



Systeme II

6. Woche Mediumzugriff in der Sicherungsschicht

Christian Schindelhauer

Technische Fakultät

Rechnernetze und Telematik

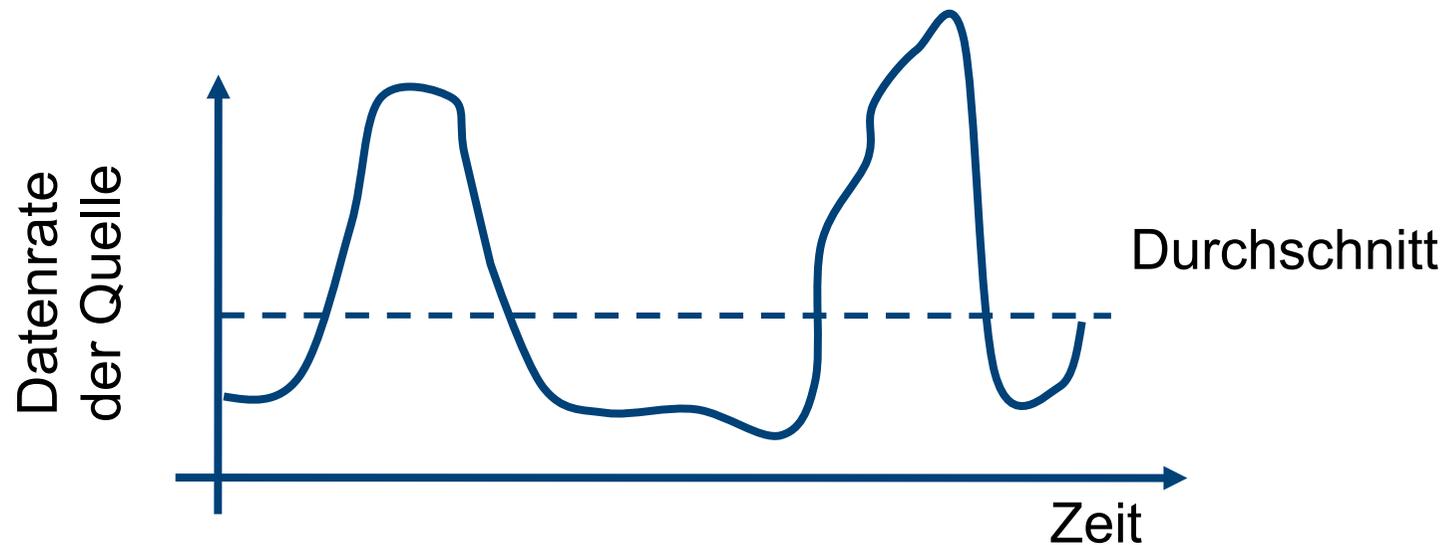
Albert-Ludwigs-Universität Freiburg

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

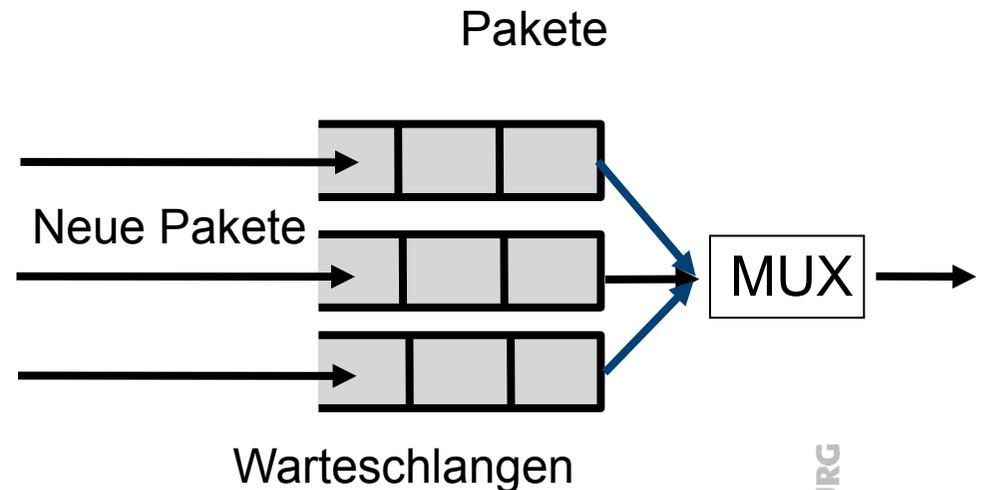
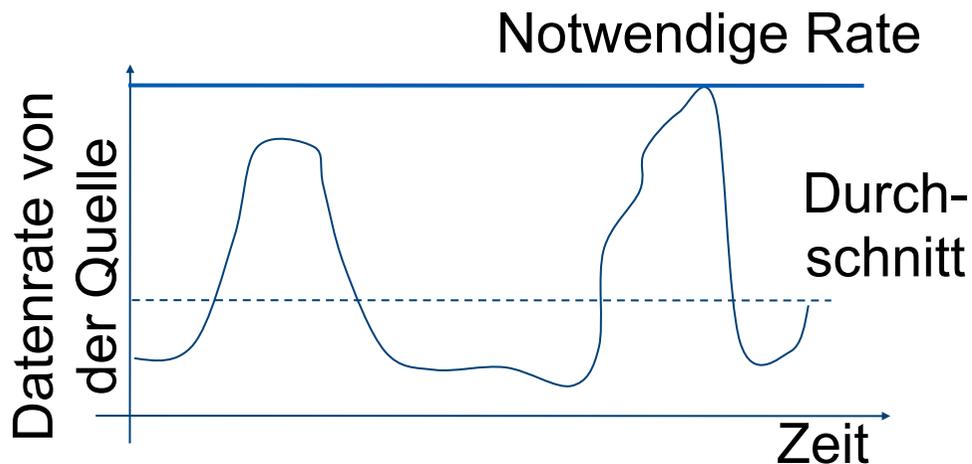
- Statisches Multiplexen
- Dynamische Kanalbelegung
 - Kollisionsbasierte Protokolle
 - Kollisionsfreie Protokolle (contention-free)
 - Protokolle mit beschränktem Wettbewerb (limited contention)
- Fallbeispiel: Ethernet

- Gegeben sei eine einzelne Leitung (Ressource)
- Mehreren Kommunikationsverbindungen 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

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



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



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

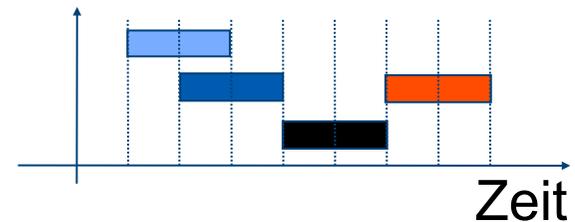
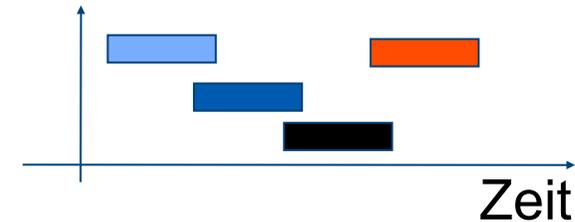
- Statisches Multiplexen
- Dynamische Kanalbelegung
 - Kollisionsbasierte Protokolle
 - Kollisionsfreie Protokolle (contention-free)
 - Protokolle mit beschränktem Wettbewerb (limited contention)
- Fallbeispiel: Ethernet

- 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)

- 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

■ 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

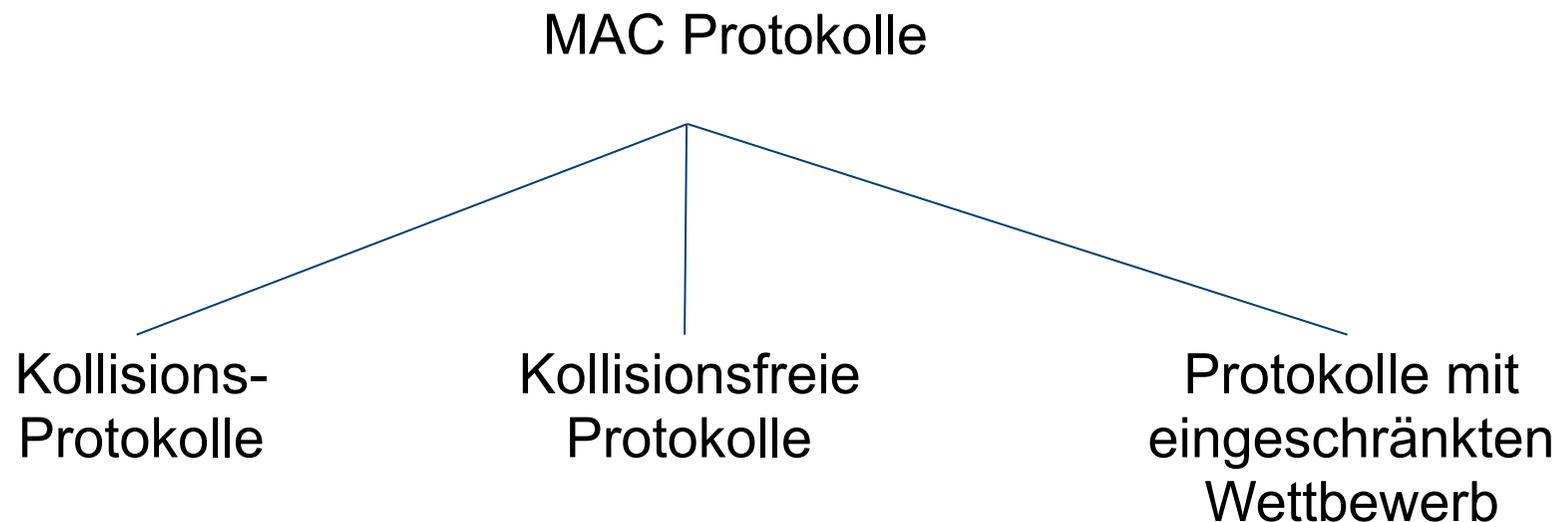


■ Träger-Messung (Carrier Sensing)

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

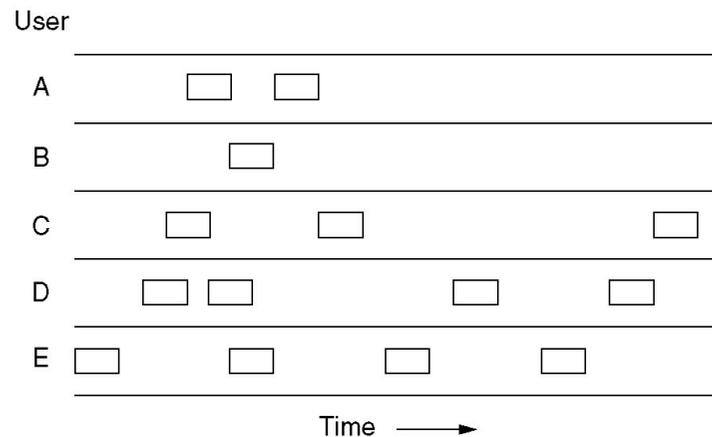
- Methoden zur Bewertung der Effizienz einer Kanalzuweisung
- Durchsatz (throughput)
 - Anzahl Pakete pro Zeiteinheit
 - Besonders bei großer Last wichtig
- Verzögerung (delay)
 - Zeit für den Transport eines Pakets
 - Muss bei geringer Last gut sein
- Gerechtigkeit (fairness)
 - Gleichbehandlung aller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

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

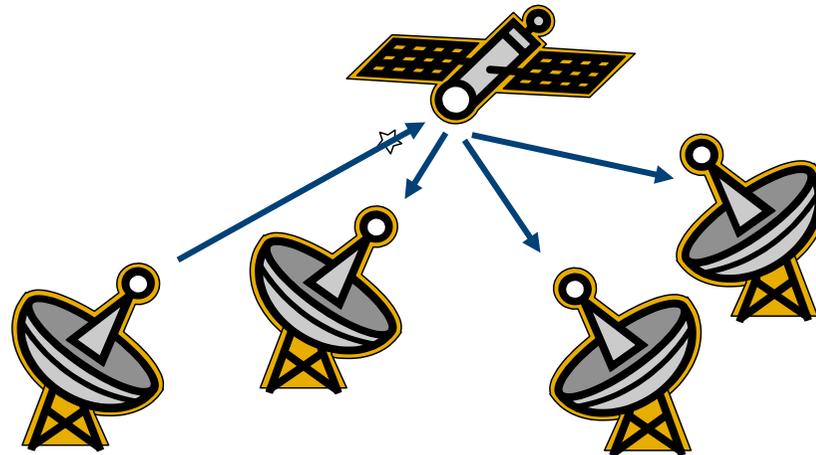


System mit Kollisionen: **Contention System**

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



Pakete werden zu beliebigen Zeiten übertragen



- Vorteile

- Einfach
- Keine Koordination notwendig

- Nachteile

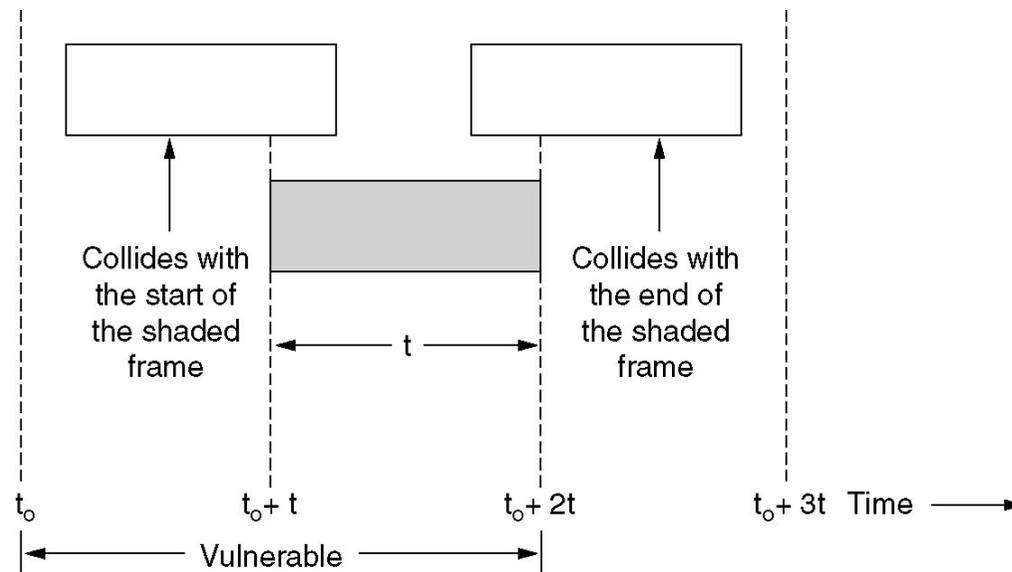
- Kollisionen
 - Sender überprüft den Kanalzustand nicht
- Sender hat keine direkte Methode den Sende-Erfolg zu erfahren
 - Bestätigungen sind notwendig
 - Diese können auch kollidieren

- 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?

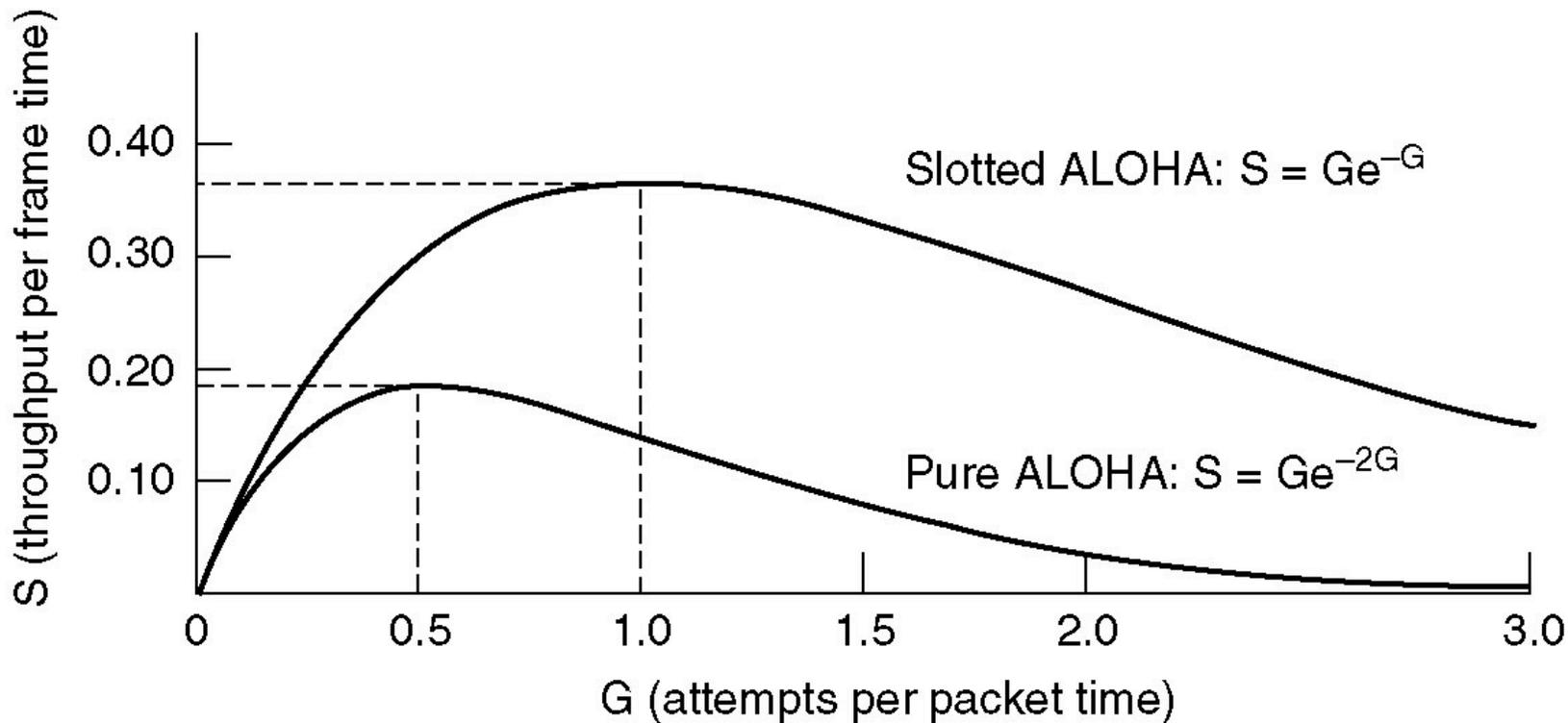
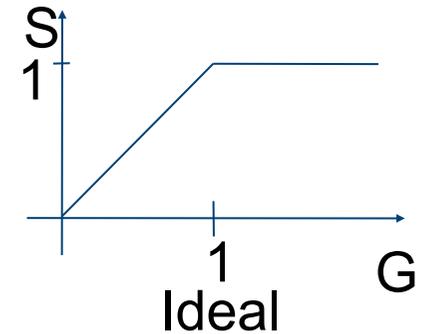
- Ein Paket X wird gestört, wenn
 - ein Paket kurz vor X startet
 - wenn ein Paket kurz vor dem Ende von X startet
- Das Paket wird erfolgreich übertragen, wenn in einem Zeitraum von zwei Paketen kein (anderes) Paket übertragen wird



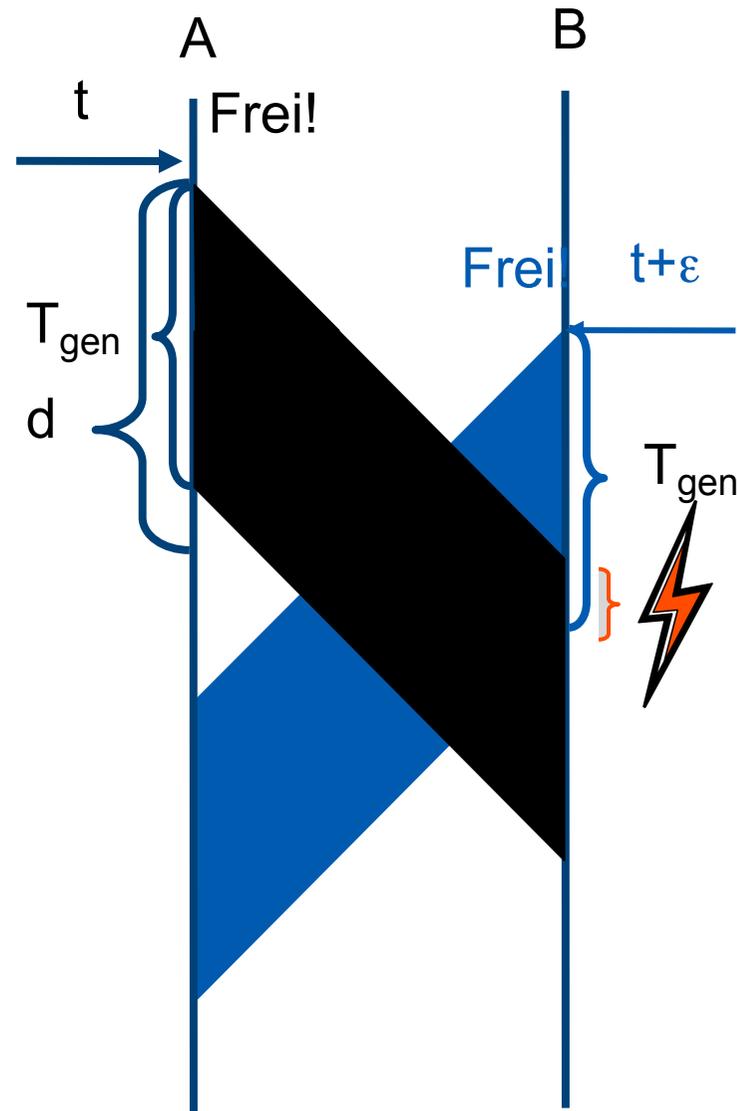
- ALOHAs Problem:
 - Lange Verwundbarkeit eines Pakets
- Reduktion durch Verwendung von Zeitscheiben (Slots)
 - Synchronisation wird vorausgesetzt
- Ergebnis:
 - Verwundbarkeit wird halbiert
 - Durchsatz wird verdoppelt
 - $S(G) = Ge^{-G}$
 - Optimal für $G=1$, $S=1/e$

Durchsatz in Abhängigkeit der Last

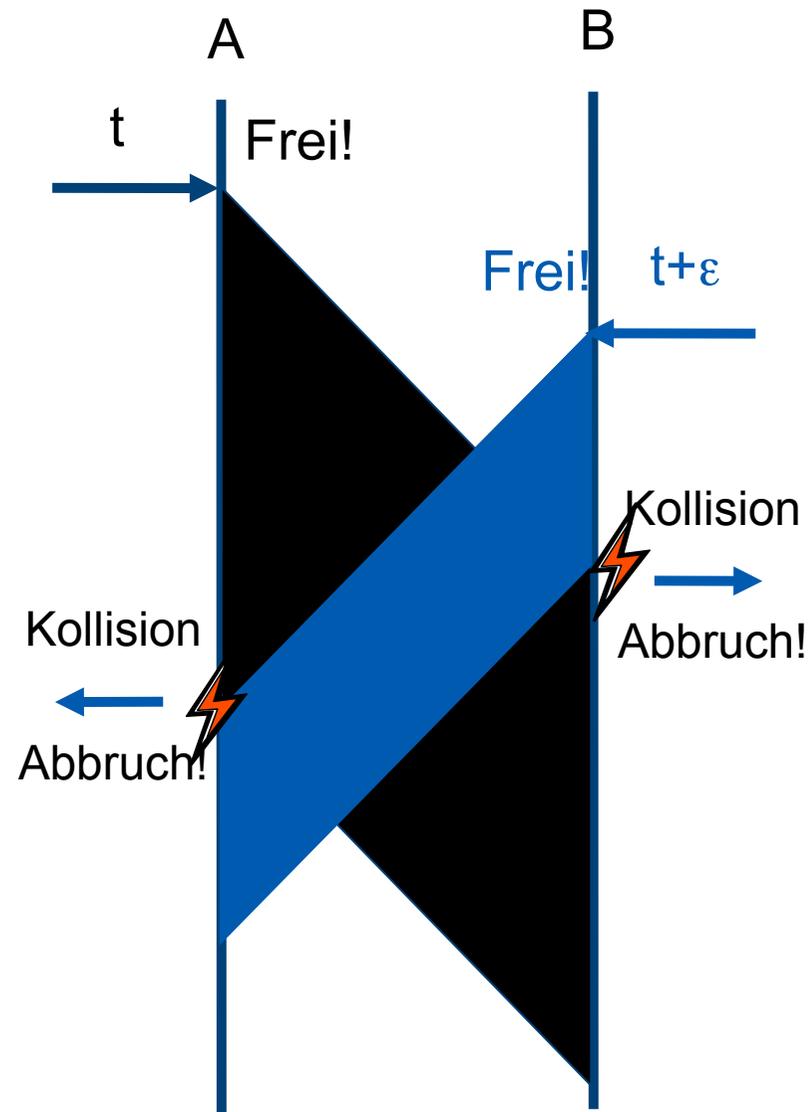
- Für (slotted) ALOHA ist eine geschlossene Darstellung in Abhängigkeit von G möglich
- Kein gutes Protokoll
 - Durchsatz bricht zusammen, wenn die Last zunimmt



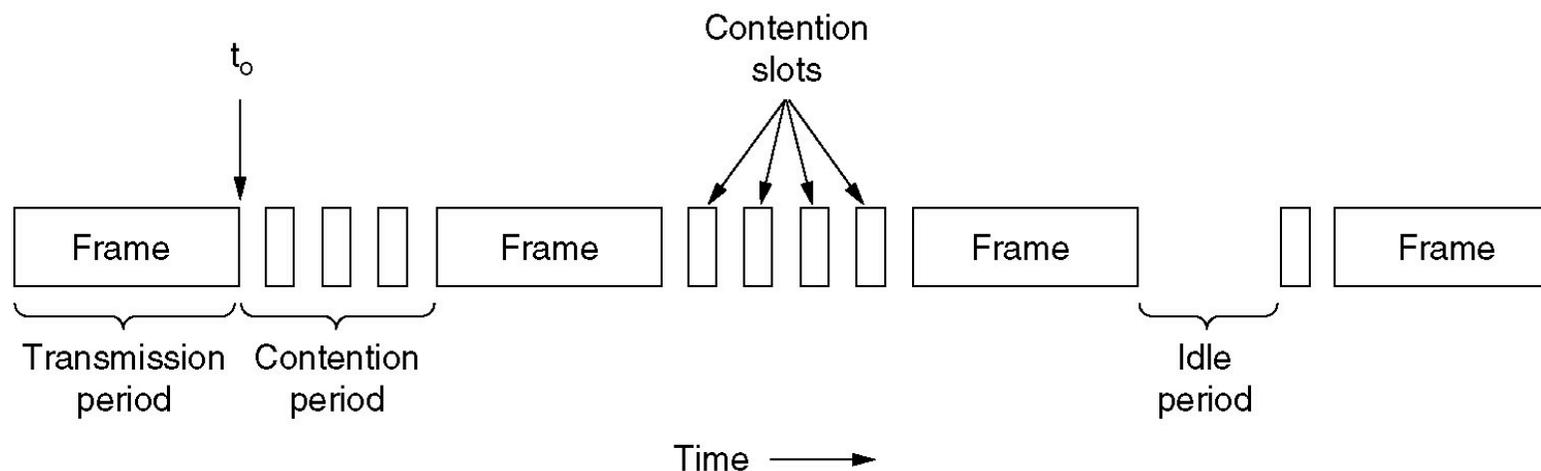
- CSMA-Problem:
 - Übertragungszeit d (propagation delay)
- Zwei Stationen
 - starten Senden zu den Zeitpunkten t und $t+\epsilon$ mit $\epsilon < d$
 - sehen jeweils einen freien Kanal
- Zweite Station
 - verursacht eine Kollision



- 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?



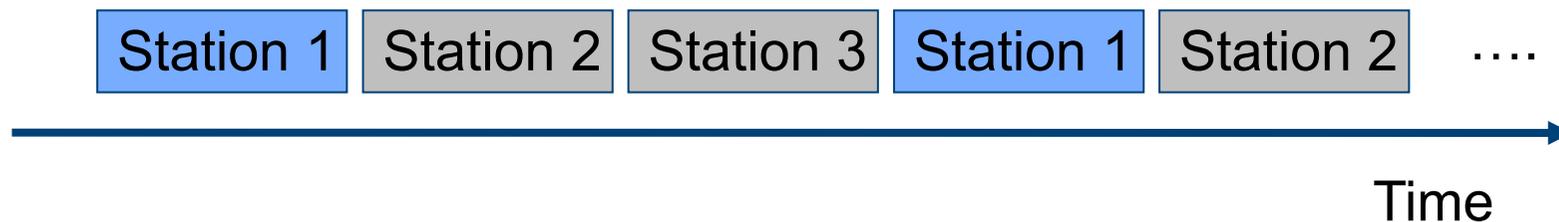
- Leer-Phase (IDLE)
 - Keine Station sendet einen Frame
 - Wettbewerbsphase (Contention Period)
 - Kollisionen entstehen, Übertragungen werden abgebrochen
 - Übertragungsphase (Transmission Period)
 - Keine Kollision, effektiver Teil des Protokolls
- Es gibt nur Wettbewerbs-, Übertragungsphasen und Leer-Phasen



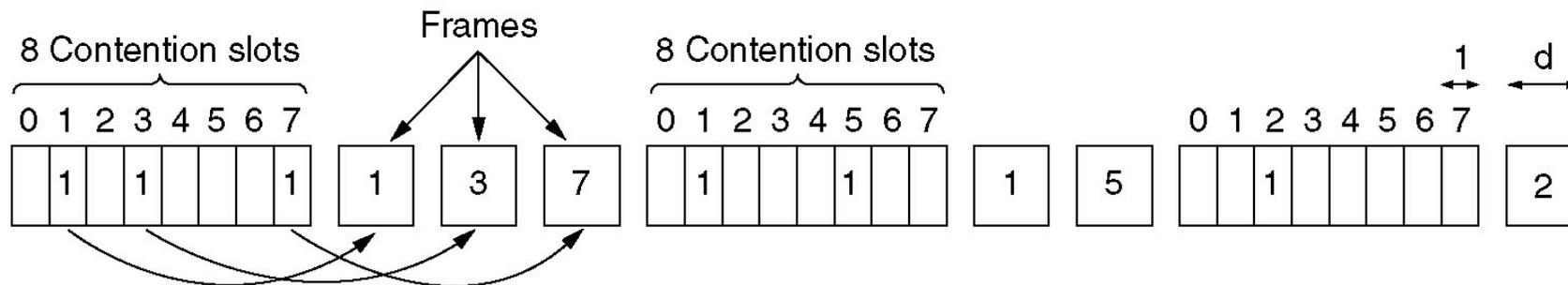
- Nach der Kollision:
- Algorithmus binary exponential backoff
 - $k:=2$
 - Solange Kollision beim letzten Senden
 - Wähle t gleichwahrscheinlich zufällig aus $\{0, \dots, k-1\}$
 - Warte t Zeit-Slots
 - Sende Nachricht (Abbruch bei Collision Detection)
 - $k:= 2 k$
- Algorithmus
 - passt Wartezeit dynamisch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

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

- 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

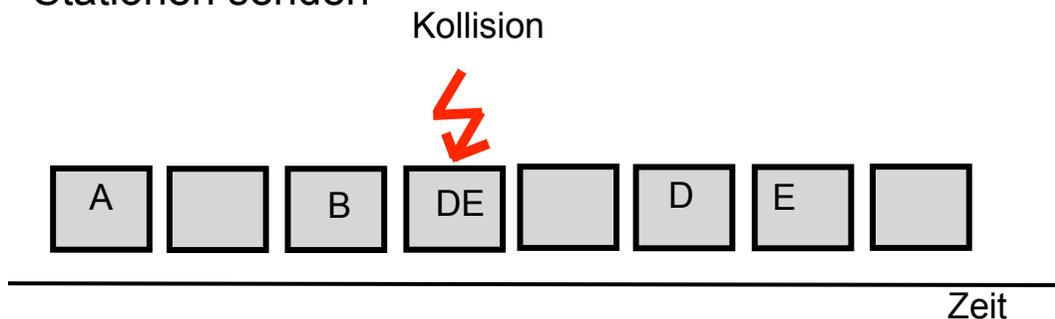
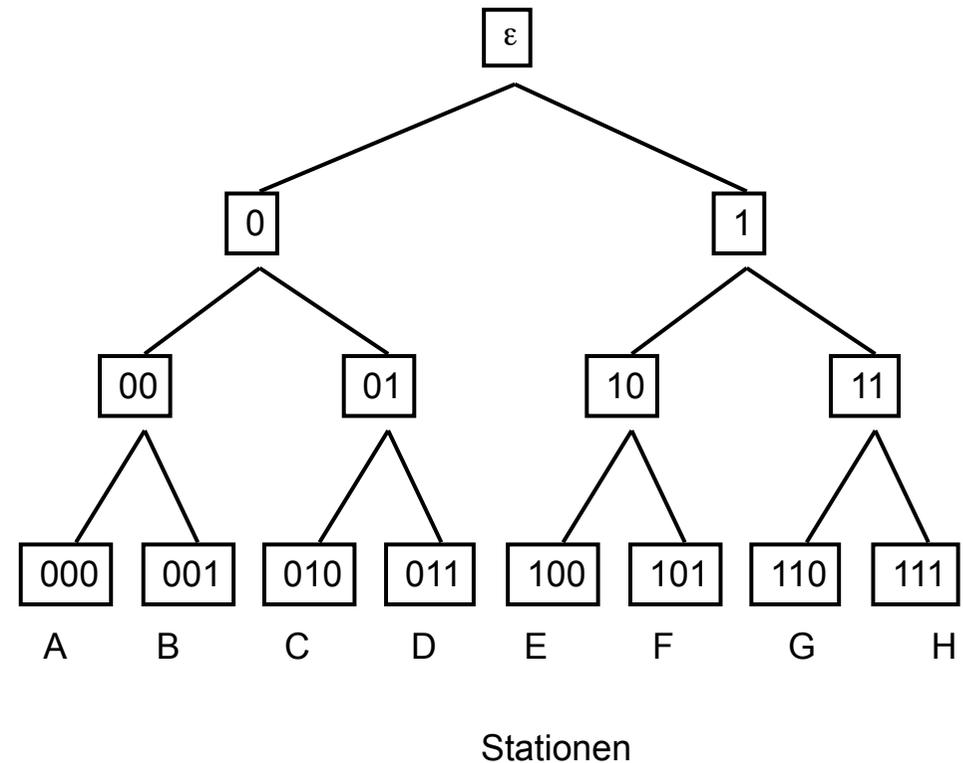


- Verhalten bei geringer Last
 - Falls keine Pakete verschickt werden, wird der (leere) Wettbewerbs-Slot wiederholt
 - Eine Station muss auf seinen Wettbewerbs-Slot warten
 - Erzeugt gewisse Verzögerung (delay)
- Verhalten bei hoher Last
 - Datenpakete dominieren die Kanalbelegung
 - Datenpakete sind 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)
- Fallbeispiel: Ethernet

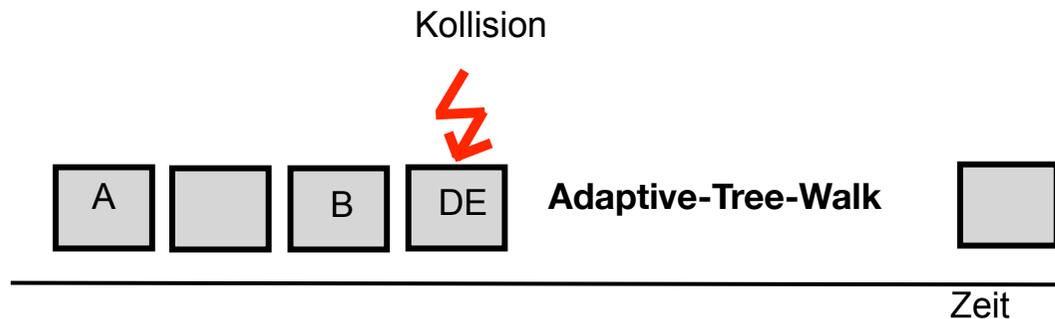
- Ziel
 - geringe Verzögerung bei kleiner Last
 - wie Kollisionsprotokolle
 - hoher Durchsatz bei großer Last
 - wie kollisionsfreie Protokolle
- Idee
 - Anpassung des Wettbewerb-Slots (contention slot) an die Anzahl der teilnehmenden Stationen
 - Mehrere Stationen müssen sich dann diese Slots teilen

- Adaptives Baumprotokoll (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

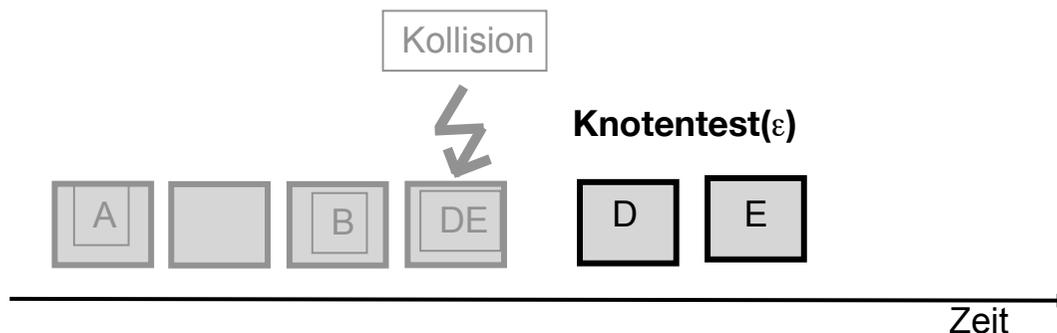
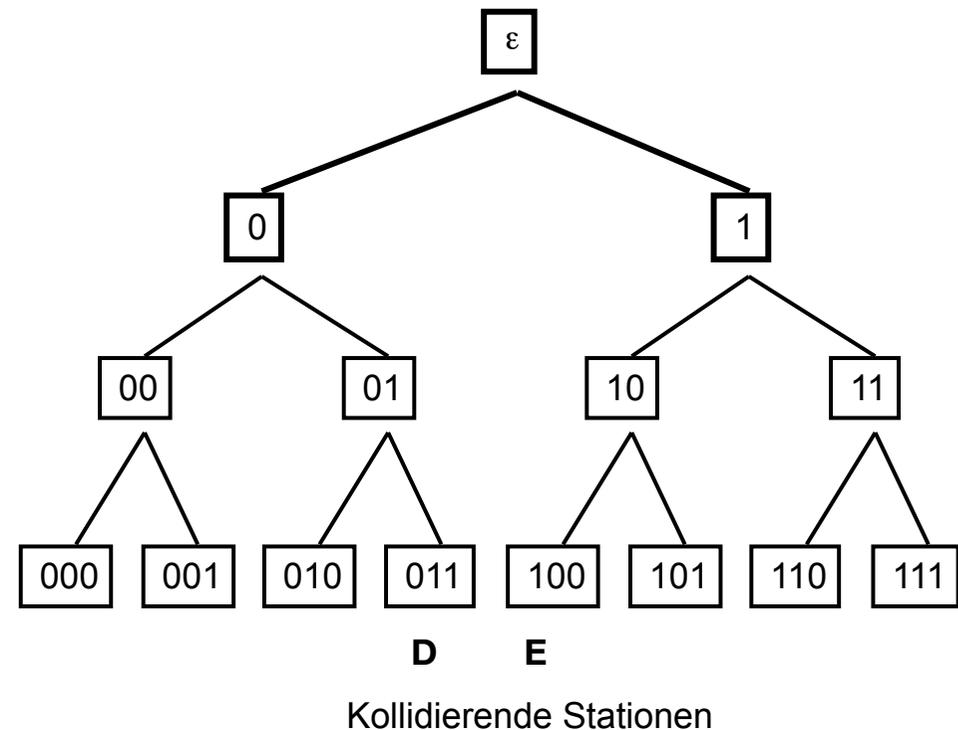


■ Basis-Algorithmus

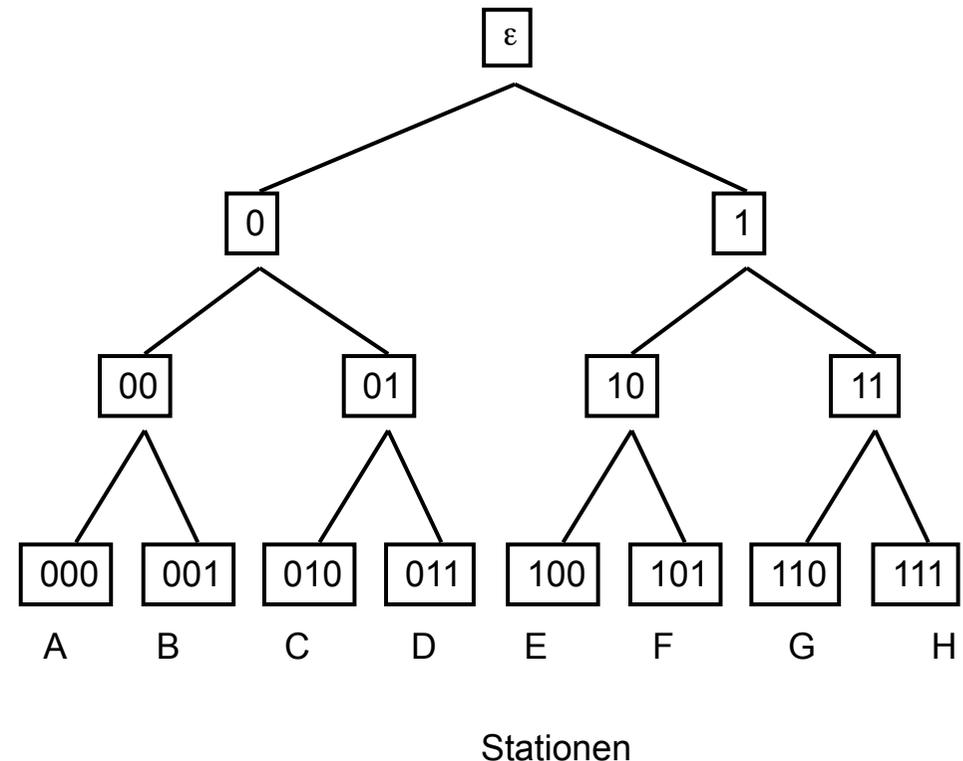
- Jeder Algorithmus sendet sofort (slotted Aloha)
- Falls eine Kollision auftritt,
 - akzeptiert keine Station mehr neue Paket aus der Vermittlungsschicht
 - Führe Adaptive-Tree-Walk(ϵ) aus



- 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

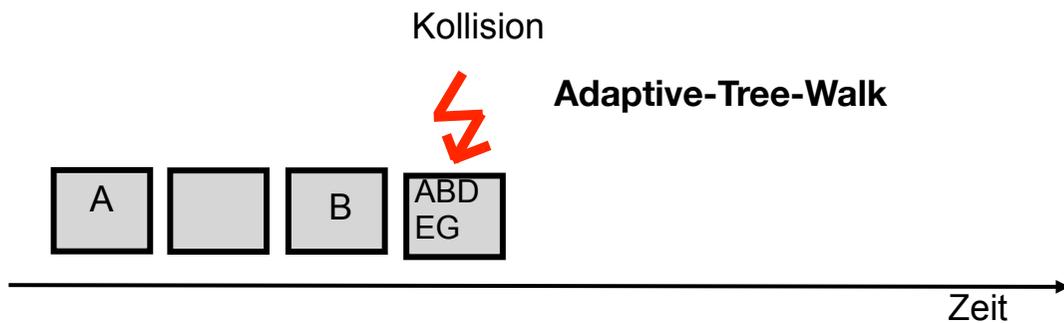
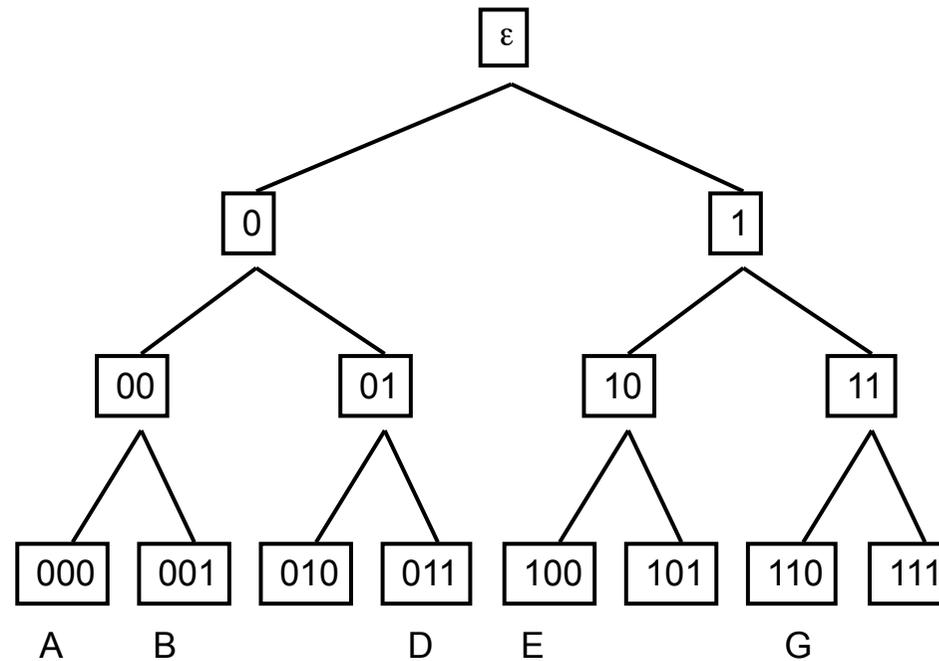


- Algorithmus Knoten-Test
 - für Knoten u des Baums und
 - kollidierende Menge S von Station
- Knoten-Test(u)
 - Betrachte zwei Slots pro Knoten des Baums
 - Im ersten Slot senden alle Knoten aus S , die mit ID $u0$ anfangen
 - Im zweiten Slot senden alle Knoten aus S , die mit ID $u1$ anfangen
- Adaptive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

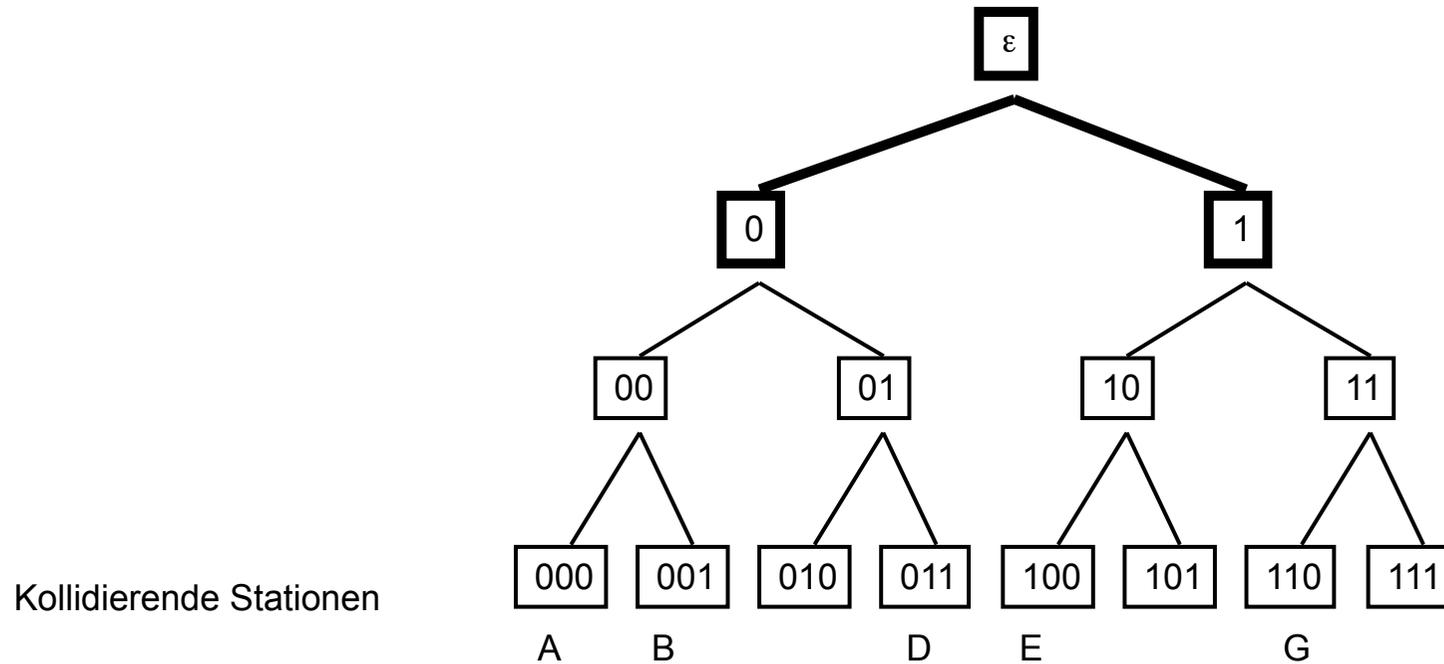


Adaptives Baumprotokoll

Beispiel (1)

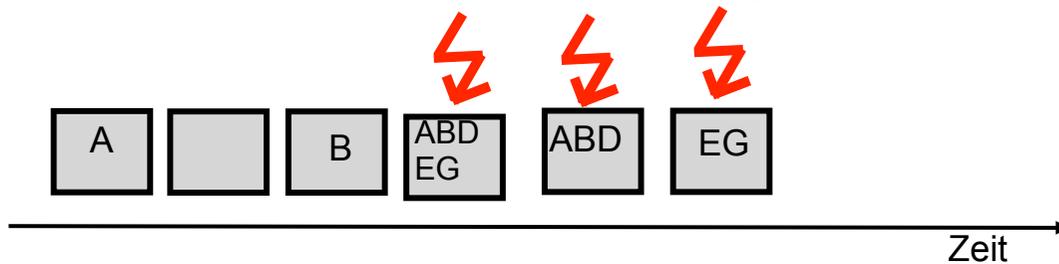


Adaptives Baumprotokoll Beispiel (2)



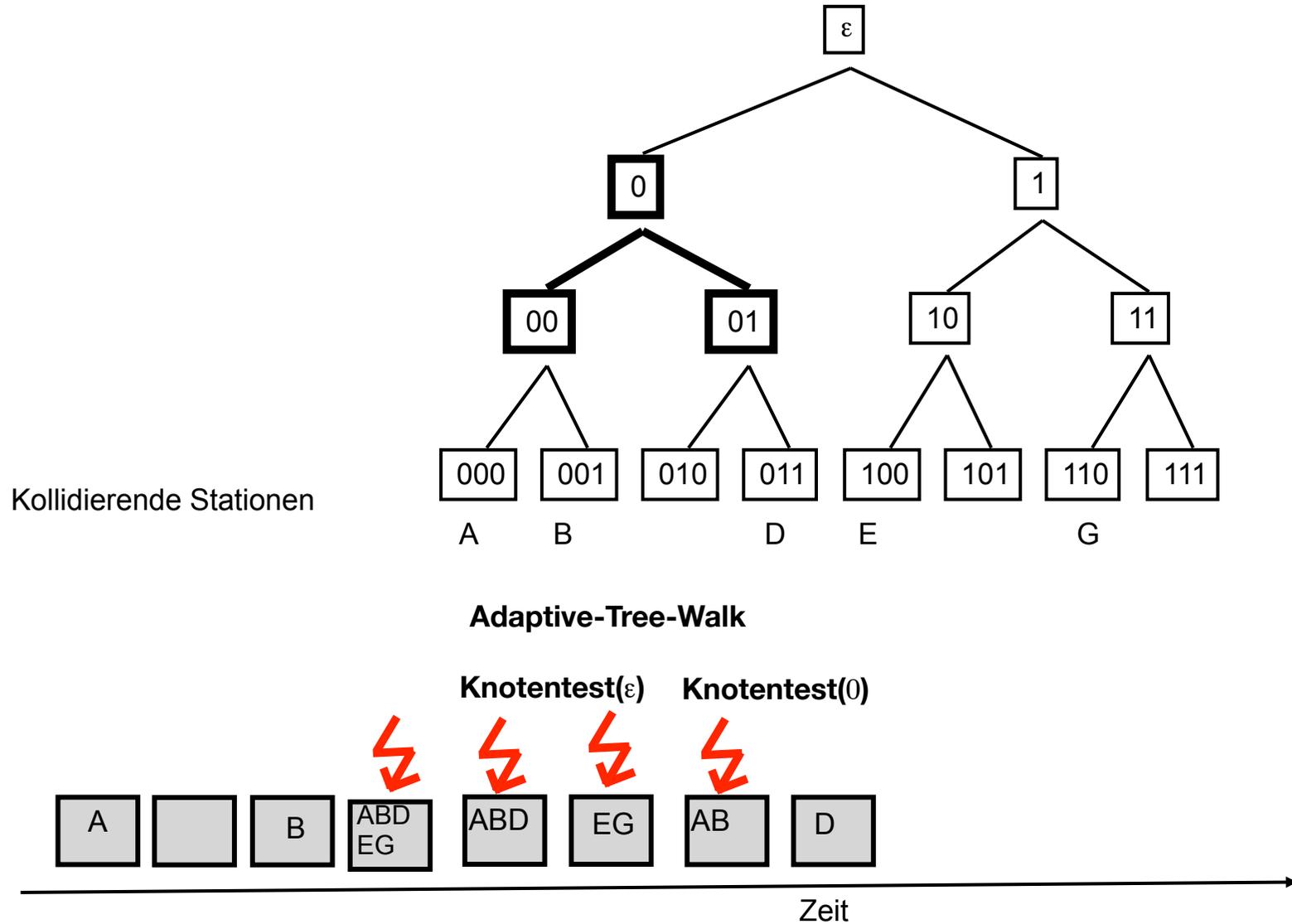
Adaptive-Tree-Walk

Knotentest(ϵ)



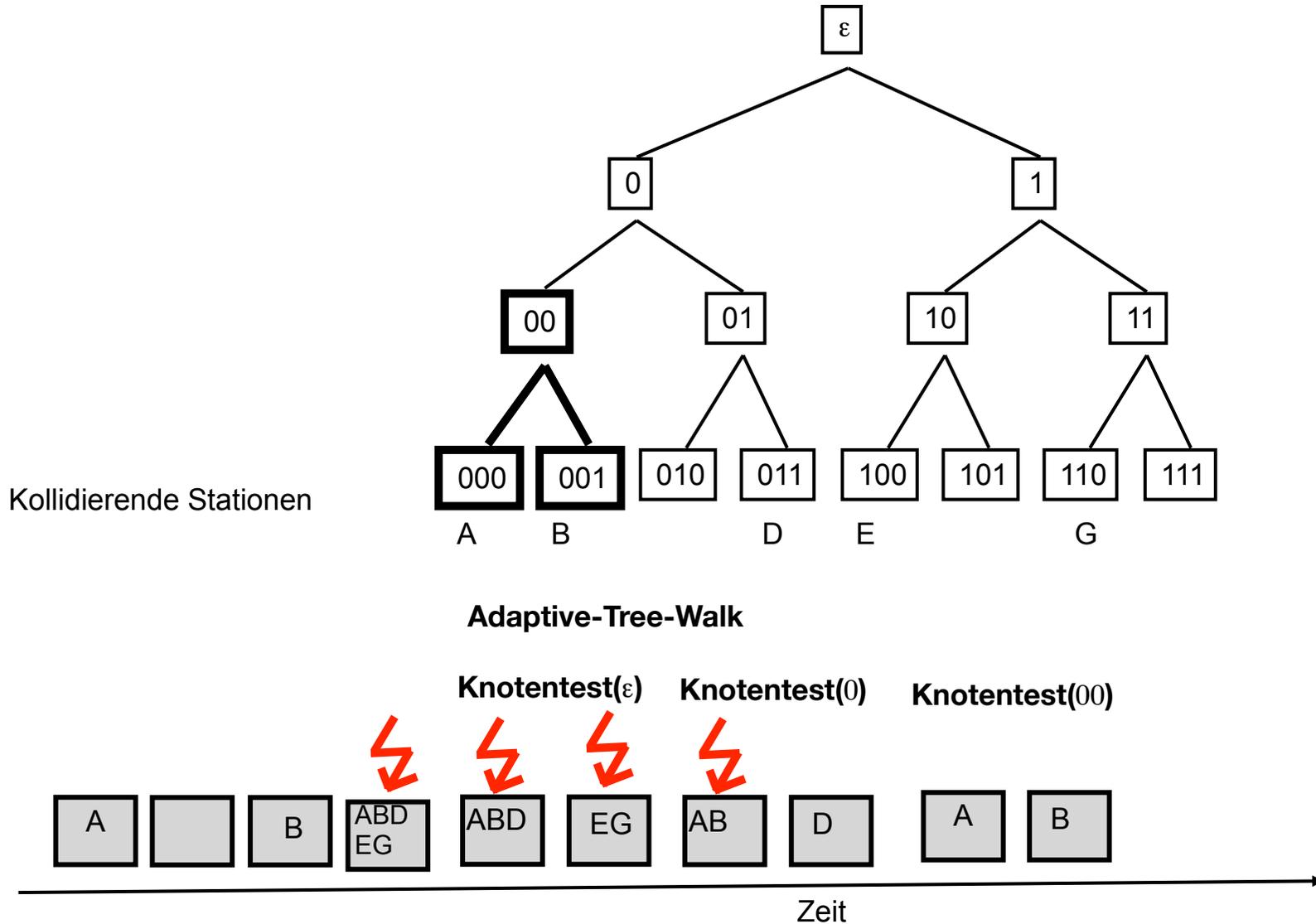
Adaptives Baumprotokoll

Beispiel (3)



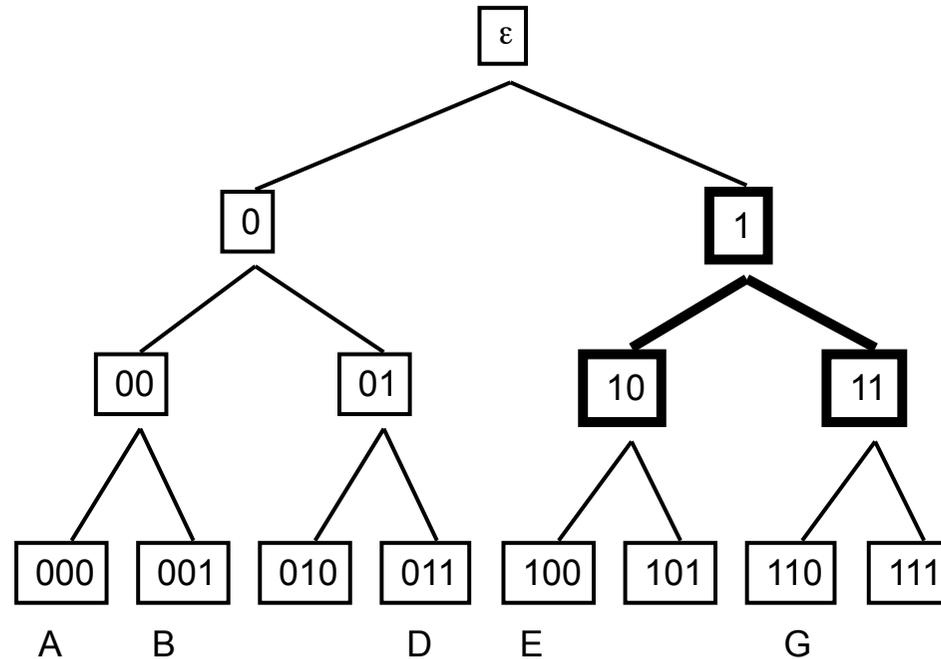
Adaptives Baumprotokoll

Beispiel (4)



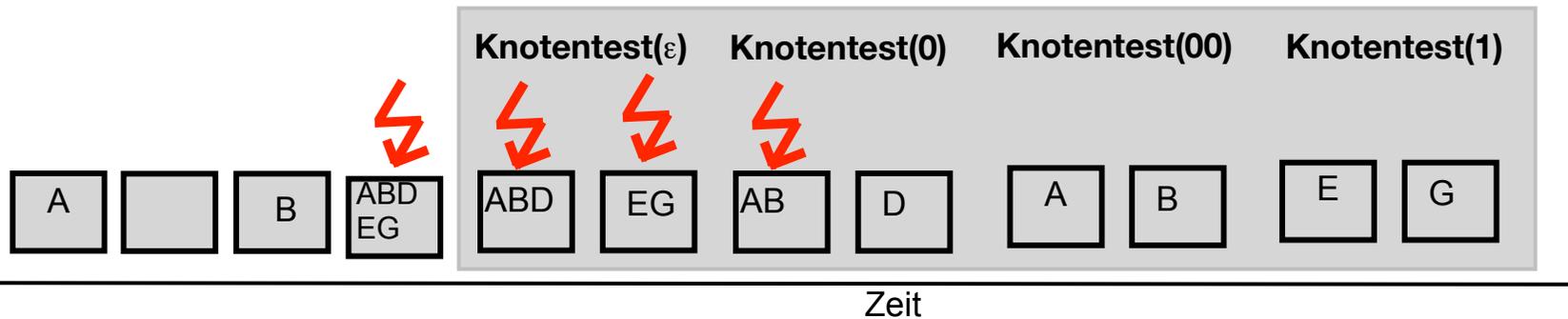
Adaptives Baumprotokoll

Beispiel (5)

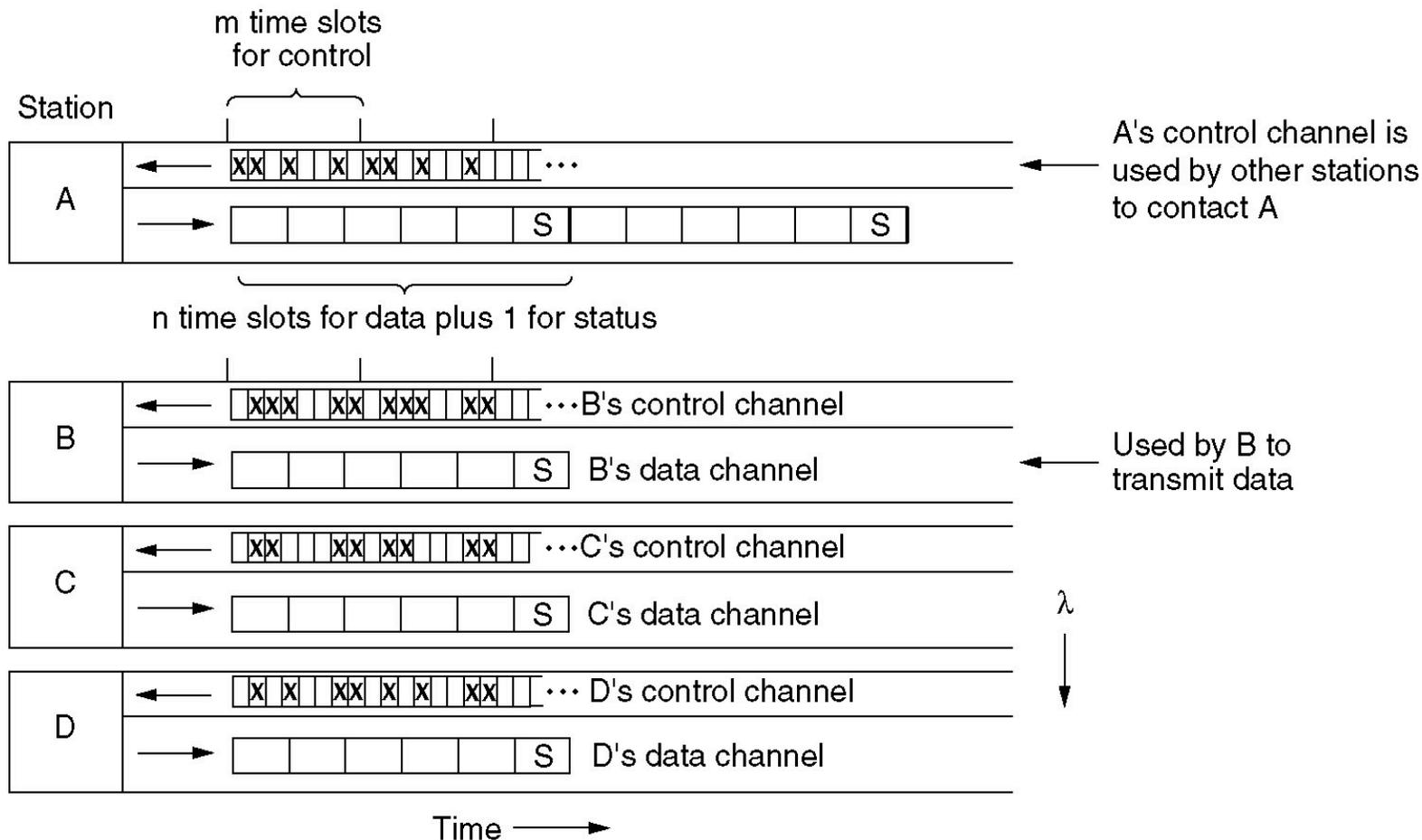


Kollidierende Stationen

Adaptive-Tree-Walk



- Wavelength division multiple access.





Systeme II

6. Woche Mediumzugriff in der Sicherungsschicht

Christian Schindelhauer

Technische Fakultät

Rechnernetze und Telematik

Albert-Ludwigs-Universität Freiburg