CoNe Freiburg

IIF INSTITUT FÜR INFORMATIK FREIBURG

Systeme II

3. Die Datensicherungsschicht

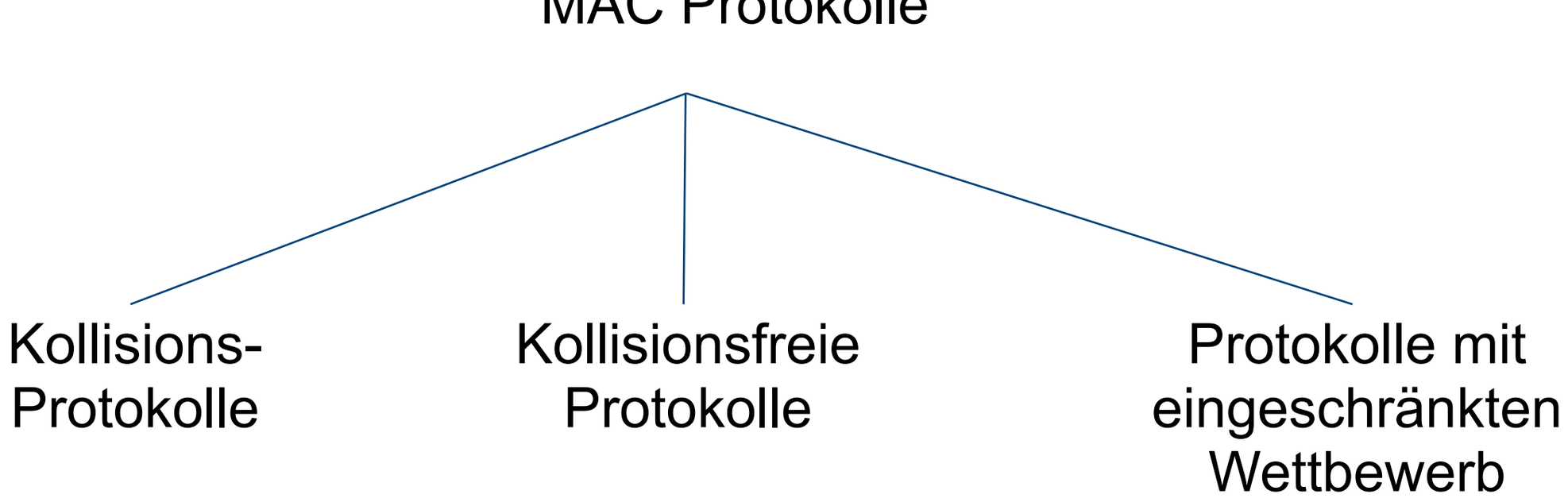
Christian Schindelhauer
Technische Fakultät
Rechnernetze und Telematik
Albert-Ludwigs-Universität Freiburg

Version 14.05.2013

UNI FREIBURG

Mögliche MAC-Protokolle

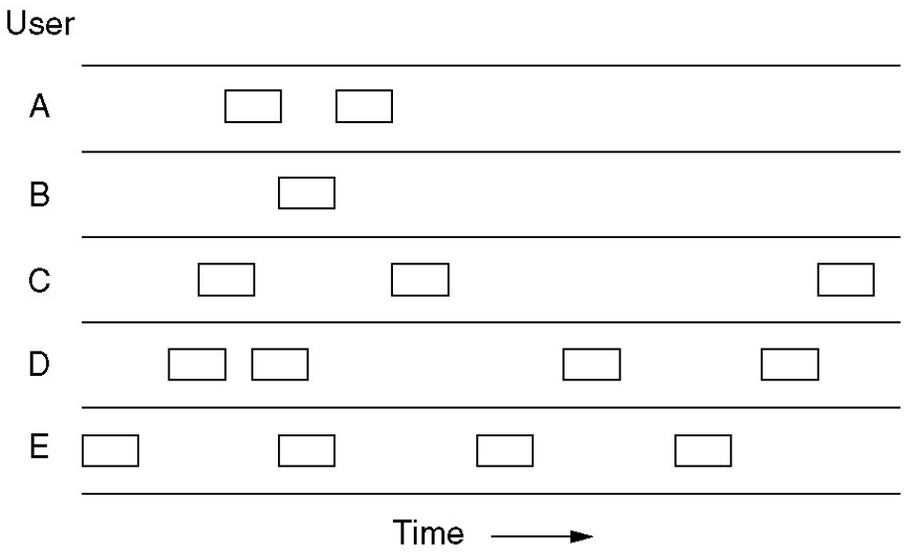
- Unterscheidung: Erlaubt das Protokoll Kollisionen?
  - Als Systementscheidung
  - Die unbedingte Kollisionsvermeidung kann zu Effizienzeinbußen führen



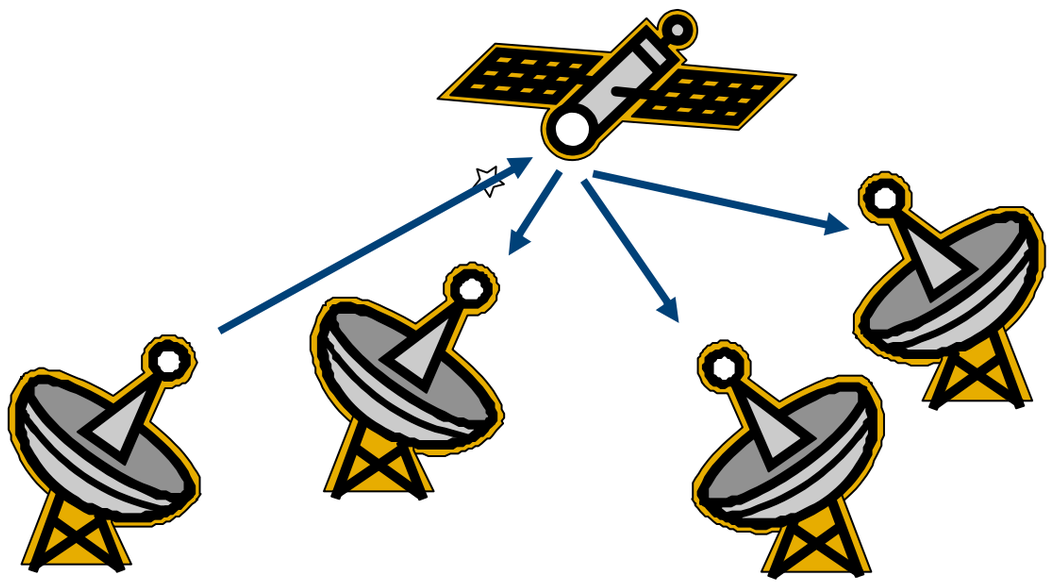
System mit Kollisionen: **Contention System**

ALOHA

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



Pakete werden zu
beliebigen Zeiten übertragen



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?

n : $n \begin{cases} \square \\ \square \\ \square \\ \square \\ \square \end{cases}$ W'keit p für Nachricht

Erwartungswert $= p \cdot n = 6 = \underline{\underline{\lambda}}$

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

$\square \quad \square \quad \square \quad \square \quad \square$ $\quad k = 3$

$\boxed{\text{X} \quad \text{X} \quad \text{X} \quad \square \quad \square}$

$n = 5$

$p = \frac{1}{10}$

$$\binom{5}{3} = \frac{5}{3!\,(5-3)!}$$

$$= \frac{5 \cdot 4 \cdot 3}{3!}$$

$$\frac{1}{10} \cdot \frac{1}{10} \cdot \frac{1}{10} \cdot \frac{9}{10} \cdot \frac{9}{10}$$

$$\left(\frac{1}{10}\right)^3 \cdot \left(\frac{9}{10}\right)^{5-3}$$

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

$$\lim_{n\to\infty} P(X=k) = \lim_{n\to\infty} \frac{n!}{k!(n-k)!} \cdot \left(\frac{\lambda}{n}\right)^k \left(1-\frac{\lambda}{n}\right)^{n-k}$$

$$\sigma = \lambda = p \cdot n$$

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

$$\frac{\lambda^k}{k!} \cdot \underbrace{\frac{n \cdot (n-1) \cdot (n-2) \cdots (n-k+1)}{n \cdot n \cdot n \cdots n}}_{\to 1} \cdot \underbrace{\left(1-\frac{\lambda}{n}\right)^{\frac{n}{\lambda}\lambda}}_{\to e} \cdot \underbrace{\left(1-\frac{\lambda}{n}\right)^{-k}}_{\to 1}$$

$$\to 1$$

$$\boxed{\frac{\lambda^k}{k!} \cdot e^{-\lambda}}$$

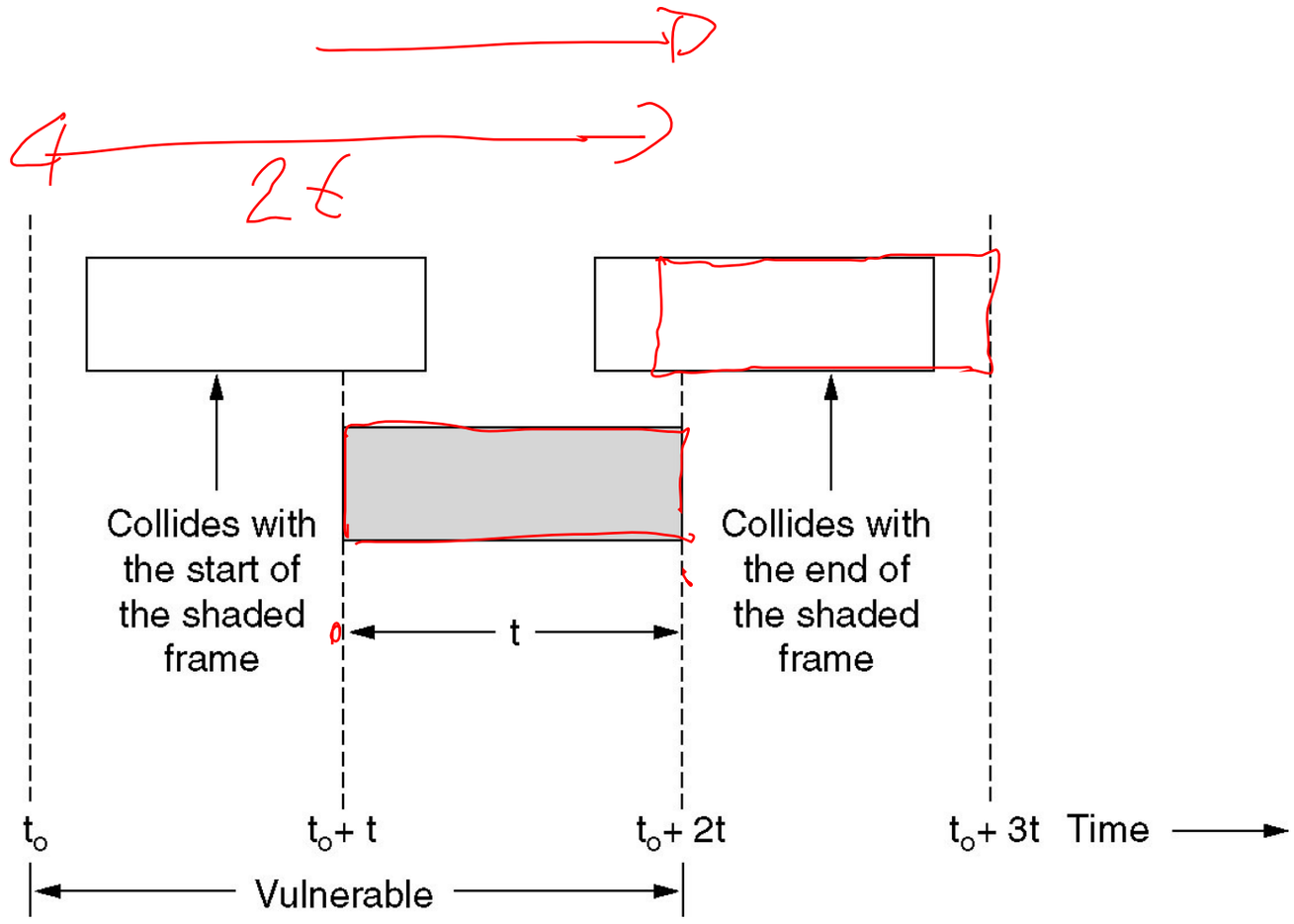
$$\lambda = \sigma$$

$$\lim_{m\to\infty} \left(1-\frac{1}{m}\right)^m \to \frac{1}{e}$$

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

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

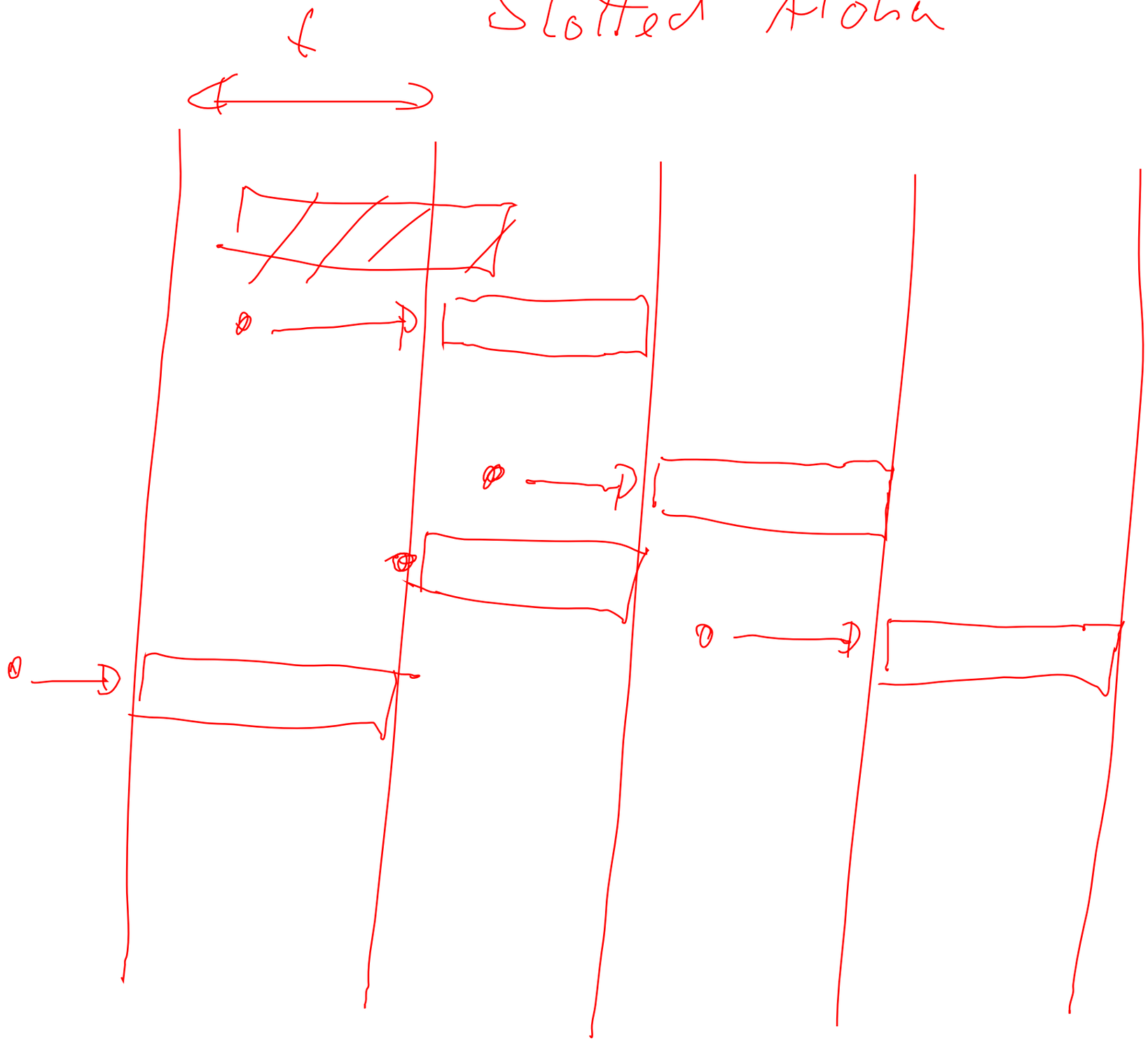


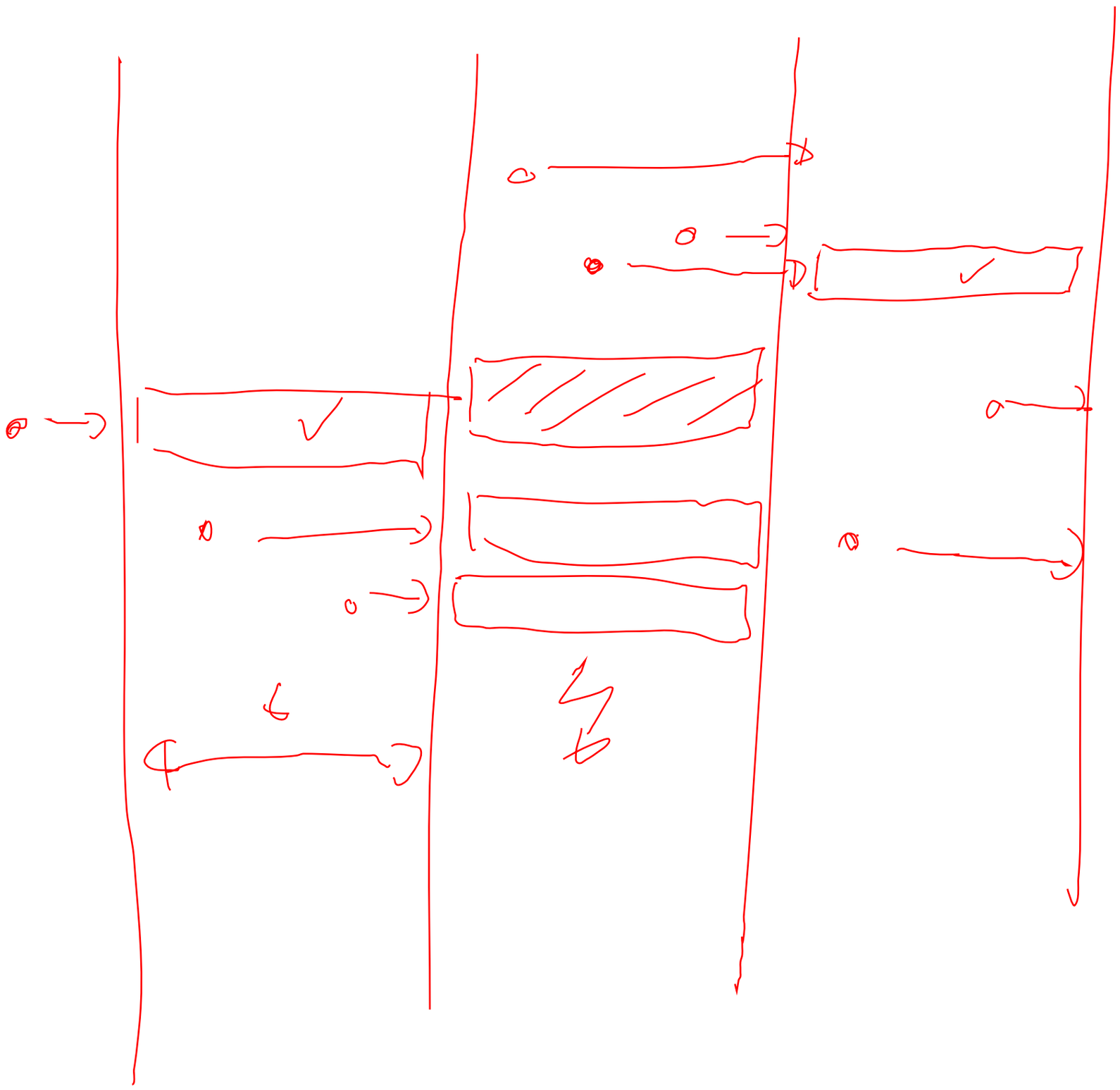
CoNe Freiburg

Slotted ALOHA

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

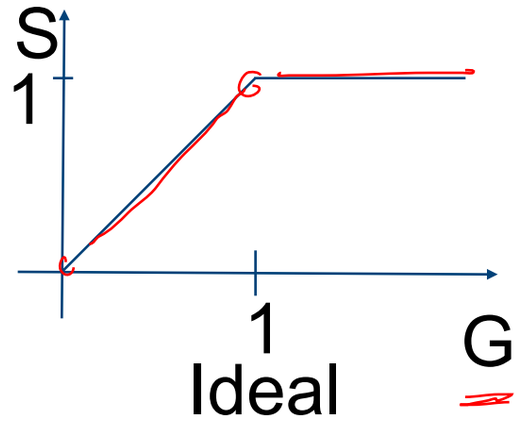
Slotted Aloha

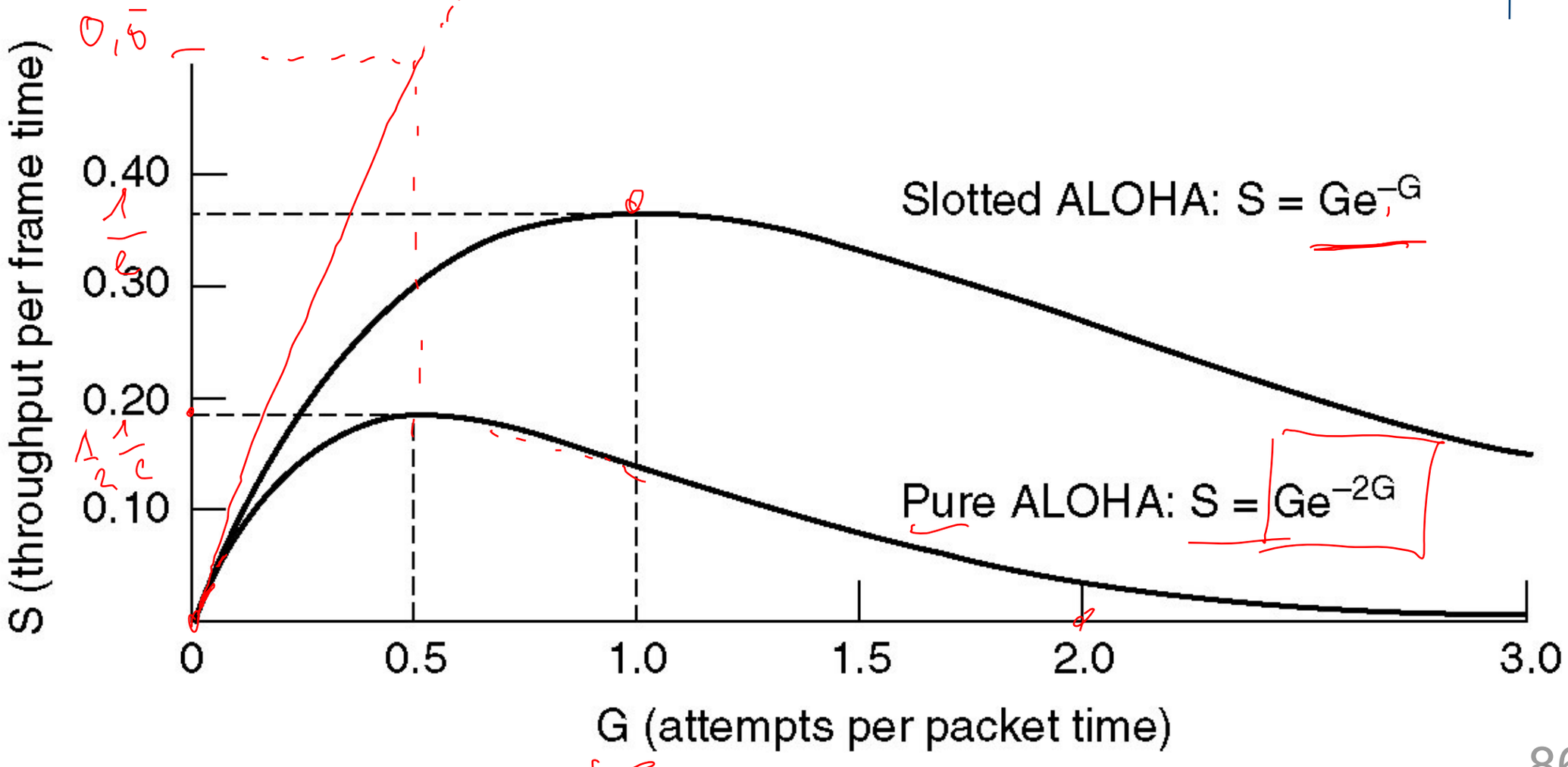




Durchsatz in Abhängigkeit der Last

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





CoNe Freiburg

Bestimmung der Warte-Zeit

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

A		x	
B	x	x	
C		x	
D	x		

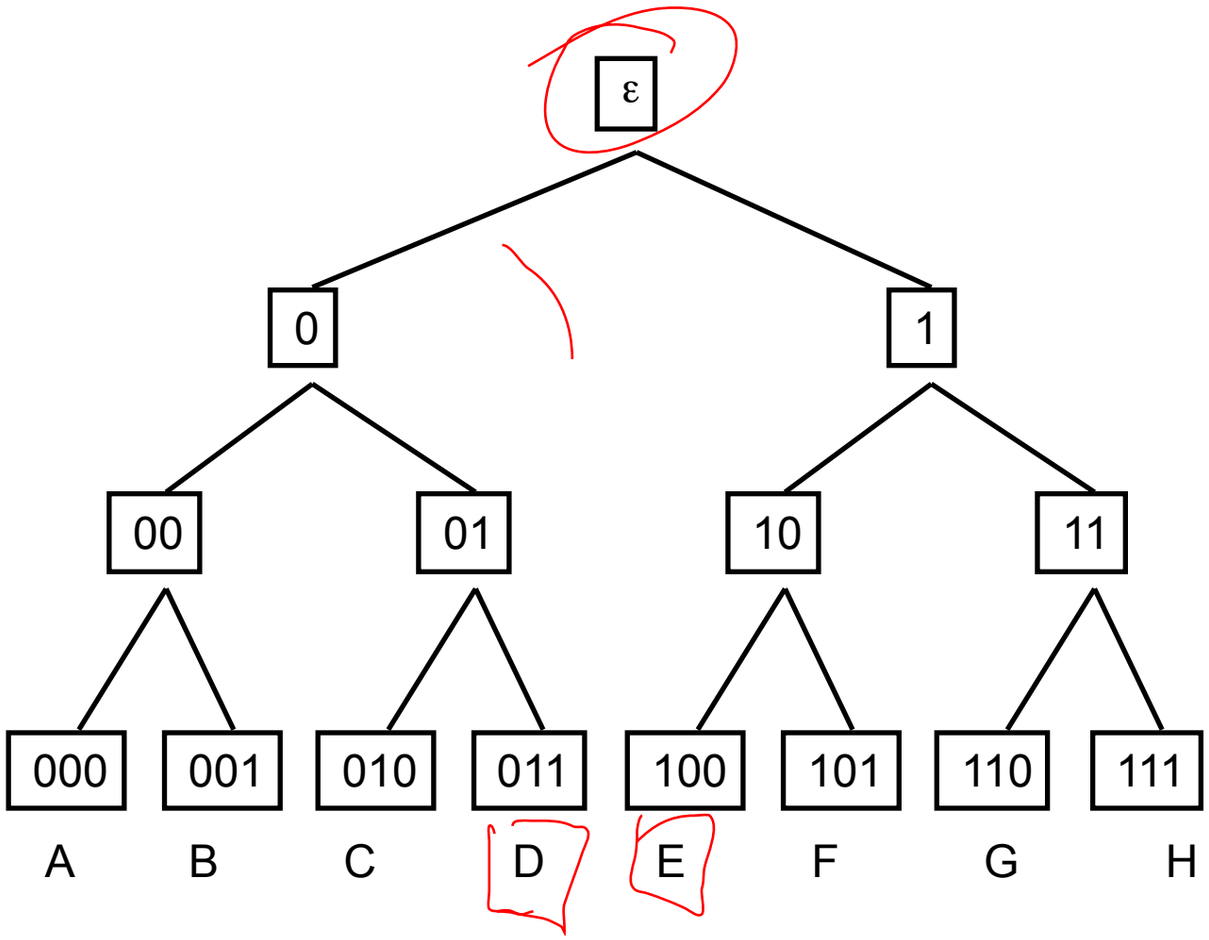
6 8 6

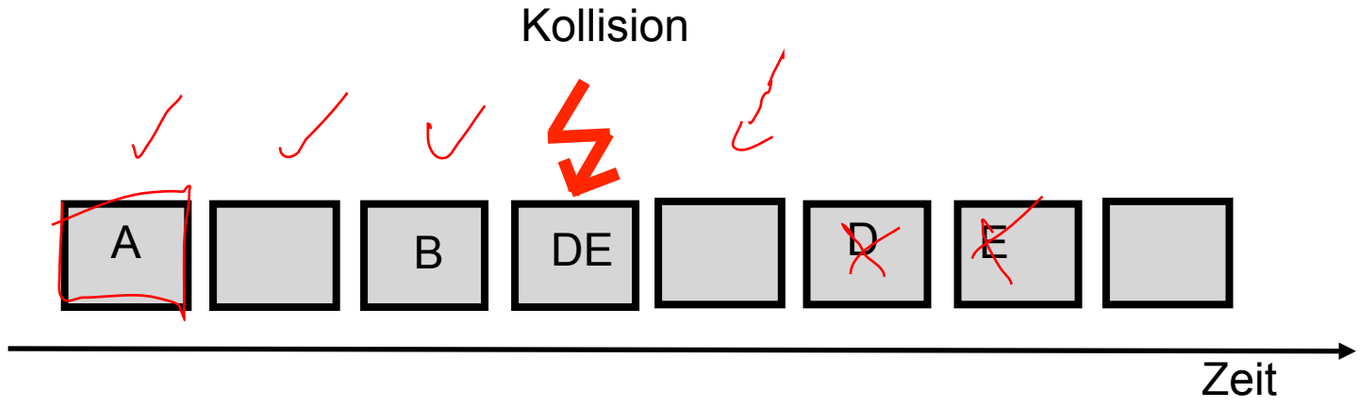
Protokolle mit beschränktem Wettbewerb

- 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

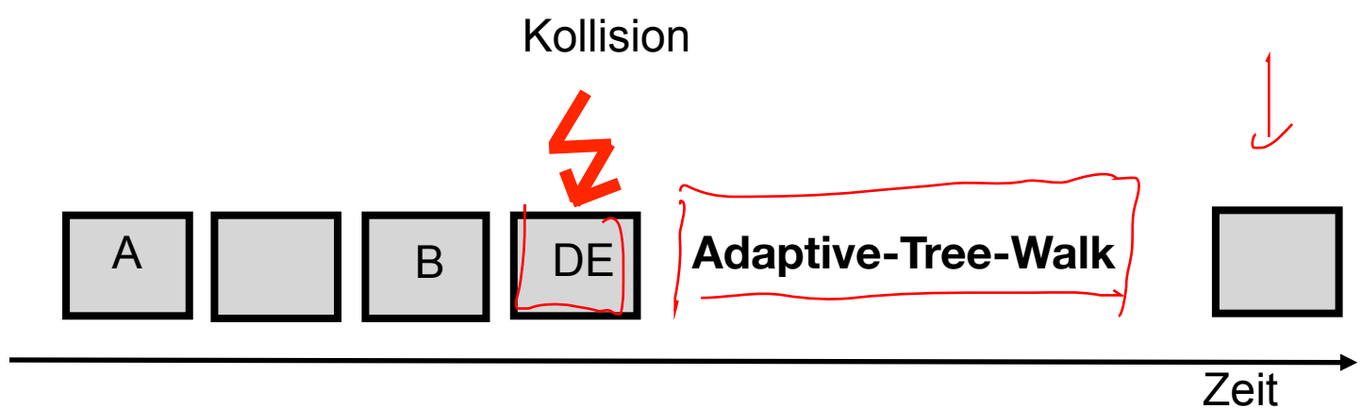
- Adaptives Baumprotokoll (adaptive tree walk)
- Ausgangspunkt:
  - Binäre, eindeutige Präsentation aller Knoten (ID)
  - Dargestellt in einem Baum
  - Synchronisiertes Protokoll
  - Drei Typen können unterschieden werden:
    - Keine Station sendet
    - Genau eine Station sendet
    - Kollision: mindestens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





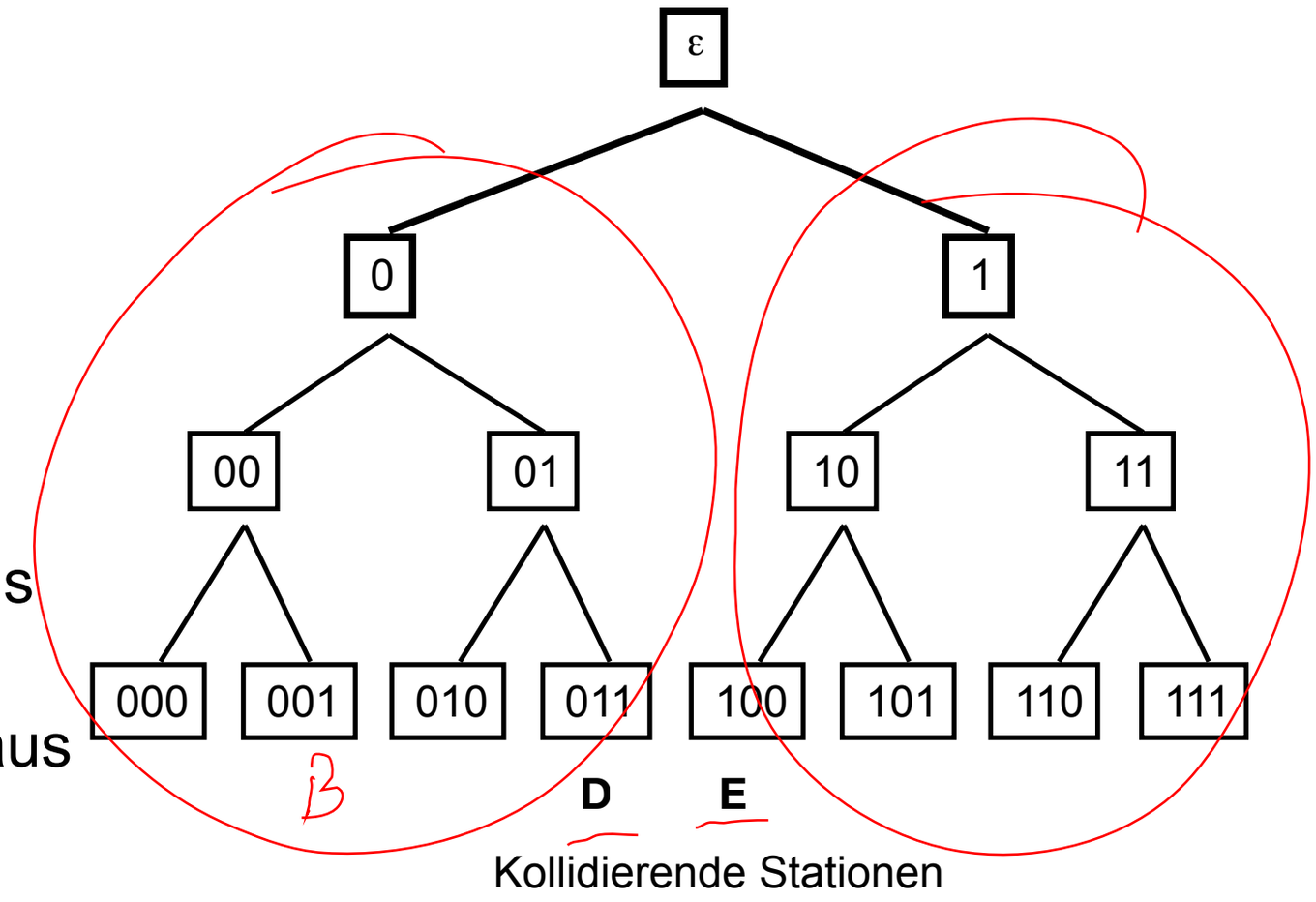
Adaptives Baumprotokoll
Basis-Algorithmus

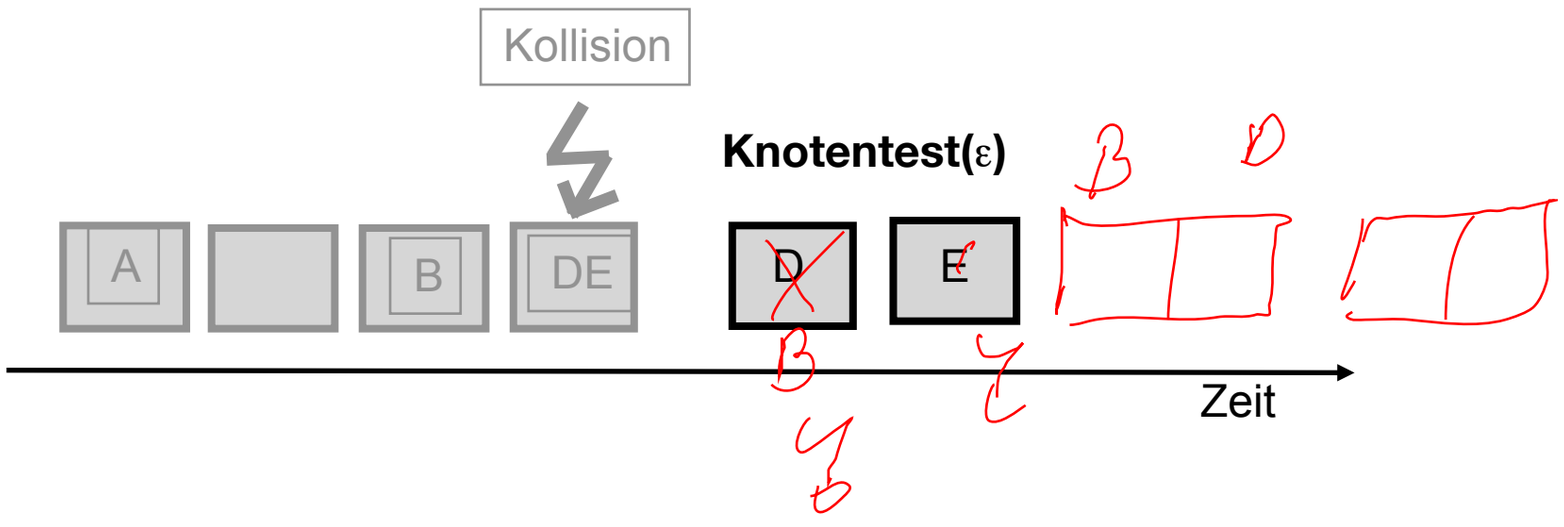
- Basis-Algorithmus
  - Jeder Algorithmus sendet sofort (slotted Aloha)
  - Falls eine Kollision auftritt,
    - akzeptiert keine Station mehr neue Paket aus der Vermittlungsschicht
    - Führe Adaptive-Tree-Walk(ε) aus



Adaptives Baumprotokoll
Knoten-Test

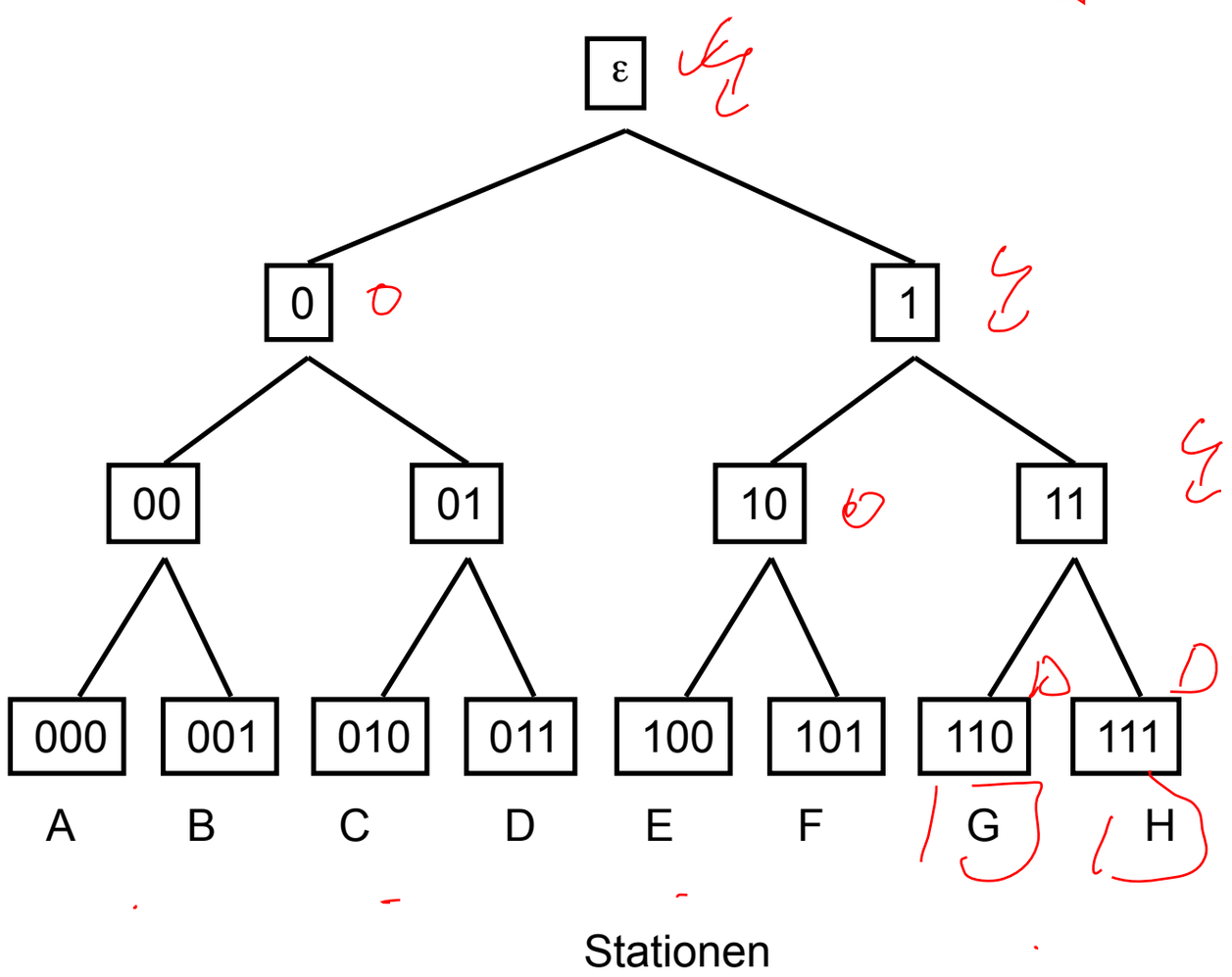
- Algorithmus Knoten-Test
  - für Knoten u des Baums und
  - kollidierende Menge S von Station
- Knoten-Test(u)
  - Betrachte zwei Slots pro Knoten des Baums
  - Im ersten Slot senden alle Knoten aus S, die mit ID u0 anfangen
  - Im zweiten Slot senden alle Knoten aus S, die mit ID u1 anfangen



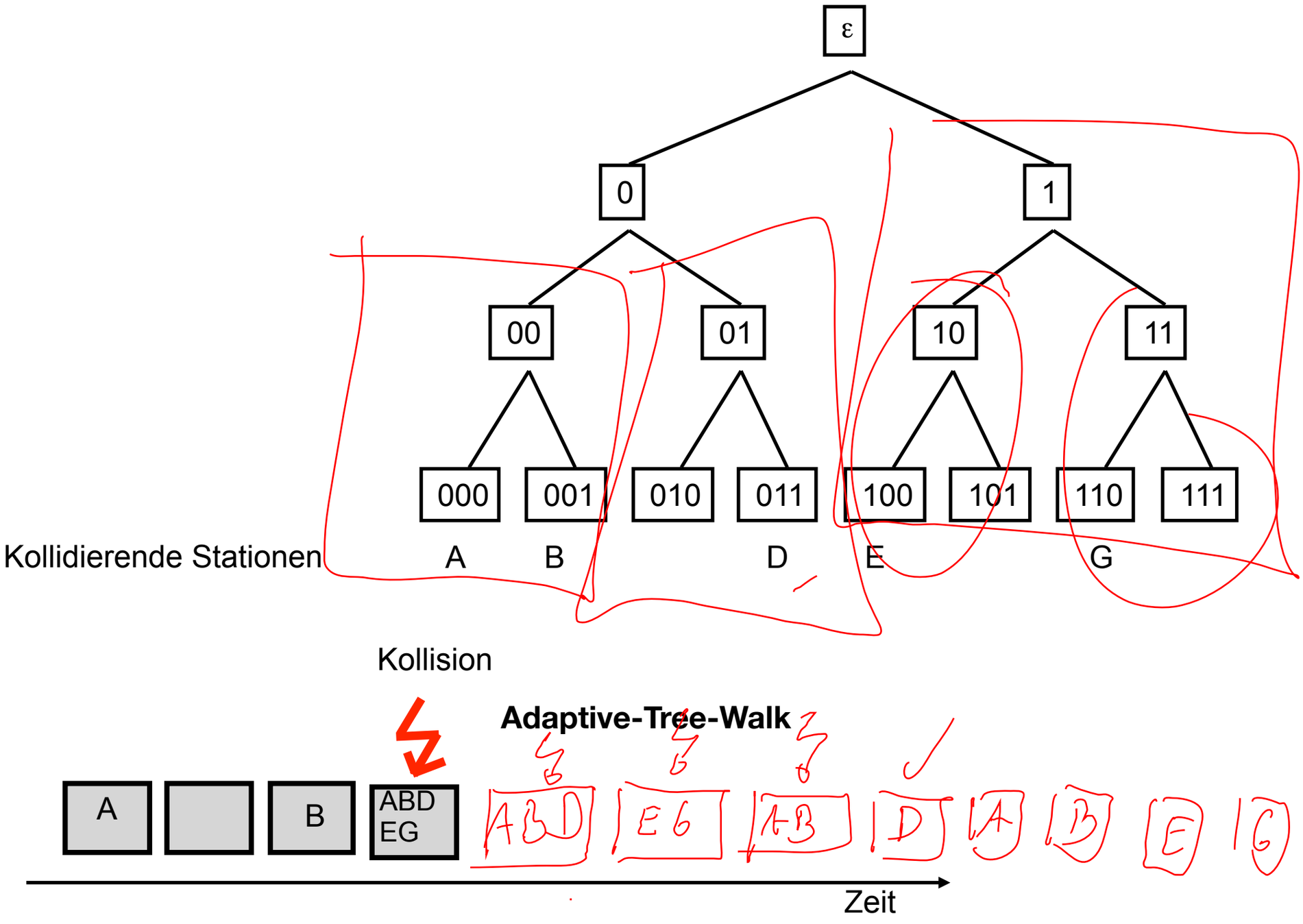


Adaptives Baumprotokoll
Kern-Algorithmus

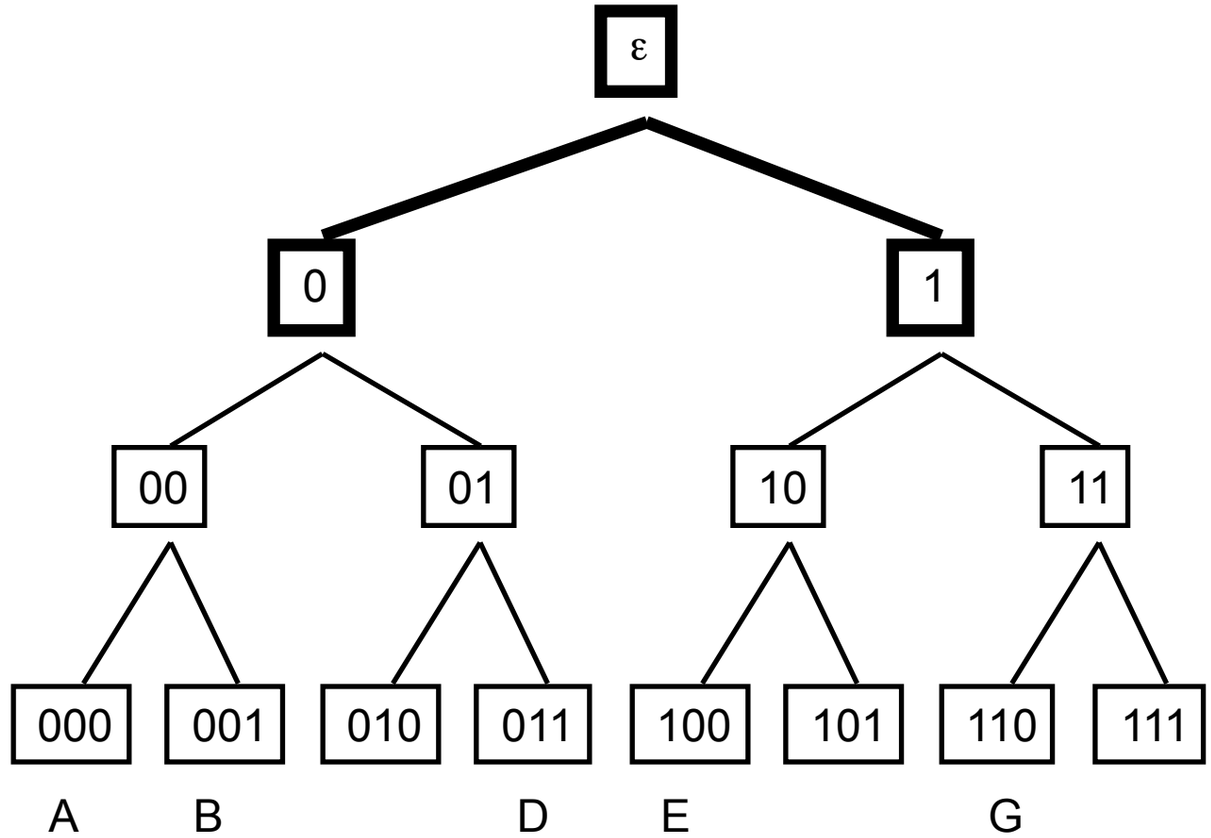
- Algorithmus Knoten-Test
  - für Knoten u des Baums und
  - kollidierende Menge S von Station
- Knoten-Test(u)
  - Betrachte zwei Slots pro Knoten des Baums
    - Im ersten Slot senden alle Knoten aus S, die mit ID u0 anfangen
    - Im zweiten Slot senden alle Knoten aus S, die mit ID u1 anfangen
- Adaptive Tree Walk(x)
  - 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



Adaptives Baumprotokoll
Beispiel (1)

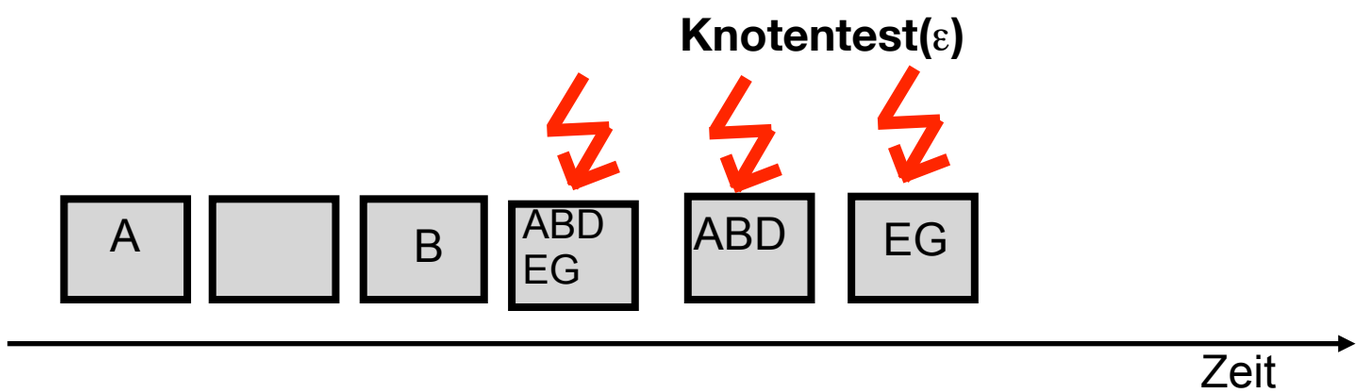


Adaptives Baumprotokoll
Beispiel (2)

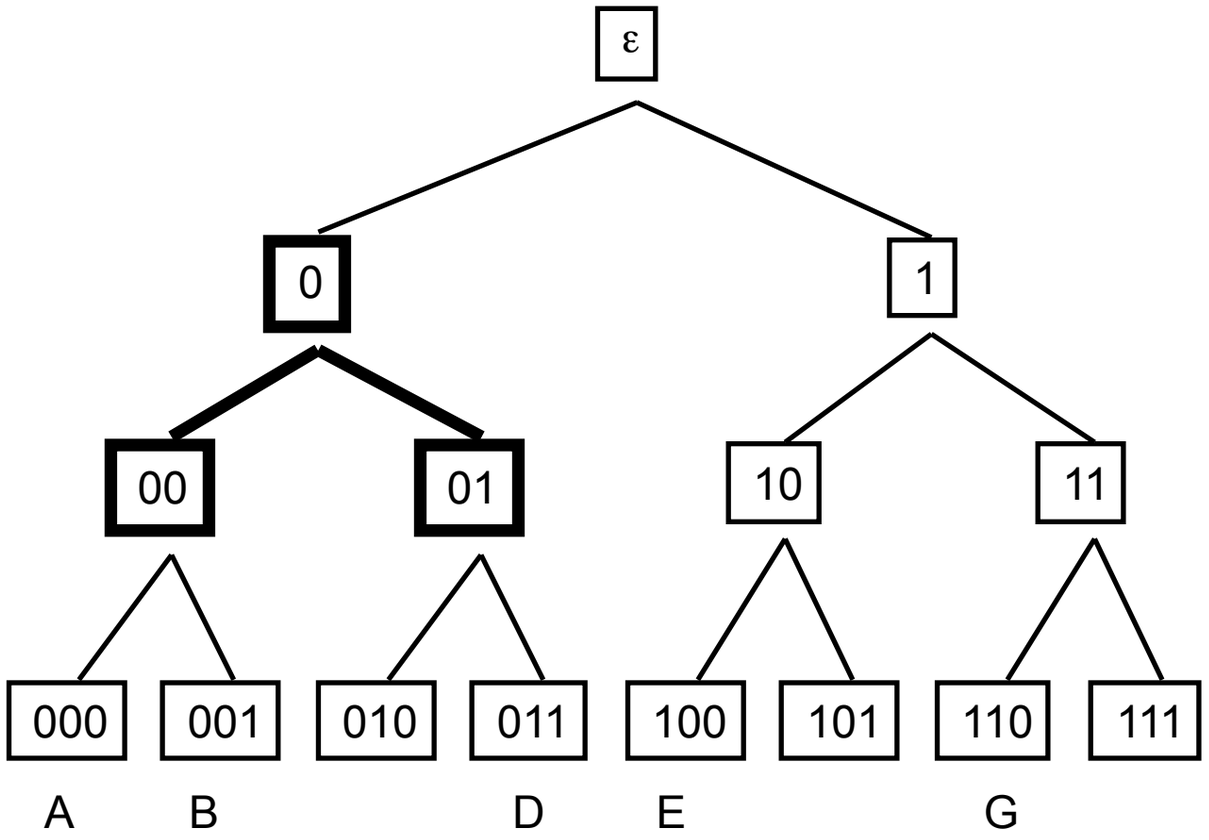


Kollidierende Stationen

Adaptive-Tree-Walk

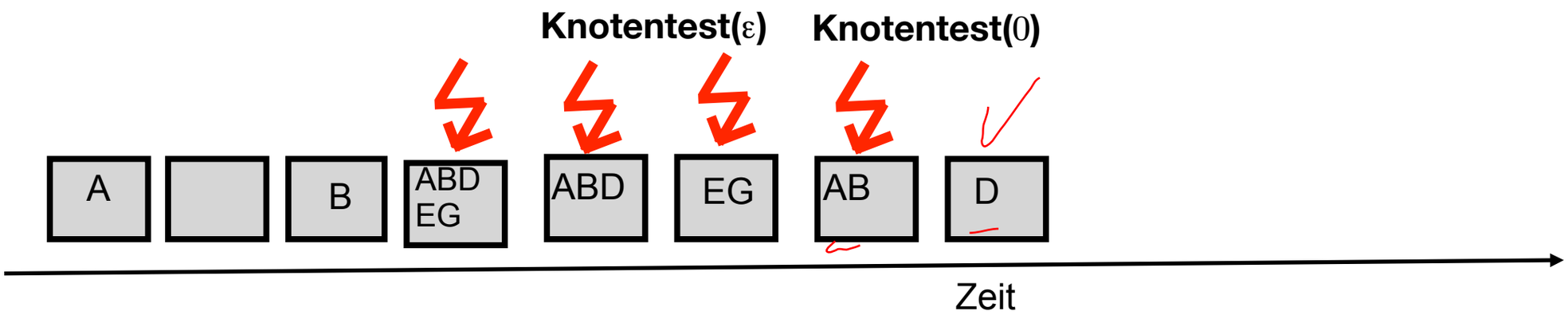


Adaptives Baumprotokoll
Beispiel (3)

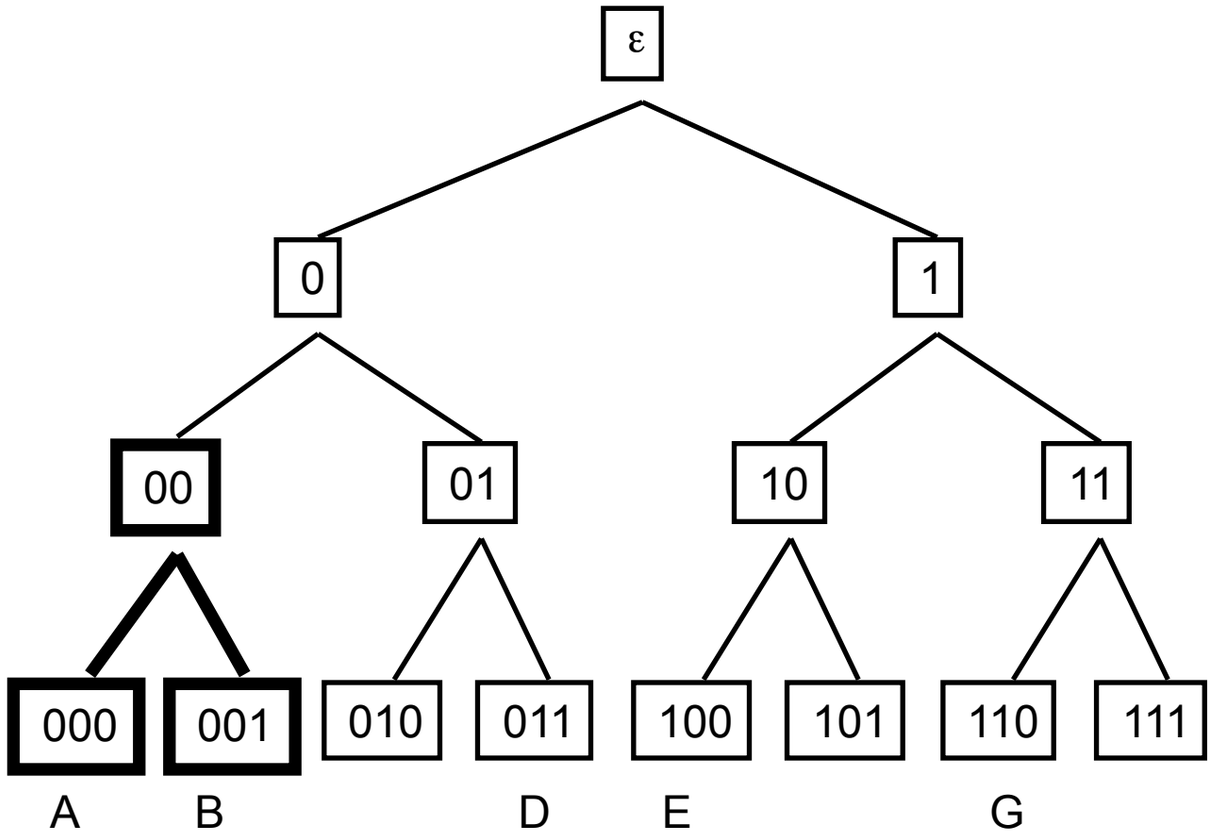


Kollidierende Stationen

Adaptive-Tree-Walk

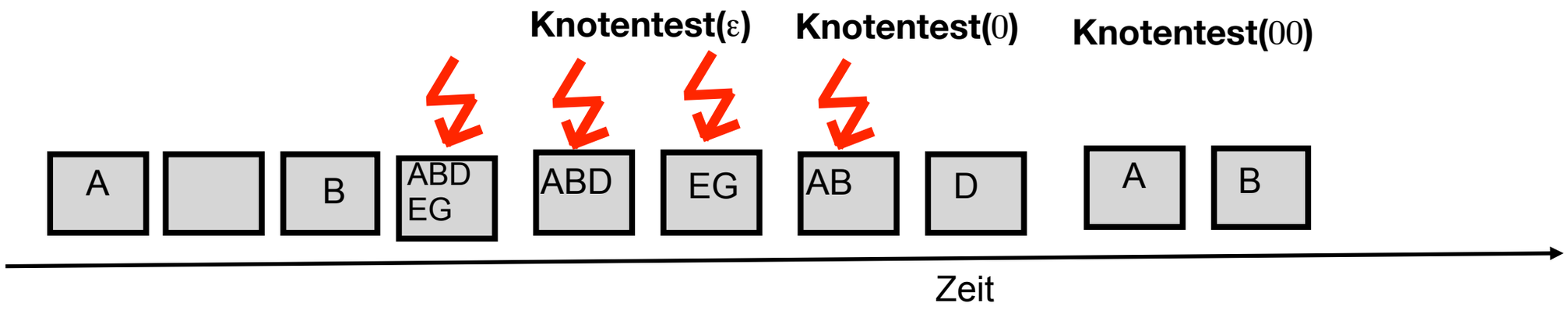


Adaptives Baumprotokoll
Beispiel (4)

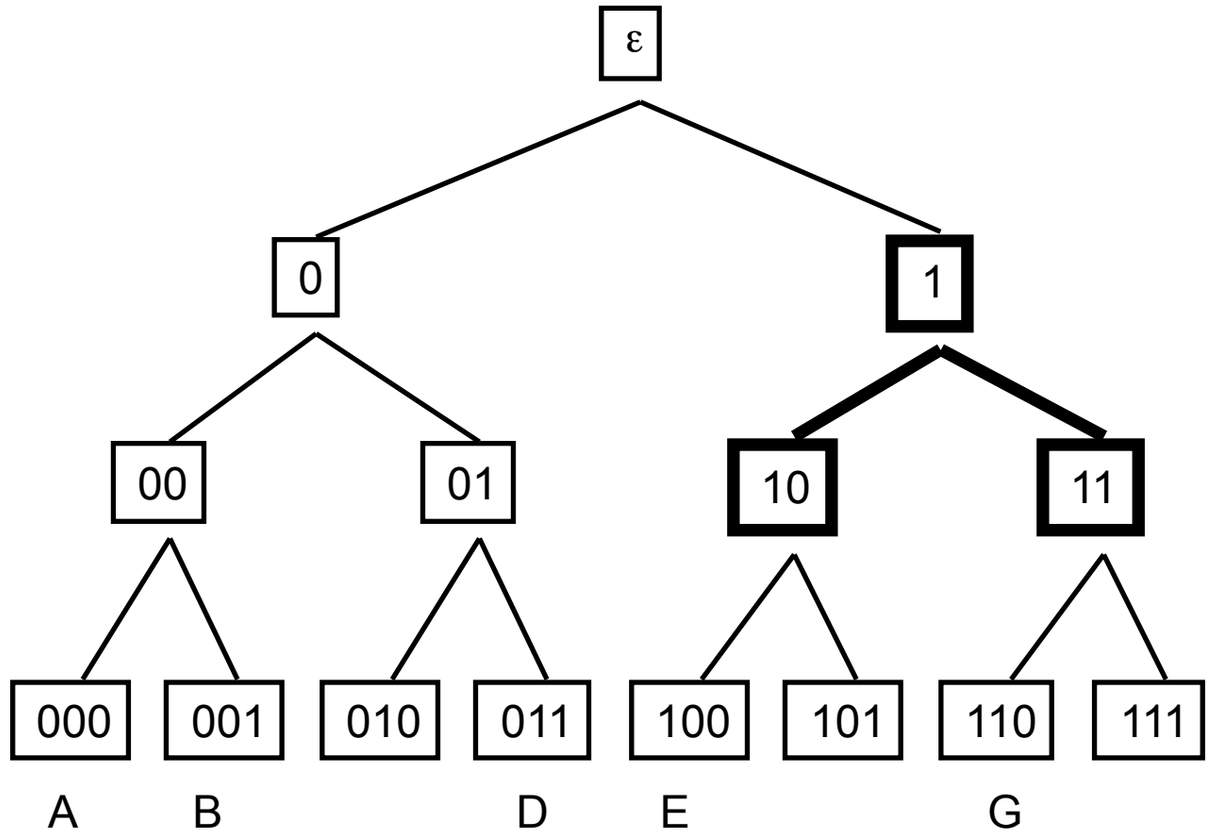


Kollidierende Stationen

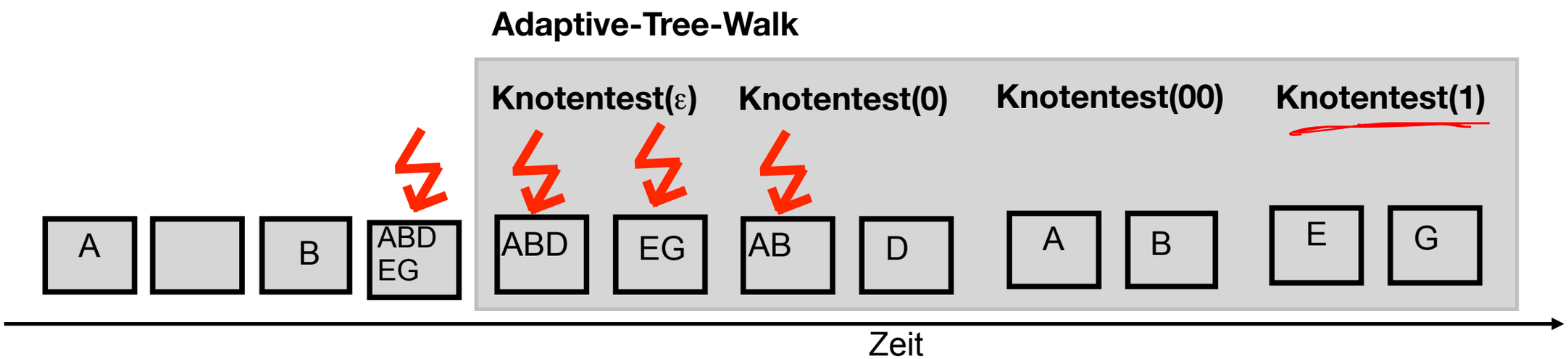
Adaptive-Tree-Walk

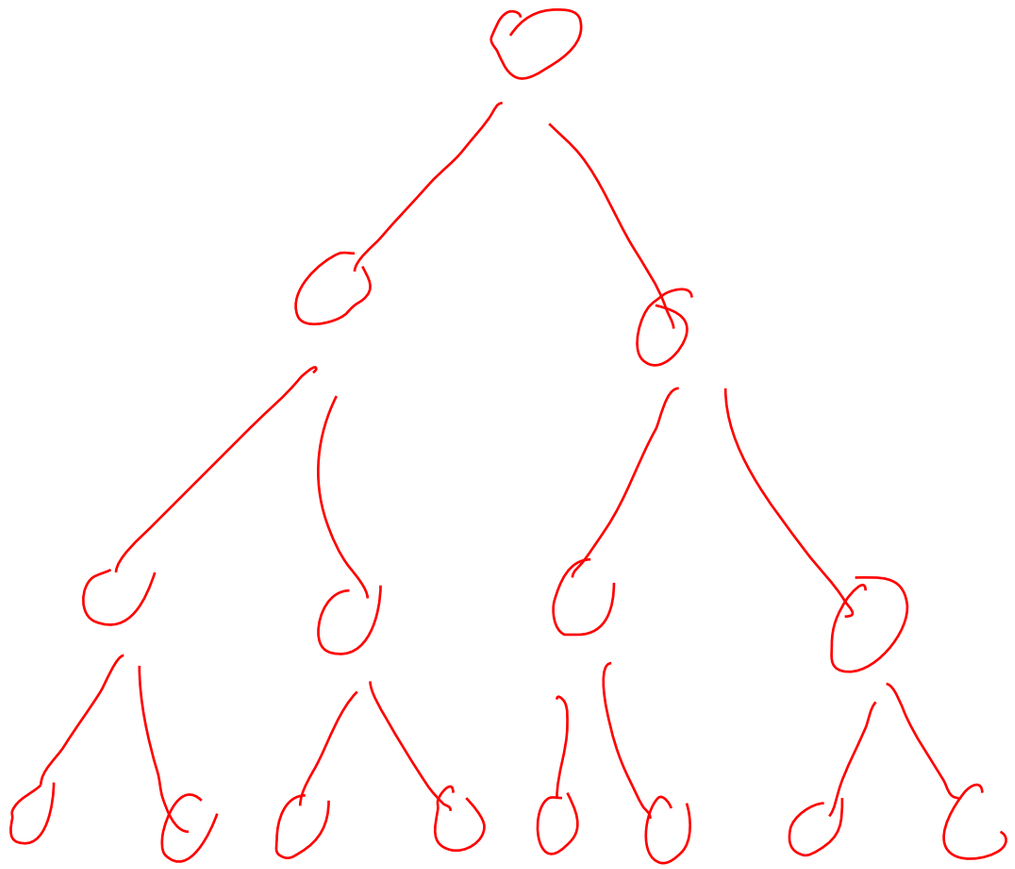


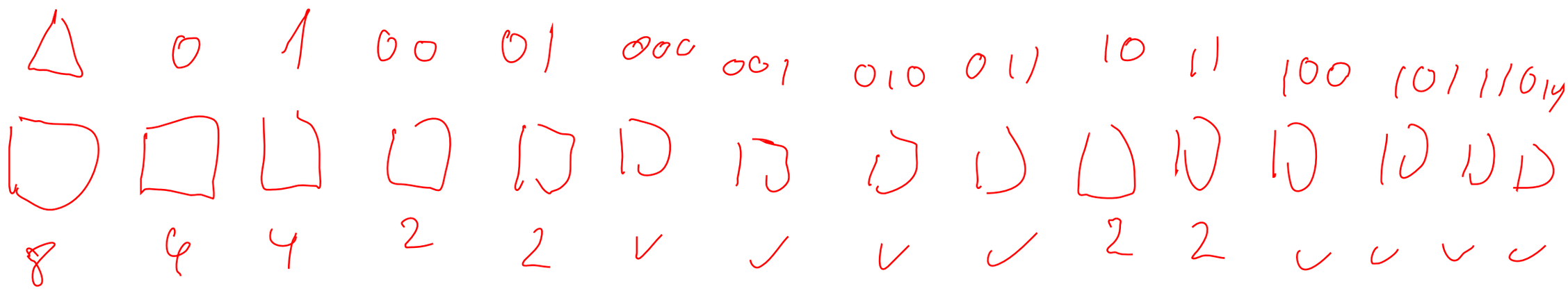
Adaptives Baumprotokoll
Beispiel (5)



Kollidierende Stationen







7 Kollisionen 8

CoNe Freiburg

IIF INSTITUT FÜR INFORMATIK FREIBURG

Systeme II

3. Die Datensicherungsschicht

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

UNI FREIBURG