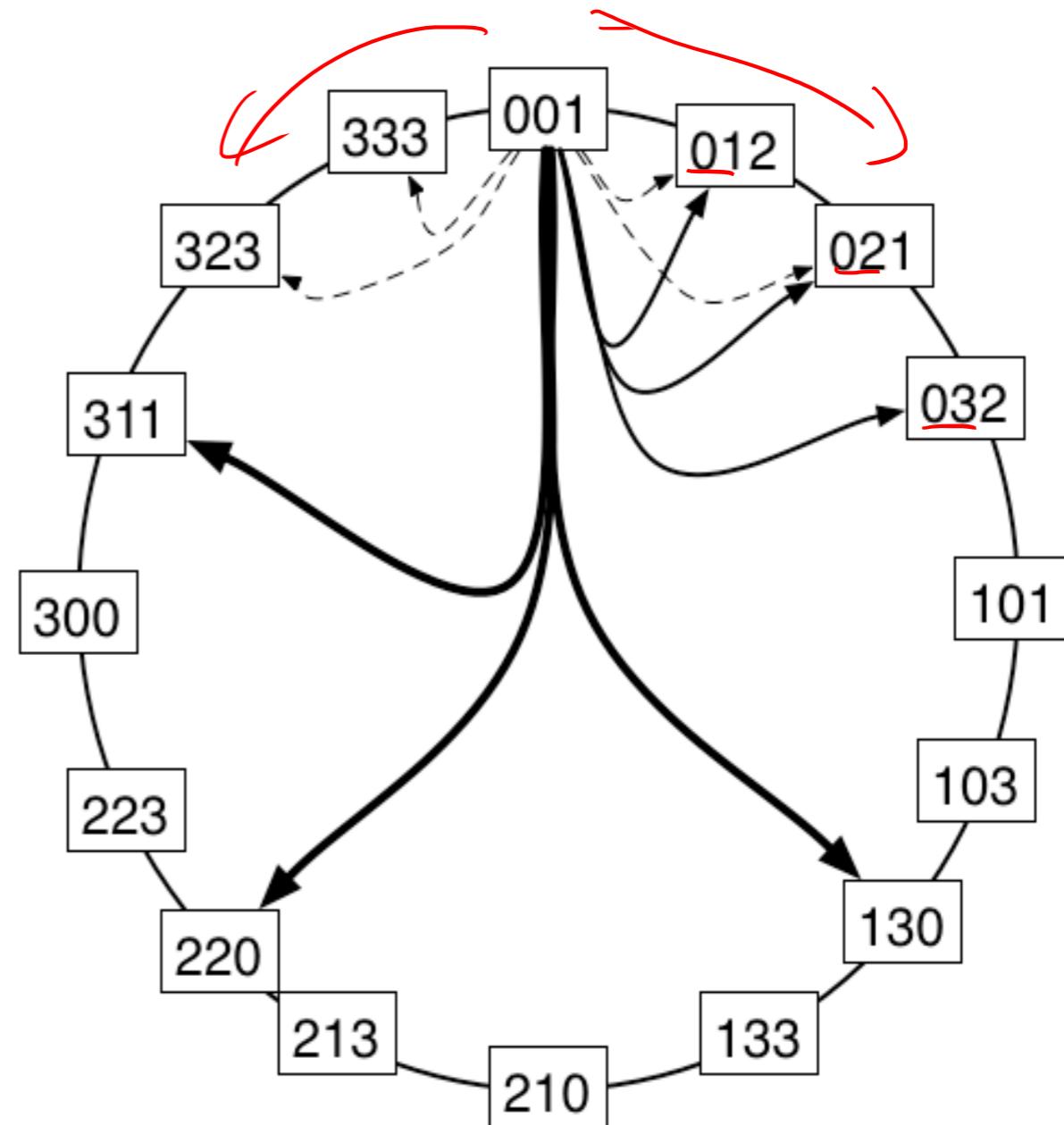
Zeiger-Struktur in Chord



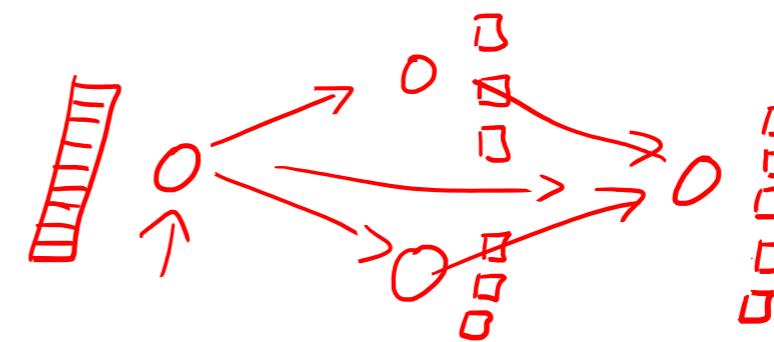
Pastry



- 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