Grundlagen Rechnernetze und Verteilte Systeme

SoSe 2013

Kapitel 2: Sicherungsschicht

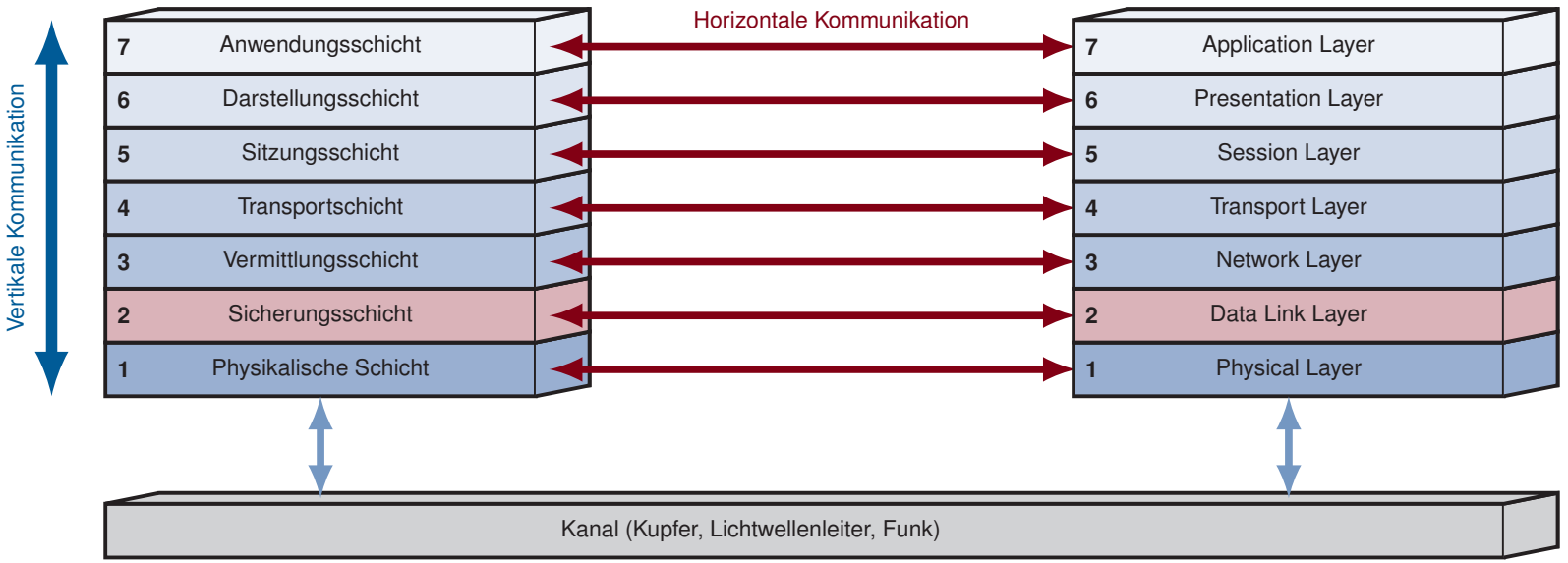
Prof. Dr.-Ing. Georg Carle

Dipl.-Ing. Stephan M. Günther, M.Sc.
Nadine Herold, M.Sc.
Dipl.-Inf. Stephan-A. Posselt

Fakultät für Informatik
Lehrstuhl für Netzarchitekturen und Netzdienste
Technische Universität München

Worum geht es in diesem Kapitel?

Einordnung im ISO/OSI-Modell



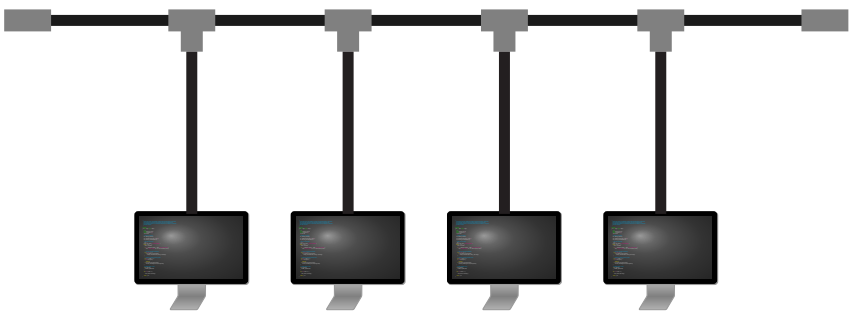
Problemstellung und Motivation

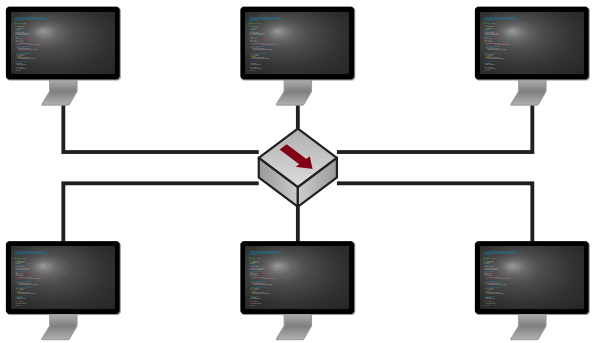
Wir beschäftigen uns zunächst mit sog. Direktverbindungsnetzen, d. h.

- alle angeschlossenen Knoten sind direkt erreichbar und
- werden mittels einfacher Adressen der Schicht 2 identifiziert,
- es findet keine Vermittlung statt,
- eine einfache Weiterleitung (in Form von „Bridging“ oder „Switching“) ist aber möglich.

Beispiele:

- einzelne lokale Netzwerke





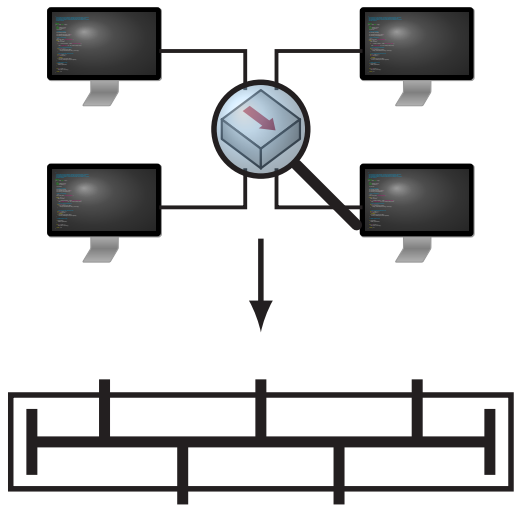
- Verbindung zwischen Basisstation und Mobiltelefon
- Bus-Systeme innerhalb eines Computers, z. B. PCIe

Die wesentlichen Aufgaben der Sicherungsschicht sind

- die Steuerung des Medienzugriffs,

Steuerung des Medienzugriffs:

- Hubs z. B. erzeugen nur auf den ersten Blick eine Sterntopologie
- Intern werden alle angeschlossenen Computer zu einem Bus verbunden
- Gleichzeitiges Senden von zwei Stationen führt zu Kollisionen und daher zum Verlust von Nachrichten

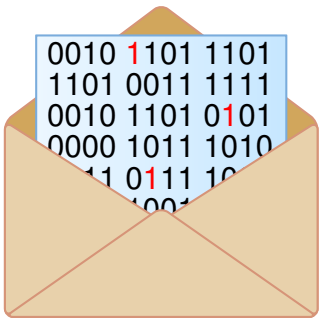


Die wesentlichen Aufgaben der Sicherungsschicht sind

- die Steuerung des Medienzugriffs,
- die Prüfung übertragener Nachrichten auf Fehler und

Prüfung übertragener Nachrichten auf Fehler:

- Trotz Kanalkodierung treten Übertragungsfehler auf
- Diese müssen erkannt werden
- Defekte Nachrichten werden nicht an höhere Schichten weitergegeben
- Die **Wiederholung** einer Übertragung ist häufig Aufgabe höherer Schichten

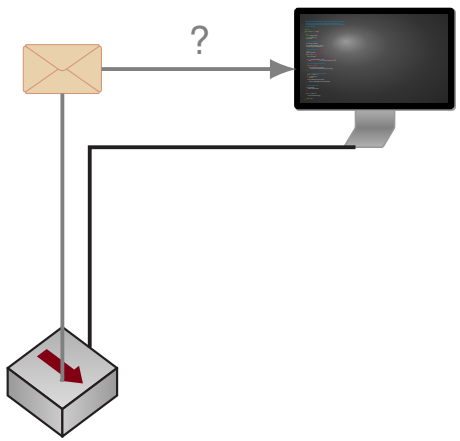


Die wesentlichen Aufgaben der Sicherungsschicht sind

- die Steuerung des Medienzugriffs,
- die Prüfung übertragener Nachrichten auf Fehler und
- die Adressierung innerhalb von Direktverbindungsnetzen.

Adressierung:

- Eine Nachricht kann von vielen Knoten empfangen werden, z. B. bei Bus-Verbindungen oder Funknetzwerken
- Der jeweilige Empfänger muss entscheiden können, ob eine Nachricht für ihn bestimmt ist



Worum geht es in diesem Kapitel?

Darstellung von Netzwerken als Graphen

Motivation

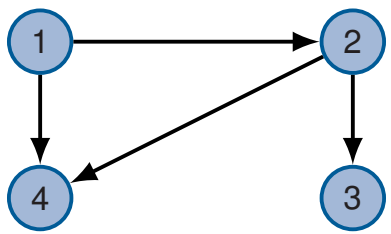
- Zur Darstellung von Netztopologien und Knotenverbindungen werden häufig gerichtete oder ungerichtete Graphen verwendet.
- Im Folgenden führen wir die entsprechende Notation und grundlegende Begriffe ein.

Gerichtete Graphen

Ein asymmetrisches Netzwerk lässt sich als gerichteter Graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ darstellen, wobei

- $\mathcal{N}$ eine Menge von Knoten (Nodes bzw. Vertices) und
- $\mathcal{A} = \{(i, j) \mid i, j \in \mathcal{N} \wedge i, j \text{ sind gerichtet verbunden}\}$ eine Menge gerichteter Kanten (Arcs) bezeichnet.

Beispiel: $\mathcal{N} = \{1, 2, 3, 4\}$, $\mathcal{A} = \{(1, 2), (2, 3), (2, 4), (1, 4)\}$

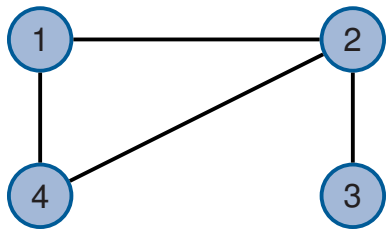


Ungerichtete Graphen

Ein symmetrisches Netzwerk lässt sich als ungerichteter Graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ darstellen, wobei

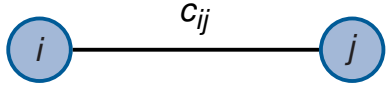
- $\mathcal{N}$ eine Menge von Knoten und
- $\mathcal{E} = \{\{i, j\} \mid i, j \in \mathcal{N} \wedge i, j \text{ sind ungerichtet verbunden}\}$ eine Menge ungerichteter Kanten (Edges) bezeichnet.

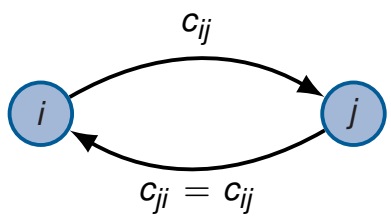
Beispiel: $\mathcal{N} = \{1, 2, 3, 4\}$, $\mathcal{E} = \{\{1, 2\}, \{2, 3\}, \{2, 4\}, \{1, 4\}\}$



Hinweis zur Notation

Ungerichtete Graphen können als gerichtete Graphen mit sym. Kanten verstanden werden. Eine ungerichtete Kante $\{i, j\}$ eines ungerichteten Graphen mit Kantenkosten c_{ij} entspricht also den beiden gerichteten Kanten (i, j) und (j, i) eines gerichteten Graphen mit Kantenkosten $c_{ji} = c_{ij}$.





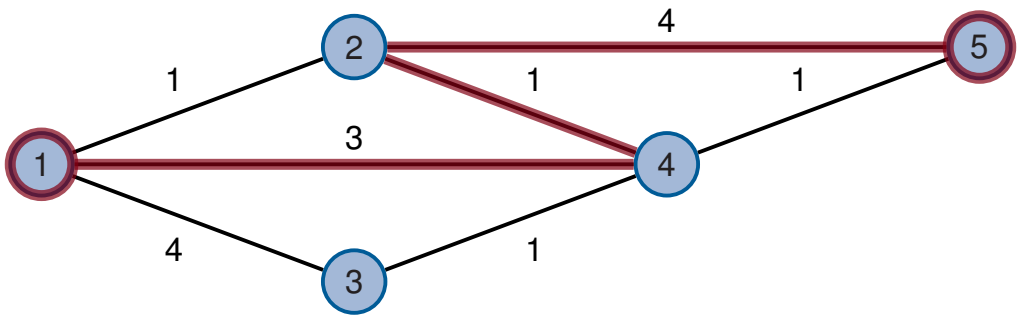
Pfade in Netzwerken

- Ein Pfad zwischen zwei Knoten[1] $s, t \in \mathcal{N}$ ist eine Menge
$$\mathcal{P}_{st} = \{(s, i), (i, j), \ldots, (k, l), (l, t)\}$$
gerichteter Kanten, die s und t miteinander verbinden.

- Die Pfadkosten entsprechen der Summe der Kantenkosten: $c(\mathcal{P}_{st}) = \sum_{(i,j)\in\mathcal{P}_{st}} c_{ij}$.

- Die Pfadlänge entspricht der Anzahl der Kanten auf dem Pfad: $l(\mathcal{P}_{st}) = |\mathcal{P}_{st}|$.
Die Pfadlänge wird auch Hop Count genannt.

Beispiel: $\mathcal{P}_{15} = \{(1, 4), (4, 2), (2, 5)\}$



$$c(\mathcal{P}_{15}) = 3 + 1 + 4 = 8$$
$$l(\mathcal{P}_{15}) = 3$$

[1] Eine Nachrichtenquelle wird häufig mit s (engl. source) abgekürzt, eine Senke mit t (engl. terminal).

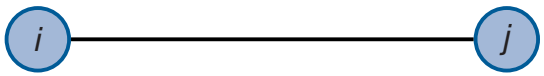
Netztopologien

Die Topologie beschreibt die Struktur, wie Knoten miteinander verbunden sind. Wir unterscheiden die

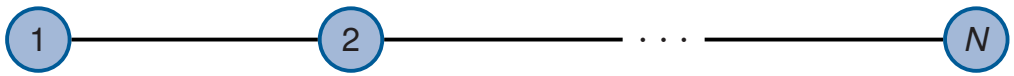
- physikalische Topologie und die
- logische Topologie.

Wichtige Topologien (Beispiele)

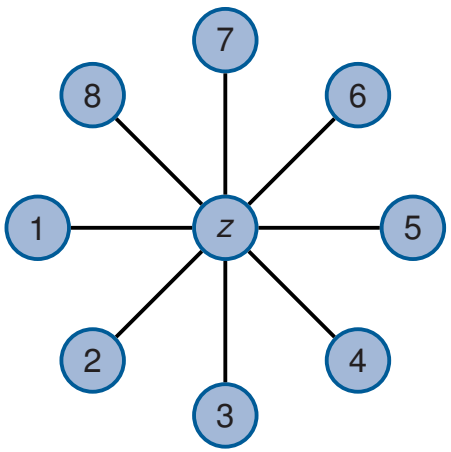
- Punkt-zu-Punkt (engl. Point-to-Point)



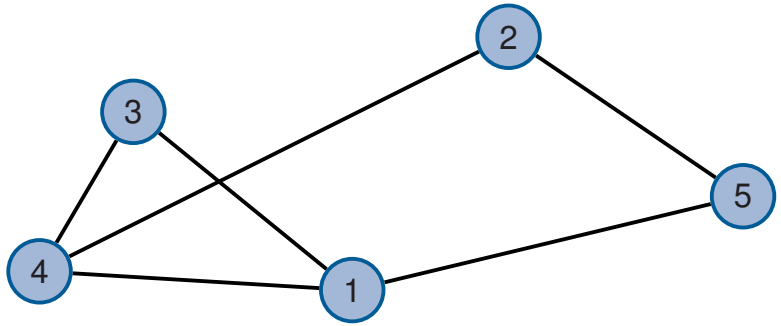
- Kette



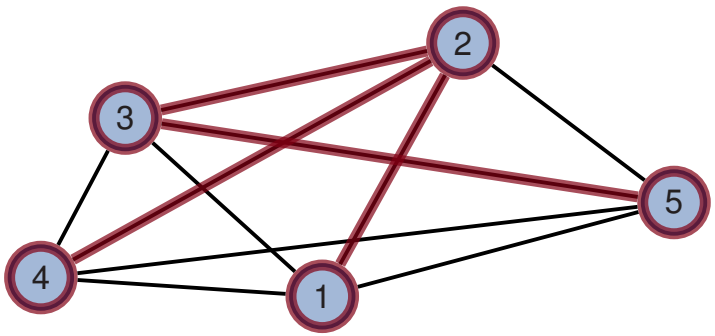
- Stern

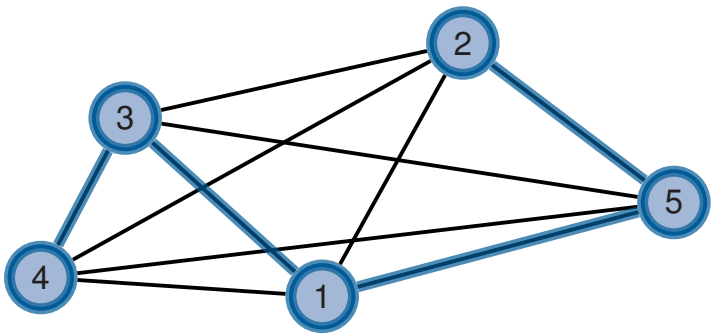


- Vermaschung (engl. Mesh)

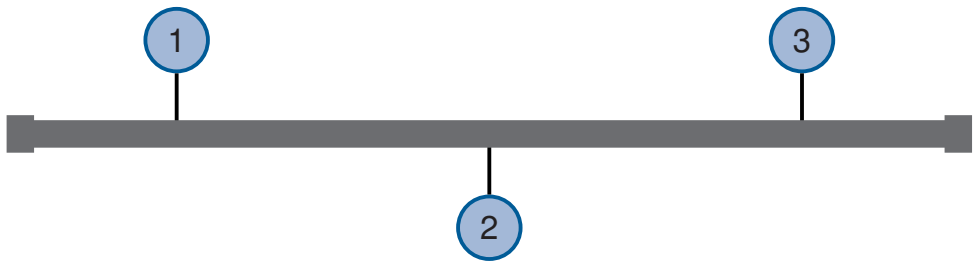


- Baum (meist logische Topologie)





- Bus



Adjazenzmatrix

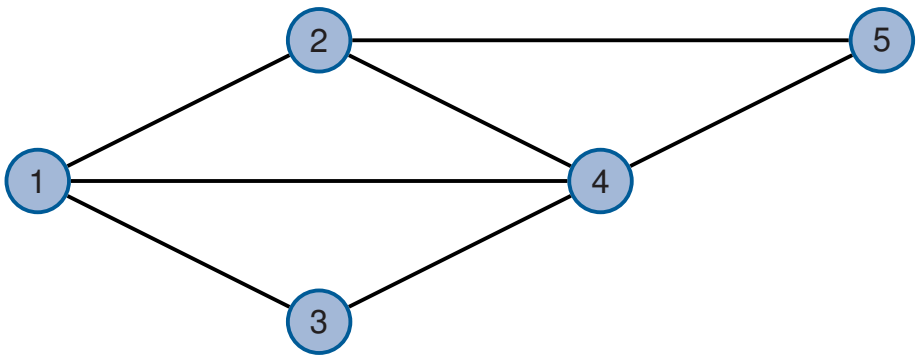
Netzwerke lassen sich leicht als Matrizen schreiben. Die Adjazenzmatrix

$$\boldsymbol{A} = (a)_{ij} = \begin{cases} 1 & \exists (i,j) \in \mathcal{E} \\ 0 & \text{sonst} \end{cases}, \quad \forall i,j \in \mathcal{N}, \qquad \boldsymbol{A} \in \{0,1\}^{N \times N}$$

gibt an, ob Knoten i mit Knoten j verbunden ist.

Beispiel:

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$



- Das Element a_{ij} der Matrix $\boldsymbol{A}$ ist 1, wenn eine Verbindung von Knoten i zu Knoten j besteht.
- $\boldsymbol{A}$ ist symmetrisch ($\boldsymbol{A} = \boldsymbol{A}^T$), wenn die Kanten ungerichtet sind, d. h. zu jeder Kante (i, j) auch eine antiparallele Kante (j, i) existiert.

Distanzmatrix

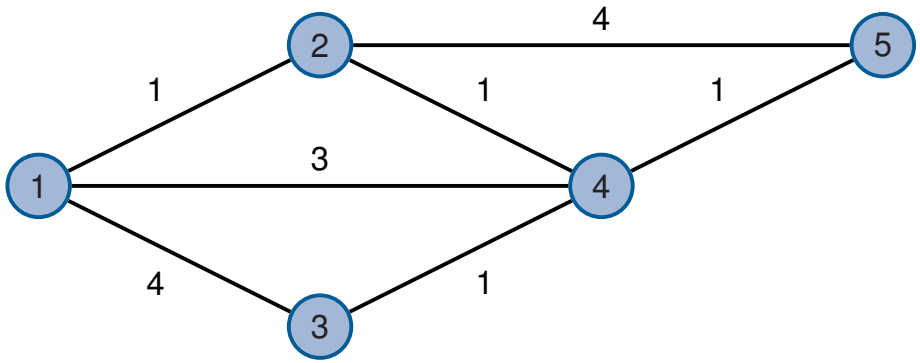
Die Distanzmatrix

$$\boldsymbol{D} = (d)_{ij} = \begin{cases} c_{ij} & \exists (i,j) \in \mathcal{E} \\ 0 & \text{wenn } i = j \ , \quad \forall i,j \in \mathcal{N}, \\ \infty & \text{sonst} \end{cases} \qquad \boldsymbol{D} \in \mathbb{R}_{0+}^{N \times N}$$

enthält die Kosten der Pfade der Länge 1 zwischen allen Knotenpaaren.

Beispiel:

$$\boldsymbol{D} = \begin{bmatrix} 0 & 1 & 4 & 3 & \infty \\ 1 & 0 & \infty & 1 & 4 \\ 4 & \infty & 0 & 1 & \infty \\ 3 & 1 & 1 & 0 & 1 \\ \infty & 4 & \infty & 1 & 0 \end{bmatrix}$$



- Das Element d_{ij} der Matrix $\boldsymbol{D}$ gibt die Distanz zwischen Knoten i und Knoten j an.
- Existiert keine direkte Verbindung zwischen i und j, so ist $d_{ij} = \infty$.
- $\boldsymbol{D}$ ist symmetrisch, wenn das Netzwerk symmetrisch ist, d. h. zu jeder Kante (i, j) auch eine antiparallele Kante (j, i) mit denselben Kosten existiert.

Frage: Wie erhält man die Matrix, welche die Kosten eines kürzesten Pfads zwischen je zwei Knoten enthält?

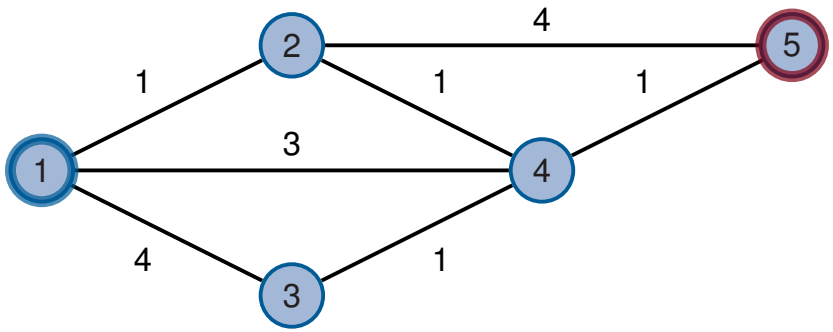
Antwort: Man potenziert $\boldsymbol{D}$ bzgl. des min-plus-Produkts

$$\boldsymbol{D}^n = \boldsymbol{D}^{n-1} \otimes \boldsymbol{D} \text{ mit } d_{ij}^n = \min_{k \in \mathcal{N}} \left\{ d_{ik}^{n-1} + d_{kj} \right\}.$$

- Die Matrix $\boldsymbol{D}^n$ enthält die Länge eines jeweils kürzesten Pfades über höchstens n Hops.
- Für ein endliches n konvergiert die Potenzreihe, so dass $\boldsymbol{D}^{n+1} = \boldsymbol{D}^n = \boldsymbol{D}^*$.

Beispiel: Wie entsteht Element $(1, 5)$ der Matrix $\boldsymbol{D}^2$?

$$\boldsymbol{D} = \begin{bmatrix} 0 & 1 & 4 & 3 & \infty \\ 1 & 0 & \infty & 1 & 4 \\ 4 & \infty & 0 & 1 & \infty \\ 3 & 1 & 1 & 0 & 1 \\ \infty & 4 & \infty & 1 & 0 \end{bmatrix}$$



Frage: Wie erhält man die Matrix, welche die Kosten eines kürzesten Pfads zwischen je zwei Knoten enthält?

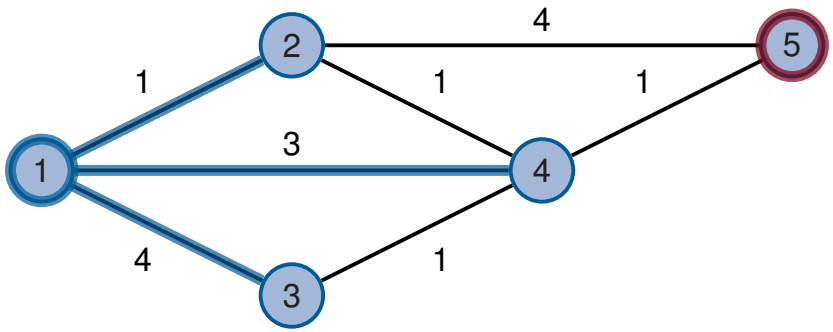
Antwort: Man potenziert $\boldsymbol{D}$ bzgl. des min-plus-Produkts

$$\boldsymbol{D}^n = \boldsymbol{D}^{n-1} \otimes \boldsymbol{D} \text{ mit } d_{ij}^n = \min_{k \in \mathcal{N}} \left\{ d_{ik}^{n-1} + d_{kj} \right\}.$$

- Die Matrix $\boldsymbol{D}^n$ enthält die Länge eines jeweils kürzesten Pfades über höchstens n Hops.
- Für ein endliches n konvergiert die Potenzreihe, so dass $\boldsymbol{D}^{n+1} = \boldsymbol{D}^n = \boldsymbol{D}^*$.

Beispiel: Wie entsteht Element $(1, 5)$ der Matrix $\boldsymbol{D}^2$?

$$\boldsymbol{D} = \begin{bmatrix} 0 & 1 & 4 & 3 & \infty \\ 1 & 0 & \infty & 1 & 4 \\ 4 & \infty & 0 & 1 & \infty \\ 3 & 1 & 1 & 0 & 1 \\ \infty & 4 & \infty & 1 & 0 \end{bmatrix}$$



- Zeile 1 gibt die Kosten eines jeweils kürzesten Pfades der Länge höchstens 1 von Knoten 1 zu allen anderen Knoten an,

Frage: Wie erhält man die Matrix, welche die Kosten eines kürzesten Pfads zwischen je zwei Knoten enthält?

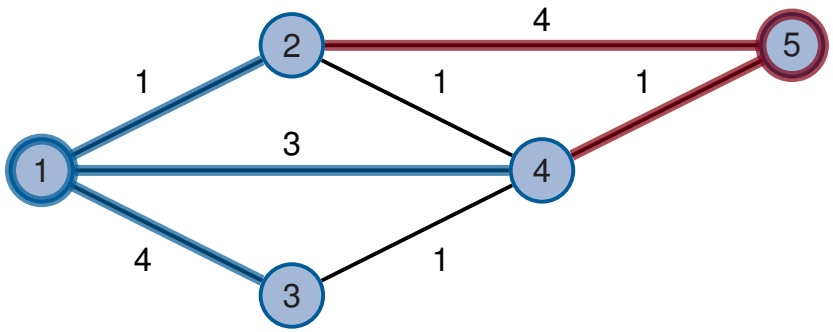
Antwort: Man potenziert $\boldsymbol{D}$ bzgl. des min-plus-Produkts

$$\boldsymbol{D}^n = \boldsymbol{D}^{n-1} \otimes \boldsymbol{D} \text{ mit } d_{ij}^n = \min_{k \in \mathcal{N}} \left\{ d_{ik}^{n-1} + d_{kj} \right\}.$$

- Die Matrix $\boldsymbol{D}^n$ enthält die Länge eines jeweils kürzesten Pfades über höchstens n Hops.
- Für ein endliches n konvergiert die Potenzreihe, so dass $\boldsymbol{D}^{n+1} = \boldsymbol{D}^n = \boldsymbol{D}^*$.

Beispiel: Wie entsteht Element $(1, 5)$ der Matrix $\boldsymbol{D}^2$?

$$\boldsymbol{D} = \begin{bmatrix} 0 & 1 & 4 & 3 & \infty \\ 1 & 0 & \infty & 1 & 4 \\ 4 & \infty & 0 & 1 & \infty \\ 3 & 1 & 1 & 0 & 1 \\ \infty & 4 & \infty & 1 & 0 \end{bmatrix}$$



- Zeile 1 gibt die Kosten eines jeweils kürzesten Pfades der Länge höchstens 1 von Knoten 1 zu allen anderen Knoten an,
- Spalte 5 gibt die Kosten an, mit denen Knoten 5 von allen anderen Knoten über einen kürzesten Pfad der Länge höchstens 1 erreicht werden kann.

Wie oft muss multipliziert werden?

- Der Wert n, so dass $\boldsymbol{D}^n = \boldsymbol{D}^{n+1} = \boldsymbol{D}^*$ gilt, ist durch den längsten einfachen Pfad im Netzwerk beschränkt.
- Der längste einfache Pfad ist durch die Anzahl N der Knoten beschränkt.

$$\Rightarrow n < N$$

Im vorherigen Beispiel reicht bereits $n = 3$ aus, obwohl $N = 5$ gilt.

Die Matrix $\boldsymbol{D}^*$ enthält die Kosten eines jeweils kürzesten Pfades zwischen je zwei Knoten und löst damit das All-pair-shortest-distance-Problem (apsd).

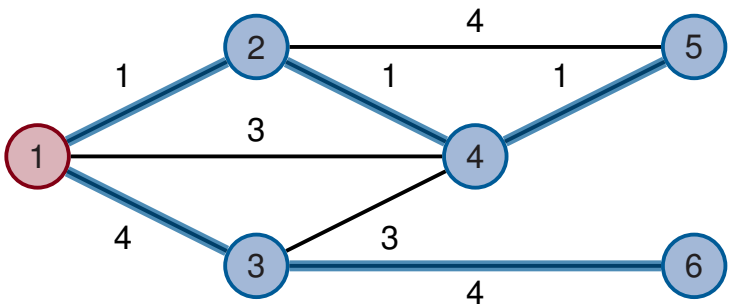
Generierung von Baumstrukturen

Ein Baum ist ein zusammenhängender aber schleifenfreier Graph. Wir unterscheiden im folgenden zwei spezielle Arten von Bäumen:

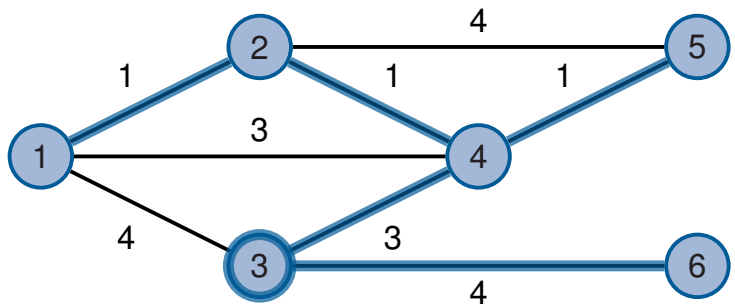
- Shortest Path Tree (SPT)
  Verbindet einen Wurzelknoten mit jeweils minimalen Kosten mit jedem anderen Knoten des Netzwerks.
- Minimum Spanning Tree (MST)
  Verbindet alle Knoten des Netzwerks mit insgesamt minimalen Kosten.

Diese Bäume minimieren unterschiedliche Metriken und sind i. A. **nicht identisch**.

Beispiel:



(a) Shortest Path Tree (SPT) mit Wurzelknoten 1



(b) Minimum Spanning Tree (MST)

In Kapitel 3 werden wir zwei Algorithmen zur Erzeugung von SPTs kennen lernen / wiederholen:

- Algorithmus von Bellman-Ford (basiert auf dem min-plus-Produkt)
- Dijkstras-Algorithmus (Greedy-Prinzip)

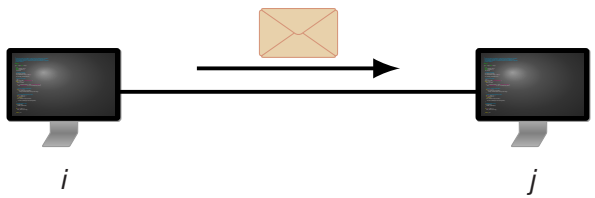
Übersicht

Verbindungscharakterisierung

Eine Verbindung zwischen zwei Knoten kann hinsichtlich einiger grundlegender Eigenschaften charakterisiert werden:

- Übertragungsrate
- Übertragungsverzögerung
- Übertragungsrichtung
- Mehrfachzugriff (Multiplexing)

Zunächst betrachten wir eine Punkt-zu-Punkt-Verbindung:



Übertragungsrate

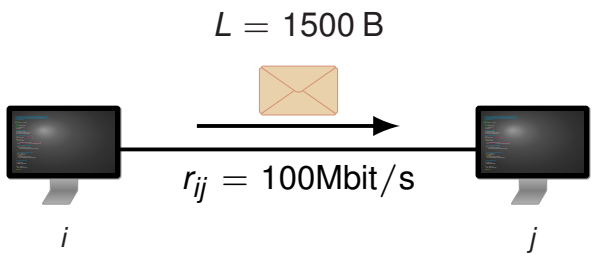
Definition (Übertragungsrate und Serialisierungszeit)

Die Übertragungsrate r in bit/s bestimmt die notwendige Zeit, um L Datenbits auf ein Übertragungsmedium zu legen. Sie bedingt die Serialisierungszeit

$$t_s = \frac{L}{r}.$$

Die Serialisierungszeit bzw. Übertragungsverzögerung wird im Englischen als Serialization Delay bzw. Transmission Delay bezeichnet (vgl. t_s).

Beispiel:



$$t_s = \frac{L}{R} = \frac{1500 \cdot 8\,\text{bit}}{100 \cdot 10^6\,\text{bit/s}} = 120\ \mu\text{s}$$

Frage: Wann empfängt Knoten j das erste Bit der Nachricht?

Ausbreitungsgeschwindigkeit

In Kapitel 1 haben wir bereits gesehen, dass Signale i. d. R. elektromagnetische Wellen sind, welche sich mit Lichtgeschwindigkeit im Medium ausbreiten.

Definition (Ausbreitungsverzögerung)

Die Ausbreitungsverzögerung über eine Distanz d rührt von der endlichen Ausbreitungsgeschwindigkeit von Signalen, welche relativ zur Lichtgeschwindigkeit im Vakuum $c \approx 300.000\,\mathrm{km/s}$ angegeben wird:

$$t_p = \frac{d}{\nu c}.$$

Der Wert $0 < \nu < 1$ ist die relative Ausbreitungsgeschwindigkeit in einem Medium. Für Kupfer gilt beispielsweise $\nu \approx 2/3$. Die Ausbreitungsverzögerung wird im Englischen als Propagation Delay bezeichnet (vgl. Benennung t_p).

Beispiel:

- Im Beispiel auf der vorherigen Folie haben wir exemplarisch die Serialisierungszeit zu $t_s = 120\,\mu\mathrm{s}$ bestimmt
- Angenommen die Knoten i und j sind $d_{ij} = 100\,\mathrm{m}$ voneinander entfernt
- Bei Lichtgeschwindigkeit benötigen Signale für diese Strecke gerade einmal 334 ns $\Rightarrow$ j empfängt bereits das erste Bit der Nachricht, wenn i gerade das 33ste Bit sendet!

Frage: Wie lange dauert es, bis j das letzte Bit der Nachricht empfangen hat?

Übertragungszeit und Nachrichtenflussdiagramm

In einem Nachrichtenflussdiagramm bzw. Weg-Zeit-Diagramm lässt sich die zeitliche Abfolge beim Senden und Empfangen von Nachrichten grafisch veranschaulichen:

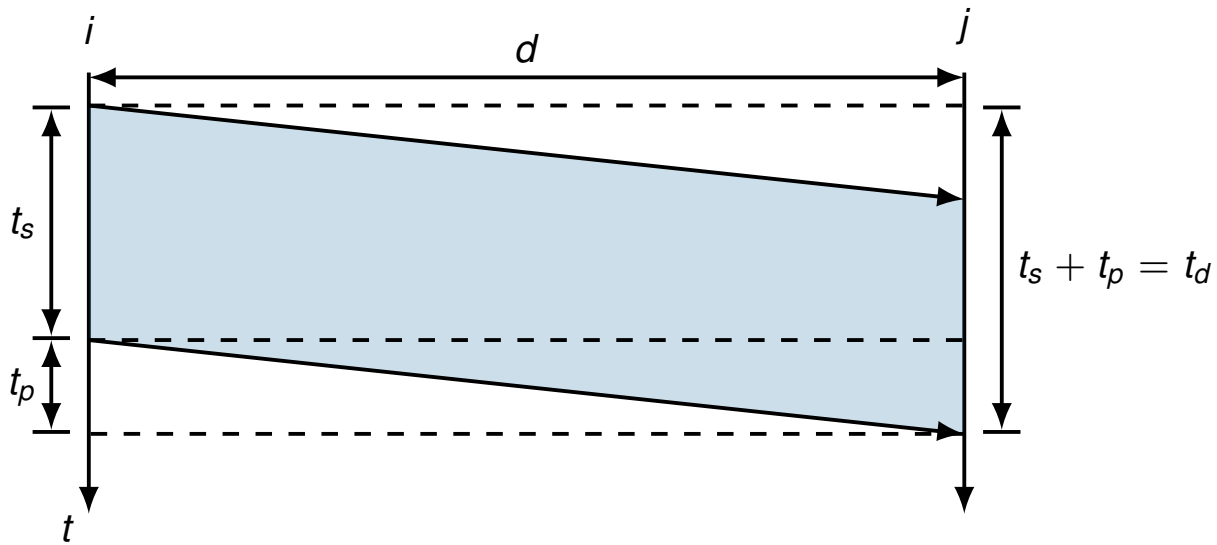


Abbildung: Nachrichtenflussdiagramm

- Die Gesamtverzögerung t_d (Delay) ergibt sich daher zu $t_d = t_s + t_p = \frac{L}{r} + \frac{d}{\nu c}$.
- Die Ausbreitungsverzögerung kann bei der Bestimmung von t_d u. U. vernachlässigt werden. Dies hängt allerdings von r, L und d ab! (s. Übung)

Bandbreitenverzögerungsprodukt

Durch die endliche Ausbreitungsverzögerung besitzt ein Übertragungskanal eine gewisse „Speicherkapazität“ C, welche als **Bandbreitenverzögerungsprodukt** bekannt ist.

Definition (Bandbreitenverzögerungsprodukt)

Als Bandbreitenverzögerungsprodukt bezeichnet man die Anzahl an Bits (Kapazität)

$$C = t_p r = \frac{d}{\nu c} r,$$

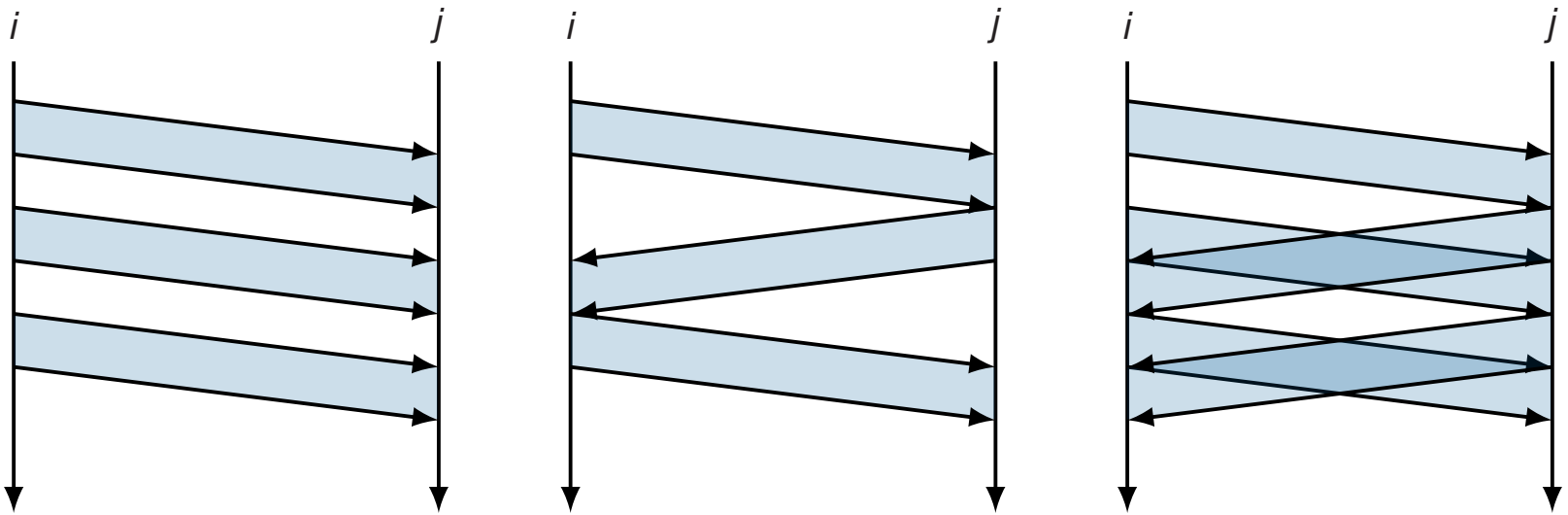
die sich in einer Senderichtung gleichzeitig auf der Leitung befinden können.

Beispiel:

- Leitung mit $r = 1\,\text{GBit/s}$
- Länge $d = 10\,\text{m}$
- $\nu = 2/3$ (Kupferleitung)
- $C = t_p \cdot r = \frac{d}{\nu c} \cdot r = \frac{10\,\text{m}}{2 \cdot 10^8 \frac{\text{m}}{\text{s}}} \cdot 10^9 \frac{\text{Bit}}{\text{s}} \approx 50\,\text{Bit}$

Übertragungsrichtung

Hinsichtlich der Übertragungsrichtung unterscheidet man:



(a) Simplex **(b)** Halbduplex **(c)** Vollduplex

Die Art der Verbindung hängt dabei ab von

- den Fähigkeiten des Übertragungskanals,
- dem Medienzugriffsverfahren und
- den Anforderungen der Kommunikationspartner.

Mehrfachzugriff (Multiplexing)

Häufig ist es von Vorteil, Nachrichten unterschiedlicher Teilnehmer gemeinsam über eine Leitung zu übertragen:

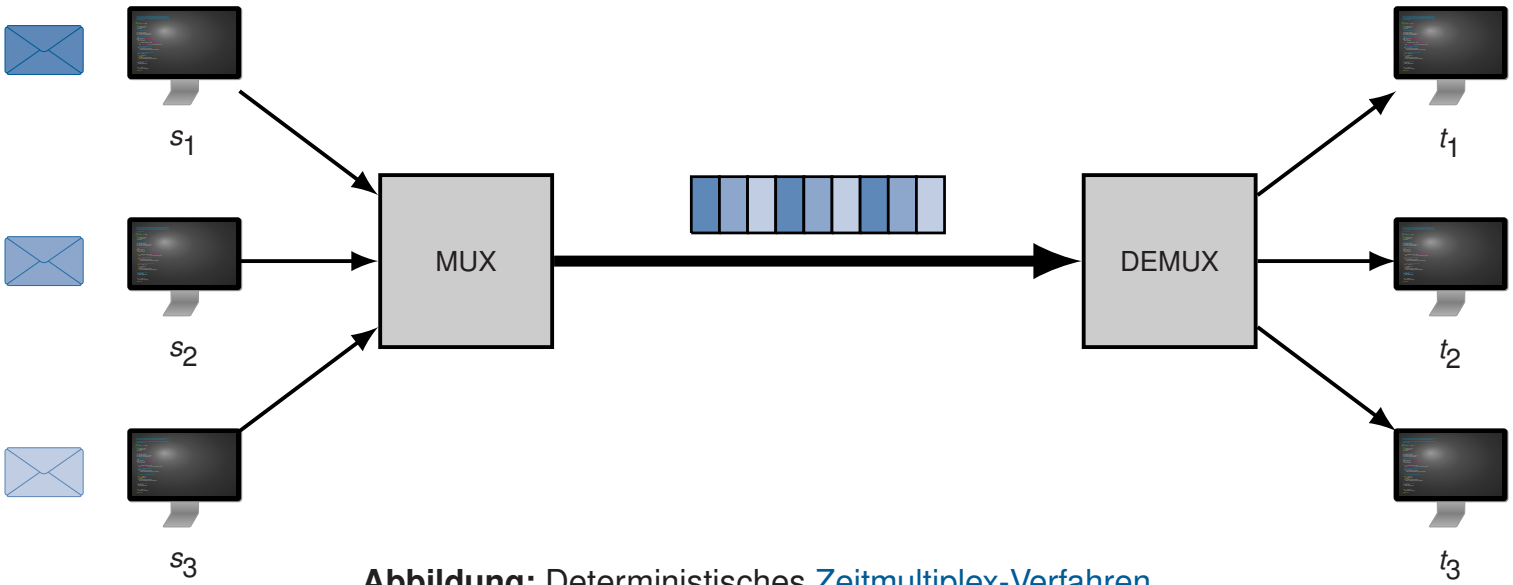


Abbildung: Deterministisches Zeitmultiplex-Verfahren

Ein anderes Zeitmultiplex-Verfahren haben wir bereits kennen gelernt:

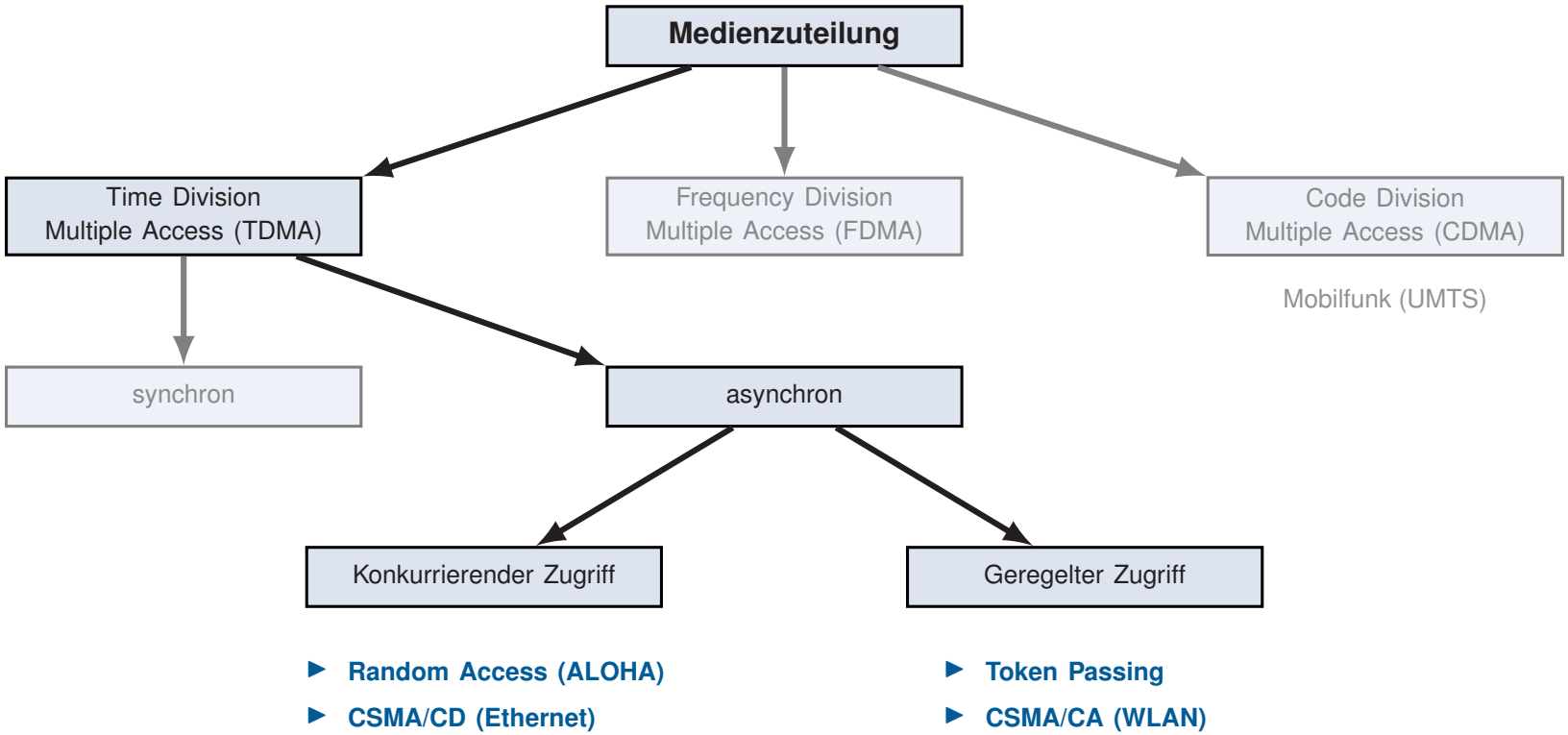
- Werden mehrere Computer mittels eines Hubs miteinander verbunden,
- so bildet das Hub ein gemeinsames geteiltes Medium,
- auf das die Computer mittels eines nicht-deterministischen Medienzugriffsverfahrens abwechselnd zugreifen.

Übersicht über Multiplex-Verfahren

- **Zeitmultiplex (Time Division Multiplex, TDM)**
  (s. vorherige Folie)
  - Deterministische Verfahren z. B. im Telefonnetz, bei ISDN-Verbindungen und im Mobilfunk
  - Nichtdeterministische Verfahren (konkurrierender Zugriff) in paketbasierten Netzwerken (z. B. Ethernet, WLAN)
- **Frequenzmultiplex (Frequency Division Multiplex, FDM)**
  Aufteilung des Kanals in unterschiedliche Frequenzbänder (spektrale Zerlegung) und Zuweisung Frequenzbänder an Kommunikationspartner (s. Kapitel 1).
  - Omnipräsent bei Funkübertragungen (z. B. unterschiedliche Radiosender)
  - Einsatz bei Glasfaserübertragungen („Modes“ mit unterschiedlicher Farbe)
  - Koexistenz von ISDN und DSL auf derselben Leitung
- **Raummultiplex (Space Division Multiplex, SDM)**
  Verwendung mehrerer paralleler Übertragungskanäle.
  - „Kanalbündelung“ bei ISDN
  - MIMO (Multiple-In Multiple-Out) bei kabellosen Übertragungen (Verwendung mehrerer Antennen schafft mehrere Übertragungskanäle)
- **Codemultiplex (Code Division Multiplex, CDM)**
  Verwendung orthogonaler Alphabete und Zuweisung der Alphabete an Kommunikationspartner.
  - Die Mobilfunktechnologie UMTS repräsentiert eine Variante von CDMA
  - Eine weitere Variante von CDMA im Mobilfunkbereich, CDMA2000, findet sich u.a. im Netz des amerikanischen Providers *Verizon* (c.f. „CDMA-iPhone“)

Mehrfachzugriff und Medienzugriffskontrolle [1]

Einige der (statischen) Multiplexing-Verfahren eignen sich auch als Mehrfachzugriffsverfahren:



Diese ausgewählten vier Zugriffsverfahren werden wir im Folgenden näher kennen lernen.

Bewertungskriterien für Medienzugriffsverfahren sind u.a.

- Durchsatz, d. h. Gesamtanzahl an Nachrichten pro Zeiteinheit, die übertragen werden können
- Verzögerung für einzelne Nachrichten
- Fairness zwischen Teilnehmern, die sich dasselbe Medium teilen
- Implementierungsaufwand für Sender und Empfänger

Problem bei synchronem TDMA

- Der Kanal wird statisch zwischen Teilnehmern aufgeteilt
- Datenverkehr ist aber stossartig bzw. burst-artig, d. h. ein Teilnehmer überträgt kurz mit hoher Bandbreite und danach längere Zeit nicht mehr
- Bandbreite steht während Ruhepausen anderen Teilnehmern nicht zur Verfügung

Lösungsidee: Asynchrones (flexibles) TDMA

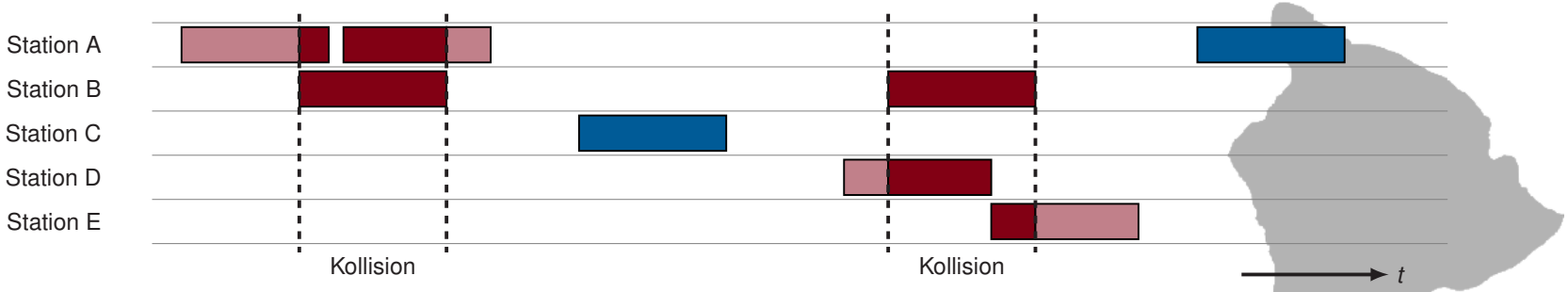
- Keine statische Aufteilung / Zuweisung von Zeitslots
- Stattdessen: Zufälliger, konkurrierender oder dynamisch geregelter Medienzugriff

Random Access (ALOHA)

- Entwickelt an der Universität von Hawaii (1971), c.f. Prof. Abramson
- Ursprünglich für kabellose Datenübertragungen
- Ziel: Verbindung von Oahu mit den anderen hawaiianischen Inseln

Funktionsweise

- Jede Station sendet an eine zentrale Station (vgl. „Basisstation“ in modernen WLANs), sobald Daten vorliegen
- Senden zwei Stationen gleichzeitig, kommt es zu Kollisionen
- Erfolgreich übertragene Nachrichten werden vom Empfänger auf anderer Frequenz quittiert („out-of-band“ Bestätigungsverfahren auf Link-Layer, keine Kollisionen zwischen Nachrichten und Bestätigungen)



Das Kanalmodell ist vergleichsweise einfach. Es existieren math. Beschreibungen für den sog. ALOHA Random Access Channel.

Erreichbarer Durchsatz mit ALOHA

Vereinfachende Annahmen:

- Mittlere bis beliebig große Anzahl an Knoten ($N > 15$)
- Gleiche, unabhängige und geringe Sendewahrscheinlichkeit auf allen Knoten
- Nachrichten konstanter Größe (Sendedauer T)

Modellierung:

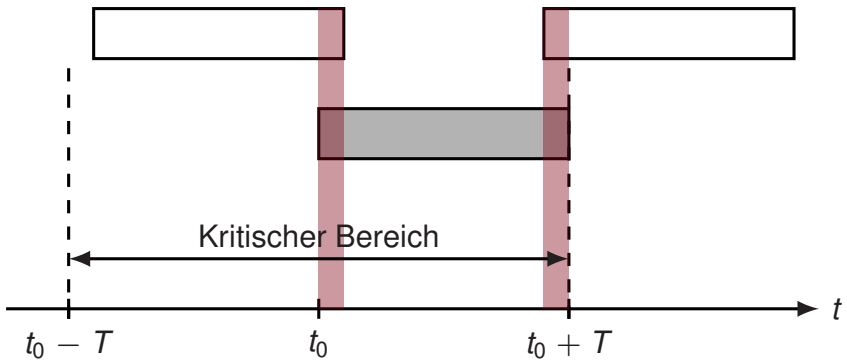
- Ob ein bestimmter Knoten i innerhalb des Zeitintervalls $[t, t + T)$ zu senden beginnt oder nicht entspricht einem Bernoulli-Experiment mit Erfolgs- bzw. Sendewahrscheinlichkeit p_i
- Da die Sendewahrscheinlichkeit für alle Knoten gleich ist, gilt $p_i = p \ \forall i = 1, \ldots, N$
- Da wir N Knoten haben, die jeweils unabhängig voneinander zu senden beginnen, wird dasselbe Bernoulli-Experiment N-mal wiederholt
- Das ist nichts anderes als eine Binomialverteilung, welche die Anzahl der Erfolge einer Serie gleichartiger und unabhängiger Versuche beschreibt
- Für sinnvoll großes N kann die Binomialverteilung durch eine Poisson-Verteilung[2] approximiert werden (s. Übung)
- Die mittlere erwartete Anzahl von Nachrichten pro Intervall ist gegeben als $Np = \lambda$

Das Ereignis X_t, dass im Intervall $[t, t + T)$ genau k Knoten senden, ist poisson-verteilt:

$$\Pr[X_t = k] = \frac{\lambda^k e^{-\lambda}}{k!}.$$

[2] Verteilung der seltenen Ereignisse

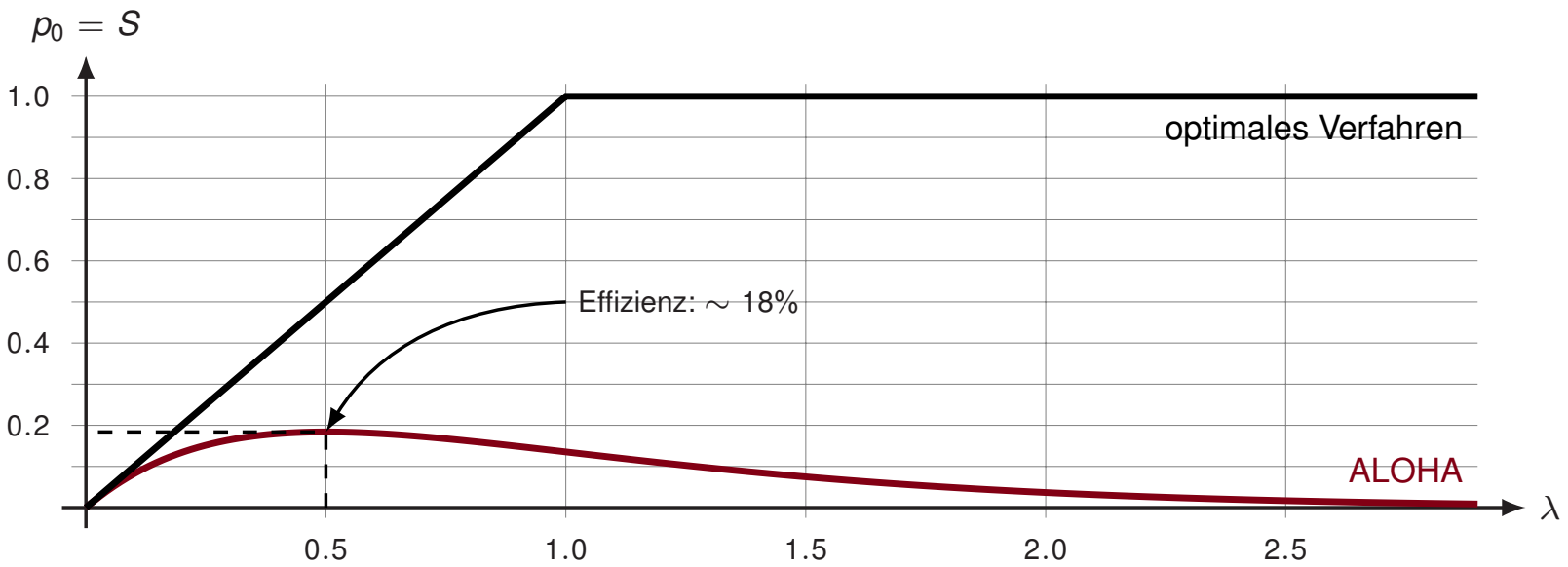
- Eine beliebige Station sende nun zum Zeitpunkt t_0 eine Nachricht
- Eine Kollision tritt genau dann auf, wenn eine andere Station im Intervall $(t_0 - T, t_0 + T]$ versucht, ebenfalls zu übertragen
- Die Übertragung ist also erfolgreich, wenn innerhalb des Intervalls $[t_0, t_0 + T]$ genau eine Übertragung stattfindet und im Intervall $(t_0 - T, t_0)$ keine Übertragung begonnen hat.



- Mit der Dichtefunktion $\Pr[X_t = k] = \frac{\lambda^k e^{-\lambda}}{k!}$ erhalten wir
- die Wahrscheinlichkeit p_0 für eine erfolgreiche Übertragung:

$$p_0 = \Pr[X_{t_0-T} = 0] \cdot \Pr[X_{t_0} = 1] = \mathrm{e}^{-\lambda} \cdot \lambda \mathrm{e}^{-\lambda} = \lambda \mathrm{e}^{-2\lambda}$$

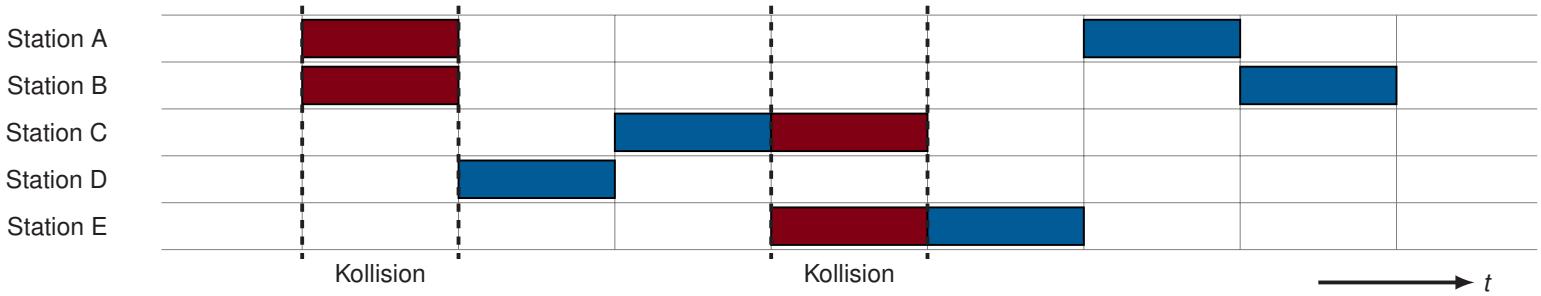
Die Erfolgswahrscheinlichkeit p_0 kann gegen die Senderate λ aufgetragen werden:



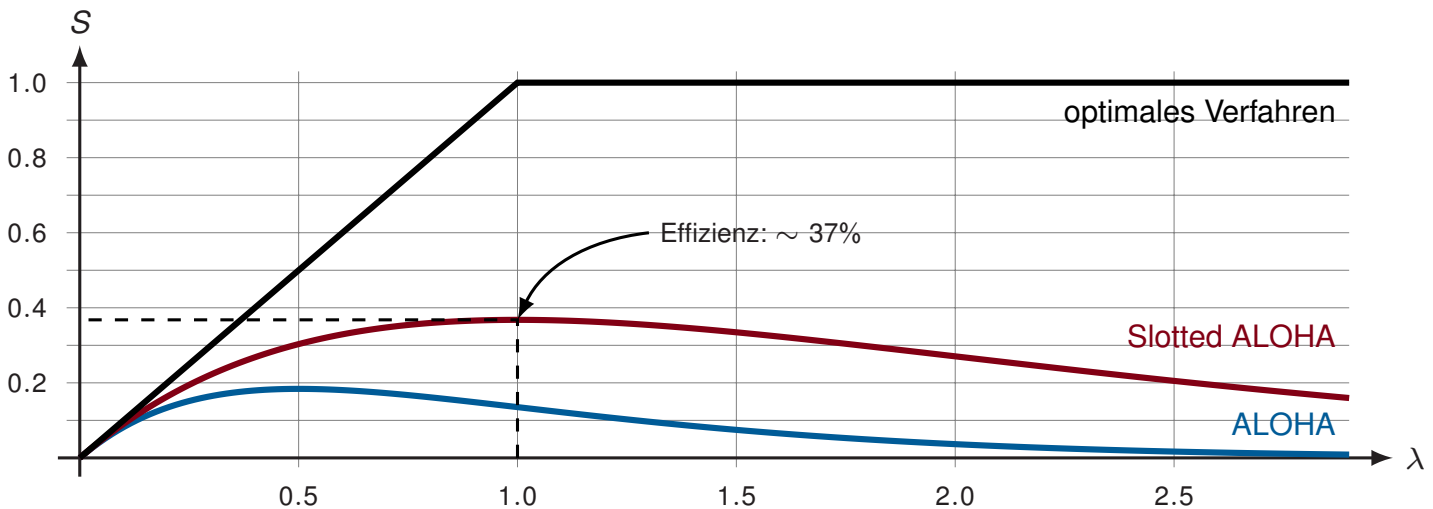
- Wir wissen, dass innerhalb eines beliebigen Intervalls $[t, t + T)$ höchstens eine Übertragung erfolgreich sein kann
- Dementsprechend entspricht die Anzahl S der erfolgreichen Nachrichten pro Intervall gleichzeitig der Wahrscheinlichkeit für eine erfolgreiche Übertragung
- Bei einem optimalen Verfahren würde die Anzahl erfolgreicher Nachrichten S linear mit der Senderate ansteigen, bis die maximale Anzahl von Nachrichten pro Zeitintervall erreicht ist (hier ist das genau eine Nachricht pro Intervall)
- Steigt die Senderate weiter, würde dies ein optimales Verfahren nicht beeinträchtigen

Variante: Slotted ALOHA

Stationen dürfen nicht mehr zu beliebigen Zeitpunkten mit einer Übertragung beginnen, sondern nur noch zu den Zeitpunkten $t = nT$, $n = 0, 1, \ldots$



Kritischer Bereich ist nur noch T anstelle von $2T \Rightarrow S = \lambda \cdot e^{-\lambda}$.



Carrier Sense Multiple Access (CSMA)

Eine einfache Verbesserung von Slotted ALOHA: „Listen Before Talk“

- Höre das Medium ab
- Beginne erst dann zu senden, wenn das Medium frei ist

Non-persistent CSMA:

1. Wenn Medium frei, übertrage im nächstmöglichen Intervall
2. Wenn belegt, warte eine feste Zeitspanne, dann (1)

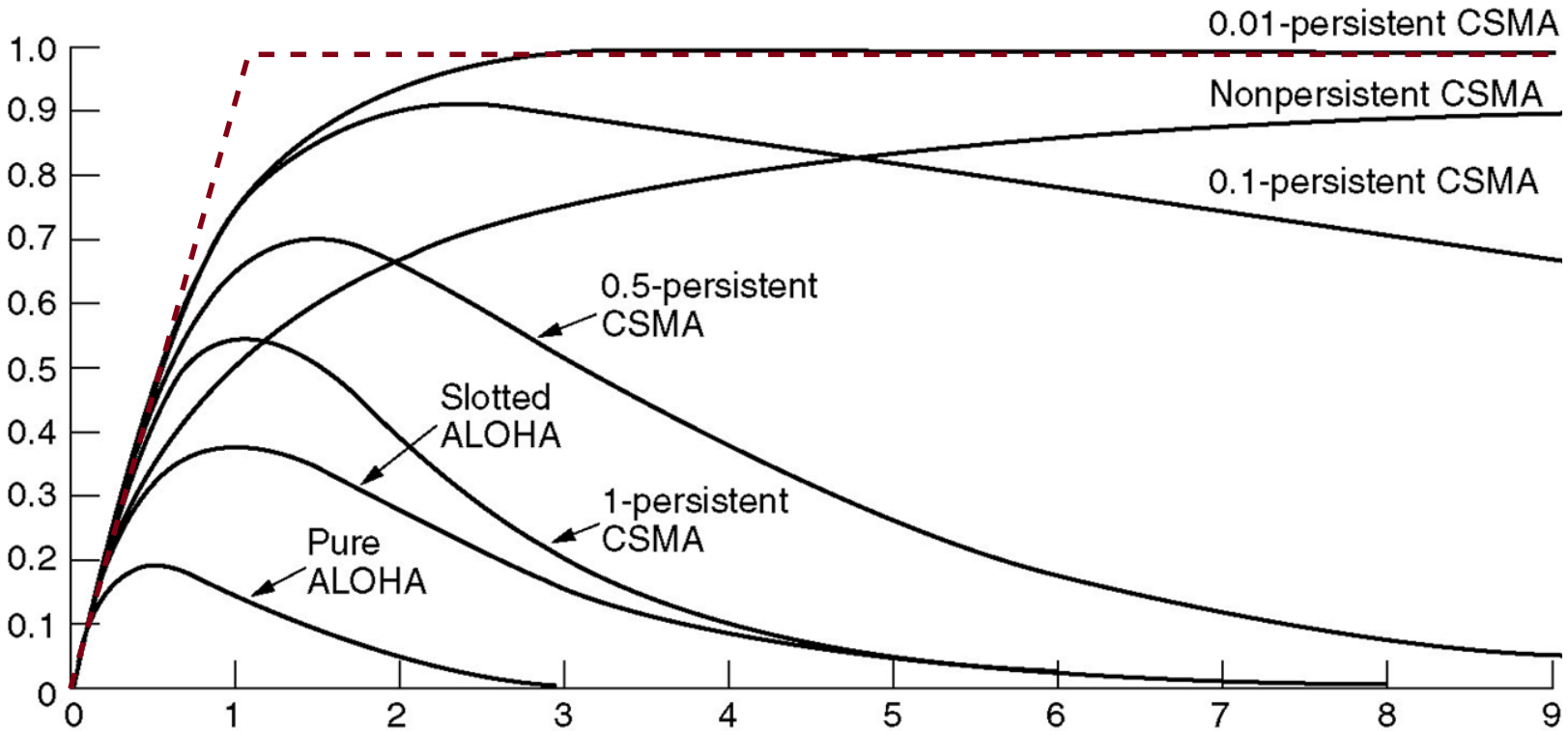
1-persistent CSMA:

1. Wenn Medium frei, übertrage im nächstmöglichen Intervall
2. Wenn Medium belegt, warte bis frei und übertrage im nächstmöglichen Intervall

p-persistent CSMA:

1. Wenn Medium frei, übertrage mit Wahrscheinlichkeit p oder verzögere mit Wahrscheinlichkeit $1 - p$ um eine Slotzeit
2. Wenn Medium belegt, warte bis frei, dann (1)
3. Wenn um eine Slotzeit verzögert, dann (1)

Alle bisherigen Verfahren im Vergleich



CSMA/CD (Collision Detection)

Ansatz

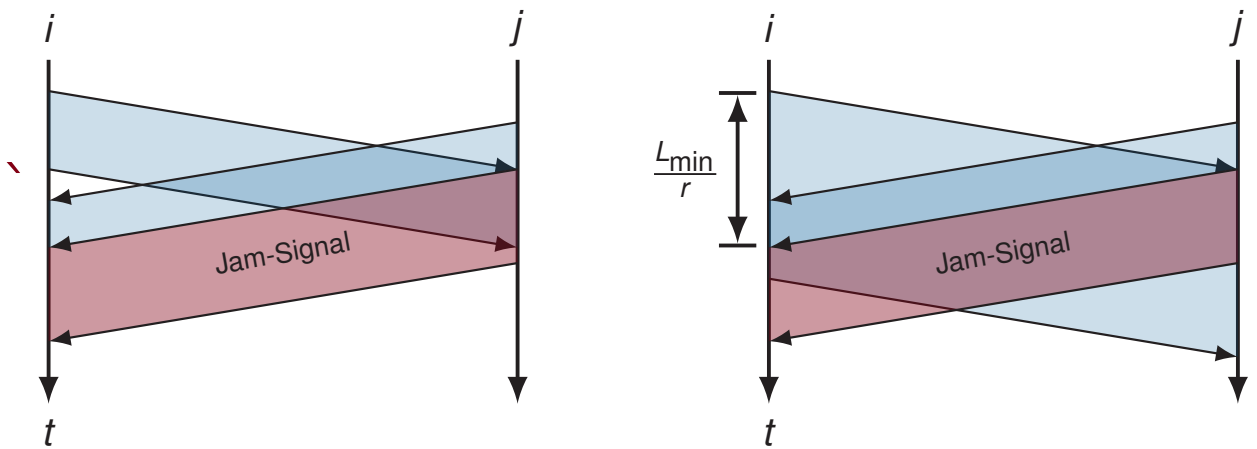
- Erkenne Kollisionen und wiederhole die Übertragung, wenn eine Kollision erkannt wird
- Verzichte auf das Senden von Bestätigungen
- Wird keine Kollision erkannt, gilt die Übertragung als erfolgreich

Problem: Der Sender muss die Kollision erkennen, während er noch überträgt

Voraussetzung für CSMA/CD [1]

Angenommen zwei Stationen i und j kommunizieren über eine Distanz d mittels CMSA/CD. Damit Kollisionen erkannt werden können, müssen Nachrichten folgende Mindestlänge L_{min} aufweisen:

$$L_{\text{min}} = \frac{2d}{\nu c} r$$



Wird 1-persistentes CSMA mit Kollisionserkennung verwendet, ergibt sich folgendes Problem:

- Die Kollision zerstört die Nachrichten beider in die Kollision verwickelten Stationen
- Mind. eine der Stationen sendet ein JAM-Signal
- Nachdem das Medium frei wird, wiederholen beide Stationen die Übertragung
  $\Rightarrow$ Es kommt sofort wieder zu einer Kollision

Lösung: Warte „zufällige“ Zeit nach einer Kollision

Binary Exponential Backoff

Bei der k-ten Wiederholung einer Nachricht

- wählt der Sender zufällig $n \in \left\{0, 1, \ldots, \min\{2^{k-1}, 1024\}\right\}$ aus und
- wartet n Slotzeiten vor einem erneuten Sendeversuch.

Die maximale Wiederholungszahl beträgt $k = 15$ (also 16 Sendeversuche).

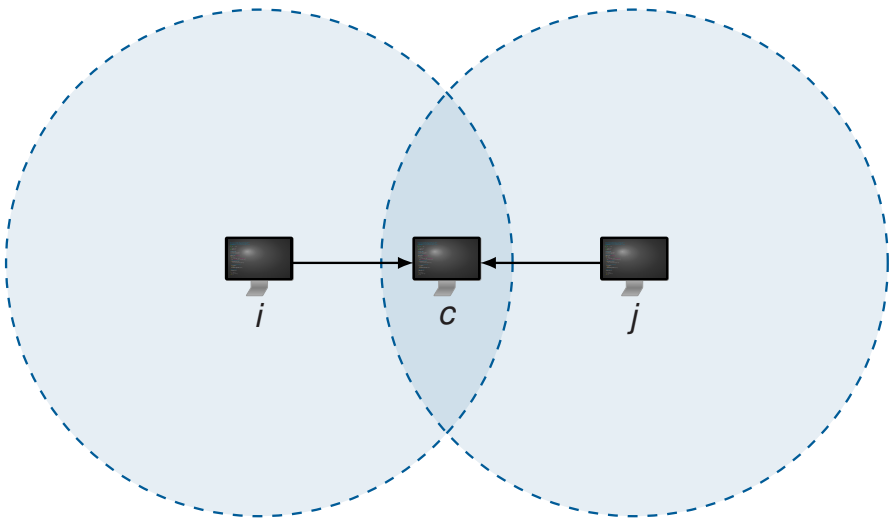
Durch die Wartezeiten, die

- zufällig gewählt und
- situationsabhängig größer werden,
- wird die Kollisionswahrscheinlichkeit bei Wiederholungen reduziert.

CSMA/CA (Collision Avoidance)

In Funknetzwerken funktioniert CSMA/CD nicht, da der Sender einer Nachricht eine Kollision auch bei ausreichender Nachrichtenlänge nicht immer detektieren kann.

„Hidden Station“:



- Knoten *i* und *j* senden gleichzeitig
- Knoten *c* erkennt die Kollision
- Weder *i* noch *j* bemerken die Kollision

CSMA/CA basiert auf *p*-persistentem CSMA, d. h.

1. Wenn Medium frei, übertrage mit Wahrscheinlichkeit *p* oder verzögere mit Wahrscheinlichkeit $1 - p$ um eine Slotzeit
2. Wenn Medium belegt, warte bis frei, dann (1)
3. Wenn um eine Slotzeit verzögert, dann (1)

Fallbeispiel: IEEE 802.11 DCF (Distributed Coordination Function)

- Festes Zeitinervall zwischen Rahmen (DCF Interframe Spacing).
- Wenn Medium mind. für DIFS idle ist, dann wähle unabhängig und gleichverteilt eine Anzahl von Backoff-Slots aus dem Intervall $\{0, 1, 2, \ldots, \min\{2^{c+n} - 1, 255\}\}$.
- c ist abhängig vom PHY (z.B. $c = 4$), n ist der Retry Counter des Binary Exponential Backoffs.

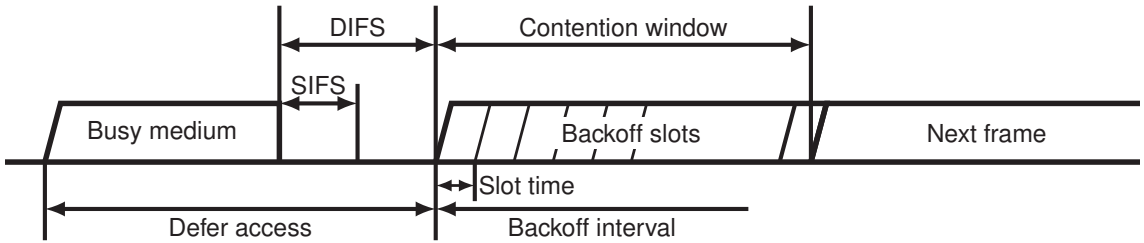


Abbildung: IEEE 802.11 DCF

- Medienzugriff hat durch festes c ein stets ein Contention Window.
- Ein Rahmen gilt in IEEE 802.11 als erfolgreich übertragen, wenn
  - im Fall von Unicasts der Empfänger eine Bestätigung schickt (Link-Layer Acknowledgements) oder
  - im Fall von Broadcasts die Übertragung eines Frames störungsfrei abgeschlossen wird.
- Da i.d.R. nicht gleichzeitig gesendet und das Medium geprüft werden kann (anders bei Ethernet), ist die zweite Bedingung praktisch bereits erfüllt, wenn ein Knoten zu senden beginnt.

Fallbeispiel: IEEE 802.11 DCF (Distributed Coordination Function)

Was passiert in der Praxis? Beispiel anhand handelsüblicher Hardware im Monitor Mode:

- Wir deaktivieren Link-Layer Bestätigungen und prüfen, wie sich die Hardware verhält.
- Ohne Bestätigungen wird es keinen Exponential Back-Off geben, da Übertragungen (einmal begonnen) nicht mehr fehlschlagen.
- Das Contention Window sollte aber $\{0, 1, 2, \ldots, 15\}$ betragen und Backoff-Slots unabhängig und gleichverteilt daraus gezogen werden.

Was wirklich passiert...

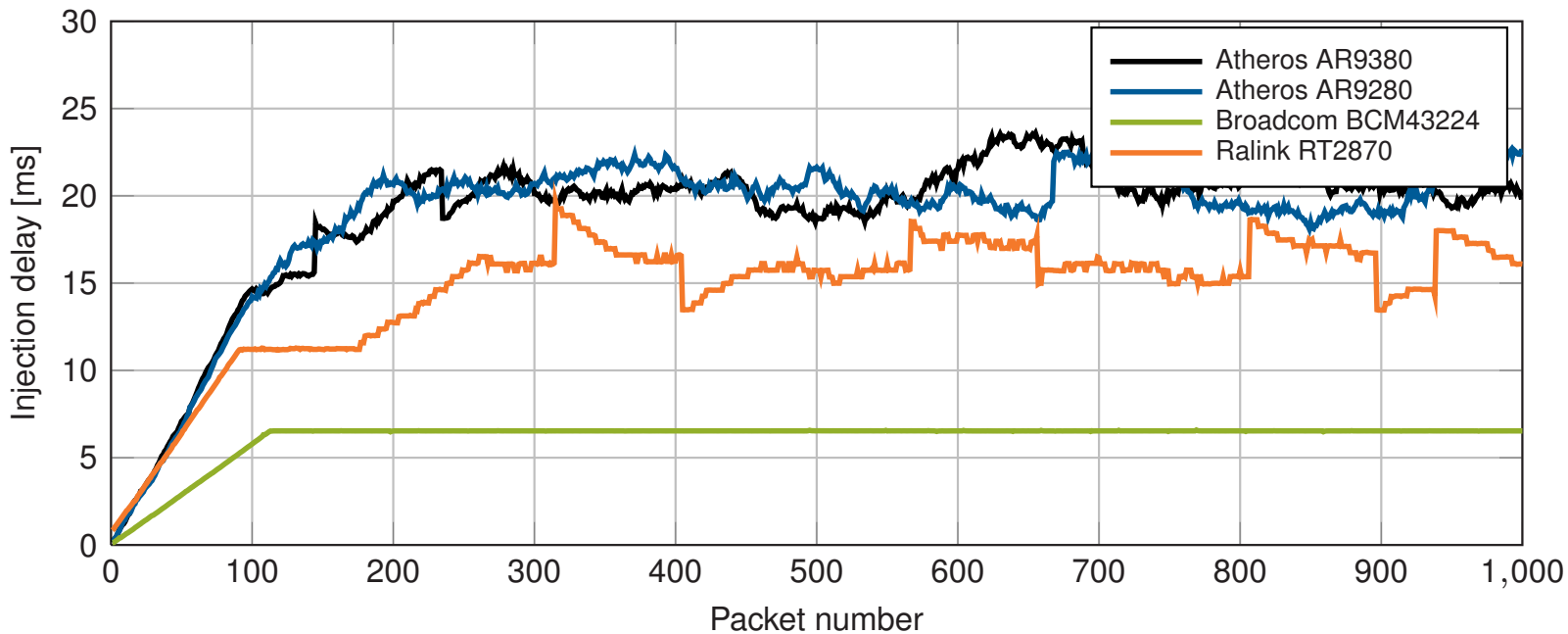


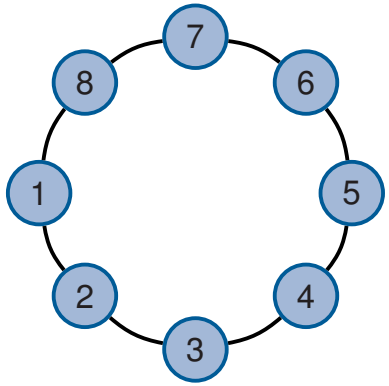
Abbildung: Injection Delay, d.h. Zeit zwischen `write()` auf RAW-Socket und Antwort vom Treiber

Erweiterung: RTS/CTS (Request to Send / Clear to Send)

- Übertragungen werden von einer Basisstation gesteuert
- Alle Knoten müssen sich in Reichweite zur Basisstation befinden
- Bevor ein Knoten eine Nachricht überträgt, wird nach dem CSMA/CA-Prinzip ein RTS an die Basisstation geschickt
- Nur wenn die Basisstation mit einem CTS antwortet, darf die Übertragung beginnen

Kollisionen mit Hidden Stations werden so vermieden, aber nicht gänzlich verhindert.

Token Passing



Idee: Kollisionsfreie Übertragung durch Weitergabe eines **Tokens**

- Stationen werden zu einem physikalischen Ring zusammengeschaltet
- Ein Token zirkuliert im Ring
- Will eine Station senden, nimmt sie das Token vom Ring und darf danach als einzige Station im Ring übertragen
- Nachdem alle Nachrichten gesendet wurden (oder nach einer definierten Zeitspanne), wird das Token wieder auf den Ring gelegt

Empfang von Nachrichten:

- Die Nachricht zirkuliert wie das Token durch den Ring
- Der Empfänger markiert die Nachricht als gelesen und schickt sie weiter
- Trifft sie wieder beim Sender ein, so nimmt dieser sie vom Netz

Was ist, wenn das Token „verloren geht“?

- Es gibt eine Monitor-Station, z. B. die Station, die das erste Token erzeugt hat
- Diese Monitor-Station erzeugt bei Bedarf neue Tokens, entfernt endlos kreisende Pakete und entfernt doppelte Token
- Fällt die Monitor-Station aus, wird von den verbleibenden Stationen eine Neue gewählt

Vorteile:

- Sehr effizient, da keine kollisionsbedingten Wiederholungen
- Garantierte maximale Verzögerung (Determinismus)

Nachteile bzw. Schwierigkeiten:

- Geht das Token verloren, muss es durch ein Neues ersetzt werden
  $\rightarrow$ eine Station muss spezielle Aufgaben übernehmen (Monitor-Station)
- Fehlerhaftes verhalten eines Knotens stört die gesamte Kommunikation im Ring
- Übertragungsverzögerung u. U. größer als bei CSMA, da Sender auf Token warten muss
- Zusammenschaltung der Stationen zu einem Ring ist u. U. aufwendig

Einsatz heute:

- Token Ring (IEEE 802.5) wurde vollständig von Ethernet (IEEE 802.3) ersetzt und spielt in lokalen Netzwerken heute keine Rolle mehr.
- FDDI (Fiber Distributed Data Interface) ist ein Sammelbegriff für Glasfaserringe bis zu einer Länge von einigen hundert Kilomtern. Diese werden z. B. als Backbone lokaler Zugangsanbieter im städtischen Maßstab eingesetzt.

Zusammenfassung

In diesem Teilkapitel haben wir einige flexible Zeitmultiplexverfahren kennengelernt, die Zugriff mehrerer Stationen auf ein gemeinsames Medium erlauben. Im Gegensatz zu statischem Zeitmultiplex wird die Kanalbandbreite nicht für inaktive Knoten reserviert.

Konkurrierender Zugriff:

- ALOHA und Slotted ALOHA
- CSMA (non-persistent, 1-persistent, *p*-persistent)
- CSMA/CD (Kollisionserkennung)
  IEEE 802.3 Ethernet

Geregelter Zugriff:

- CSMA/CA (Kollisionsvermeidung)
  IEEE 802.11 WLAN
- Token Passing (Kollisionsverhinderung)
  IEEE 802.5 Token Ring, Fiber Distributed Data Interface (FDDI)

Übersicht

Motivation

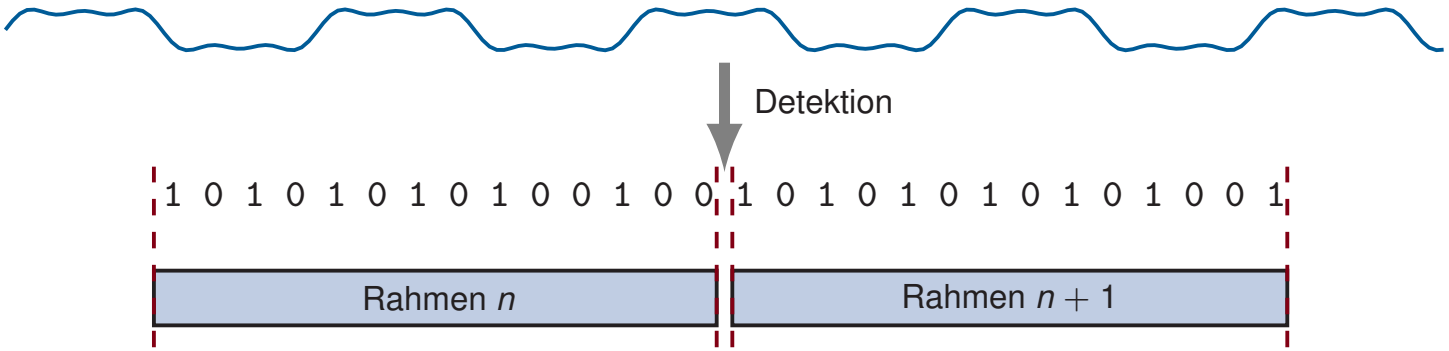
Bislang haben wir nur von Nachrichten gesprochen, ohne uns Gedanken über deren Format zu machen. Aus Sicht der physikalischen Schicht ist eine Nachricht lediglich eine Folge von Bits. Für eine Betrachtung der Sicherungsschicht reicht diese Vorstellung aber nicht mehr aus.

Im Folgenden wollen wir uns Gedanken machen,

- wie einzelne Nachrichten auseinandergehalten werden können,
- welche zusätzlichen Informationen Protokolle der Sicherungsschicht benötigen und
- wie Übertragungsfehler, die trotz Kanalkodierung auftreten, erkannt werden können.

Im Kontext der Sicherungsschicht bezeichnen wir Nachrichten fortan als Rahmen (engl. Frame).

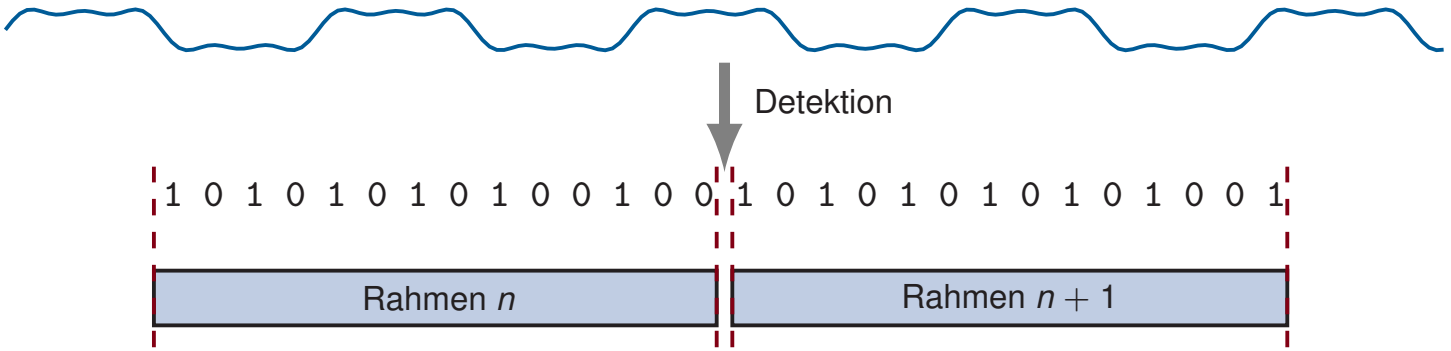
Erkennung von Rahmengrenzen



Wie kann der Empfänger Rahmen erkennen, insbesondere wenn

- Rahmen unterschiedliche Größen haben und
- nicht ständig Nutzdaten auf der Leitung liegen (Idle-Perioden)?

Erkennung von Rahmengrenzen



Wie kann der Empfänger Rahmen erkennen, insbesondere wenn

- Rahmen unterschiedliche Größen haben und
- nicht ständig Nutzdaten auf der Leitung liegen (Idle-Perioden)?

Es gibt viele Möglichkeiten:

- Längenangabe der Nutzdaten
- Steuerzeichen (Start / Ende)
- Begrenzungsfelder und „Bit-Stopfen“
- Coderegelverletzung

Ziel aller Verfahren zur Rahmenbegrenzung ist die Erhaltung der **Codetransparenz**, d. h. die Übertragung beliebiger Zeichenfolgen zu ermöglichen.

Längenangabe der Nutzdaten

Idee:

- Am Anfang des Rahmens steht die Länge der nachfolgenden Nutzdaten (oder die Gesamtlänge des Rahmens).
- Voraussetzung: Das Längenfeld und damit der Beginn einer Nachricht muss eindeutig zu erkennen sein

Längenangabe der Nutzdaten

Idee:

- Am Anfang des Rahmens steht die Länge der nachfolgenden Nutzdaten (oder die Gesamtlänge des Rahmens).
- Voraussetzung: Das Längenfeld und damit der Beginn einer Nachricht muss eindeutig zu erkennen sein

Wie kann der Beginn eines Rahmens erkannt werden?

- Durch Steuerzeichen (Start / Ende)
- Durch Voranstellen von Begrenzungsfeldern
- Durch Verlust des Trägersignals zwischen den Rahmen (Coderegelverletzung)

Steuerzeichen

In Kapitel 1 haben wir bereits den 4B5B-Code kennengelernt, welcher in Kombination mit Leitungscodes wie MLT-3 auf der physikalischen Schicht eingesetzt wird.

- Je 4 bit Eingabe werden auf 5 bit Ausgabe abgebildet
- Einem Rahmen werden die Startsymbole J/K vorangestellt
- Nach einem Rahmen werden die Endsymbole T/R eingefügt

Eingabe	Ausgabe	Bedeutung	Eingabe	Ausgabe	Bedeutung
0000	11110	Hex data 0	-	00000	Quiet (Signalverlust)
0001	01001	Hex data 1	-	11111	Idle (Pause)
0010	10100	Hex data 2	-	11000	Start #1 (J)
0011	10101	Hex data 3	-	10001	Start #2 (K)
0100	01010	Hex data 4	-	01101	End (T)
0101	01011	Hex data 5	-	00111	Reset (R)
⋮	⋮	⋮	-	11001	Set
1111	11101	Hex data F	-	00100	Halt

Beispiel:

Eingabe:			1011	0101	0110		
Ausgabe:	11000	10001	10111	01011	01110	01101	00111

Steuerzeichen werden nicht nur auf Schicht 1/2 verwendet. Auf Schicht 6 (Darstellungsschicht) wird der **ASCII-Code (American Standard Code for Information Interchange)** verwendet (7 bit Codeworte):

ASCII (hex)	Zeichen	ASCII (hex)	Zeichen	ASCII (hex)	Zeichen	ASCII (hex)	Zeichen
00	NUL	20	SP	40	@	60	‘
01	SOH	21	!	41	A	61	a
02	STX	22	„	42	B	62	b
03	ETX	23	#	43	C	63	c
04	EOT	24	$	44	D	64	d
05	ENQ	25	%	45	E	65	e
06	ACK	26	&	46	F	66	f
07	BEL	27	’	47	G	67	g
08	BS	28	(	48	H	68	h
09	TAB	29	)	49	I	69	i
0A	LF	2A	*	4A	J	6A	j
0B	VT	2B	+	4B	K	6B	k
0C	FF	2C	,	4C	L	6C	l
0D	CR	2D	-	4D	M	6D	m
0E	SO	2E	.	4E	N	6E	n
0F	SI	2F	/	4F	O	6F	o
10	DLE	10	0	50	P	70	p
11	DC1	11	1	51	Q	71	q
12	DC2	12	2	52	R	72	r
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Was ist, wenn Steuerzeichen zufällig in den Nutzdaten vorkommen?

1. Im Fall des 4B5B-Code kann das nicht passieren:
   - 4 bit Datenworte werden injektiv auf 5 bit Datenworte abgebildet
   - Einige der verbleibenden 5 bit Worte werden als Steuerzeichen verwendet

Was ist, wenn Steuerzeichen zufällig in den Nutzdaten vorkommen?

1. Im Fall des 4B5B-Code kann das nicht passieren:
   - 4 bit Datenworte werden injektiv auf 5 bit Datenworte abgebildet
   - Einige der verbleibenden 5 bit Worte werden als Steuerzeichen verwendet

2. Der ASCII-Code ist lediglich eine Interpretationsvorschrift:
   - Einige Codeworte sind Textzeichen (Ziffern, Zahlen, ...), andere Steuerzeichen
   - Um ein Steuerzeichen als Datum übertragen zu können, wird dieses durch ein spezielles Steuerzeichen markiert: Escape Character
   - Soll dieses spezielle Steuerzeichen selbst übertragen werden, so wird es verdoppelt
   - Dieses Vorgehen bezeichnet man als **Character Stuffing**

Beispiele für Character Stuffing:

Meist wird automatisch für **Codetransparenz** (d.h. Übertragung beliebiger Sequenzen von Zeichen) gesorgt, so dass sich der Benutzer nicht darum kümmern muss. Das trifft nicht auf Porgrammiersprachen zu:

```
System.out.println("Ein \" muss escaped werden");
```

Innerhalb des auszugebenden Strings müssen Anführungszeichen mittels eine Backslashs escaped werden.

Weitere Beispiele:

- Bash (Control + C)
- Telnet-Verbindungen
- Texteditoren (Emacs)

Begrenzungsfelder und Bit-Stopfen

Idee:

- Markiere Start und Ende einer Nachricht mit einer bestimmten Bitfolge
- Stelle sicher, dass die Markierung nicht zufällig in den Nutzdaten vorkommt (Bit-Stopfen, engl. Bit Stuffing)

Begrenzungsfelder und Bit-Stopfen

Idee:

- Markiere Start und Ende einer Nachricht mit einer bestimmten Bitfolge
- Stelle sicher, dass die Markierung nicht zufällig in den Nutzdaten vorkommt (Bit-Stopfen, engl. Bit Stuffing)

Beispiel:

- Start- / Endemarkierung sei `01111110`
- Um das Auftreten der Markierung in Nutzdaten zu verhindern, füge nach fünf aufeinanderfolgenden 1-en eine 0 ein

Eingabe:		`1100101111110111111`	
Ausgabe:	`01111110`	`11001011111`**`0`**`101111101`	`01111110`

- Empfänger entfernt nach fünf aufeinanderfolgenden 1-en die darauf folgende 0

Coderegelverletzung

Viele Leitungscodes (z. B. RZ und Manchester) besitzen unabhängig von den zu übertragenden Daten bestimmte Signalwechsel.

Idee:

- Lasse bestimmte Signalwechsel aus
- Auf diese Art wird ein ungültiges (im Code nicht existierendes) Symbol erzeugt
- Dieses kann verwendet werden, um Start und Ende von Rahmen zu markieren

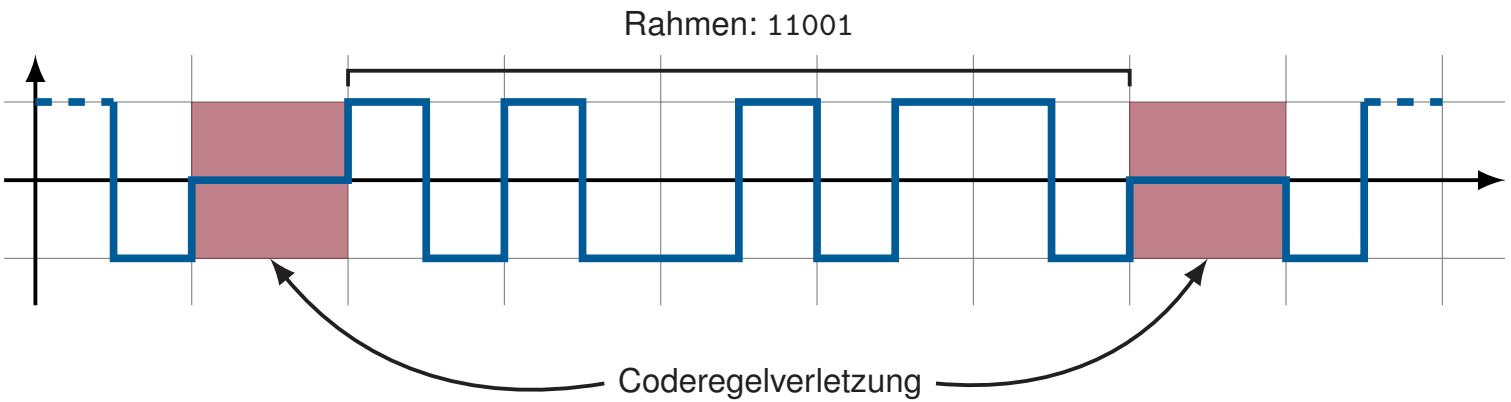
Coderegelverletzung

Viele Leitungscodes (z. B. RZ und Manchester) besitzen unabhängig von den zu übertragenden Daten bestimmte Signalwechsel.

Idee:

- Lasse bestimmte Signalwechsel aus
- Auf diese Art wird ein ungültiges (im Code nicht existierendes) Symbol erzeugt
- Dieses kann verwendet werden, um Start und Ende von Rahmen zu markieren

Beispiel: Manchester-Code



Fallbeispiele

IEEE 802.3a/i (Ethernet): 10 Mbit/s

- Als Leitungscode wird der Manchester-Code verwendet.
- Das Ende eines Frames wird durch Coderegelverletzung angezeigt.

IEEE 802.3u (FastEthernet): 100 Mbit/s

- Als Leitungscode wird MLT-3 in Kombination mit dem 4B5B-Code verwendet.
- Start und Ende von Rahmen werden durch Steuerzeichen des 4B5B-Codes markiert.

Zusätzlich wird bei IEEE 802.3a/i/u jedem Rahmen noch eine **Präambel** vorangestellt. Diese dient allerdings nur der Taktsynchronisierung zwischen Sender und Empfänger.

Übersicht

Adressierung und Fehlererkennung

Bislang wissen wir,

- wie ein binärer Datenstrom übertragen wird und
- wie der Empfänger Rahmengrenzen wiedererkennt.

Wir wissen aber noch nicht,

- wie Nutzdaten, die von Schicht 3 und höher kommen, von der Sicherungsschicht behandelt werden,
- wie der Empfänger eines Rahmens adressiert wird und
- wie aus den Nutzdaten und protokollspezifischen Informationen ein Rahmen entsteht.

Adressierung

In Direktverbindungsnetzen

- sind angeschlossene Knoten direkt erreichbar,
- es findet also keine Vermittlung (engl. Routing) zwischen Knoten statt.

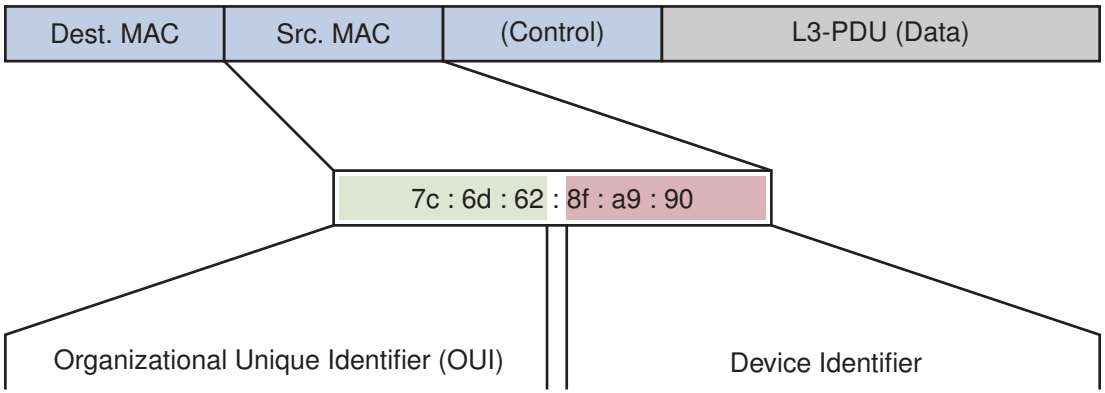
Anforderungen an Adressen auf Schicht 2:

- **Eindeutige Identifikation** eines Knotens im Direktverbindungsnetz
- Eine **Broadcast-Adresse**, mittels der **alle** Knoten im Direktverbindungsnetz adressiert werden können.

Adressen auf Schicht 2 bezeichnet man allgemein als **MAC-Adressen**, wobei MAC für Media Access Control steht.

Beispiel:

MAC-Adressen haben häufig den folgenden Aufbau:

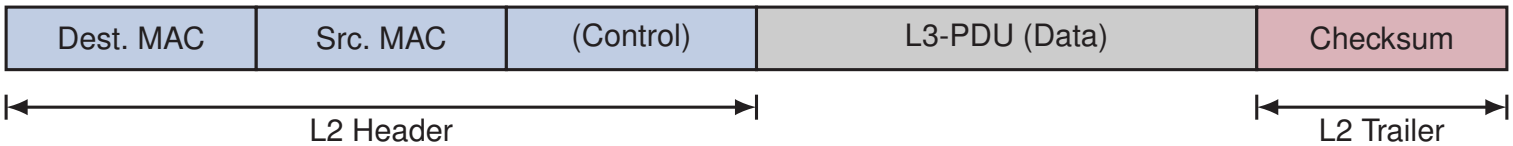


- Netzwerkkarten besitzen ab Werk eine MAC-Adresse. Diese ist meist im ROM (Read Only Memory) der Netzwerkkarte hinterlegt.
- Die Auftrennung in OUI und Device ID ermöglicht es den Herstellern von Netzwerkkarten, eindeutige MAC-Adressen vergeben.
- Daher ist möglich, den Hersteller einer Netzwerkkarte anhand deren MAC-Adresse zu identifizieren (z. B. 7c:6d:62 $\hat{=}$ Apple).
- Als Broadcast-Adresse ist ff:ff:ff:ff:ff:ff („all ones“) definiert.

Fehlererkennung

- Trotz Kanalkodierung können Übertragungsfehler (Bitfehler) auftreten.
- Es kann daher passieren, dass eine fehlerhafte Payload an höhere Schichten weitergeleitet wird.

Um die Wahrscheinlichkeit für derartige Fehler zu minimieren, werden **fehlererkennende Codes** eingesetzt (sog. **Prüfsummen**):



Im Gegensatz zur Kanalkodierung dient die Prüfsumme eines Schicht-2-Protokolls üblicherweise nicht der Fehlerkorrektur sondern lediglich der Fehlererkennung.

Cyclic Redundancy Check (CRC) [2]

- CRC berechnet zu einem gegebenen Datenblock (z. B. L3-PDU) eine Checksumme fester Länge.
- Deren Länge wird durch den Grad eines **Generatorpolynoms** $g(x)$ bestimmt.
- Dessen Koeffizienten stammen aus dem finiten Feld $\mathrm{GF}(2) = \{0, 1\}$.
- Die Berechnung der Checksumme ist eine Polynomdivision über GF(2), weswegen Addition und Subtraktion sich auf bitweise XOR-Operationen reduzieren.

Ethernet verwendet CRC-32:

$$g(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

Beispiel: Generator $g(x) = x^3 + x^2 + 1$, Daten $m(x) = x^7 + x^5 + x^2 + 1$

1. Koeffizienten bestimmen: $g(x) \mathrel{\hat{=}} 1101$ und $m(x) \mathrel{\hat{=}} 10100101$

Beispiel: Generator $g(x) = x^3 + x^2 + 1$, Daten $m(x) = x^7 + x^5 + x^2 + 1$

1. Koeffizienten bestimmen: $g(x) \mathrel{\hat{=}} 1101$ und $m(x) \mathrel{\hat{=}} 10100101$

2. $\text{grad}(g(x)) = 3 \Rightarrow$ Daten mit x^3 multiplizieren. Dies entspricht dem „Anhängen“ von 3 Nullen: $m'(x) = m(x) \cdot x^3 \mathrel{\hat{=}} 10100101000$

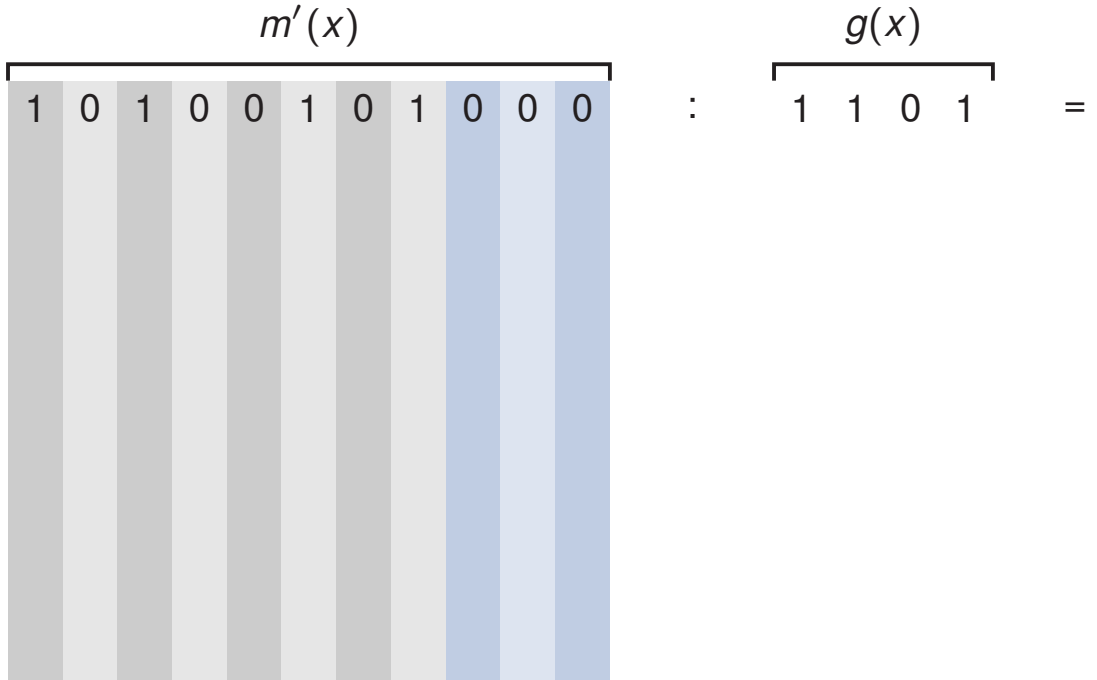
Beispiel: Generator $g(x) = x^3 + x^2 + 1$, Daten $m(x) = x^7 + x^5 + x^2 + 1$

1. Koeffizienten bestimmen: $g(x) \mathrel{\hat{=}} 1101$ und $m(x) \mathrel{\hat{=}} 10100101$

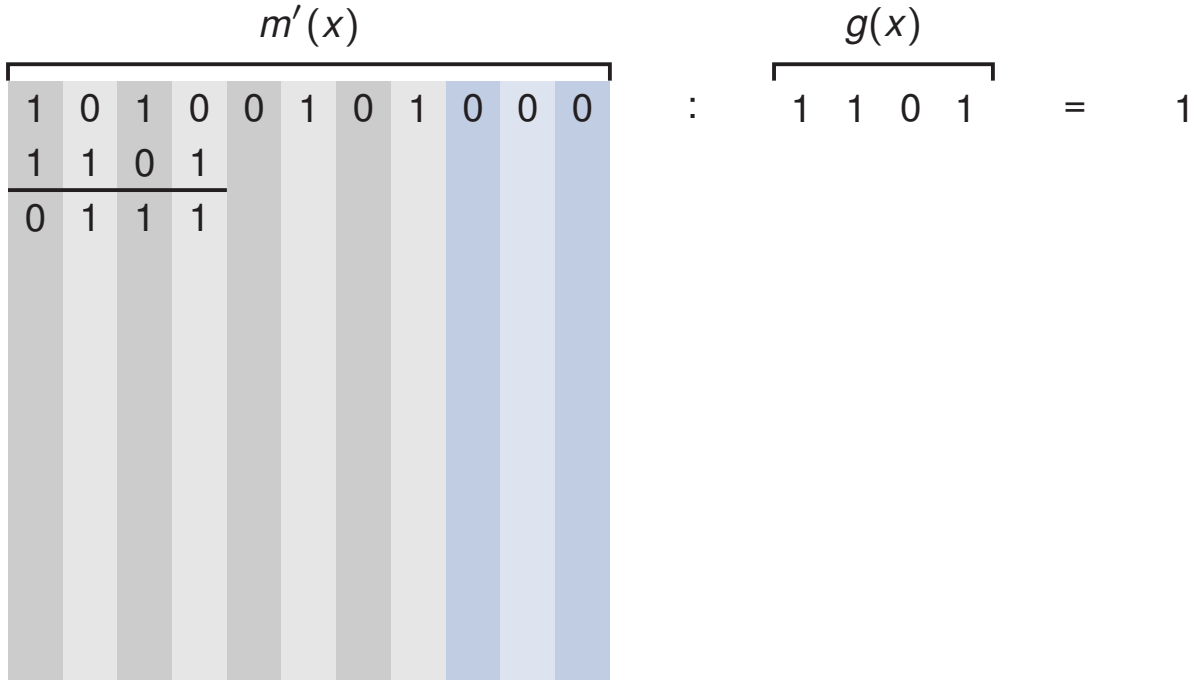
2. $\text{grad}(g(x)) = 3 \Rightarrow$ Daten mit x^3 multiplizieren. Dies entspricht dem „Anhängen“ von 3 Nullen: $m'(x) = m(x) \cdot x^3 \mathrel{\hat{=}} 10100101000$

3. Polynomdivision $m'(x)/g(x)$ ausführen und den Rest $r(x)$ bestimmen.

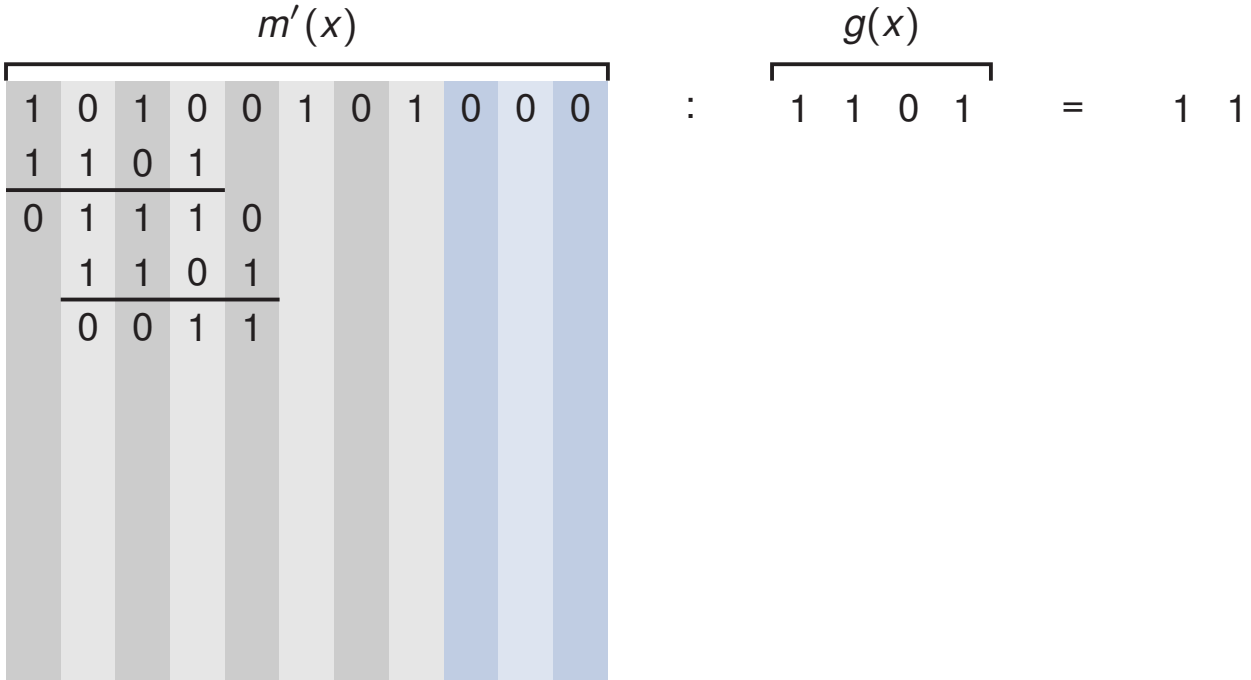
Beispiel: Generator $g(x) = x^3 + x^2 + 1$, Daten $m(x) = x^7 + x^5 + x^2 + 1$



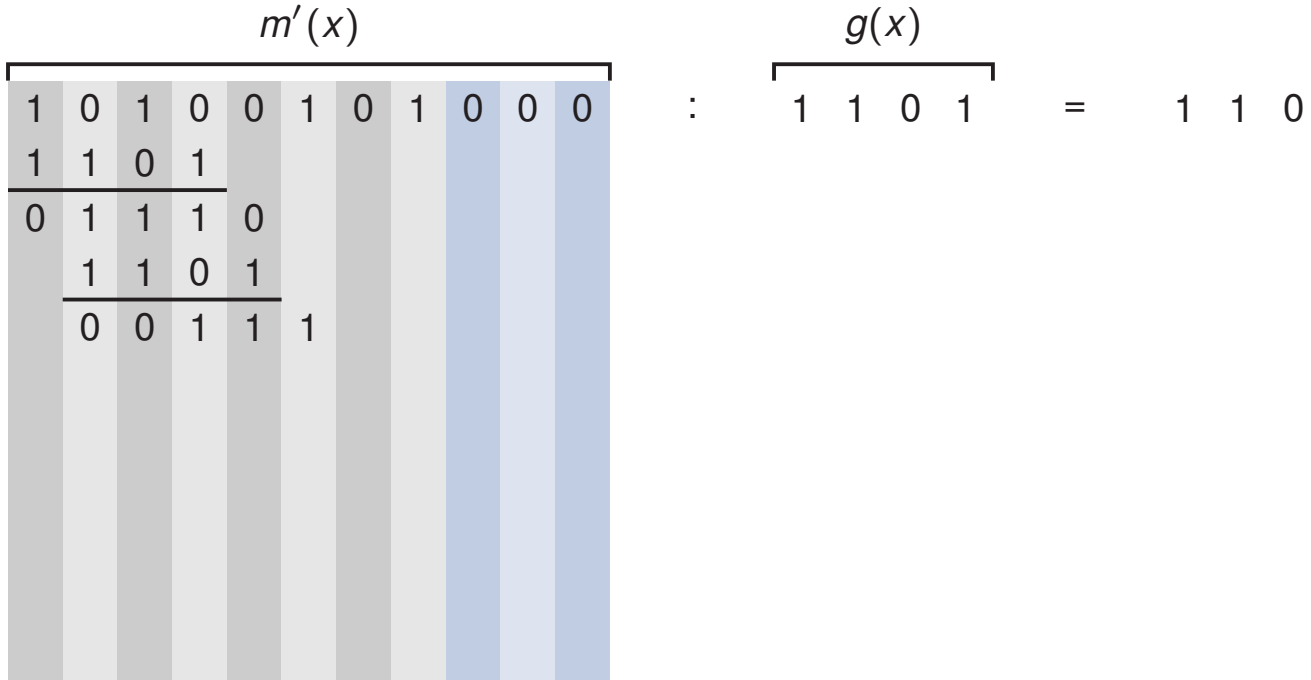
Beispiel: Generator $g(x) = x^3 + x^2 + 1$, Daten $m(x) = x^7 + x^5 + x^2 + 1$



Beispiel: Generator $g(x) = x^3 + x^2 + 1$, Daten $m(x) = x^7 + x^5 + x^2 + 1$



Beispiel: Generator $g(x) = x^3 + x^2 + 1$, Daten $m(x) = x^7 + x^5 + x^2 + 1$

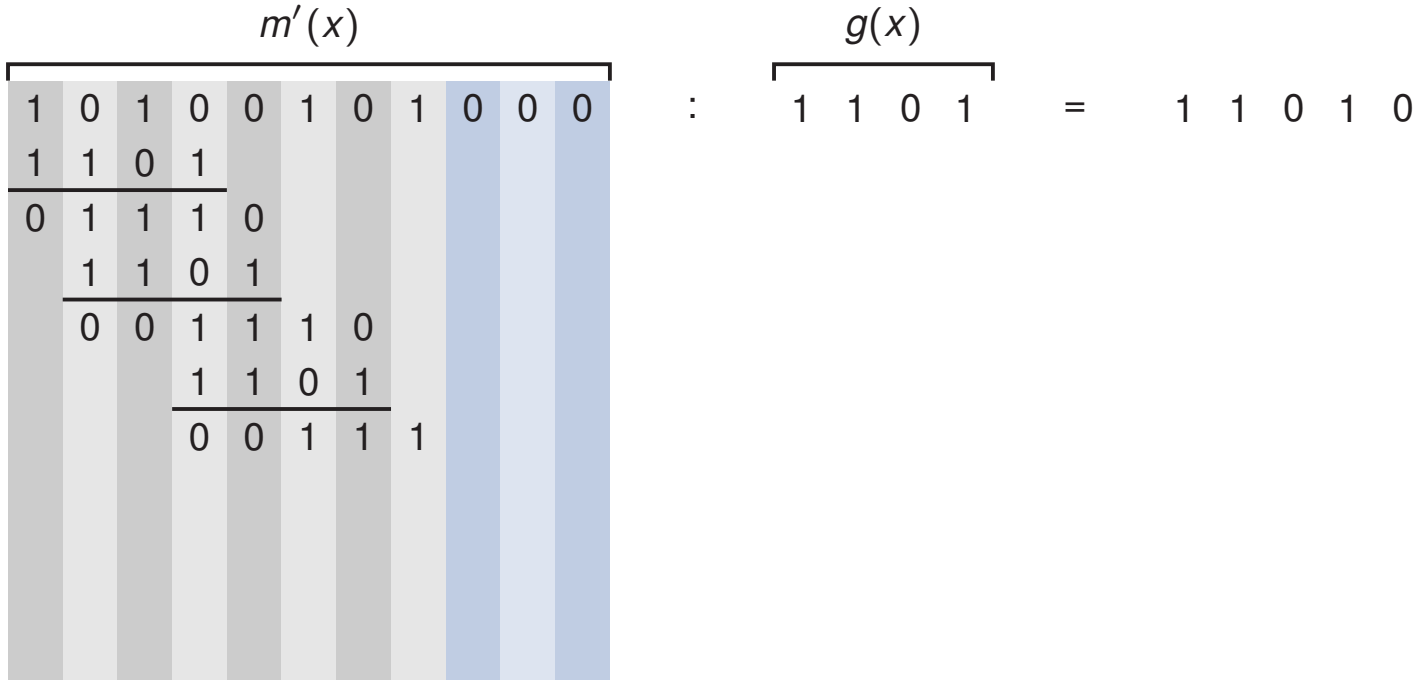


Beispiel: Generator $g(x) = x^3 + x^2 + 1$, Daten $m(x) = x^7 + x^5 + x^2 + 1$

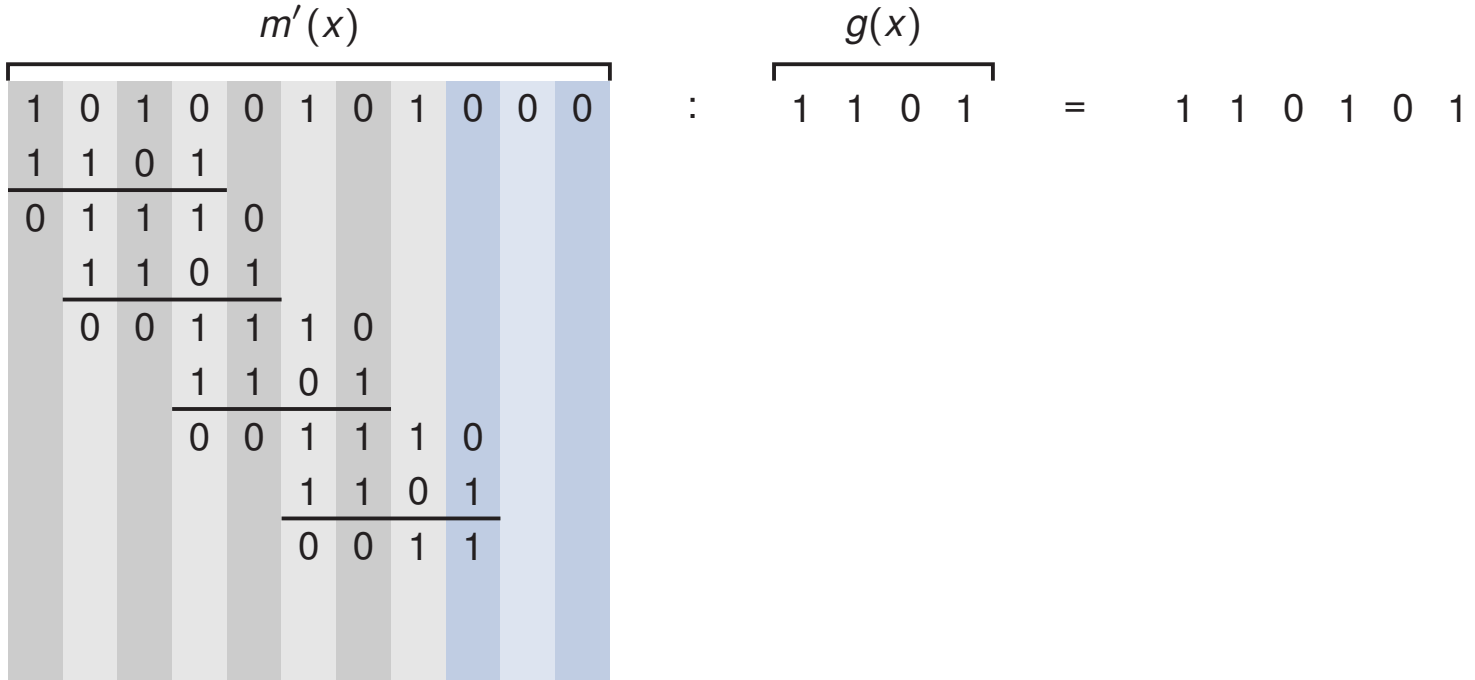
$m'(x)$												$g(x)$				
1	0	1	0	0	1	0	1	0	0	0	:	1	1	0	1	=
1	1	0	1													
0	1	1	1	0												
	1	1	0	1												
	0	0	1	1	1	0										
			1	1	0	1										
			0	0	1	1										

1 1 0 1

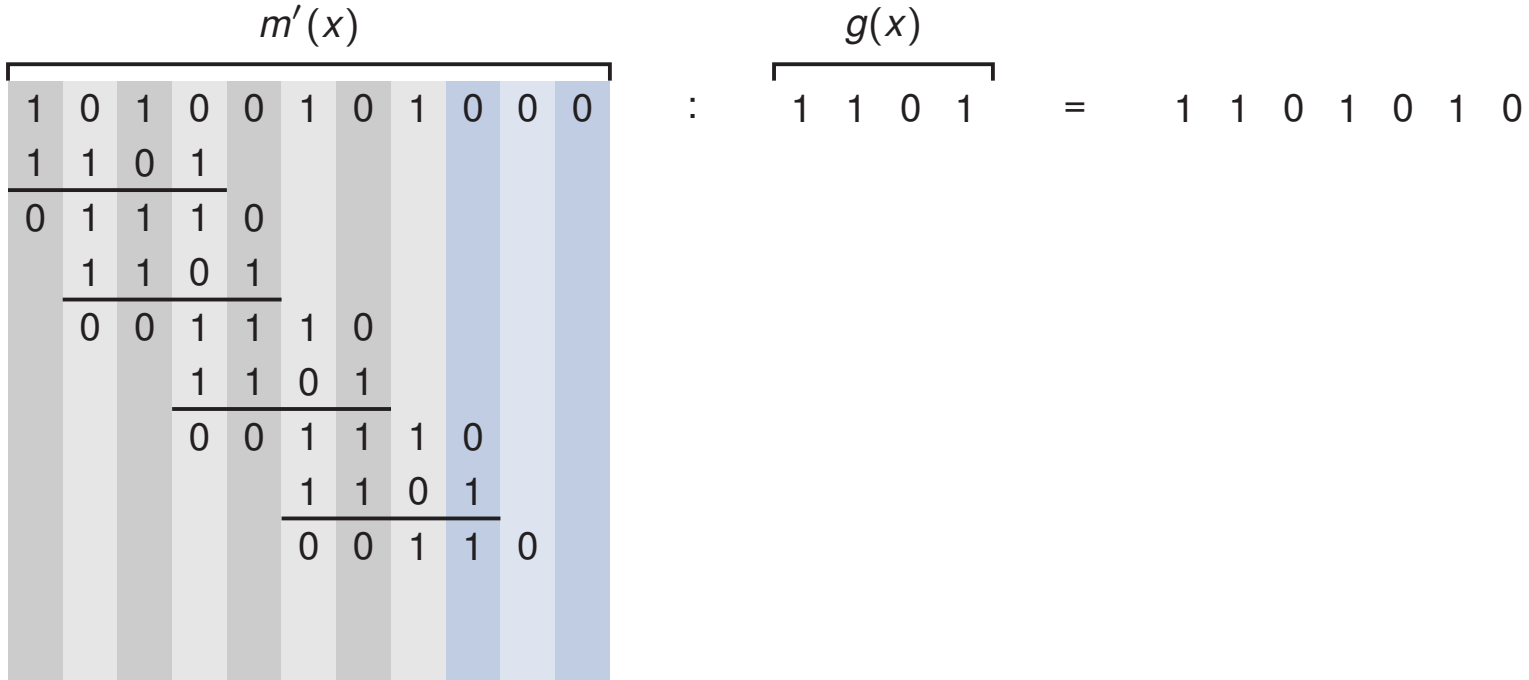
Beispiel: Generator $g(x) = x^3 + x^2 + 1$, Daten $m(x) = x^7 + x^5 + x^2 + 1$



Beispiel: Generator $g(x) = x^3 + x^2 + 1$, Daten $m(x) = x^7 + x^5 + x^2 + 1$



Beispiel: Generator $g(x) = x^3 + x^2 + 1$, Daten $m(x) = x^7 + x^5 + x^2 + 1$



Beispiel: Generator $g(x) = x^3 + x^2 + 1$, Daten $m(x) = x^7 + x^5 + x^2 + 1$

$m'(x)$: $g(x)$ = 1 1 0 1 0 1 0 1

1	0	1	0	0	1	0	1	0	0	0
1	1	0	1							
0	1	1	1	0						
	1	1	0	1						
	0	0	1	1	1	0				
			1	1	0	1				
			0	0	1	1	1	0		
					1	1	0	1		
					0	0	1	1	0	0
							1	1	0	1
							0	0	0	1

$= r(x)$

Beispiel: Generator $g(x) = x^3 + x^2 + 1$, Daten $m(x) = x^7 + x^5 + x^2 + 1$

1. Koeffizienten bestimmen: $g(x) \mathrel{\hat{=}} 1101$ und $m(x) \mathrel{\hat{=}} 10100101$

2. $\text{grad}(g(x)) = 3 \Rightarrow$ Daten mit x^3 multiplizieren. Dies entspricht dem „Anhängen“ von 3 Nullen: $m'(x) = m(x) \cdot x^3 \mathrel{\hat{=}} 10100101000$

3. Polynomdivision $m'(x)/g(x)$ ausführen und den Rest $r(x)$ bestimmen.

Beispiel: Generator $g(x) = x^3 + x^2 + 1$, Daten $m(x) = x^7 + x^5 + x^2 + 1$

1. Koeffizienten bestimmen: $g(x) \mathrel{\hat{=}} 1101$ und $m(x) \mathrel{\hat{=}} 10100101$

2. $\text{grad}(g(x)) = 3 \Rightarrow$ Daten mit x^3 multiplizieren. Dies entspricht dem „Anhängen“ von 3 Nullen: $m'(x) = m(x) \cdot x^3 \mathrel{\hat{=}} 10100101000$

3. Polynomdivision $m'(x)/g(x)$ ausführen und den Rest $r(x)$ bestimmen.

4. Die zu sendende Nachricht ist $s(x) = m'(x) + r(x)$. Die Addition reduziert sich auf ein XOR, da wir auf GF(2) arbeiten.

$$\begin{array}{rcccccccccc} & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & = m'(x) \\ \oplus & & & & & & & & & 0 & 0 & 1 & = r(x) \\ \hline & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & = s(x) \end{array}$$

Beispiel: Generator $g(x) = x^3 + x^2 + 1$, Daten $m(x) = x^7 + x^5 + x^2 + 1$

1. Koeffizienten bestimmen: $g(x) \mathrel{\hat{=}} 1101$ und $m(x) \mathrel{\hat{=}} 10100101$
2. $\text{grad}(g(x)) = 3 \Rightarrow$ Daten mit x^3 multiplizieren. Dies entspricht dem „Anhängen“ von 3 Nullen: $m'(x) = m(x) \cdot x^3 \mathrel{\hat{=}} 10100101000$
3. Polynomdivision $m'(x)/g(x)$ ausführen und den Rest $r(x)$ bestimmen.
4. Die zu sendende Nachricht ist $s(x) = m'(x) + r(x)$. Die Addition reduziert sich auf ein XOR, da wir auf GF(2) arbeiten.

Der Empfänger prüft die Nachricht, indem er $r'(x) = (s(x) + e(x))/g(x)$ bestimmt, wobei $e(x)$ für mögliche Übertragungsfehler steht:

- $r'(x) \neq 0$ besagt, dass sicher ein Fehler aufgetreten ist
- $r'(x) = 0$ besagt, dass mit hoher Wahrscheinlichkeit kein Fehler aufgetreten ist

Welche Fehler erkennt CRC?

Sei N die Länge der Checksumme, also $N = \text{grad}(g(x))$. Dann werden die folgenden Fehler erkannt:

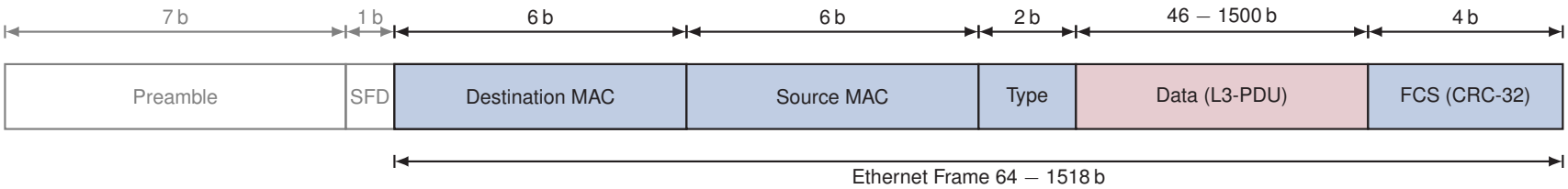
- Alle 1 bit-Fehler
- Isolierte 2 bit-Fehler, d. h. Fehler an den Bitstellen i und j wobei $i > j$
- Alle Burst-Fehler, deren Länge kleiner ist als N
- Einige Burst-Fehler, die länger sind als N

Welche Fehler erkennt CRC nicht zuverlässig oder gar nicht?

- Fehler, die länger sind als N
- Fehler, die aus mehreren Bursts bestehen
- Alle Fehler, die ein Vielfaches des Generatorpolynoms sind

Fallbeispiel: IEEE 802.3u (FastEthernet)

Frame vor der 4B5B-Kodierung:

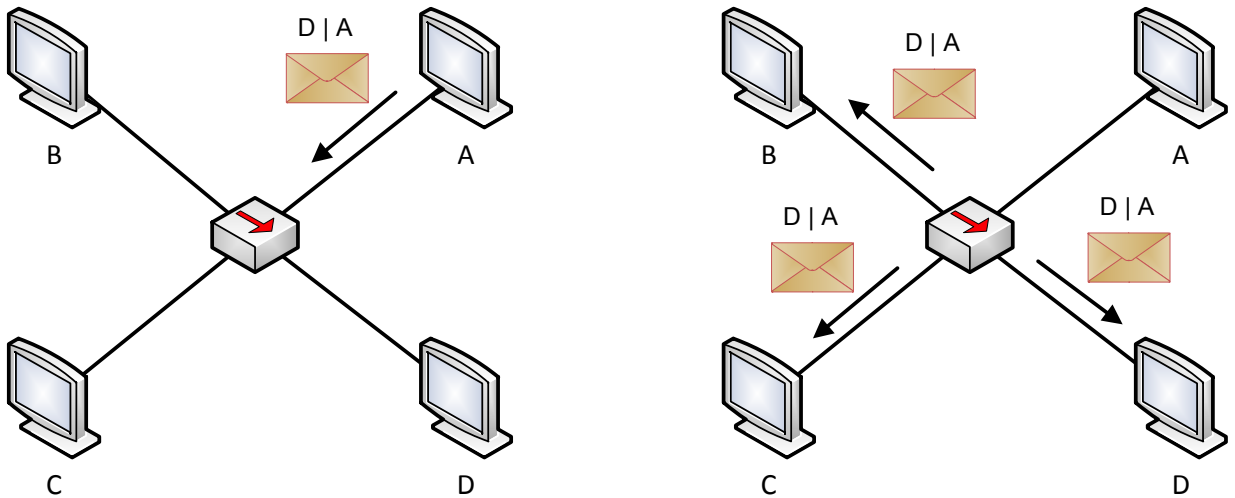


- Präambel und Start Frame Delimiter (SFD) dienen der Taktsynchronisation
- Das erste Byte der Präambel wird durch das J/K-Symbol des 4B5B-Codes ersetzt (Start of Frame)
- Nach der Frame Check Sequence (FCS) wird das T/R-Symbol des 4B5B-Codes eingefügt (End of Frame)
- Zwischen J/K und T/R liegende Daten werden gemäß des 4B5B-Codes kodiert
- Das Typfeld gibt die Art des Frames an
  (z. B. 0x0800 $\hat{=}$ IPv4 Payload, 0x0806 $\hat{=}$ ARP)
- Das Datenfeld muss (vor der Kodierung) mind. 46 b lang sein – andernfalls wird es bis zu diesem Wert gepadded

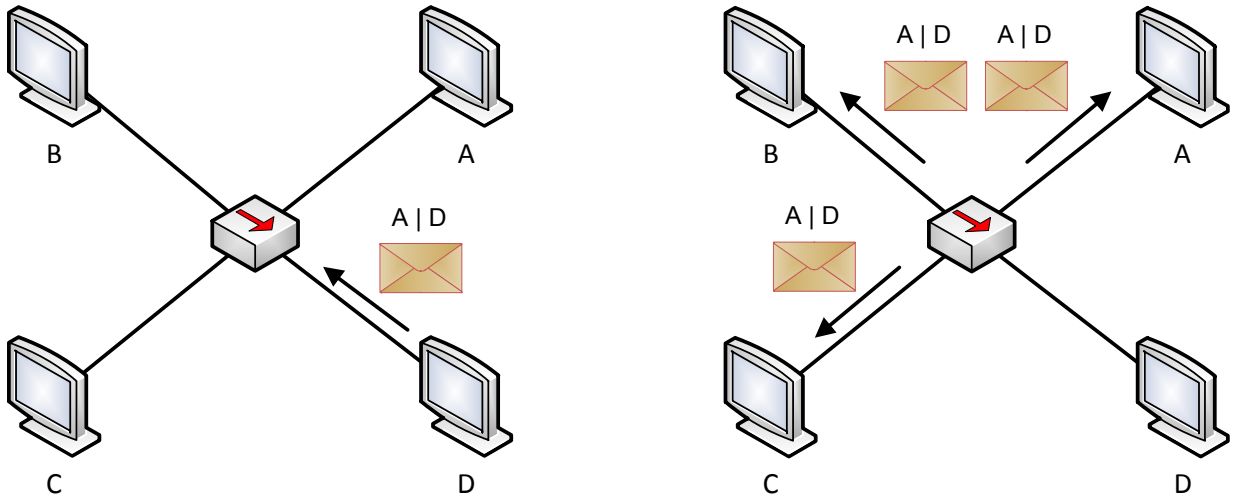
Übersicht

Verbindung auf Schicht 1: Hub [3]

- Knoten A sendet einen Rahmen an Knoten D
- Der Hub verbindet die einzelnen Links zu einem gemeinsamen Bus
- Der Rahmen erreicht alle Knoten
- Es darf folglich zu jedem Zeitpunkt nur ein Knoten senden, andernfalls treten Kollisionen auf



- Knoten D antwortet auf den Rahmen von A
- Auch die Antwort erreicht alle Knoten



Definition (Collision Domain)

Unter einer Kollisions-Domäne versteht man den Teil eines Direktverbindungsnetzes, innerhalb dem eine Kollision bei gleichzeitiger Übertragung mehrerer Knoten auftreten kann. Dieser wird häufig auch als Segment bezeichnet.

Sind Hubs mehr als nur Sternverteiler?

Man unterscheidet aktive und passive Hubs:

- Aktive Hubs (Repeater) verstärken die Signale auf der physikalischen Schicht, ohne dabei die in Rahmen enthaltenen Felder wie Adressen oder Checksummen zu prüfen
- Passive Hubs sind wirklich nur Sternverteiler – man könnte genauso gut die einzelnen Adern der Patchkabel verlöten

Kann man Hubs kaskadieren? Ja, aber es gilt bei Ethernet mit Baumtopologie (802.3a/i) die 5-4-3-Regel:

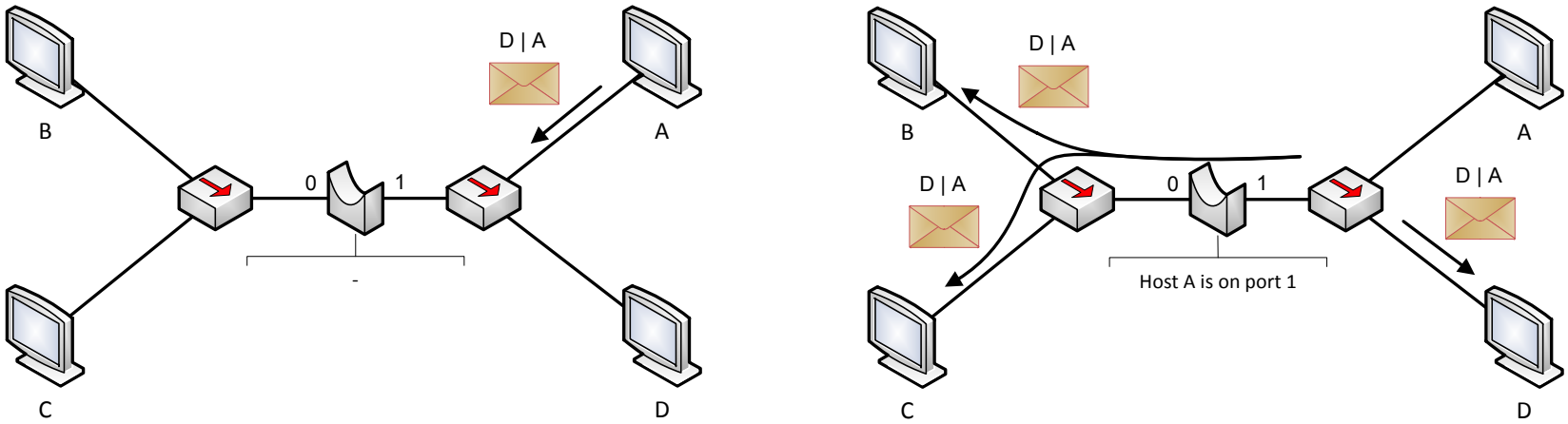
- Nicht mehr als 5 Abschnitte,
- verbunden durch 4 Repeater,
- wobei nur in 3 Abschnitten aktive Endgeräte enthalten sein dürfen.

Jeder Abschnitt soll aufgrund der Dämpfung bei 802.3a (10BASE-2) nicht länger als 185 m sein, bei 802.3i (10BASE-T) nicht länger als 100 m zwischen Hub und Endgerät (Dämpfung).
Aufgrund einer sicheren Kollissionserkennung ergibt sich bei 100BASE-TX eine maximale Ausdehnung von 500 m ($\rightarrow$ Übung).

Können Hubs unterschiedliche Medientypen miteinander verbinden?

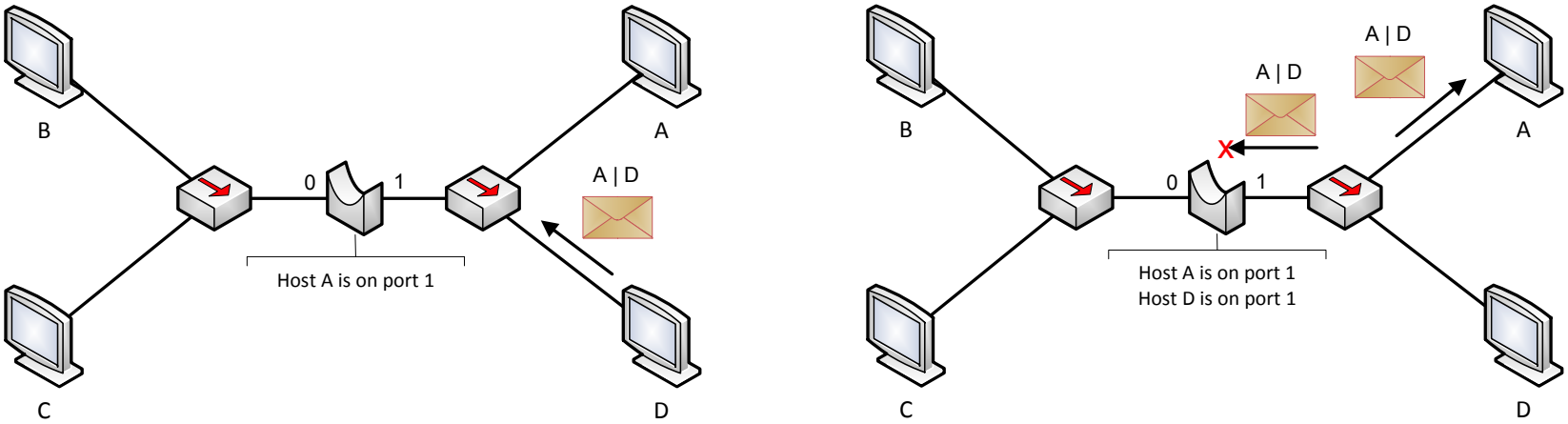
- Ja, wenn auf allen Abschnitten dasselbe Medienzugriffsverfahren genutzt wird (beispielsweise Verbindung Ethernet über BNC- und Patch-Kabel mit jeweils gleicher Datenrate).
- Unterschiedliche Zugriffsverfahren können nicht gekoppelt werden.

Verbindung auf Schicht 2: Brücke [3]



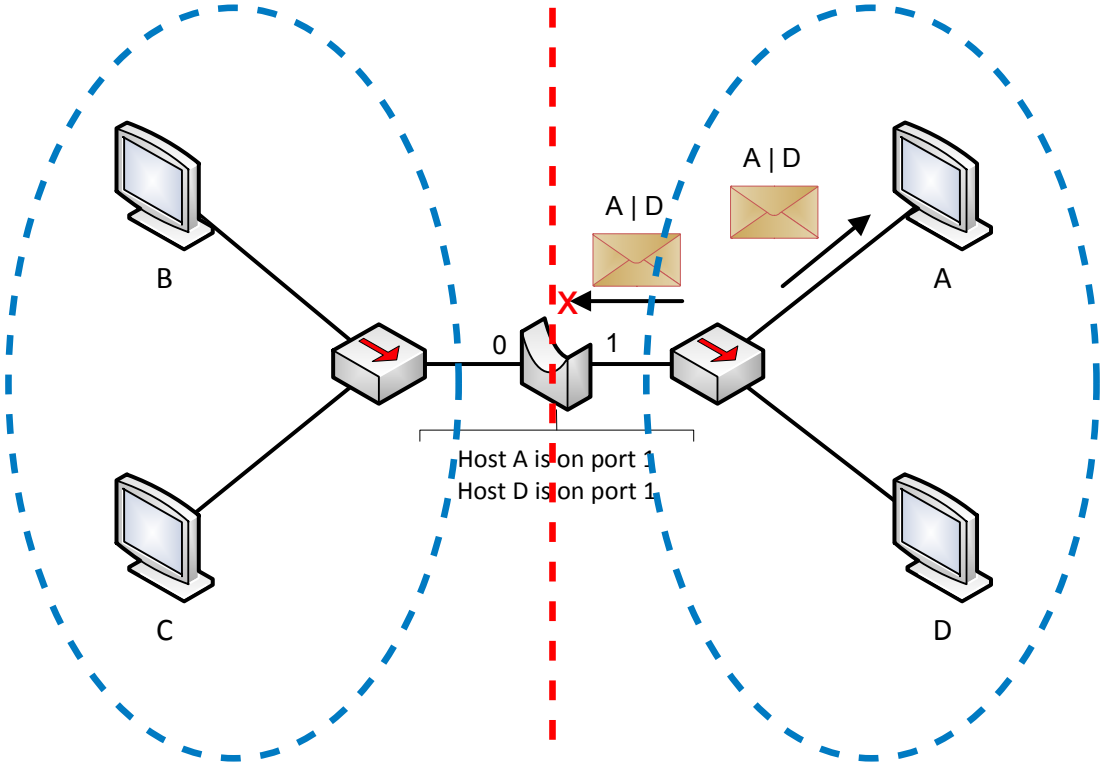
- Zwei Gruppen von Hosts, die jeweils über Hubs verbunden sind, werden im obigen Beispiel durch eine Brücke (engl. Bridge) gekoppelt
- Die Brücke arbeitet zunächst wie ein Hub mit 2 Ports (Learning-Phase)
- Dabei merkt sich die Brücke, über welchen Port ein Rahmen empfangen wurde
- So ordnet sie ihren beiden Ports 0 und 1 die MAC-Adressen der Knoten zu, die an den jeweiligen Port angeschlossen sind

Verbindung auf Schicht 2: Brücke



- Die Ziel-Adresse eingehender Rahmen wird mit den Einträgen in der Switching-Table verglichen
- Ist ein Eintrag vorhanden, wir der Rahmen nur an den betreffenden Ziel-Port weitergeleitet
- Ist kein Eintrag vorhanden, so wird der Rahmen an alle Ports weitergeleitet
- Einträge erhalten einen Zeitstempel (Timestamp) und werden nach einem festen Zeitintervall invalidiert

- Eine Brücke unterbricht Kollisionsdomänen (auch als Segmente bezeichnet)
- Wenn die Brücke alle angeschlossenen Geräte kennt, darf in jedem der beiden Segmente jeweils ein Knoten zur selben Zeit senden

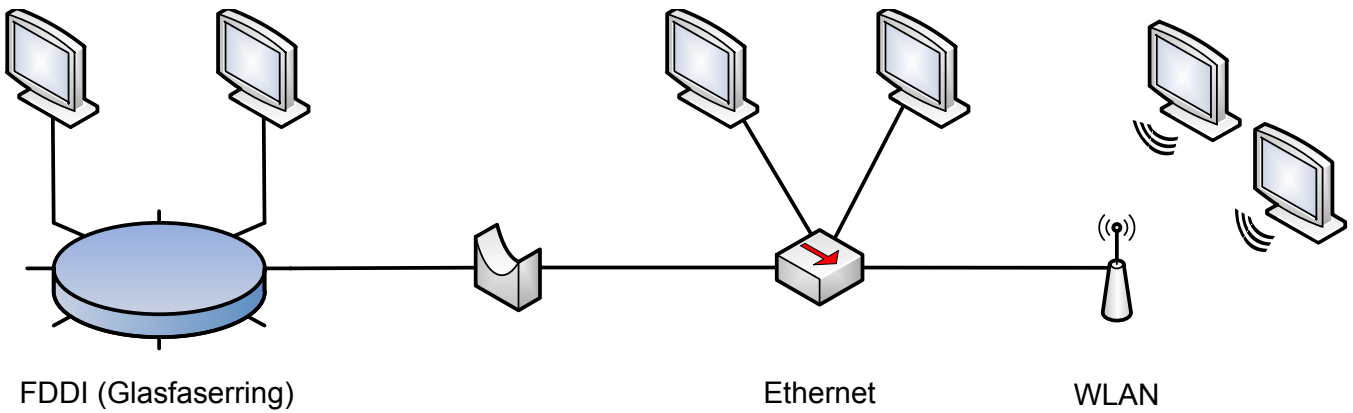


Brücken können auch genutzt werden, um Netzsegmente mit unterschiedlichen Zugriffsverfahren zu koppeln. Derartige Brücken bezeichnet man auch als Translation Bridges:

- FDDI-Ethernet Bridge zwischen Token Passing und CSMA/CD
- WLAN Access Point zwischen CSMA/CD und CSMA/CA

Diese Kopplung ist transparent, d. h.

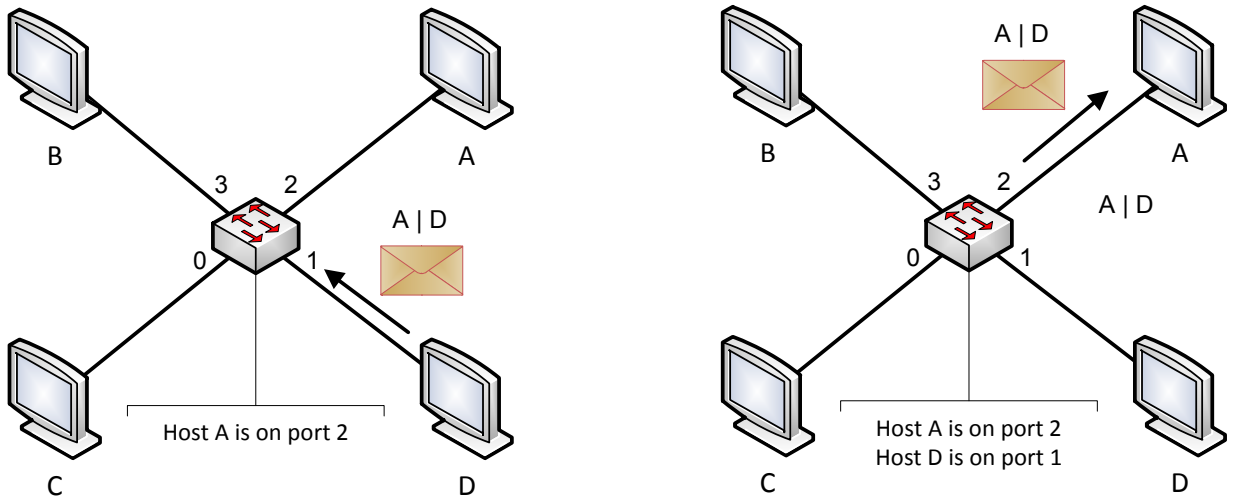
- angeschlossene Stationen bemerken nicht, dass eine Brücke verwendet wird und
- im normalen Betrieb wird ein Host niemals direkt mit der Brücke kommunizieren. (Ausnahme: Management-Anwendung kommuniziert mit Brücke)



Voraussetzung für Translation Bridges:
Die MAC-Adressen müssen „kompatibel“ sein, um Empfänger über ihre MAC-Adressen identifizieren zu können.

Switches – Brücken mit mehreren Ports

Ähnlich wie Brücken speichern Switches in einer Tabelle, welche Hosts an welchen Port angeschlossen sind:



Mit Switches ist ein kollisionsfreier Betrieb auch bei Ethernet möglich:

- An jedem Port darf höchstens ein Host (oder ein anderes Switch) angeschlossen sein
- Die Hosts müssen Vollduplex unterstützen

Anmerkungen

- Brücken und Switches sind für Hosts transparent, d. h. ein Host weiß nicht, dass er über eine Brücke mit anderen Hosts kommuniziert.
- Sender- und Empfänger-Adresse werden von Brücken nicht verändert
- Brücken schränken nicht die Erreichbarkeit innerhalb des Direktverbindungsnetzes ein
- Ein Broadcast (MAC-Adresse ff:ff:ff:ff:ff:ff) wird von allen Hosts empfangen (man spricht daher auch von Broadcast-Domänen im Unterschied zu einer Kollisions-Domäne)
- Brücken und Switches benötigen zur Erfüllung ihrer grundlegenden Aufgaben keine eigene Adresse
- Weiterleitungsentscheidungen werden aber auf Basis der Ziel-Adresse und der aktuellen Switching-Tabelle getroffen

Ferner unterscheidet man zwischen zwei unterschiedlichen Switching-Arten:

- Store-and-Forward: Eingehende Rahmen werden vollständig empfangen und deren FCS geprüft. Falls der Ausgangsport belegt ist, kann eine begrenzte Anzahl von Rahmen gepuffert werden.
- Cut-Through: Beginne mit der Serialisierung des Rahmens, sobald der Ausgangsport bestimmt wurde. Die FCS wird in diesem Fall nicht geprüft.

Schleifen auf Schicht 2

Problembeschreibung

- Schleifen auf Schicht 1 bedeuten Kurzschluss.
- Schleifen auf Schicht 2 führen dazu, dass mehrere Kopien eines Rahmens erzeugt werden und im Netzwerk zirkulieren.

Wie entstehen Schleifen?

- Auch wenn Direktverbindungsnetze räumlich begrenzt sind, kann man schnell den Überblick verlieren und ungewollt Schleifen erzeugen.
- Manchmal erzeugt man Schleifen auch absichtlich, so dass redundante Pfade entstehen. Fällt eine Verbindung aus, kann der Verkehr umgeleitet werden.

Wie werden Schleifen vermieden?

- Switches unterstützen das sog. Spanning Tree Protocol (STP)
- Ziel ist die Deaktivierung redundanter Pfade, so dass alle Netzsegmente schleifenfrei erreichbar sind
- Fällt eine Verbindung aus, wird ggf. einer dieser Pfade reaktiviert

Zusammenfassung

Wir sollten wissen,

- wie Netzwerke als Graphen dargestellt werden können,
- was der Unterschied zwischen einem MST und einem SPT ist,
- welche unterschiedlichen Medienzugriffsverfahren es gibt,
- wie diese Kollisionen vermeiden oder mit ihnen umgehen,
- warum die maximale Länge eines Ethernet-Segments 500 m beträgt,
- wie Knoten in Direktverbindungsnetzen adressiert werden,
- wie MAC-Adressen bei Ethernet aufgebaut sind,
- wie mehrere Direktverbindungsnetze zu einem größeren miteinander verbunden werden können,
- worin der Unterschied zwischen Hubs, Bridges und Switches besteht,
- wie Switches lernen, an welchem Port, welche Geräte angeschlossen sind und wie Weiterleitungsentscheidungen getroffen werden und
- was eine Kollisions-Domäne bzw. eine Broadcast-Domäne ist.

Literaturhinweise und Quellenangaben I

[1] Stein, E.: Taschenbuch Rechnernetze und Internet, Kapitel Konzepte: Lokale Netzwerke, Seiten 191–218.
Fachbuchverlag Leipzig, 2. Auflage, 2004.
Auszug s. Moodle/SVN.

[2] Stein, E.: Taschenbuch Rechnernetze und Internet, Kapitel Fehlererkennung durch CRC, Seiten 86–87.
Fachbuchverlag Leipzig, 2. Auflage, 2004.
Auszug s. Moodle/SVN.

[3] Stein, E.: Taschenbuch Rechnernetze und Internet, Kapitel Netzaufbau, Seiten 200–203.
Fachbuchverlag Leipzig, 2. Auflage, 2004.
Auszug s. Moodle/SVN.