

Systeme II

7. Die Datensicherungsschicht (Teil 5)

Thomas Janson[°], Kristof Van Laerhoven*, Christian
Ortolf[°]

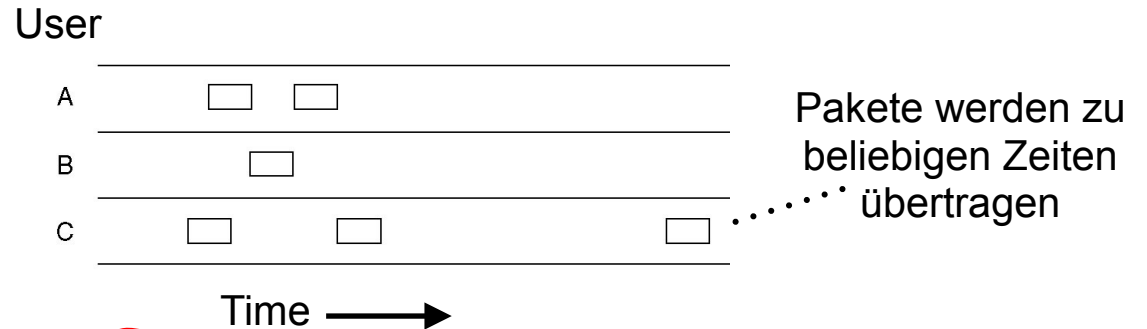
Folien: Christian Schindelbauer[°]

Technische Fakultät

[°]: Rechnernetze und Telematik, *: Eingebettete Systeme

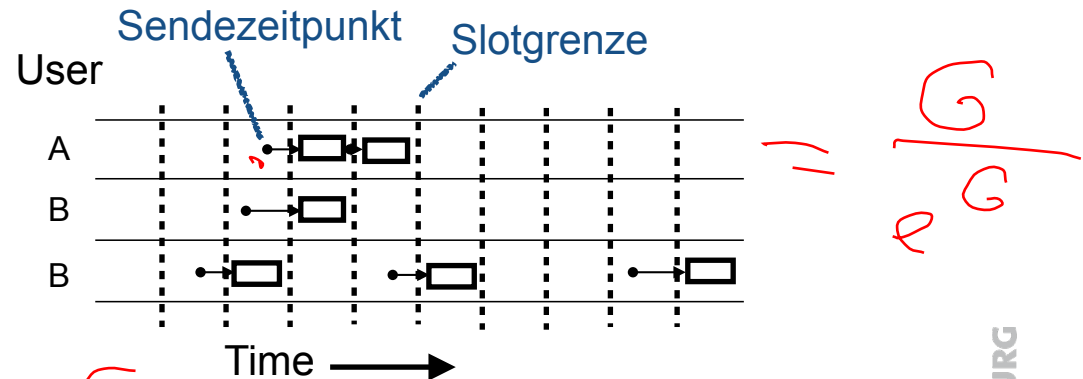
ALOHA Algorithmus

- Sobald ein Paket vorhanden ist, wird es gesendet
- Durchsatz: $S(G) = G \cdot e^{-2G} = G$
- G = erwartete Anzahl Pakete



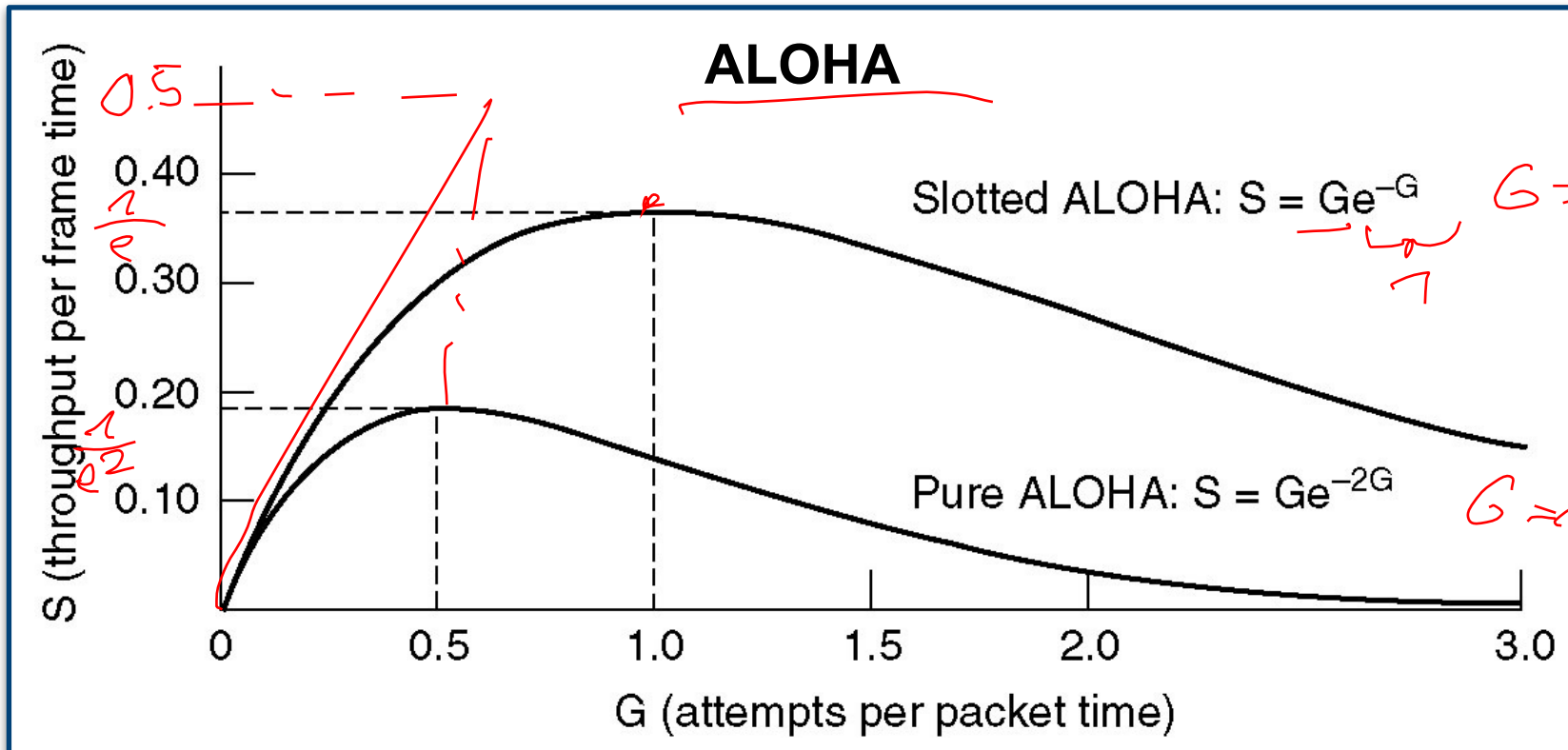
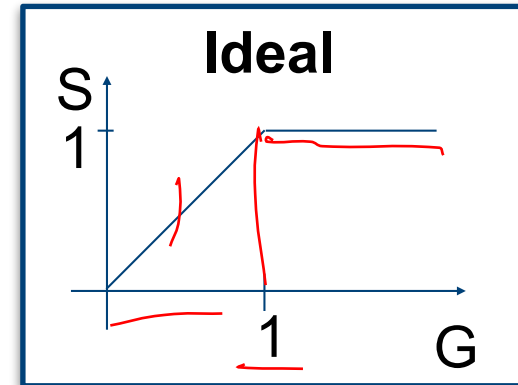
Slotted ALOHA

- Paket wird nur in synchronisierten Slots gesendet
- Verwundbarkeit halbiert
- Durchsatz: $S(G) = G \cdot e^{-G} = G$

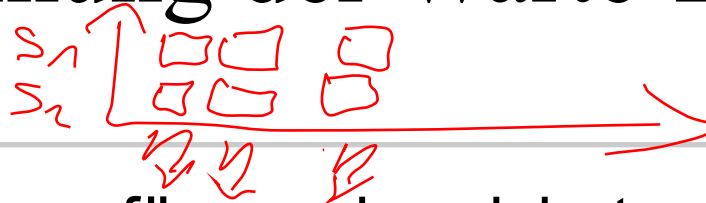


Durchsatz in Abhängigkeit der Last

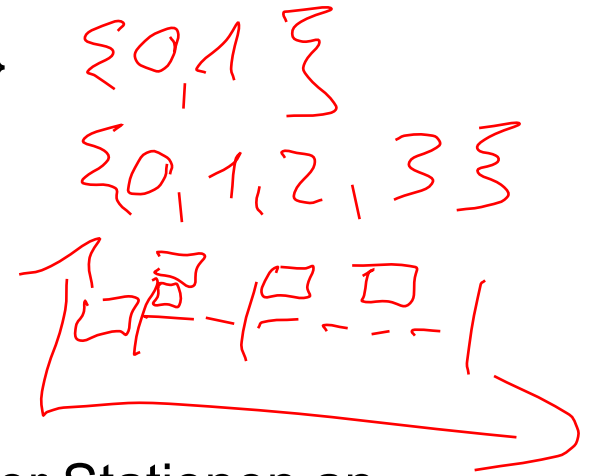
- Durchsatz $S(G)$ für G Versuche Pakete zu schicken pro Zeiteinheit der Paketlänge
- für steigende Last mit steigendem G mehr Kollisionen und Durchsatz bricht zusammen
 - kein gutes Protokoll



Bestimmung der Warte-Zeit



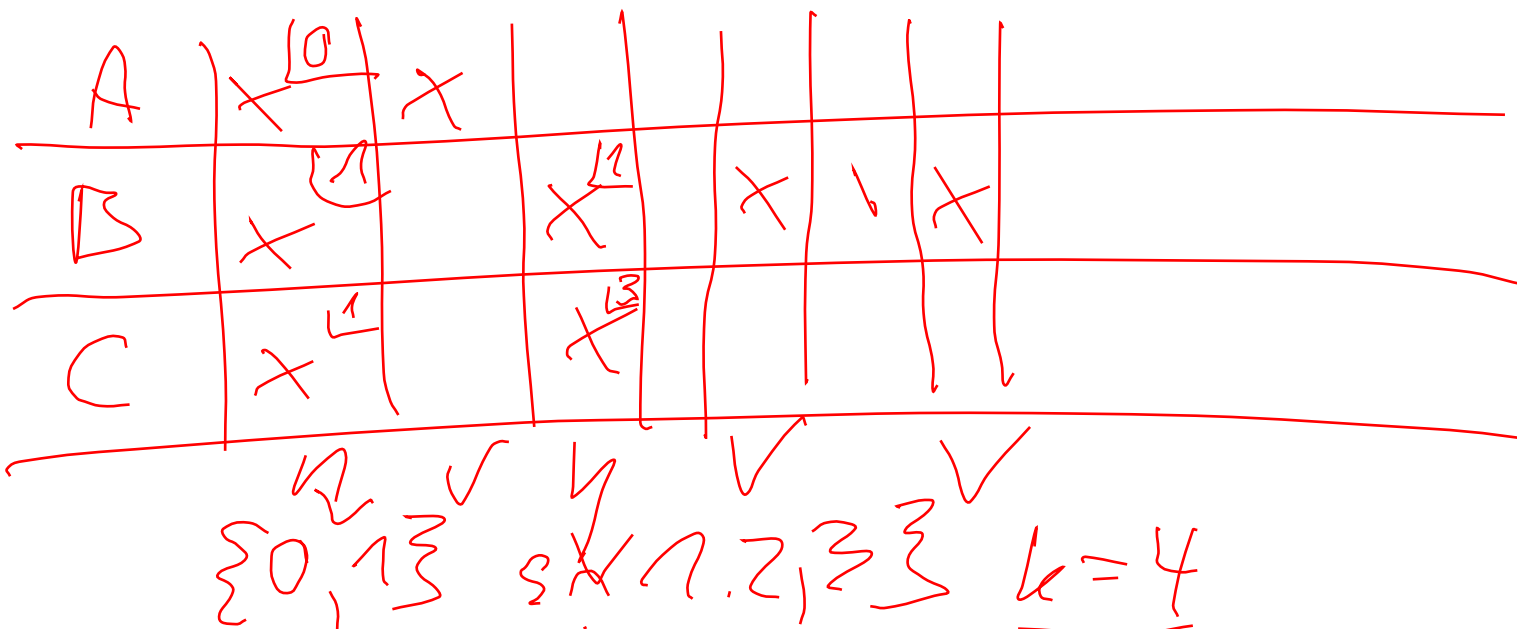
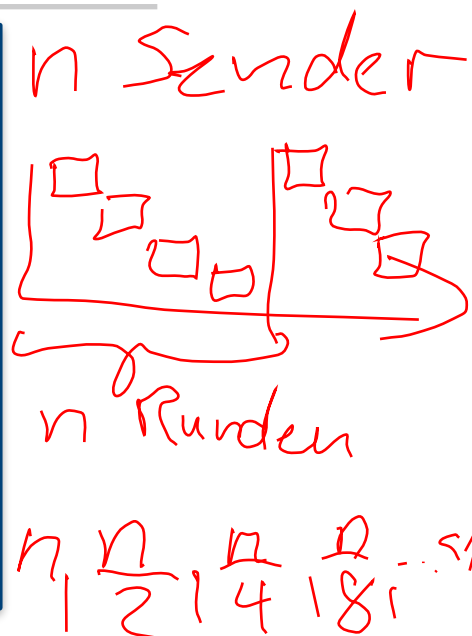
- Kollisionsvermeidung für synchronisiertes Senden in Zeit-Slots
- Algorithmus binary exponential backoff
 - $k := 2$
 - Solange Kollision beim letzten Senden
 - Wähle t gleichwahrscheinlich zufällig aus $\{0, \dots, k-1\}$
 - Warte t Zeit-Slots
 - Sende Nachricht (Abbruch bei Collision Detection)
 - $k := 2 \cdot k$
- Algorithmus
 - passt Wartezeit dynamisch an die Anzahl beteiligter Stationen an
 - sorgt für gleichmäßige Auslastung des Kanals
 - ist fair (auf lange Sicht)



Bestimmung der Warte-Zeit

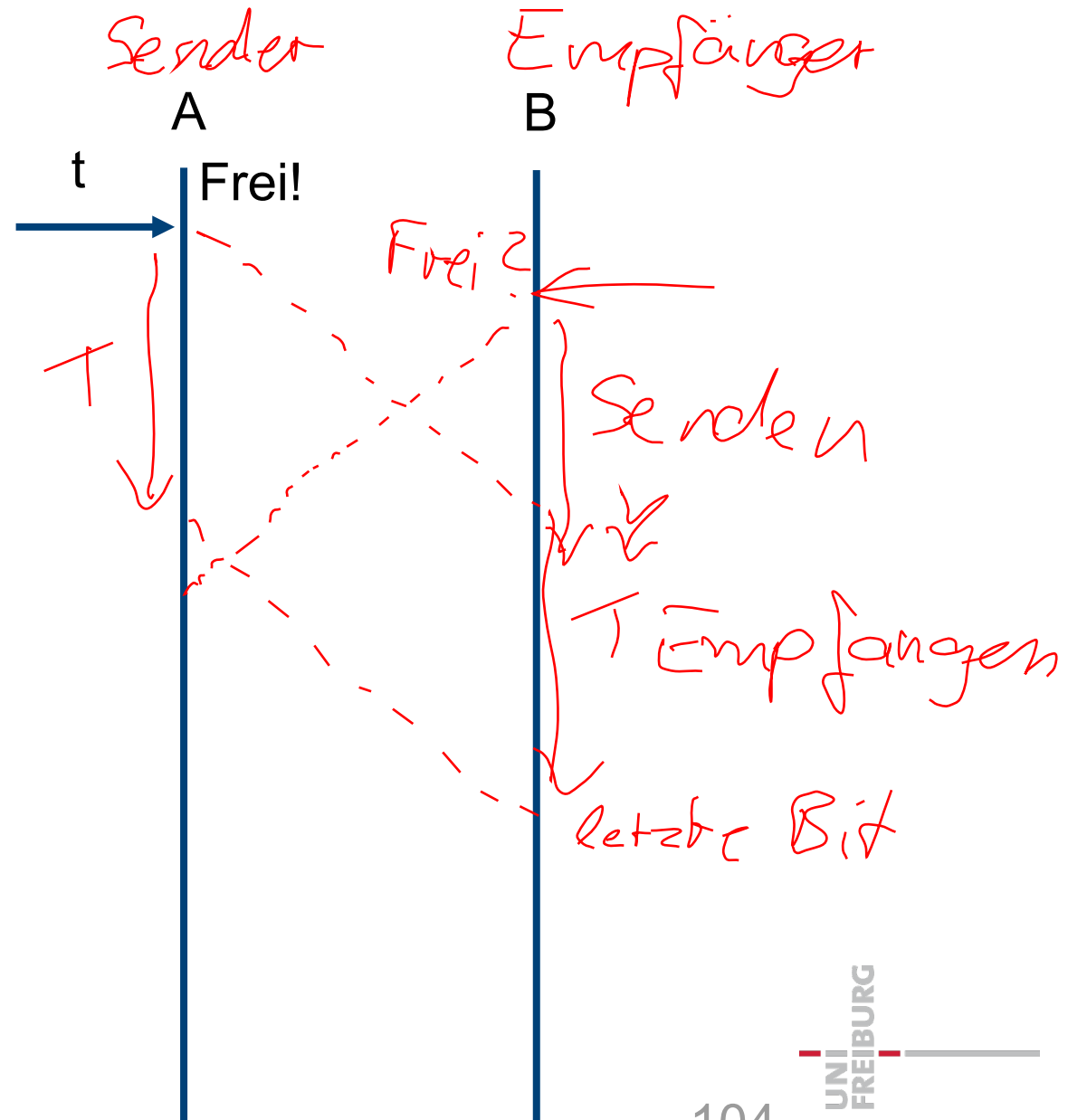
Algorithmus binary exponential backoff

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



$2^i \leq n$
 $\Rightarrow 2^i \geq n$
 $i \geq \log_2 n$

- 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 $t+\epsilon$ mit $\epsilon < d$
 - sehen jeweils einen freien Kanal
- Zweite Station
 - verursacht dann eine Kollision



- Carrier Sense Multiple Access:

- Erst senden wenn der Kanal frei ist

- CSMA-Problem:

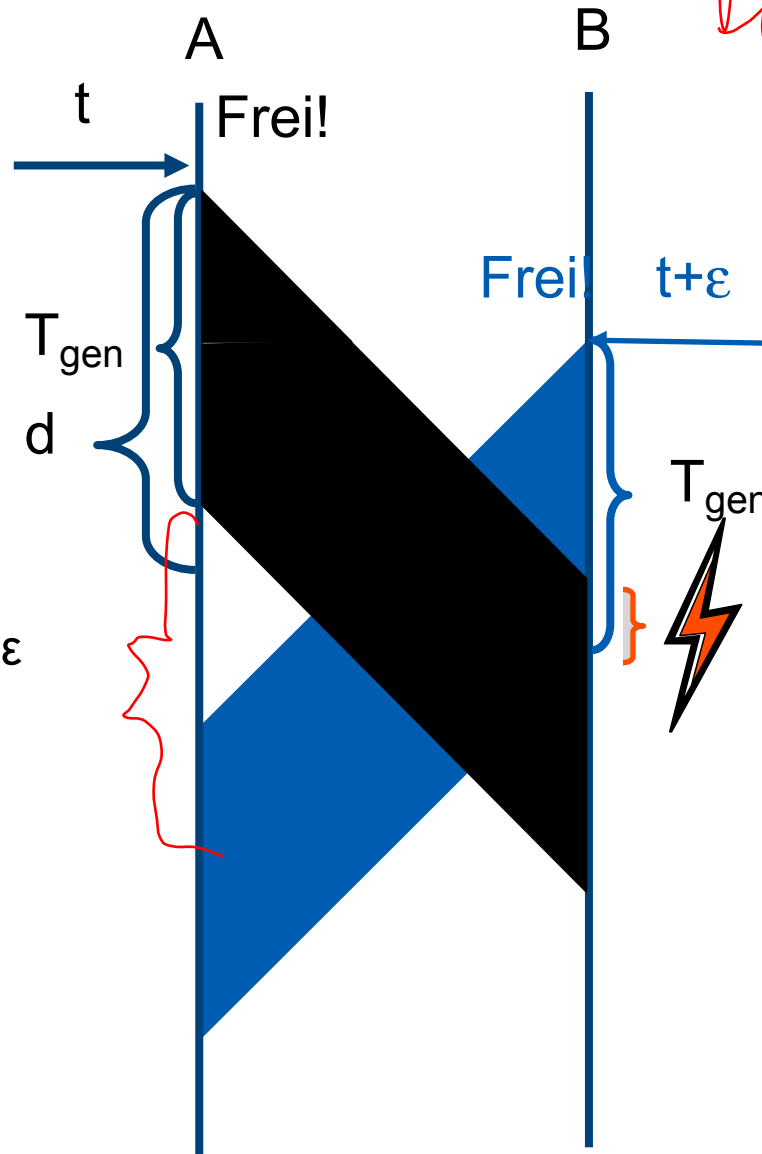
- Übertragungszeit d (propagation delay)

- Zwei Stationen

- starten Senden zu den Zeitpunkten t und $t+\epsilon$ mit $\epsilon < d$
- sehen jeweils einen freien Kanal

- Zweite Station

- verursacht dann eine Kollision



Datenrate 10 MBit/s
 10^7 Bit/s

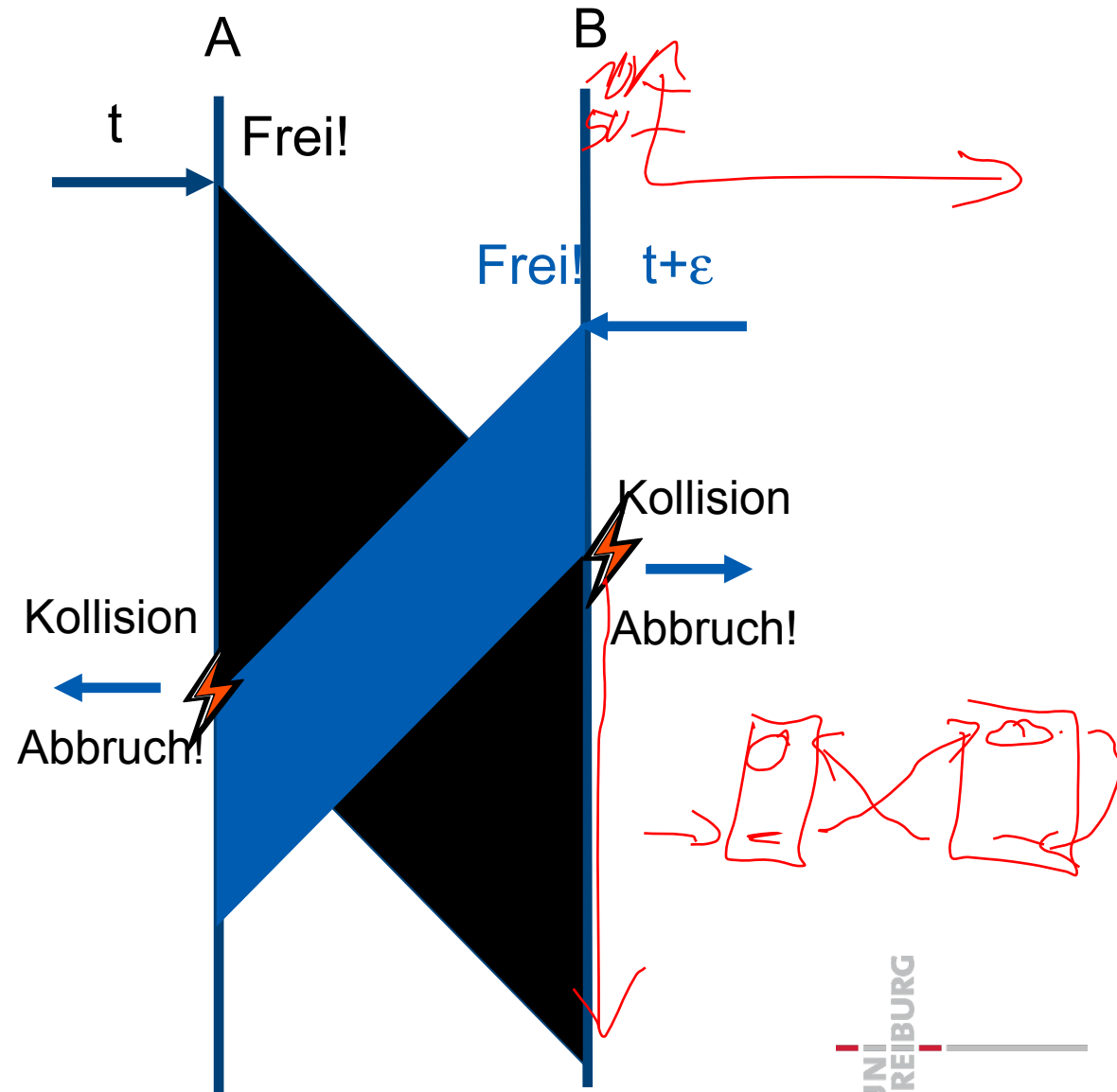
Frame 100 Bit

$$T_{Gen} = \frac{10^2}{10^7} = 10^{-5} s = 10 \mu s$$

$$c = 3 \cdot 10^8 \text{ m/s}$$

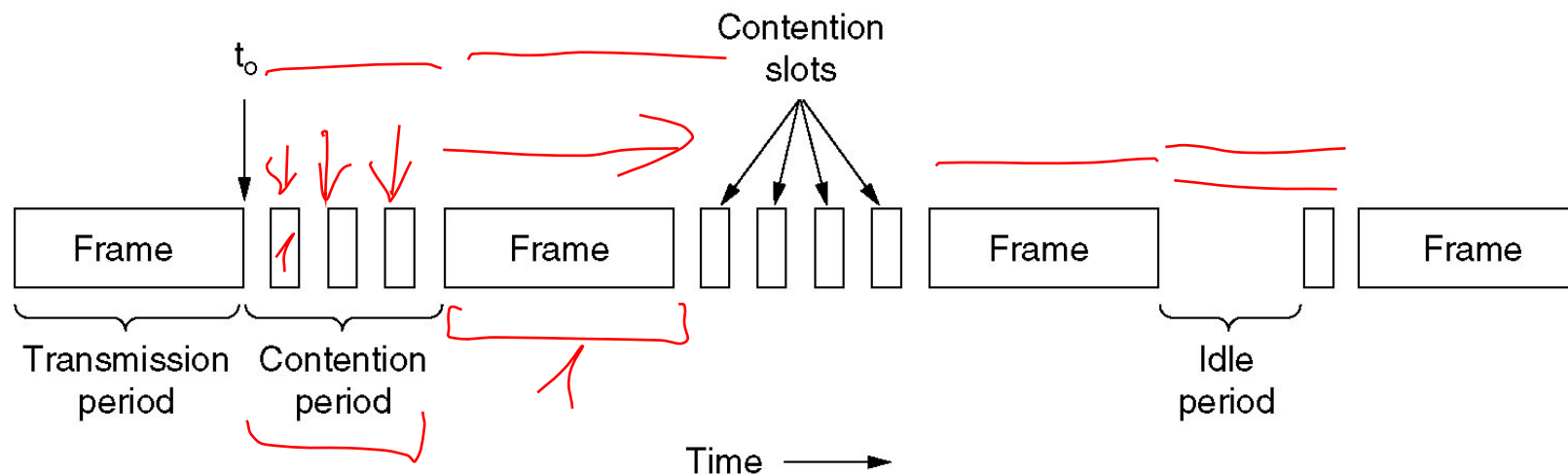
$$s = c \cdot t = 3 \text{ km}$$

- Falls Kollisionserkennung (collision detection) möglich ist,
 - dann beendet der spätere Sender seine Übertragung
 - Zeitverschwendung 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?



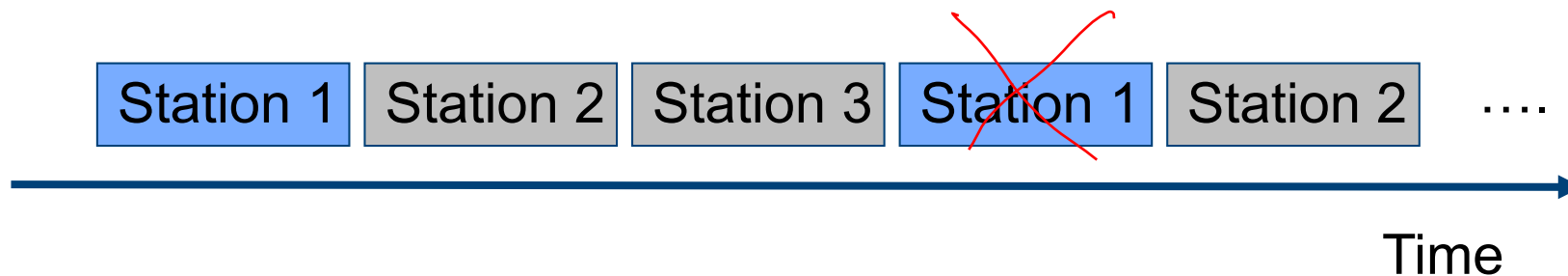
- Wettbewerbsphase (Contention Period)
 - Kollisionen entstehen, Übertragungen werden abgebrochen
- Übertragungsphase (Transmission Period)
 - Keine Kollision, effektiver Teil des Protokolls
- Leer-Phase (IDLE)
 - Keine Station sendet einen Frame

! Es gibt nur Wettbewerbs-, Übertragungsphasen und Leer-Phasen

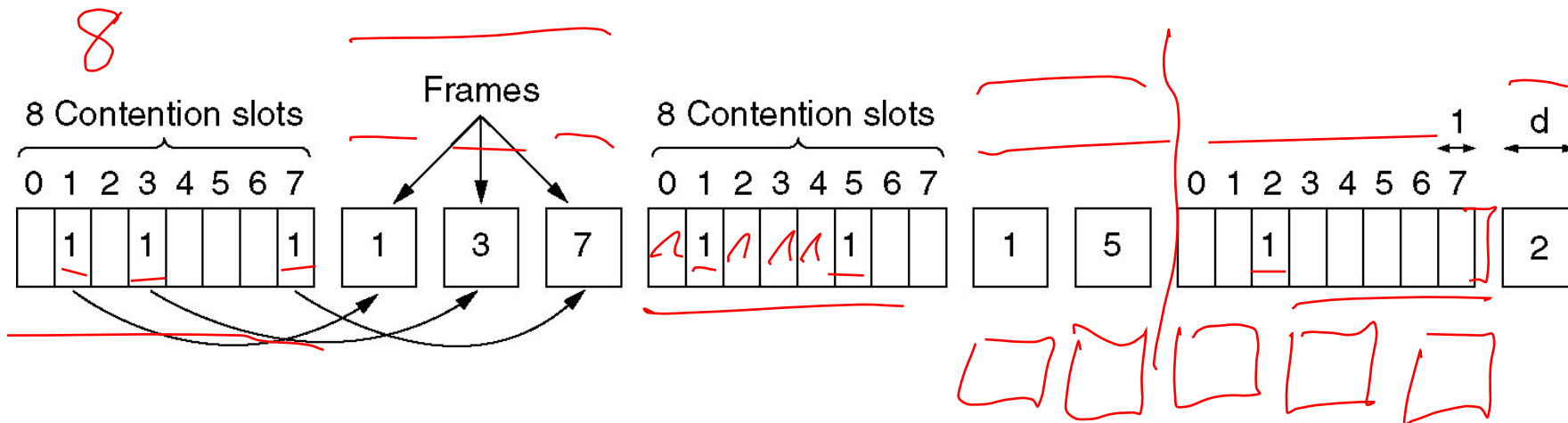


- Statisches Multiplexen
- Dynamische Kanalbelegung
 - Kollisionsbasierte Protokolle
 - Kollisionsfreie Protokolle (contention-free)
 - Protokolle mit beschränktem Wettbewerb (limited contention)

- 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?



- 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



- 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!

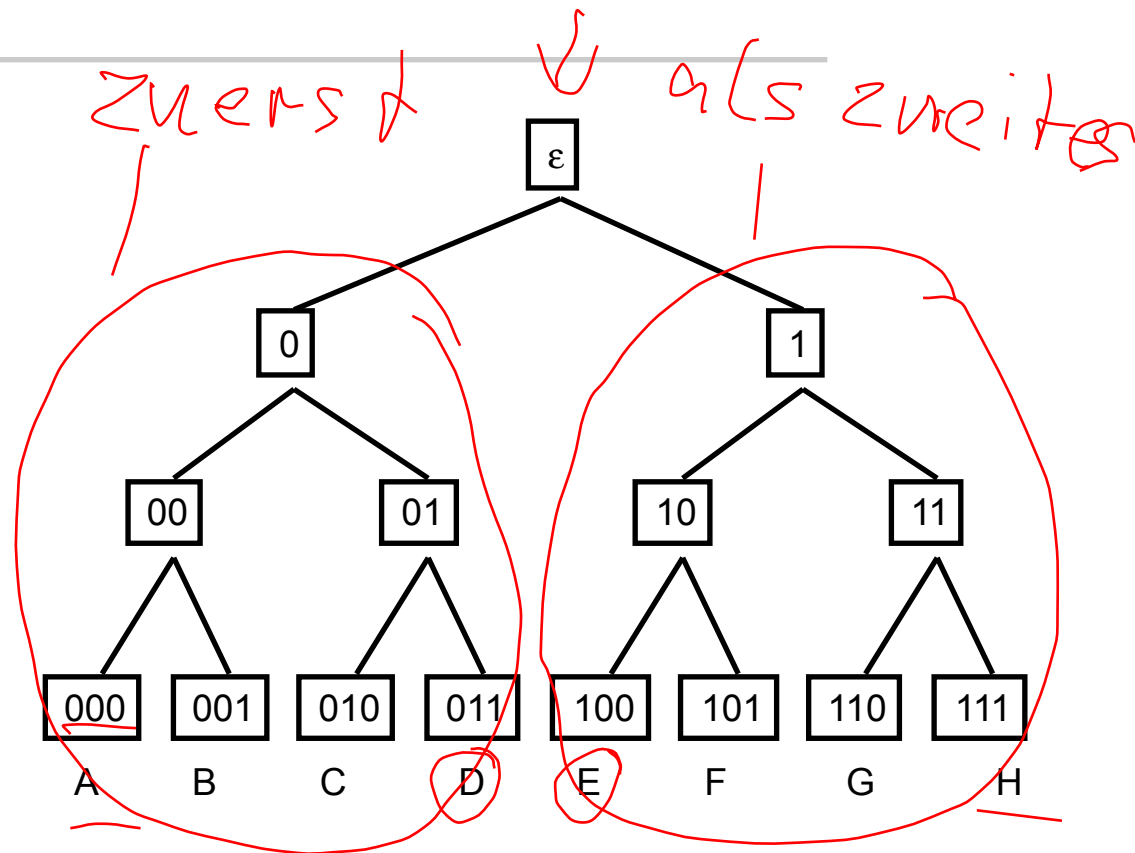
- Statisches Multiplexen
- Dynamische Kanalbelegung
 - Kollisionsbasierte Protokolle
 - Kollisionsfreie Protokolle (contention-free)
 - Protokolle mit beschränktem Wettbewerb (limited contention)

- 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

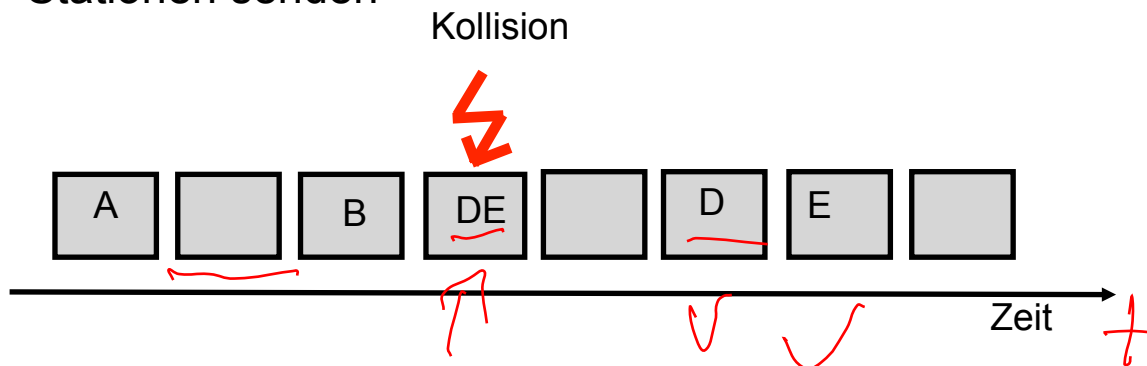
Adaptives Baumprotokoll

Voraussetzung

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

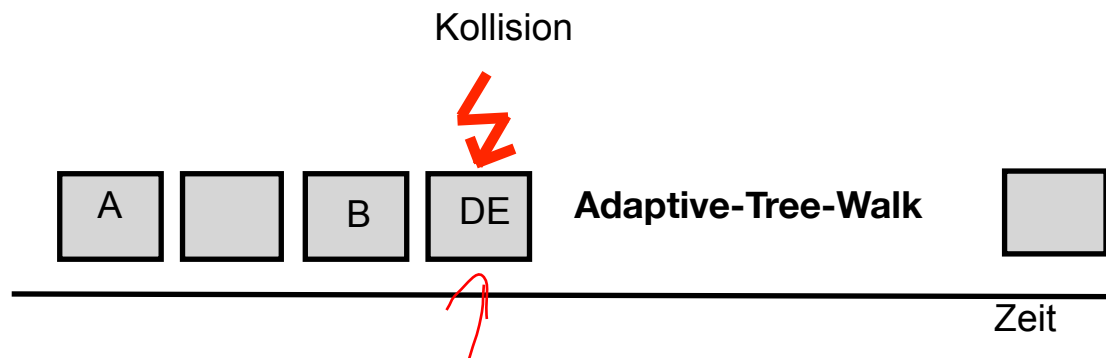


Stationen



■ Basis-Algorithmus

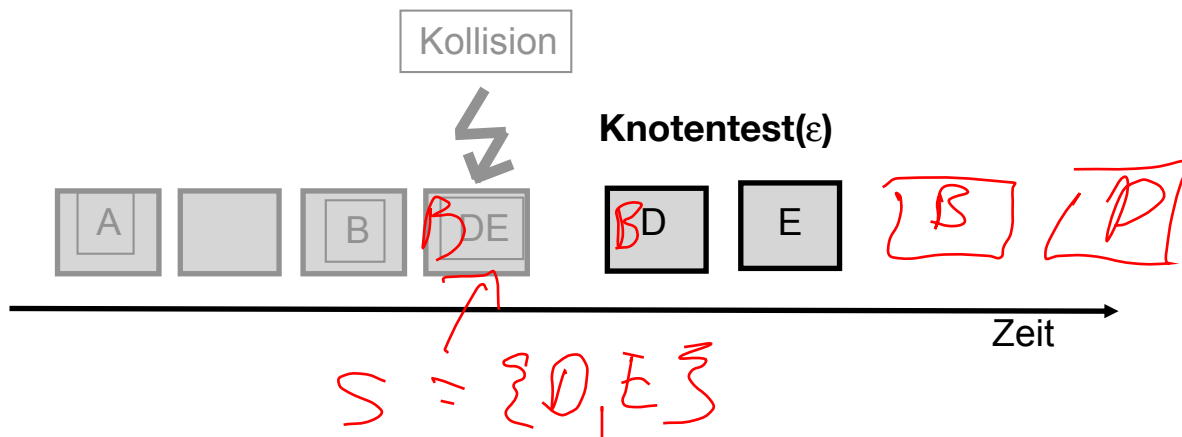
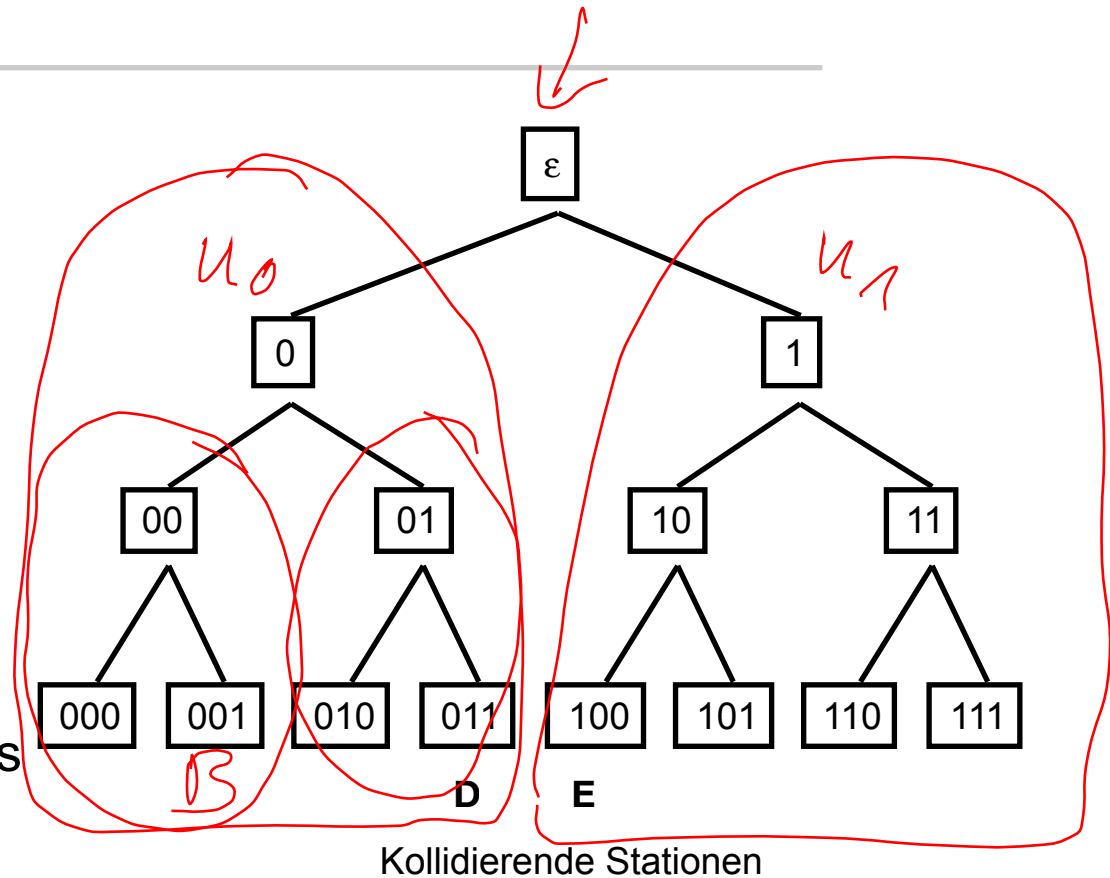
- Jeder Algorithmus sendet sofort (slotted Aloha)
- Falls eine Kollision auftritt,
 - akzeptiert keine Station mehr neue Paket aus der Vermittlungsschicht
 - Führe Adaptive-Tree-Walk(ϵ) aus



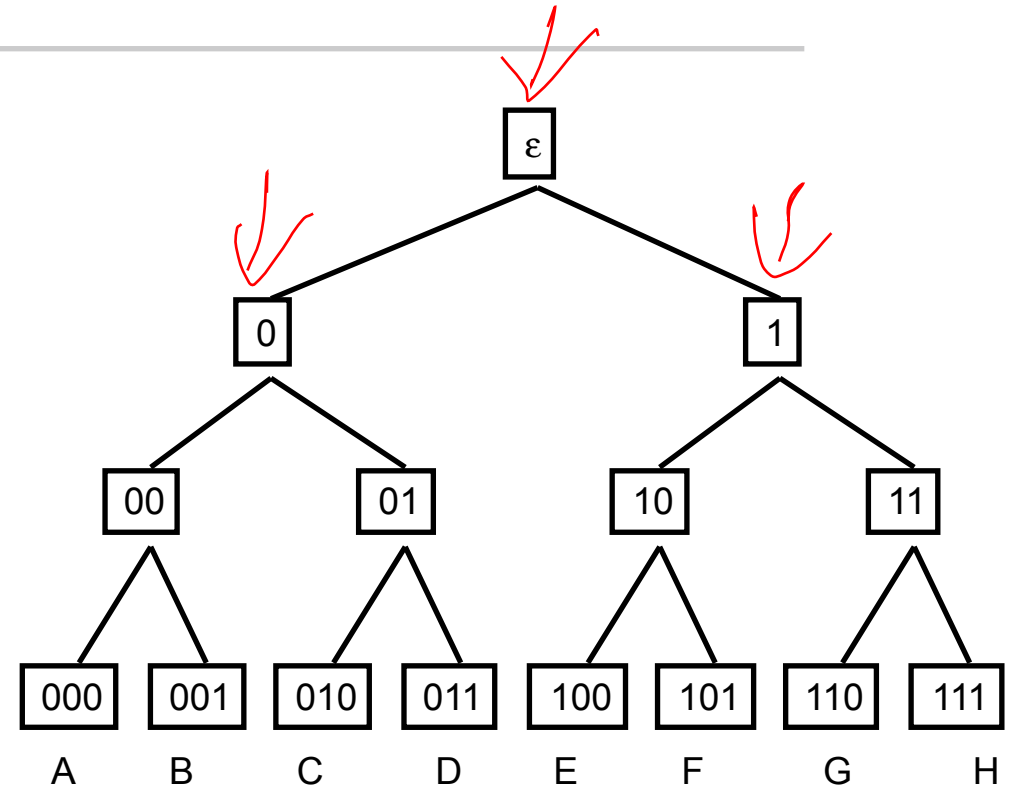
Adaptives Baumprotokoll

Knoten-Test

- Algorithmus Knoten-Test
 - für Knoten u des Baums und
 - kollidierende Menge S von Station
- 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

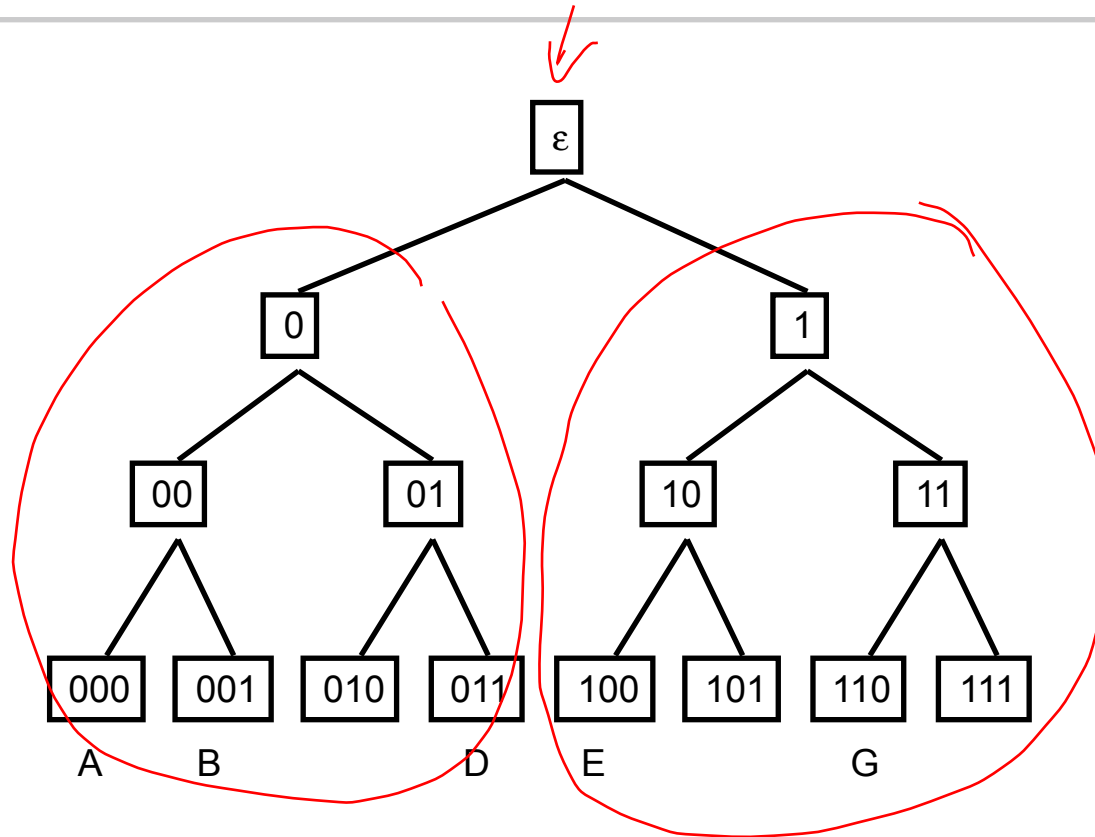


- Algorithmus Knoten-Test
 - für Knoten u des Baums und
 - kollidierende Menge S von Station
- Knoten-Test(u)
 - Betrachte zwei Slots pro Knoten des Baums
 - Im ersten Slot senden alle Knoten aus S , die mit ID $u0$ anfangen
 - Im zweiten Slot senden alle Knoten aus S , die mit ID $u1$ anfangen
- Adaptive Tree Walk(x) ^{ϵ}
 - Führe Knoten-Test(x) aus
 - Falls Kollision im ersten Slot,
 - führe Adaptive-Tree-Walk($x0$) aus
 - Falls Kollision im zweiten Slot,
 - Führe Adaptive-Tree-Walk($x1$) aus



Stationen

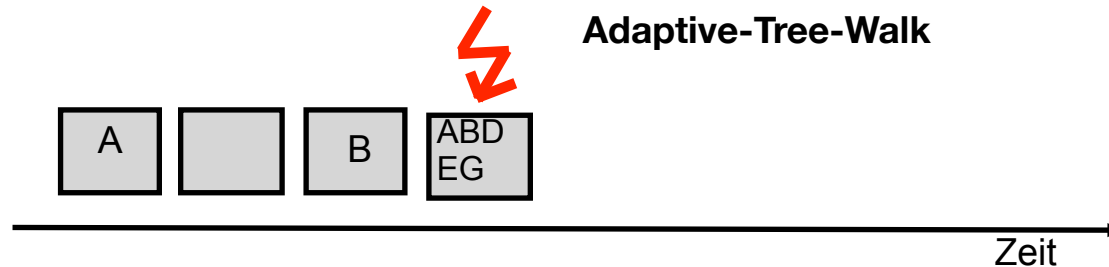
Adaptives Baumprotokoll Beispiel (1)



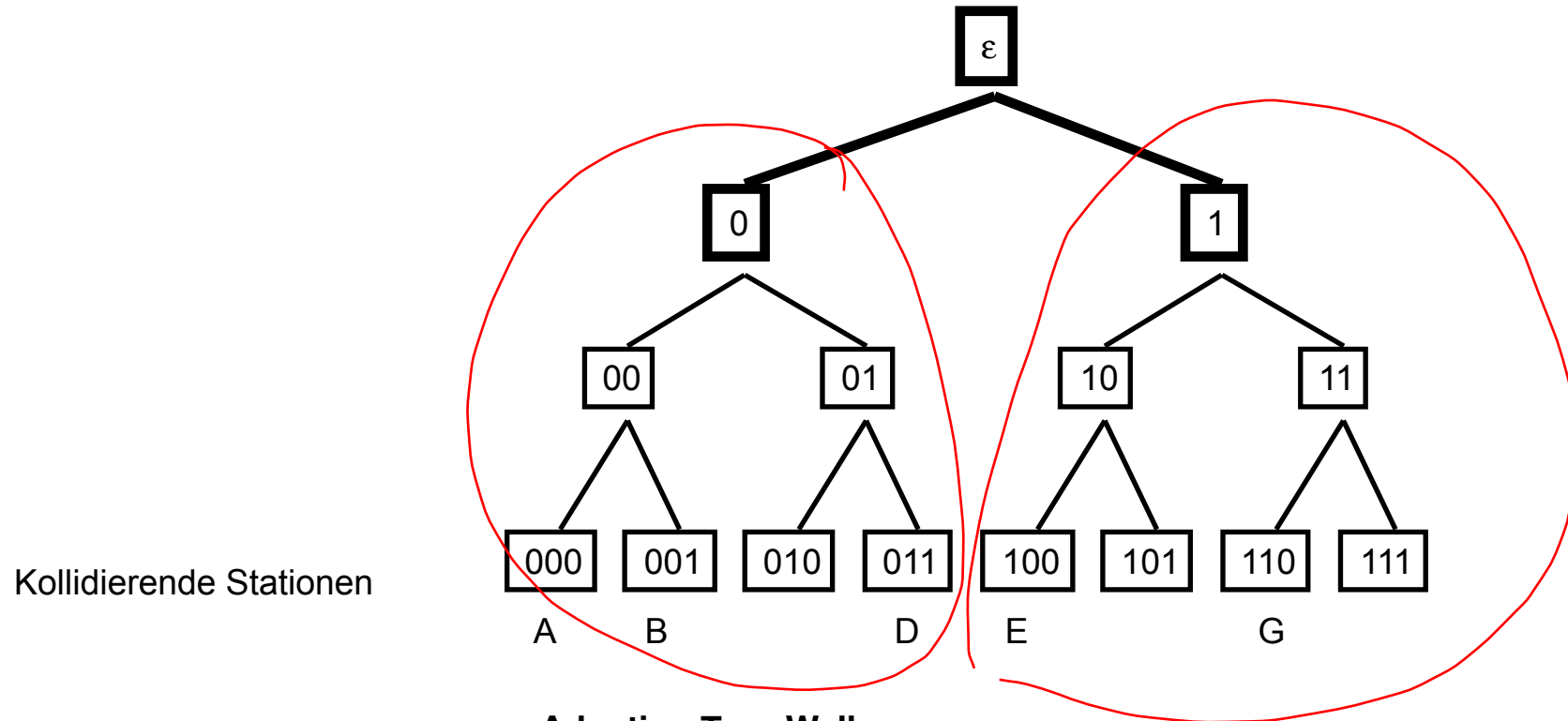
Kollidierende Stationen

Kollision

Adaptive-Tree-Walk



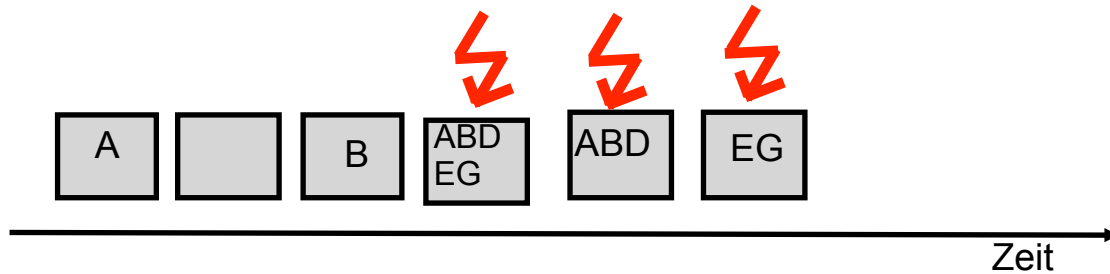
Adaptives Baumprotokoll Beispiel (2)



Kollidierende Stationen

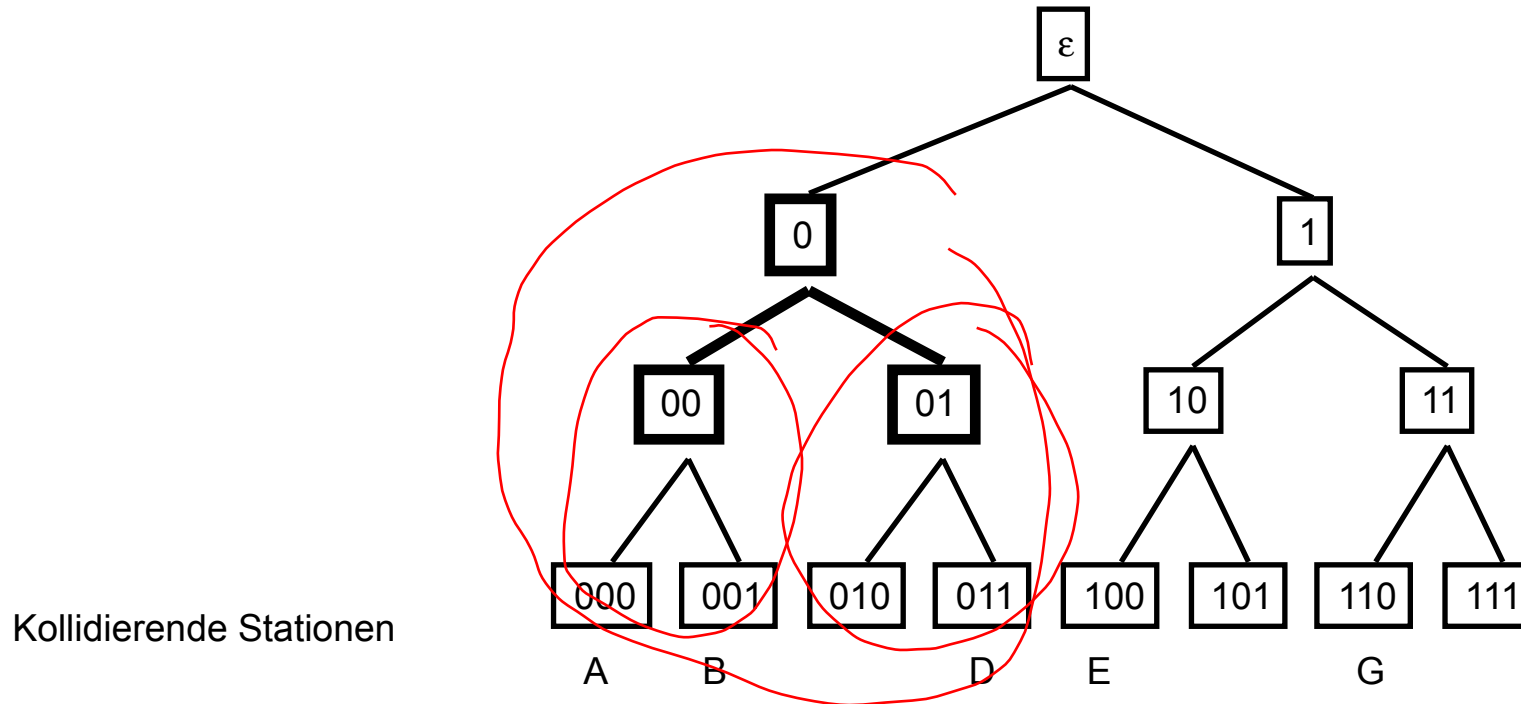
Adaptive-Tree-Walk

Knotentest(ϵ)

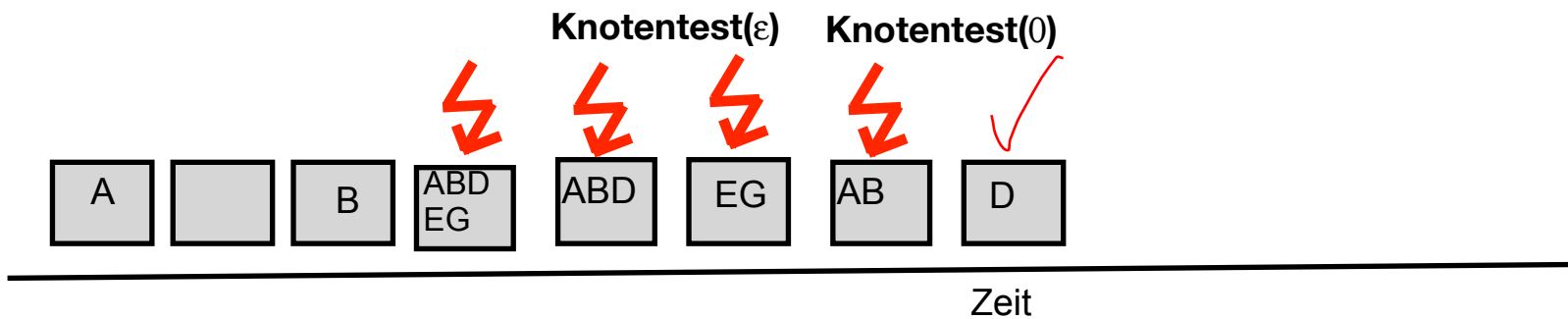


Adaptives Baumprotokoll

Beispiel (3)

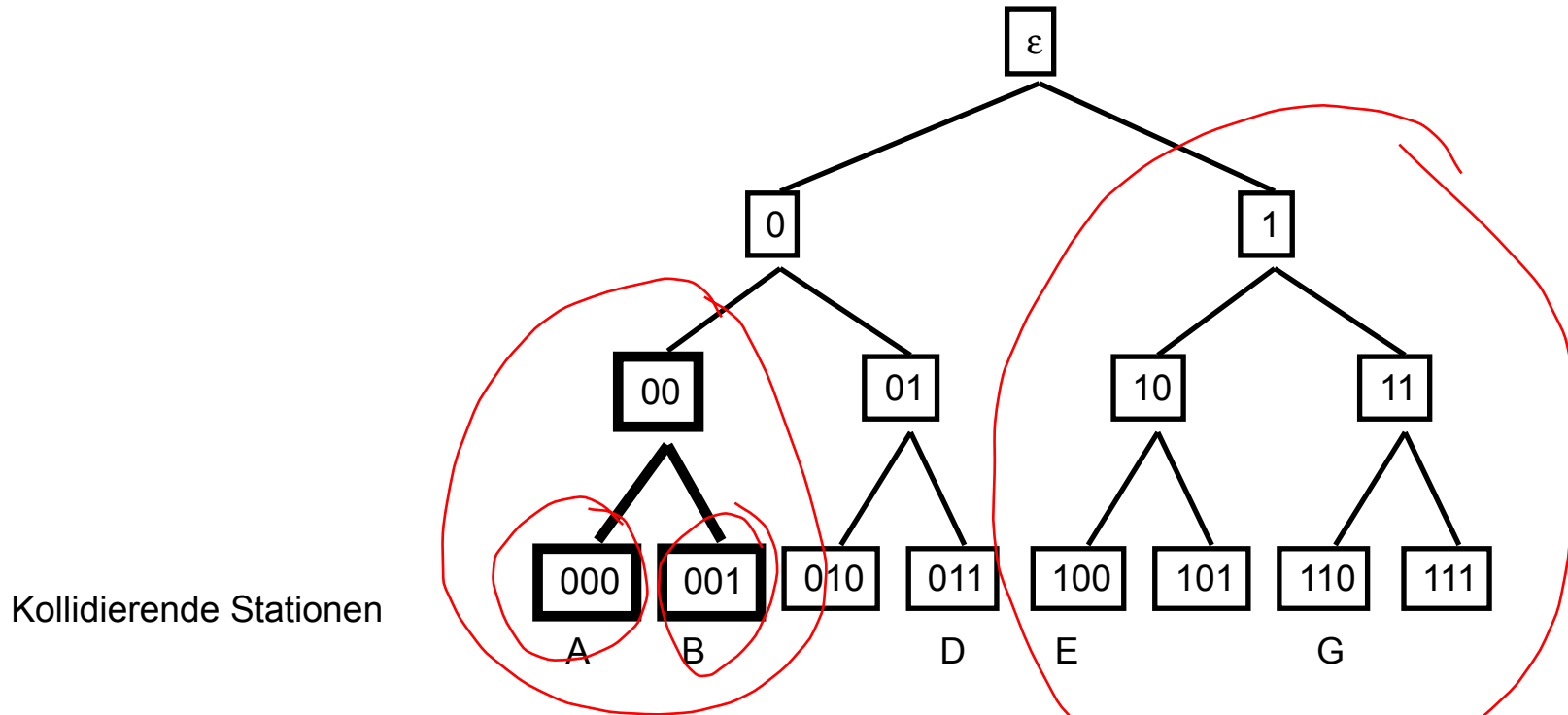


Adaptive-Tree-Walk



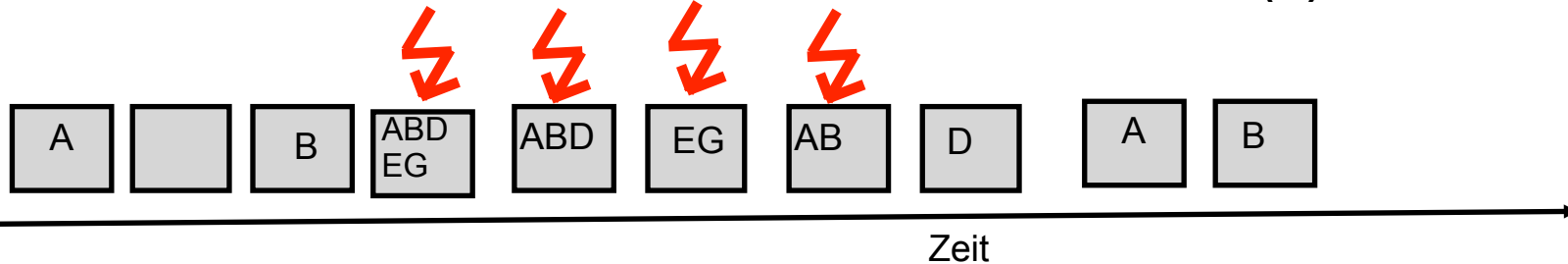
Adaptives Baumprotokoll

Beispiel (4)



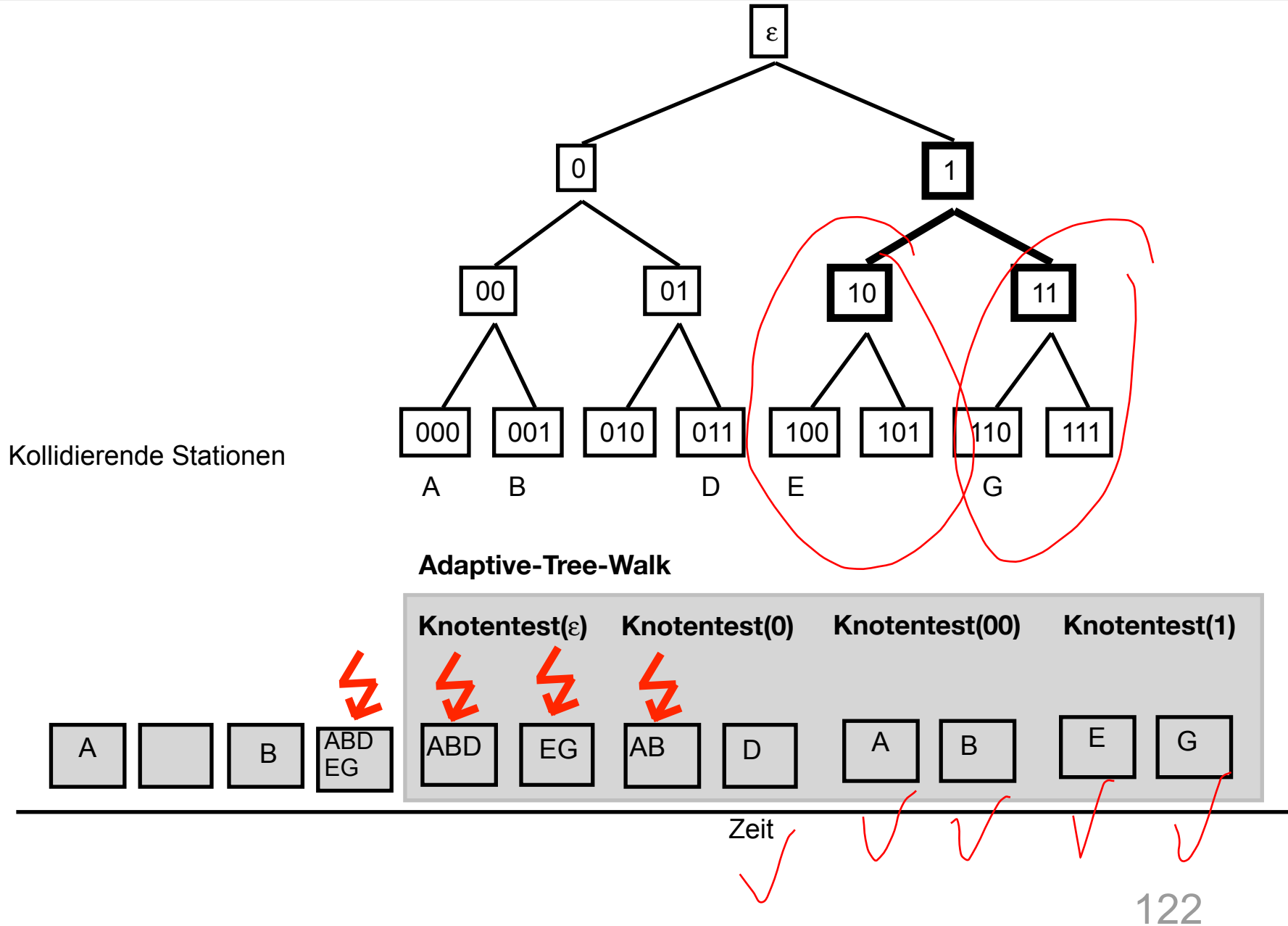
Adaptive-Tree-Walk

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



Adaptives Baumprotokoll

Beispiel (5)

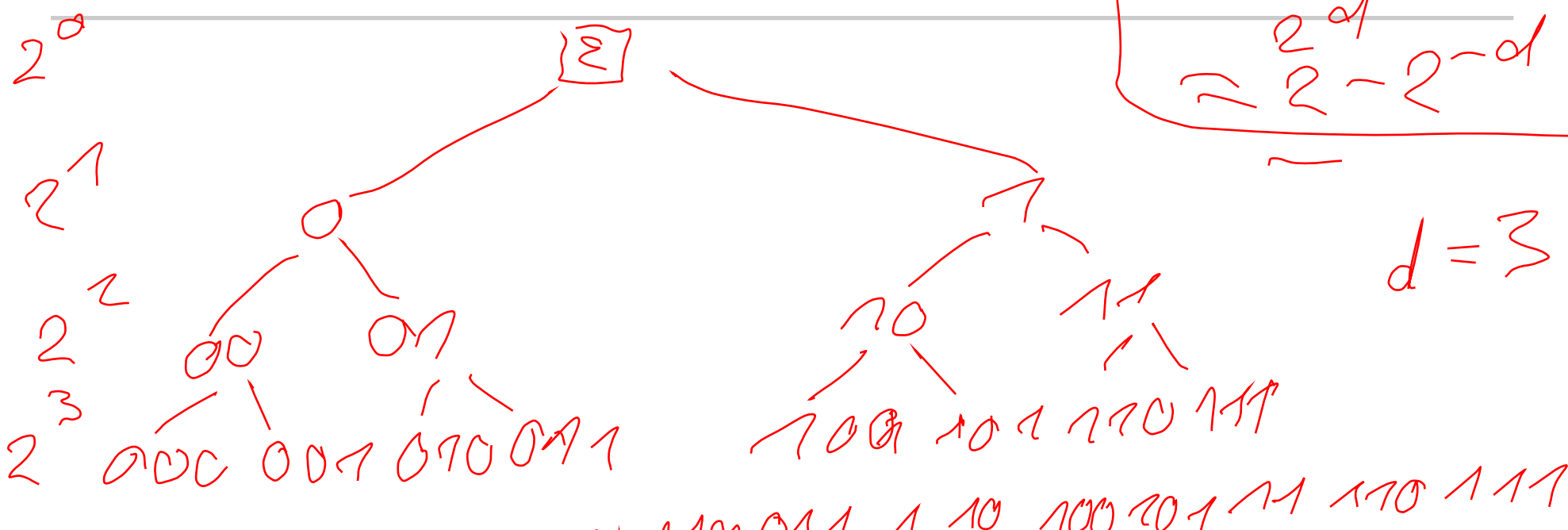


Adaptives Baumprotokoll

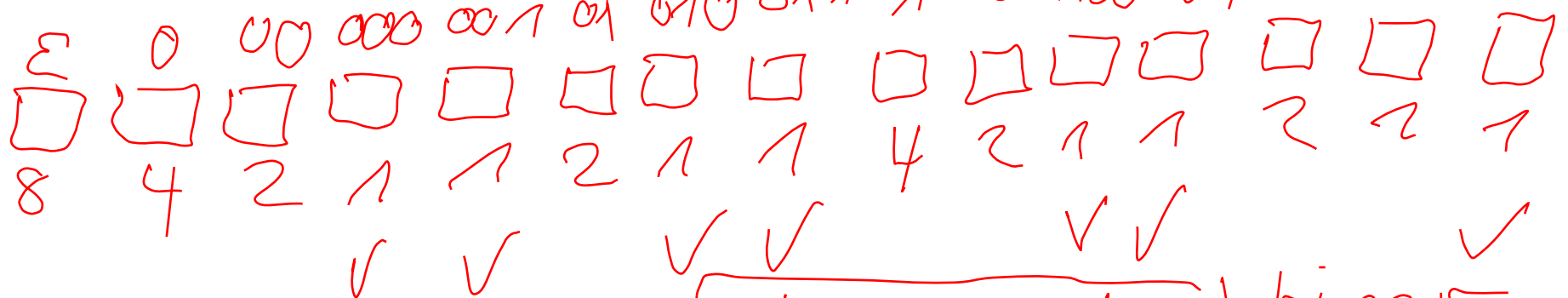
Beispiel: alle senden

Overhead

$$\frac{2^d + 2^d - 1}{2^d} \approx 2 - 2^{-d}$$

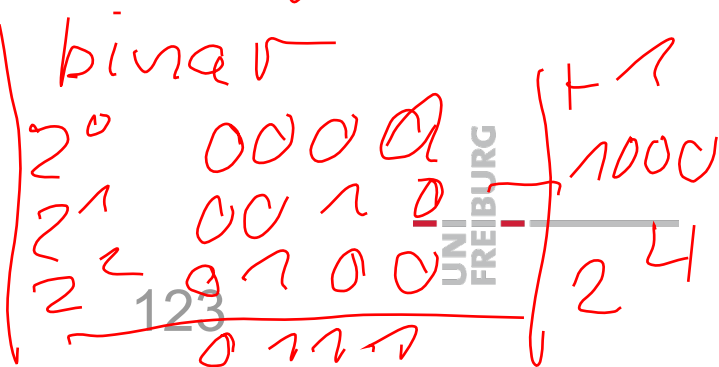


$d=3$



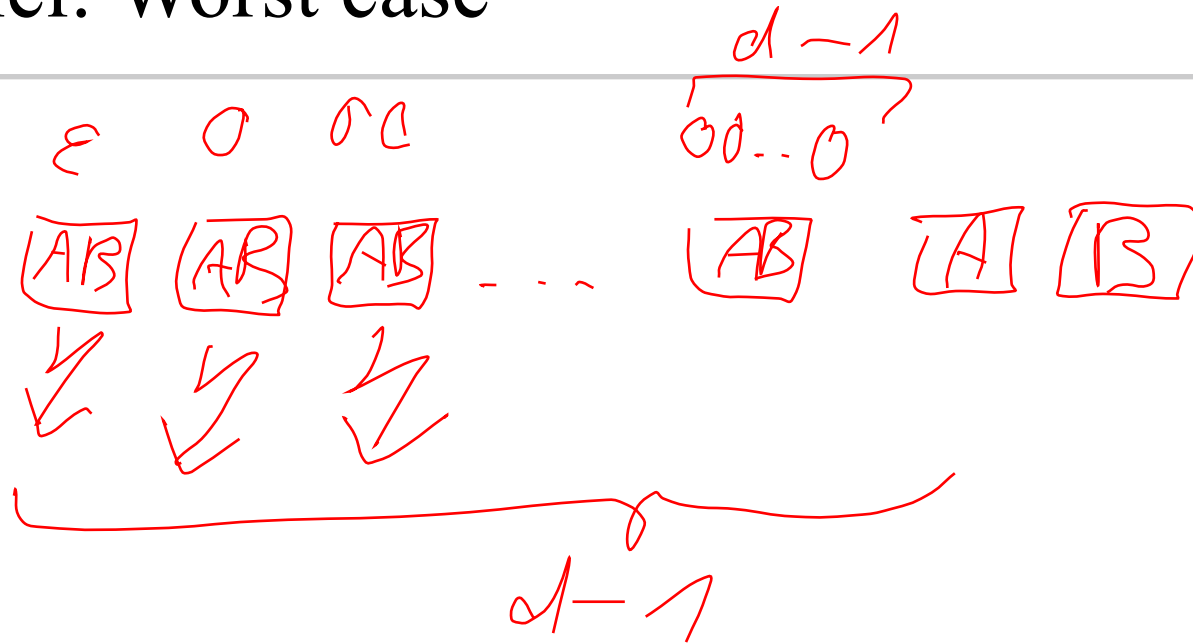
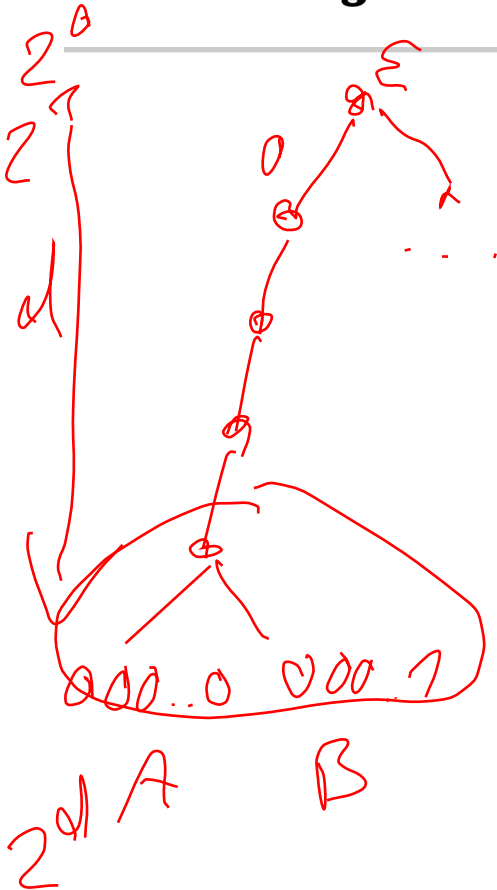
8 kollisionsfrei
 7 Kollisionen

$$\sum_{i=0}^{d-1} 2^i = 2^d - 1$$



Adaptives Baumprotokoll

Beispiel: Worst case



$$2^d = n \text{ Stationen}$$

$$d = \log_2 n$$

Systeme II

7. Die Datensicherungsschicht (Teil 5)

Thomas Janson[°], Kristof Van Laerhoven*, Christian
Ortolf[°]

Folien: Christian Schindelbauer[°]

Technische Fakultät

[°]: Rechnernetze und Telematik, *: Eingebettete Systeme

Systeme II

8. Die physikalische Schicht

Thomas Janson[°], Kristof Van Laerhoven*, Christian
Ortolf[°]

Folien: Christian Schindelhauer[°]

Technische Fakultät

[°]: Rechnernetze und Telematik, *: Eingebettete Systeme

■ ISO-Definition

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

■ Information

- Menschliche Interpretation,
 - z.B. schönes Wetter

■ Daten

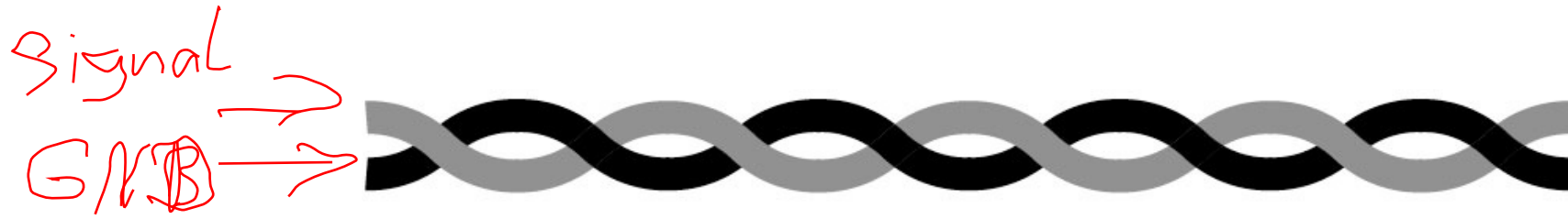
- Formale Präsentation,
 - z.B. 8 Grad Celsius, Niederschlagsmenge 0 cm,
Wolkenbedeckung 40%

■ 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

Twisted Pair



(a)

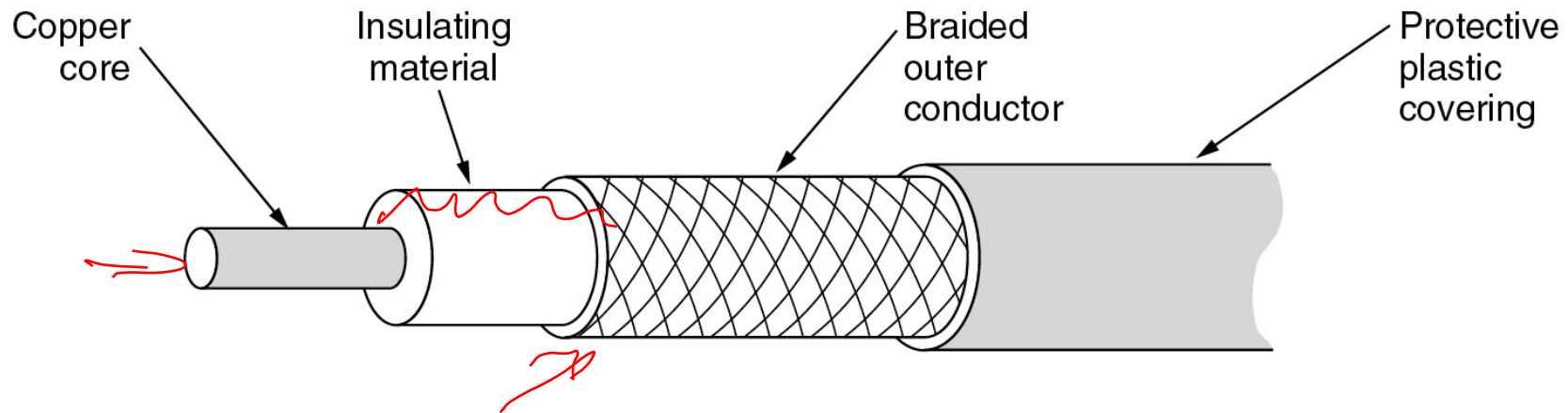


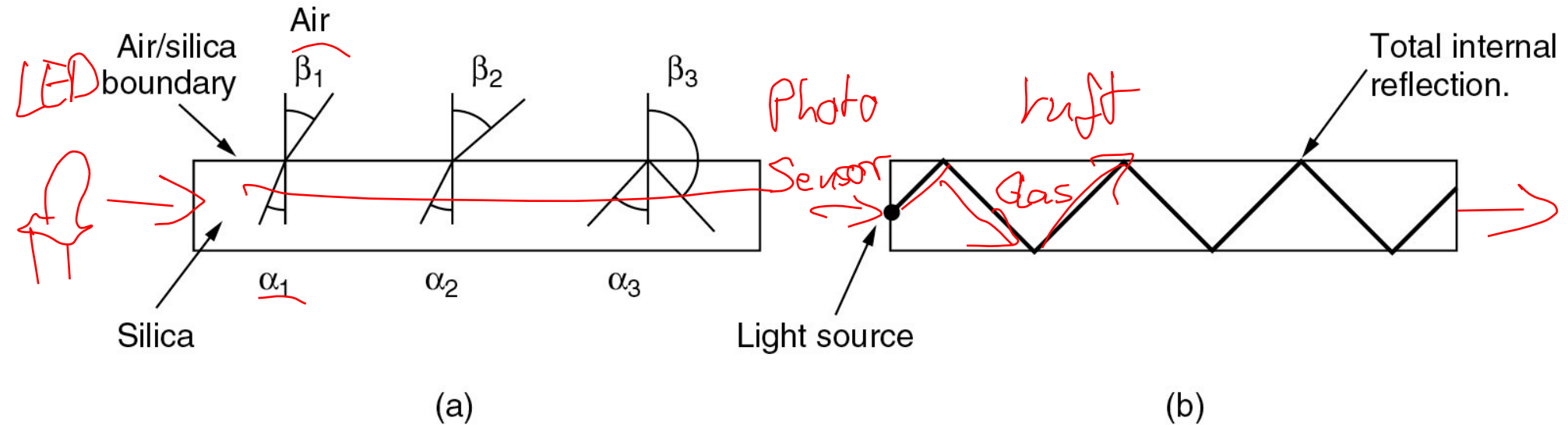
(b)

(a) Category 3 UTP.

(b) Category 5 UTP.

Koaxialkabel

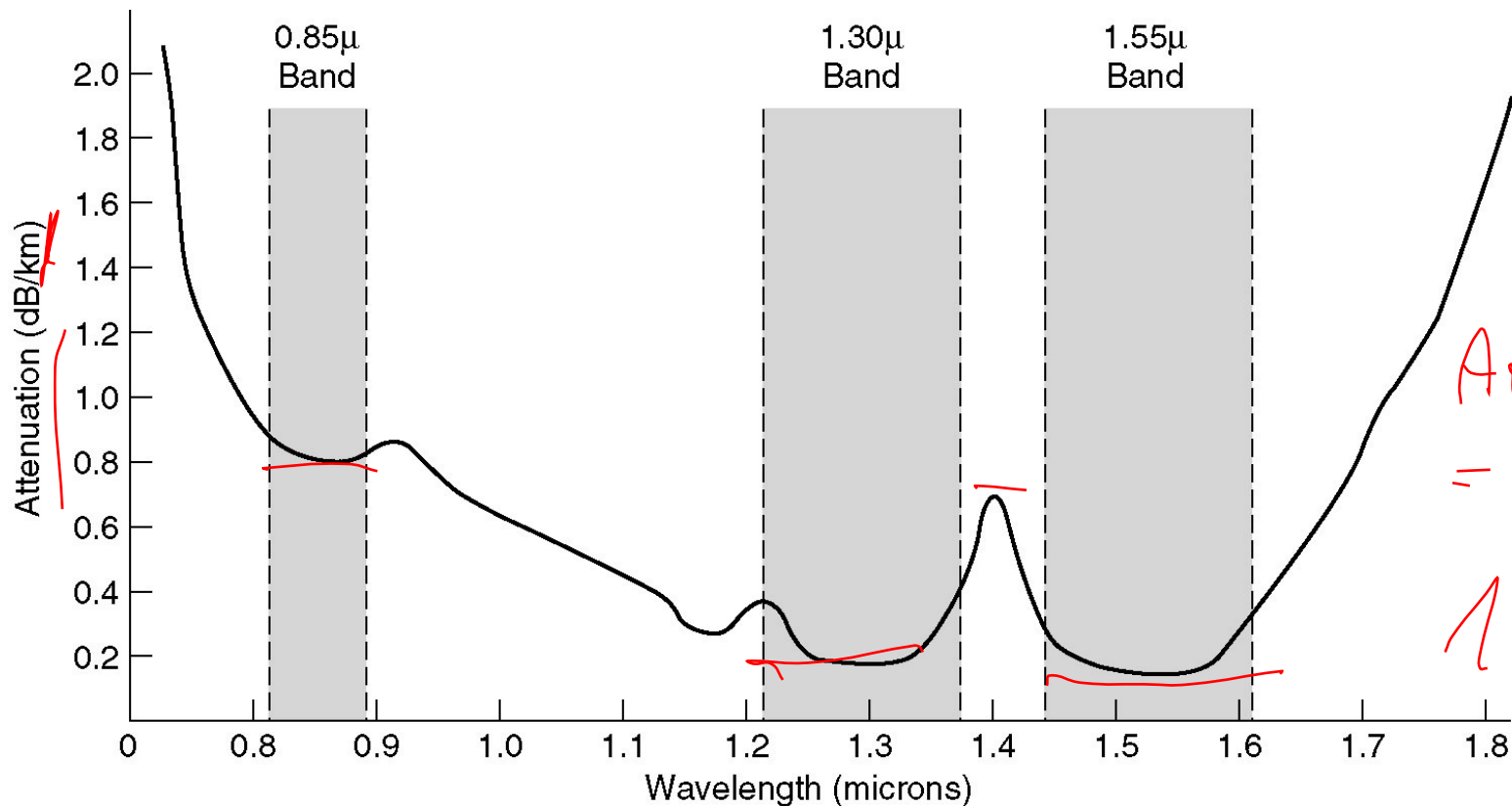




Gesetz von Snellius:
$$\frac{\sin \alpha}{\sin \beta} = \frac{c_{\text{Glas}}}{c_{\text{Luft}}}$$

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

■ Dämpfung von Infrarotlicht in Glasfaser



3 dB $\frac{1}{2}$

6 dB $\frac{1}{4}$

9 dB $\frac{1}{8}$

Attenuation (dB)
 $= 10 \cdot \log_{10} \left(\frac{in}{out} \right)$

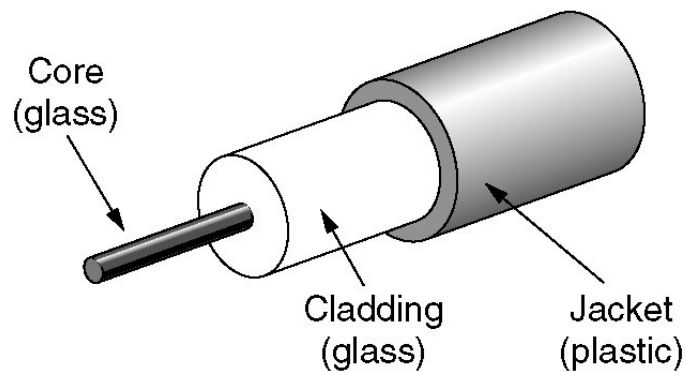
1 Watt, 3 dB

$\frac{1}{2}$ Watt

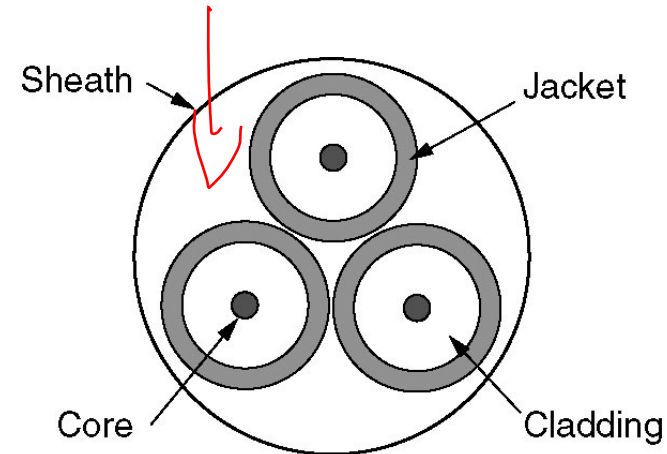
0.85 μm
 $0.85 \cdot 10^{-6} m$



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

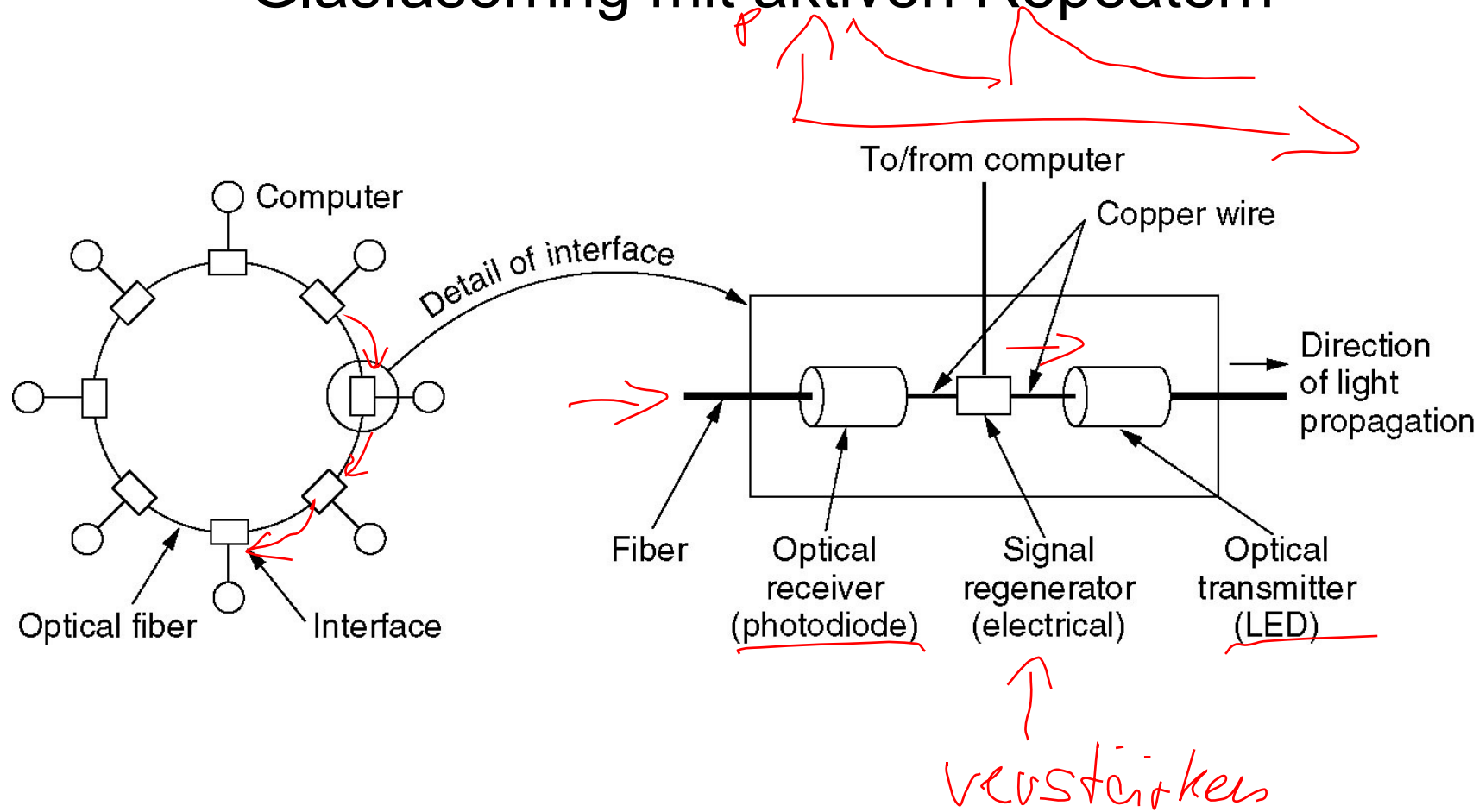


(a)

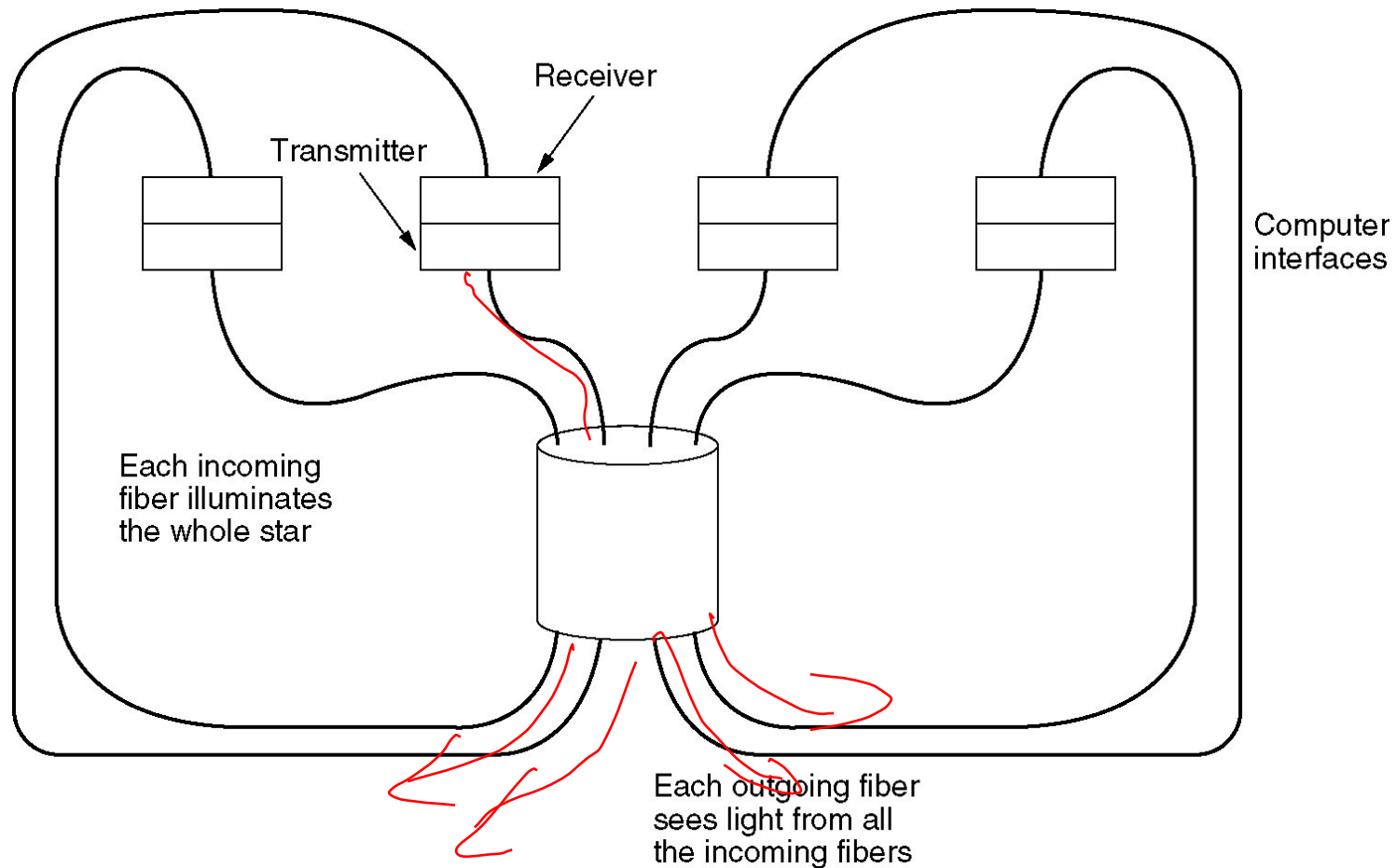


(b)

Glasfaserring mit aktiven Repeatern

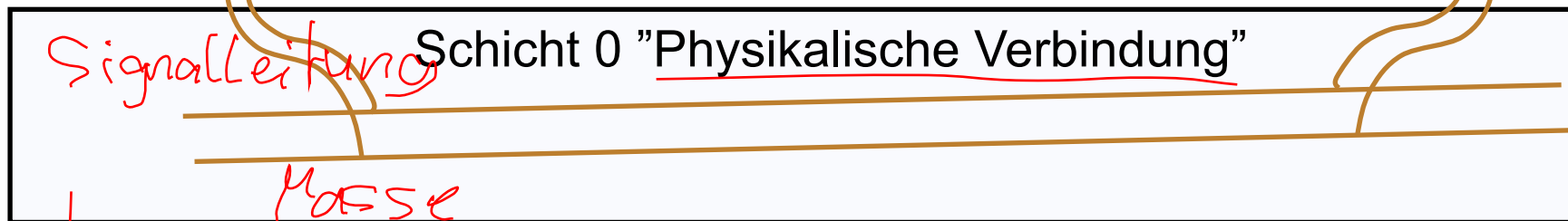
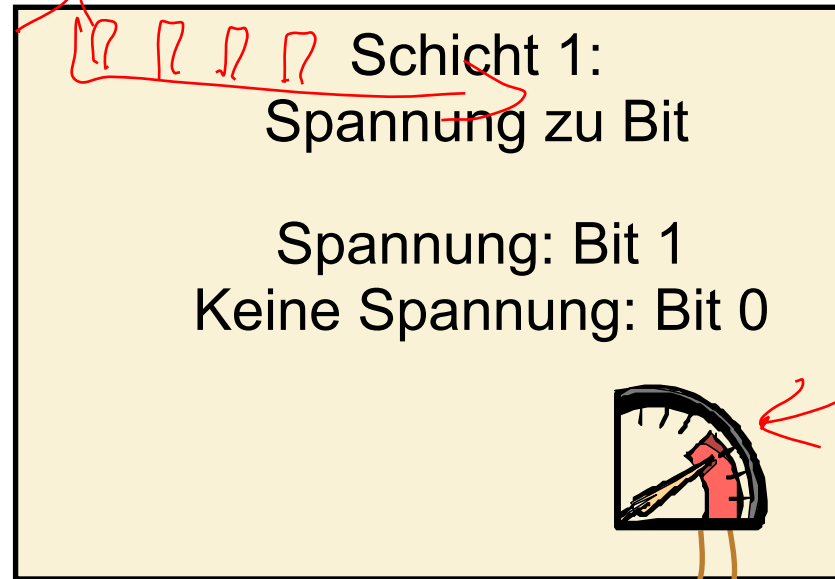
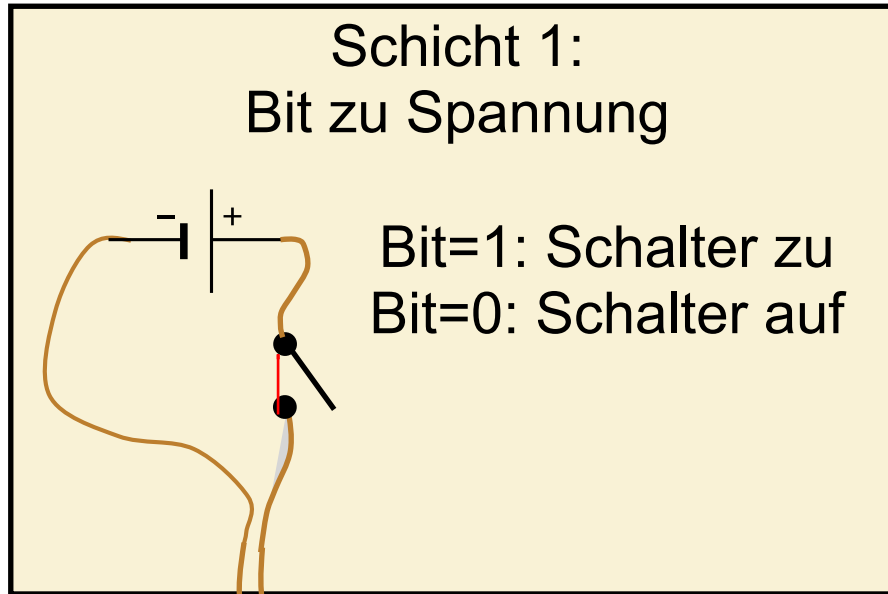
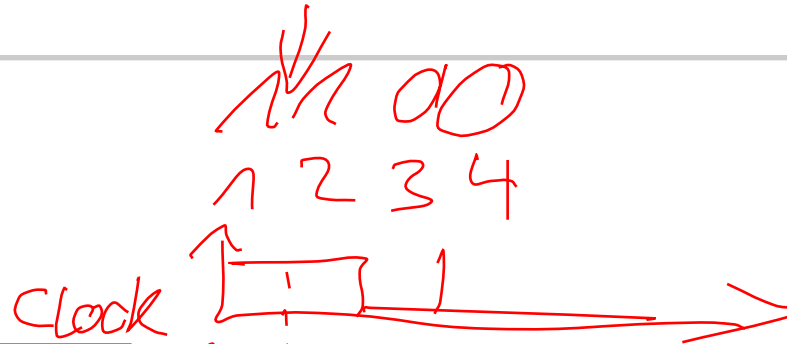


Eine passive Sternverbindung in einem Glasfasernetz



Die einfachste Bitübertragung

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

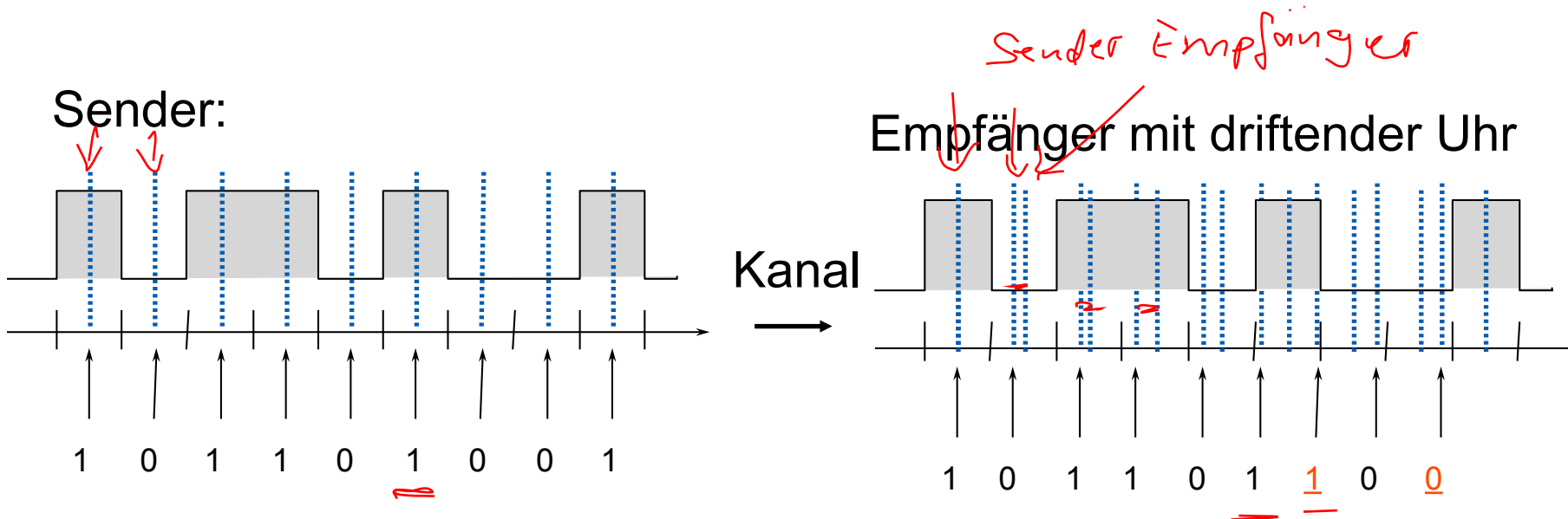


Clock

(aus Vorlesung von Holger Karl)

- 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*

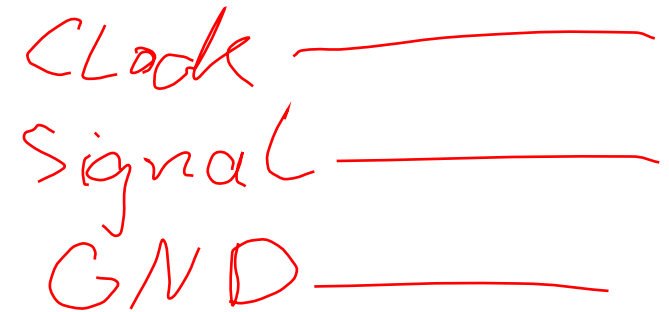
- Was passiert wenn man einfach Uhren benutzt
- Problem
 - Die Uhren driften auseinander
 - Keine zwei (bezahlbare Uhren) bleiben perfekt synchron
- Fehler by Synchronisationsverlust (NRZ):



- Ohne Kontrolle keine Synchronisation

- Lösung: explizites Uhrensinal

- Benötigt parallele Übertragung über Extra-Kanal
- Muss mit den Daten synchronisiert sein
- Nur für kurze Übertragungen sinnvoll



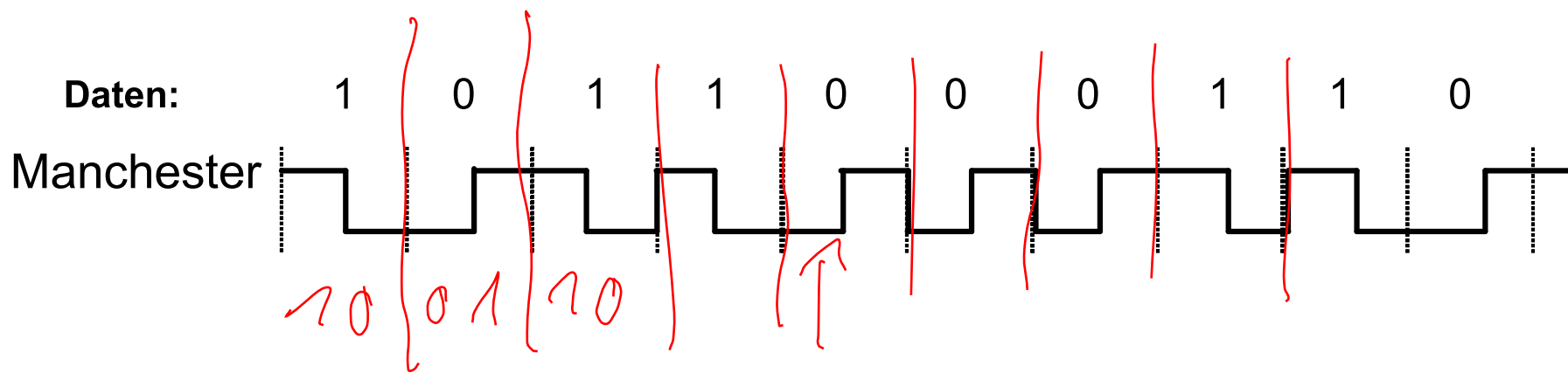
- Synchronisation an kritischen Zeitpunkten

- z.B. Start eines Symbols oder eines Blocks
- Sonst läuft die Uhr völlig frei
- Vertraut der kurzzeitig funktionierenden Synchronität der Uhren

- Uhrensinal aus der Zeichenkodierung



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



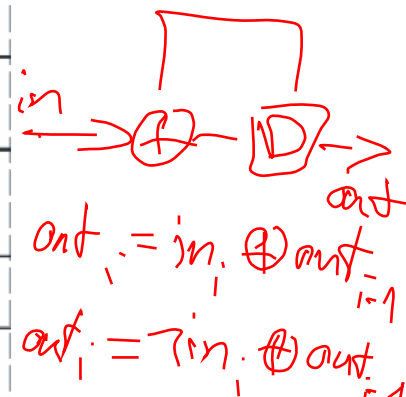
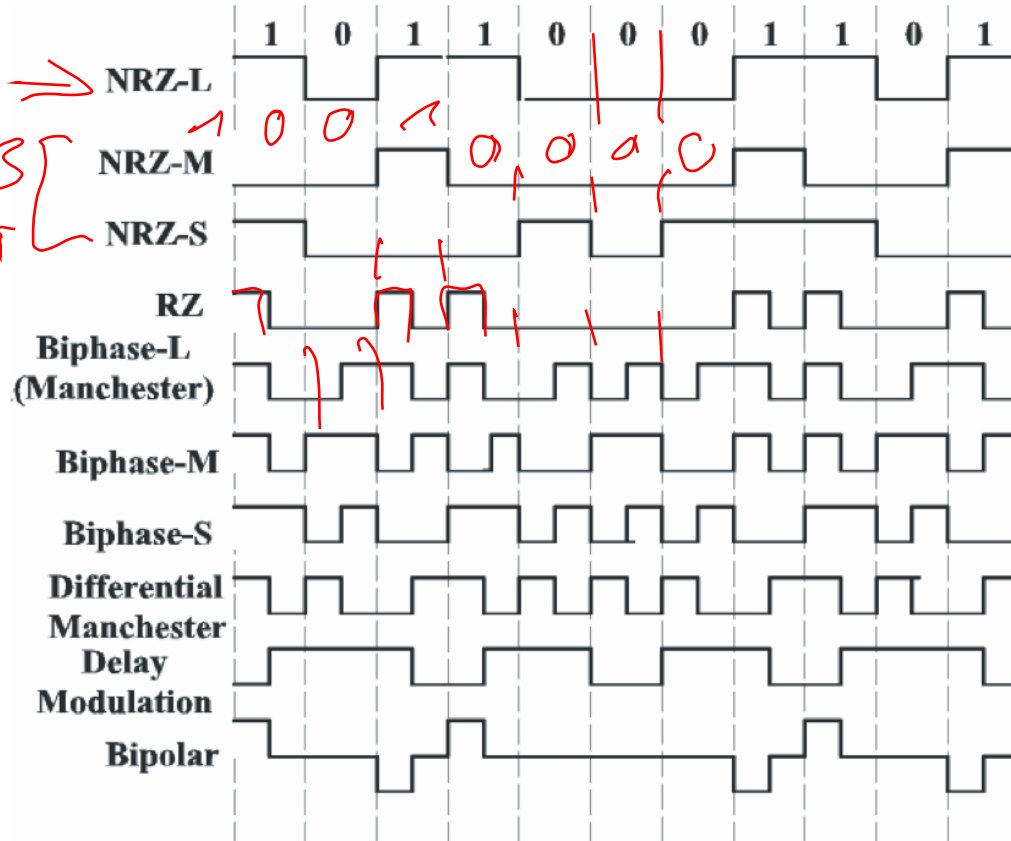
- Das Signal beinhaltet die notwendige Information zur Synchronisation

00000000 00 1111 11

- 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

nicht selbst
faktoren

USB
ethernet



selbst faktoren

Systeme II

8. Die physikalische Schicht

Thomas Janson[°], Kristof Van Laerhoven*, Christian
Ortolf[°]

Folien: Christian Schindelhauer[°]

Technische Fakultät

[°]: Rechnernetze und Telematik, *: Eingebettete Systeme