Lehrstuhl für Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München

Grundlagen Rechnernetze und Verteilte Systeme (GRNVS)

IN0010 – SoSe 2019

Prof. Dr.-Ing. Georg Carle

Dr.-Ing. Stephan Günther, Johannes Naab, Henning Stubbe

Lehrstuhl für Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München

Kapitel 2: Sicherungsschicht

Darstellung von Netzwerken als Graphen

Verbindungscharakterisierung, Mehrfachzugriff, Medienzugriffskontrolle

Rahmenbildung, Adressierung und Fehlererkennung

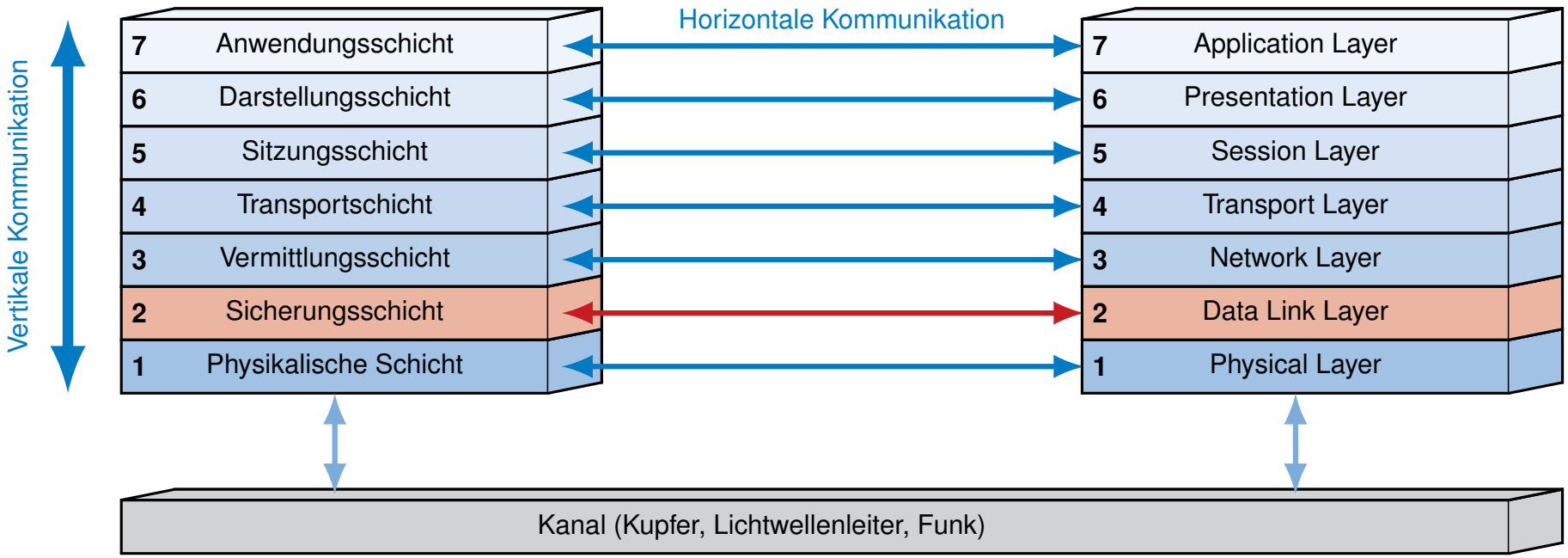
Verbindung auf Schicht 1 und 2

Zusammenfassung

Literaturangaben

Kapitel 2: Sicherungsschicht

Einordnung im ISO/OSI-Modell



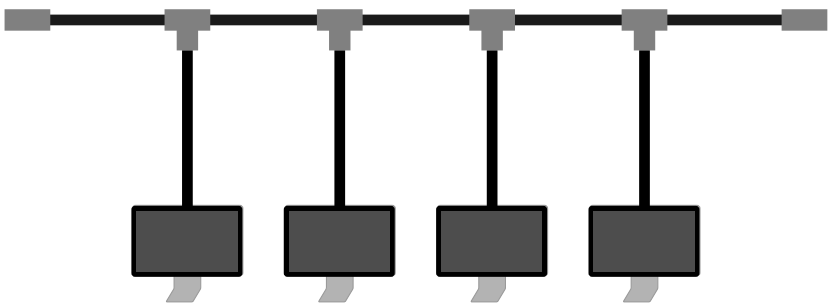
Kapitel 2: Sicherungsschicht

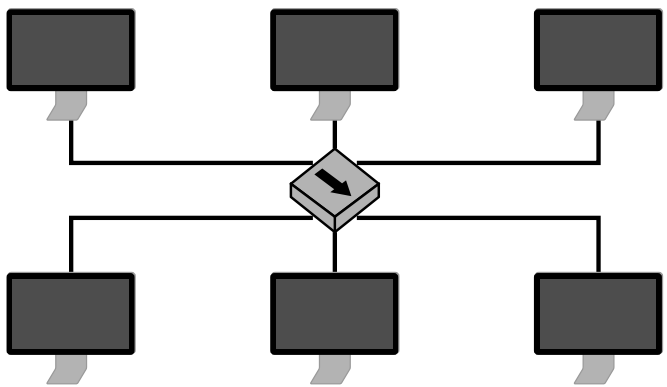
Wir beschäftigen uns zunächst mit sog. Direktverbindungsnetzen, d. h.

- 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 (hier Verbindung mittels Bus / Hub, aber auch mittels Switch möglich)





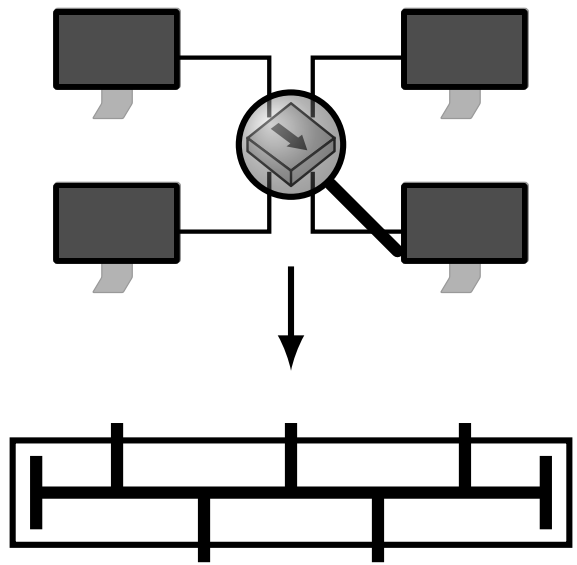
- Verbindung zwischen Basisstation und Mobiltelefon
- Bus-Systeme innerhalb eines Computers, z. B. USB, PCIe etc.

Die wesentlichen Aufgaben der Sicherungsschicht sind

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

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

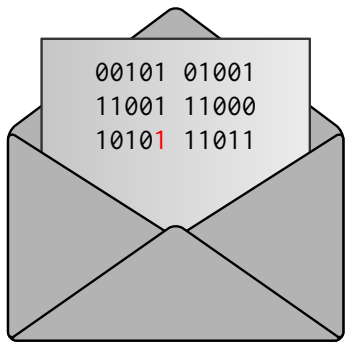


Die wesentlichen Aufgaben der Sicherungsschicht sind

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

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

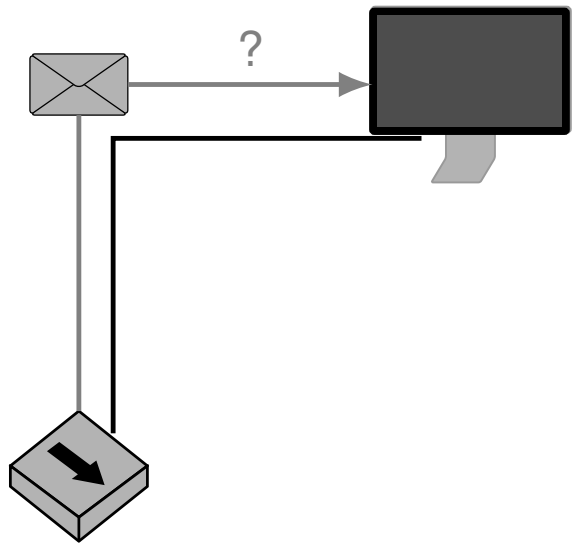


Die wesentlichen Aufgaben der Sicherungsschicht sind

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

Adressierung innerhalb von Direktverbindungsnetzen:

- 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



Kapitel 2: Sicherungsschicht

Darstellung von Netzwerken als Graphen

- Gerichtete Graphen
- Ungerichtete Graphen
- Pfade in Netzwerken
- Netztopologien
- Adjazenz- und Distanzmatrix
- Erzeugung von Baumstrukturen

Verbindungscharakterisierung, Mehrfachzugriff, Medienzugriffskontrolle

Rahmenbildung, Adressierung und Fehlererkennung

Verbindung auf Schicht 1 und 2

Zusammenfassung

Literaturangaben

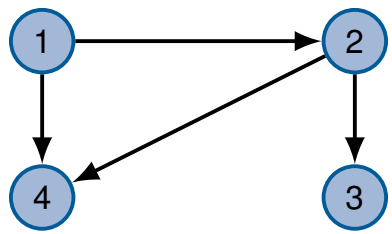
Motivation

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

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

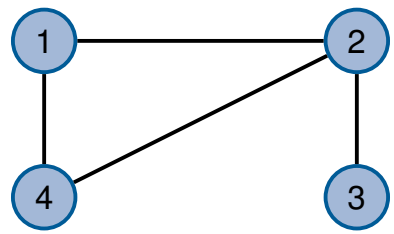


Ungerichtete Graphen

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

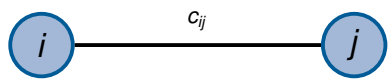
- $\mathcal{N}$ eine Menge von Knoten und
- $\mathcal{E} = \{\{i,j\} \mid i,j \in \mathcal{N} \wedge i, j \text{ sind ungerichtet verbunden}\}$ eine Menge ungerichteter Kanten (Edges) bezeichnet.

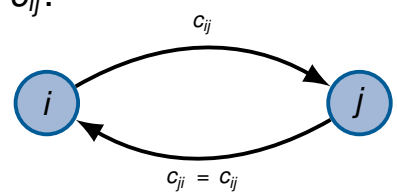
Beispiel: $\mathcal{N} = \{1,2,3,4\}$, $\mathcal{E} = \{\{1,2\},\{2,3\},\{2,4\},\{1,4\}\}$



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

Auch Pfade lassen sich mittels Graphen abbilden:

- Ein Pfad zwischen zwei Knoten[1] $s,t \in \mathcal{N}$ ist eine Menge

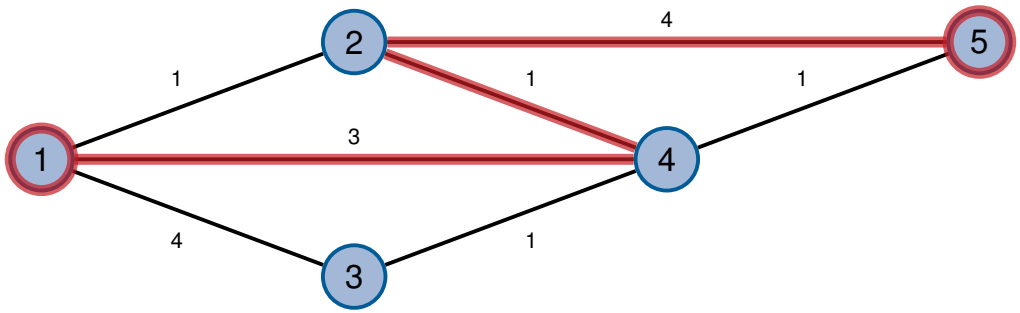
$$\mathcal{P}_{st} = \{(s,i),(i,j), \ldots ,(k,l),(l,t)\}$$

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



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

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

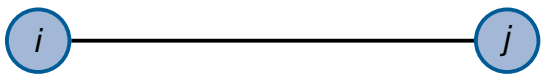
Netztopologien

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

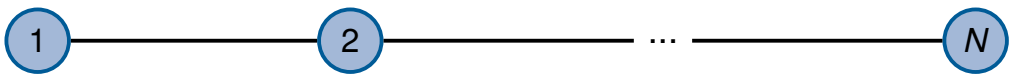
- physikalische Topologie und die
- logische Topologie.

Wichtige Topologien (Beispiele)

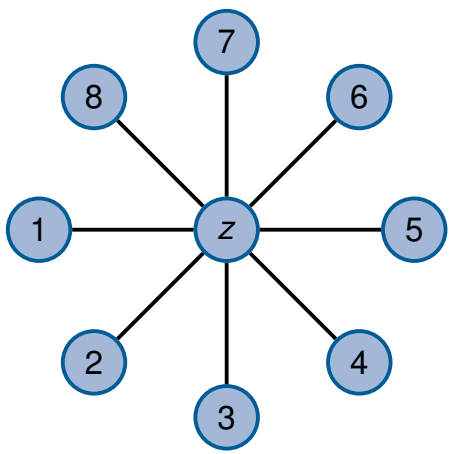
- Punkt-zu-Punkt (engl. Point-to-Point)



- Kette

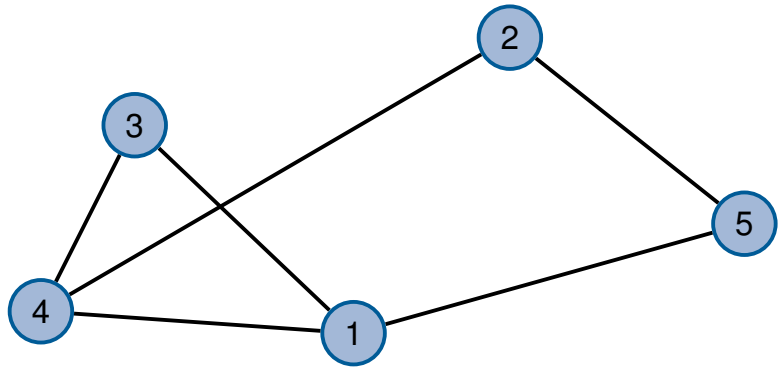


- Stern

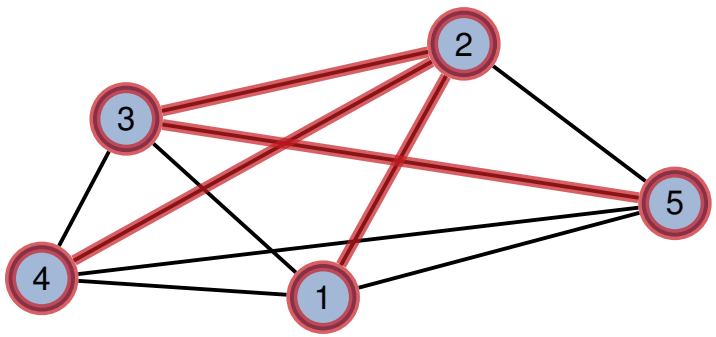


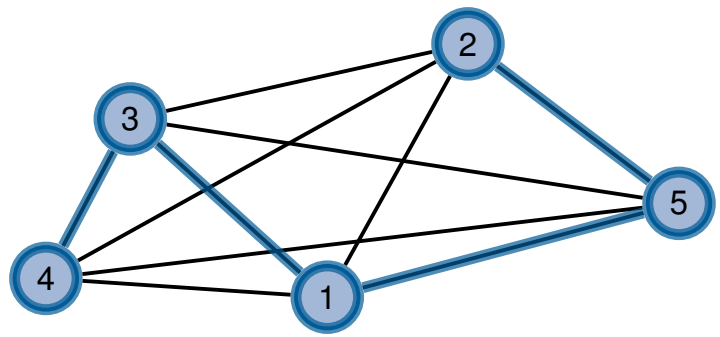
Netztopologien

- Vermaschung (engl. Mesh)

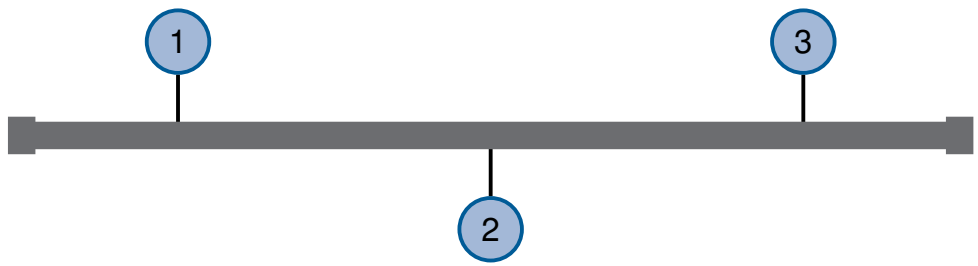


- Baum (meist logische Topologie)





- Bus



Adjazenzmatrix

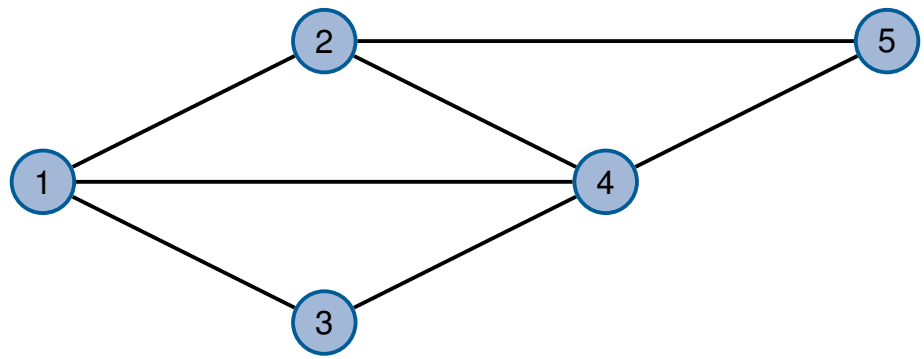
Netzwerke lassen sich leicht als Matrizen schreiben. Die Adjazenzmatrix

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

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

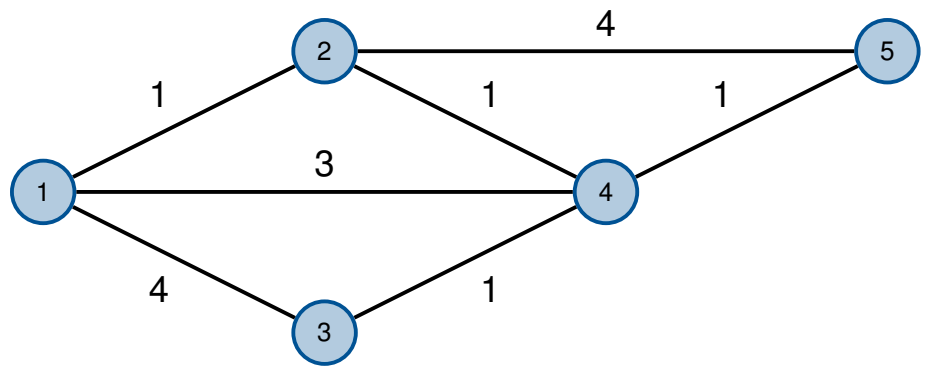
Die Distanzmatrix

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

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.

Adjazenz- und Distanzmatrix

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

Adjazenz- und Distanzmatrix

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 $\boldsymbol{D}^{n+1} = \boldsymbol{D}^n = \boldsymbol{D}^*$.

Adjazenz- und Distanzmatrix

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

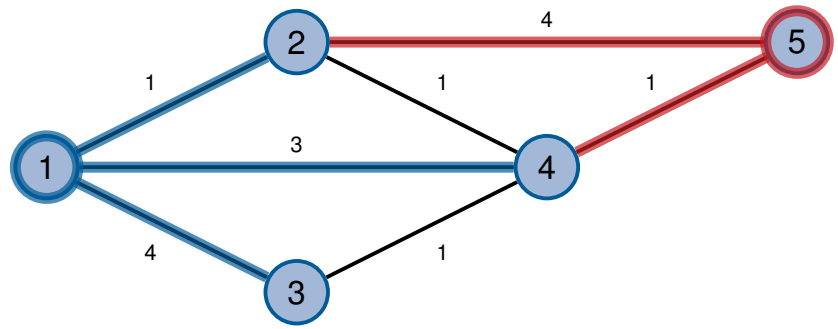
Antwort: Man potenziert $\boldsymbol{D}$ bzgl. des min-plus-Produkts

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

- 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).

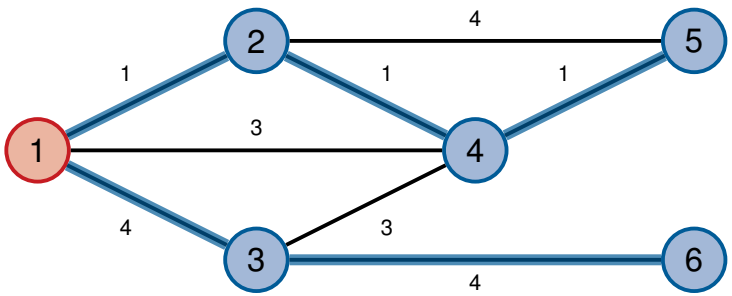
Erzeugung von Baumstrukturen

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

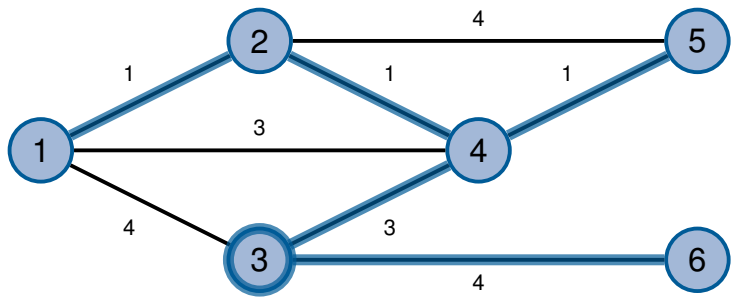
- Shortest Path Tree (SPT)
  Verbindet einen Wurzelknoten mit jeweils minimalen Kosten mit jedem anderen Knoten des Netzwerks.
- Minimum Spanning Tree (MST)
  Verbindet alle Knoten des Netzwerks mit insgesamt minimalen Kosten.

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)

Kapitel 2: Sicherungsschicht

Darstellung von Netzwerken als Graphen

Verbindungscharakterisierung, Mehrfachzugriff, Medienzugriffskontrolle

- Verbindungscharakterisierung
- Medienzugriff
- ALOHA und Slotted ALOHA
- CSMA, CSMA/CD, CSMA/CA
- Token Passing
- Zusammenfassung

Rahmenbildung, Adressierung und Fehlererkennung

Verbindung auf Schicht 1 und 2

Zusammenfassung

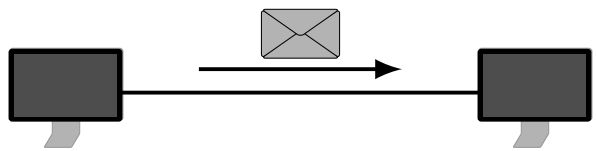
Literaturangaben

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

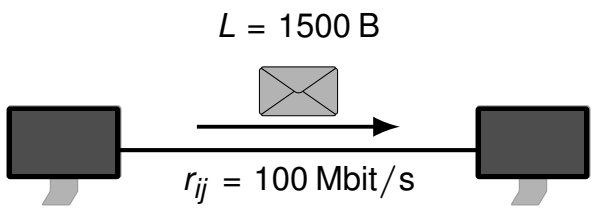
Übertragungsrate und Serialisierungszeit

Die Übertragungsrate r in bit/s bestimmt die notwendige Zeit, um L Datenbits auf ein Übertragungsmedium zu legen. Diese Zeit, auch Serialisierungszeit genannt, beträgt:

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

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.

Ausbreitungsverzögerung
Die Ausbreitungsverzögerung über eine Distanz d wird bestimmt von der endlichen Ausbreitungsgeschwindigkeit von Signalen, welche relativ zur Lichtgeschwindigkeit im Vakuum $c \approx 300\,000\,\mathrm{km/s}$ angegeben wird:

$$t_p = \frac{d}{\nu c_0}.$$

Der Wert $0 < \nu < 1$ ist die relative Ausbreitungsgeschwindigkeit in einem Medium. Für typische isolierte Kupferleitungen 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 im Vakuum 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:

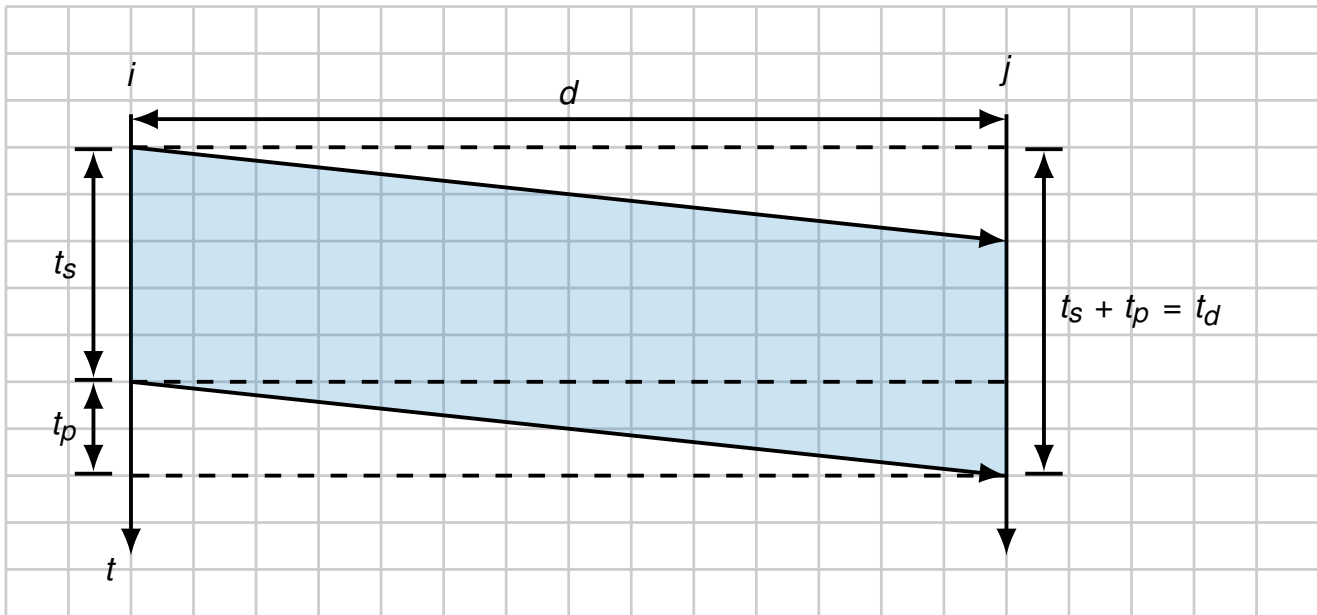


Abbildung 1: Nachrichtenflussdiagramm

- Die Gesamtverzögerung t_d (Delay) ergibt sich daher zu $t_d = t_s + t_p = \frac{L}{r} + \frac{d}{\nu c_0}$.
- 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.

Bandbreitenverzögerungsprodukt
Als Bandbreitenverzögerungsprodukt bezeichnet man die Anzahl an Bits (Kapazität)

$$C = t_p r = \frac{d}{\nu c_0} r,$$

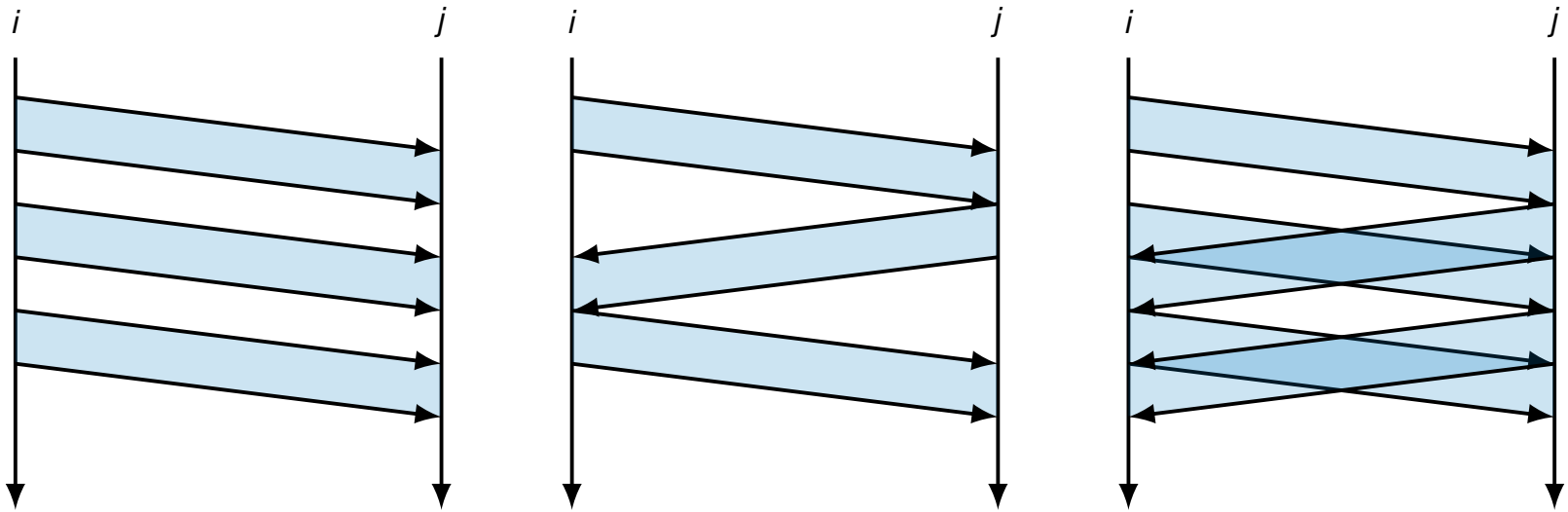
die sich in einer Senderichtung gleichzeitig auf der Leitung befinden können.

Beispiel:

- Leitung mit $r = 1\,\text{Gbit/s}$
- Länge $d = 10\,\text{m}$
- $\nu = 2/3$ (Kupferleitung)
- $C = t_p \cdot r = \frac{d}{\nu c_0} \cdot r = \frac{10\,\text{m}}{2/3 \cdot 3 \cdot 10^8\,\text{m/s}} \cdot 10^9\,\text{bit/s} \approx 50\,\text{bit}$

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:

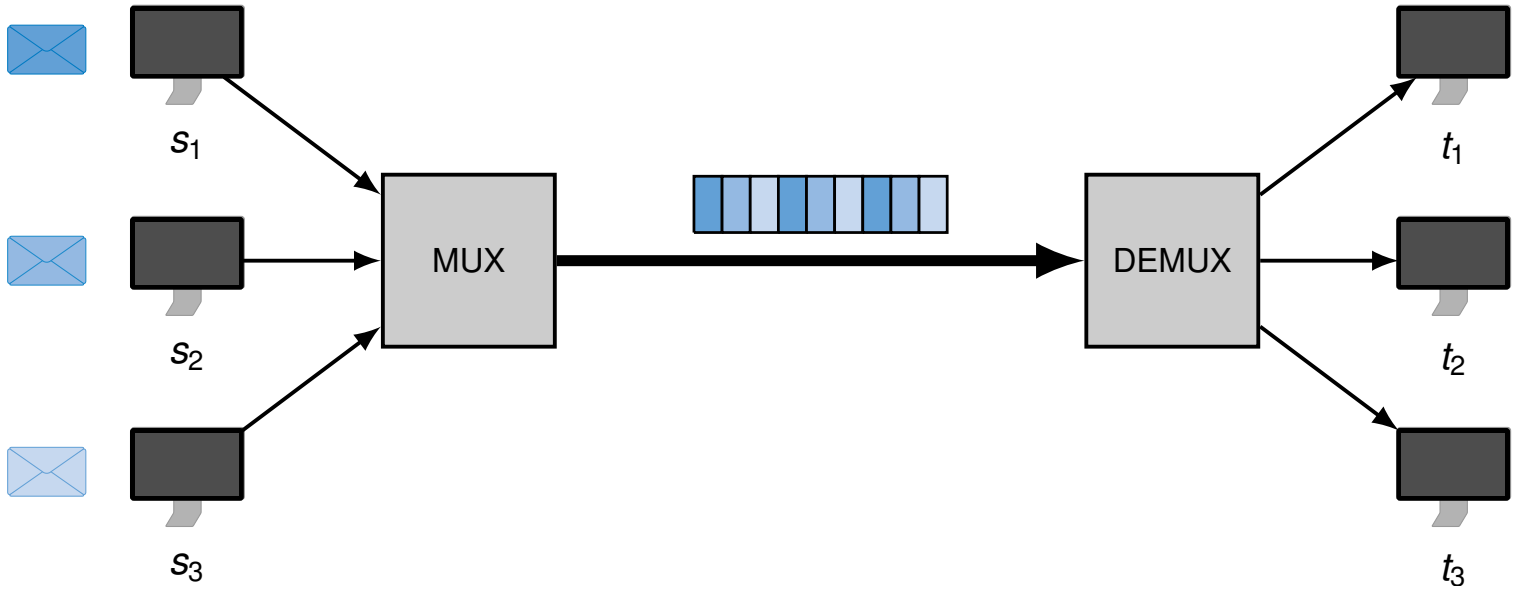


Abbildung 2: Deterministisches Zeitmultiplex-Verfahren

Ein anderes Zeitmultiplex-Verfahren haben wir bereits kennen gelernt:

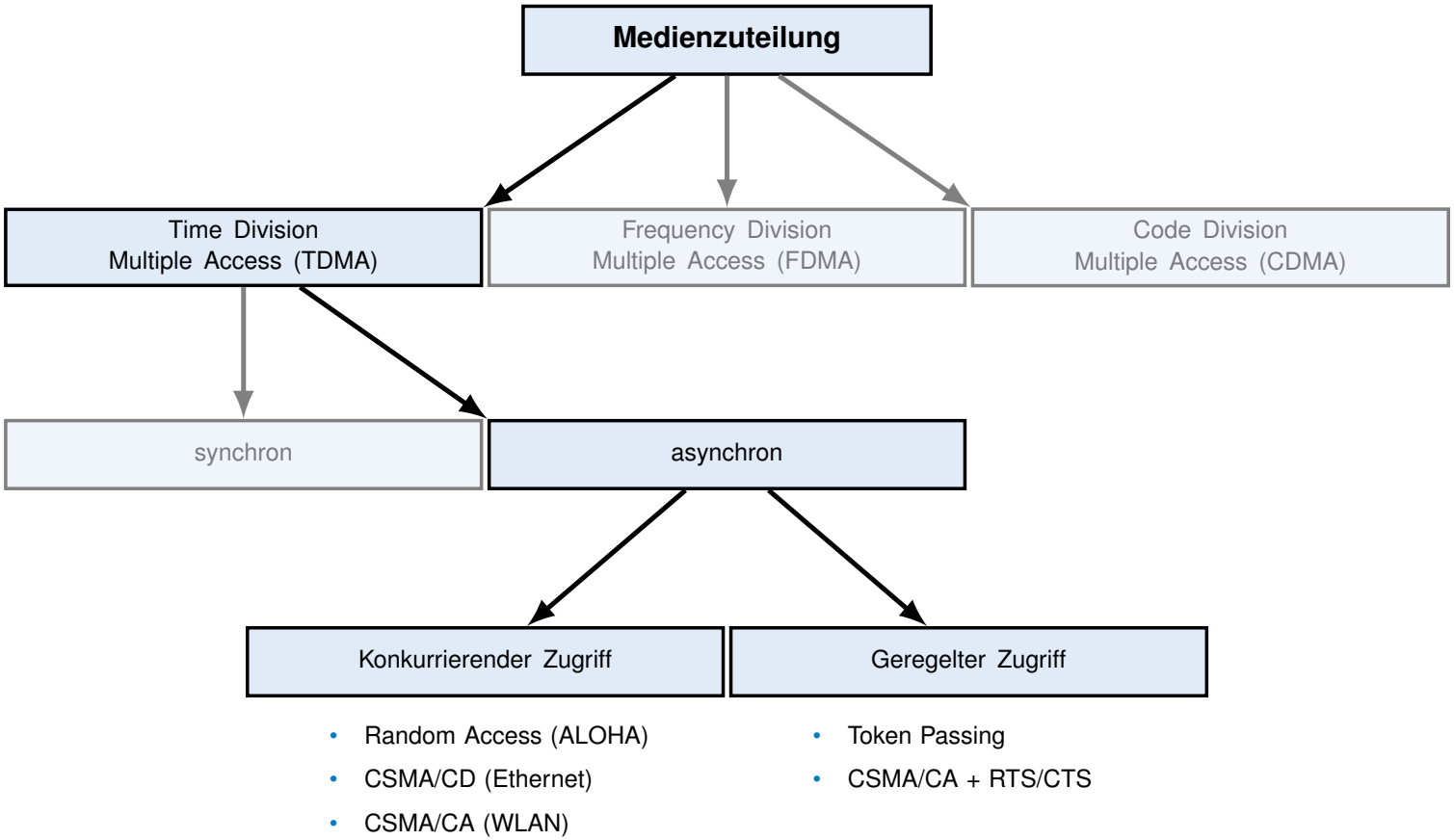
- Werden mehrere Computer mittels eines Hubs miteinander verbunden,
- so bildet das Hub ein gemeinsames geteiltes Medium,
- auf das die Computer mittels eines nicht-deterministischen Medienzugriffsverfahrens abwechselnd zugreifen.

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“ (Link Aggregation) bei Ethernet
  - 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* („CDMA-iPhone“)

Mehrfachzugriff und Medienzugriffskontrolle [2]

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



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

Bewertungskriterien für Medienzugriffsverfahren sind unter anderem:

- 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

Lösungsansatz: Asynchrones (flexibles) TDMA

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

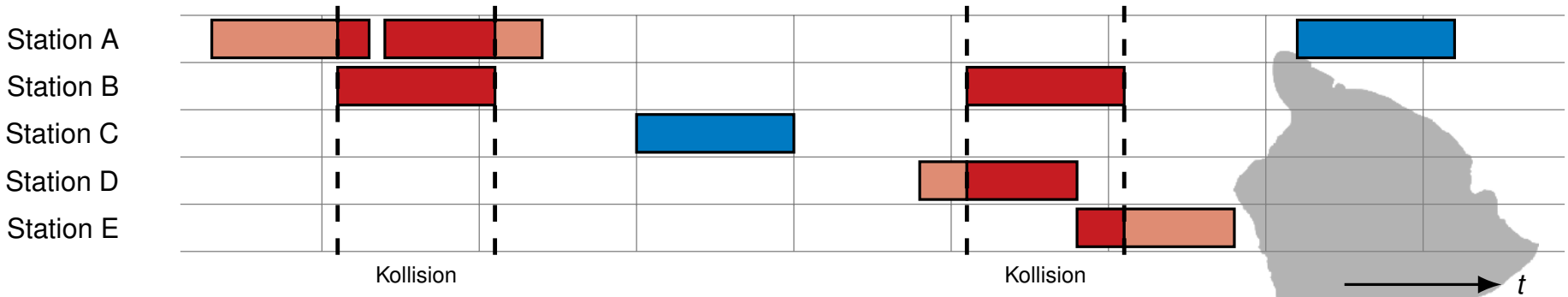
ALOHA und Slotted ALOHA

Random Access (ALOHA)

- Entwickelt an der Universität von Hawaii (1971), cf. 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 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 $p_i = p \quad \forall i = 1, \ldots, N$
- 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 $Np = \lambda$

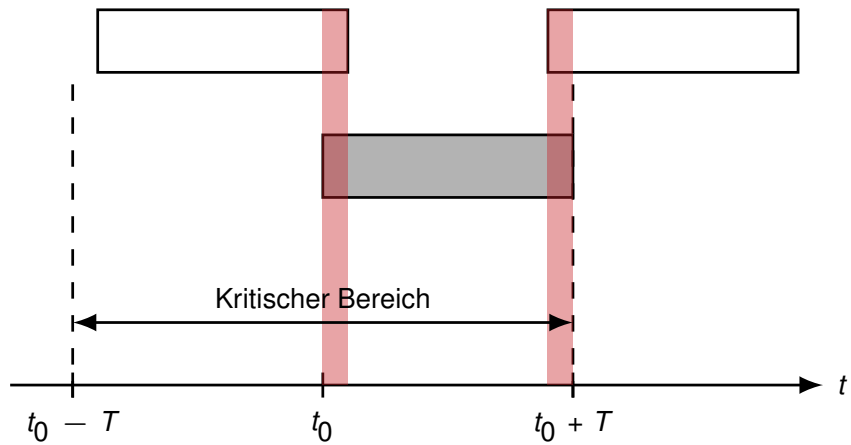
Das Ereignis X_t, dass im Intervall $[t,t+T)$ genau k Knoten senden, ist poisson-verteilt:

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

[2] Verteilung der seltenen Ereignisse

ALOHA und Slotted ALOHA

- Eine beliebige Station sende nun zum Zeitpunkt t_0 eine Nachricht
- Eine Kollision tritt genau dann auf, wenn mindestens 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.

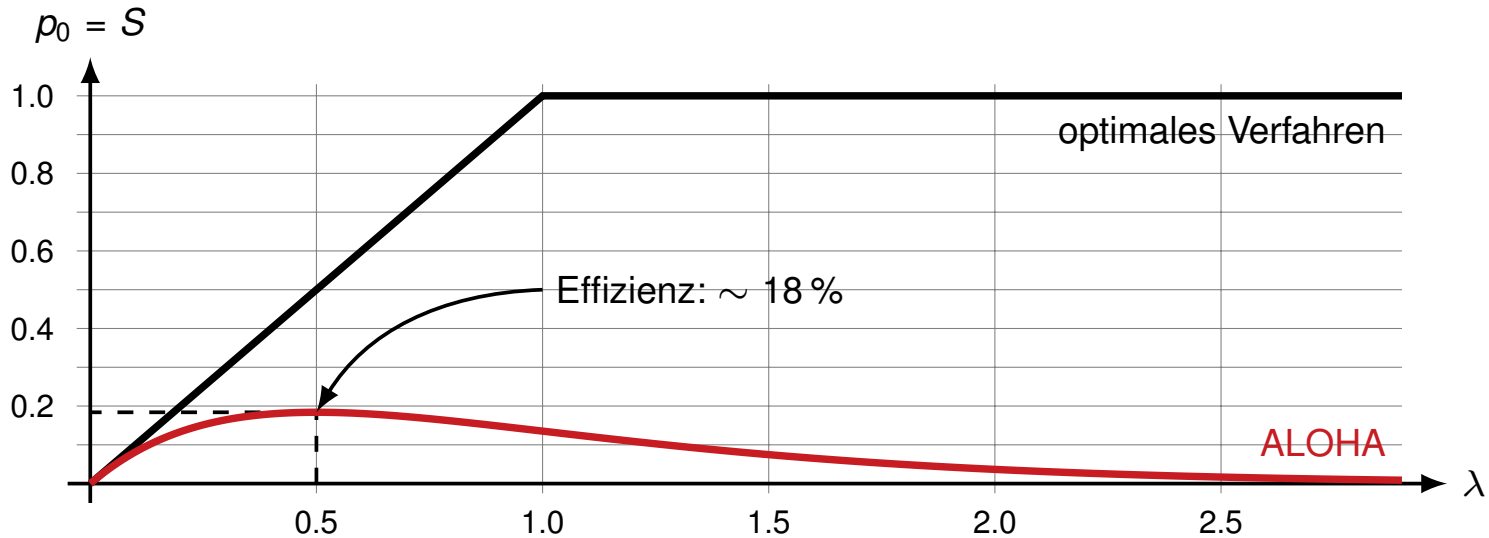


Mit der Dichtefunktion $\Pr[X_t = k] = \frac{\lambda^k e^{-\lambda}}{k!}$ der Poisson-Verteilung erhalten wir die Wahrscheinlichkeit p_0 für eine erfolgreiche Übertragung:

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

ALOHA und Slotted ALOHA

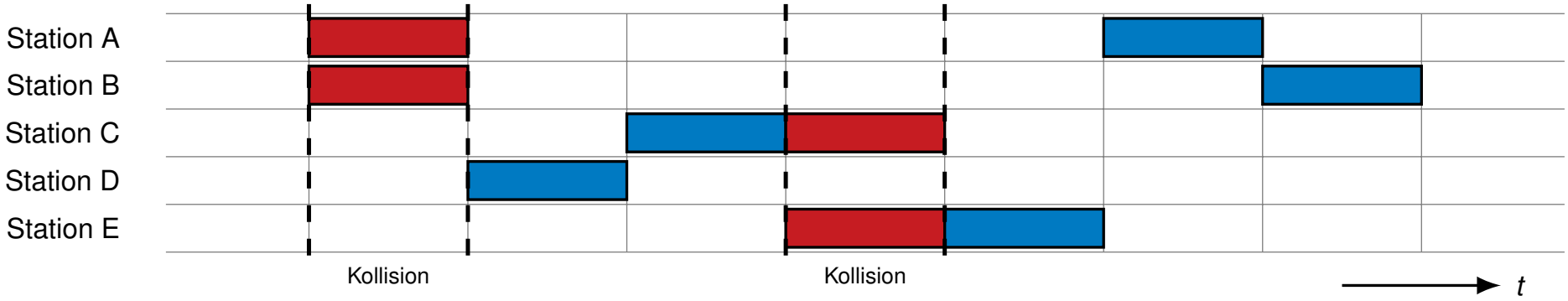
Die Erfolgswahrscheinlichkeit p_0 kann gegen die Senderate λ aufgetragen werden:



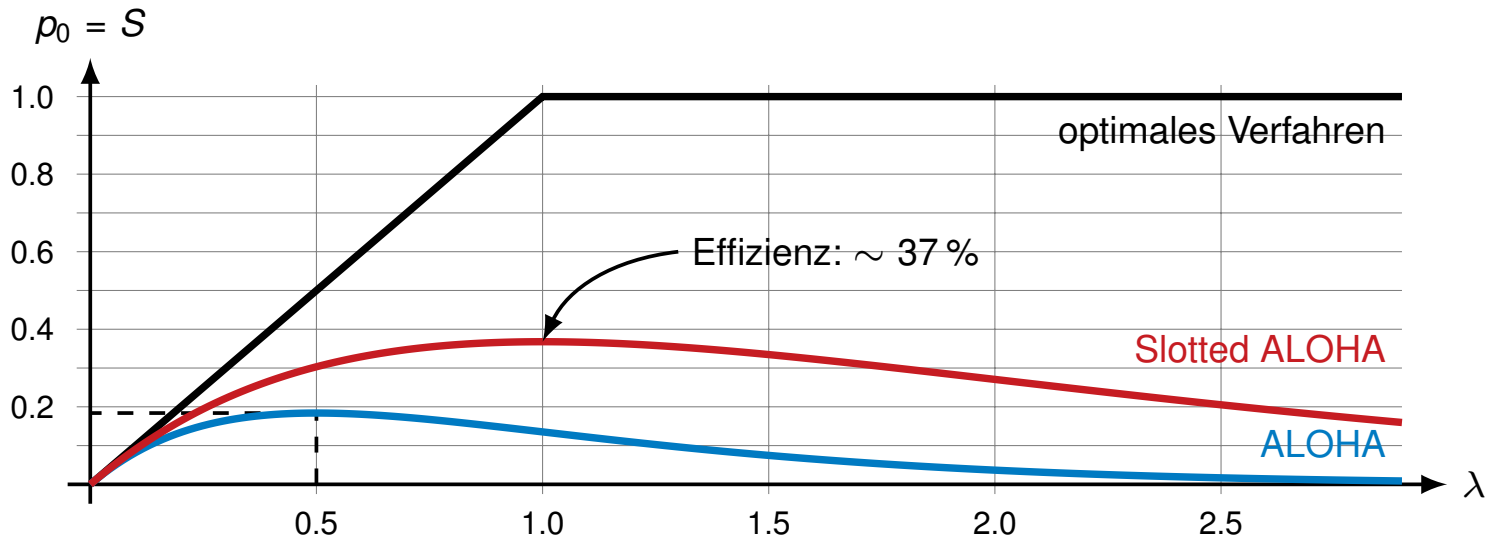
- Wir wissen, dass innerhalb eines beliebigen Intervalls $[t,t + T)$ höchstens eine Übertragung erfolgreich sein kann.
- Dementsprechend entspricht die Anzahl S der erfolgreichen Nachrichten pro Intervall gleichzeitig der Wahrscheinlichkeit für eine erfolgreiche Übertragung.
- Bei einem optimalen Verfahren 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.

ALOHA und Slotted ALOHA

Variante: Slotted ALOHA Stationen dürfen nicht mehr zu beliebigen Zeitpunkten mit einer Übertragung beginnen, sondern nur noch zu den Zeitpunkten $t = nT$, $n = 0,1, ...$



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



CSMA, CSMA/CD, CSMA/CA

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

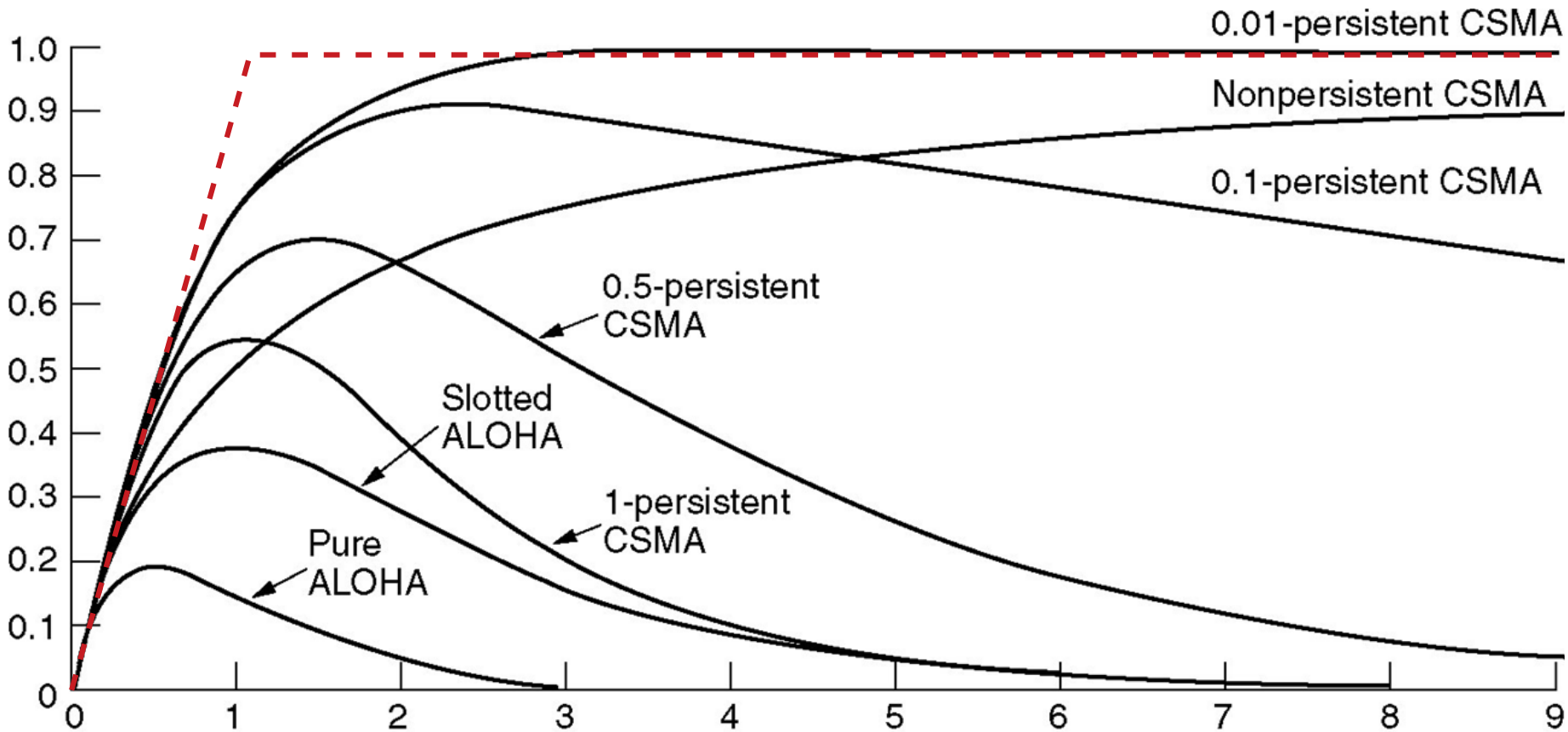
Verschiedene Varianten:

- 1-persistentes CSMA
  1. Wenn Medium frei, beginne Übertragung
  2. Wenn Medium belegt, warte bis frei und beginne dann Übertragung

- p-persistentes CSMA
  1. Wenn Medium frei, übertrage mit Wahrscheinlichkeit p oder verzögere mit Wahrscheinlichkeit $1 - p$ um eine feste Zeit dann 1.
  2. Wenn Medium belegt, warte bis frei, dann 1.

- nicht-persistentes CSMA
  1. Wenn Medium frei, beginne Übertragung
  2. Wenn belegt, warte eine zufällig gewählte Zeitspanne dann 1.

Alle bisherigen Verfahren im Vergleich



CSMA/CD (Collision Detection)

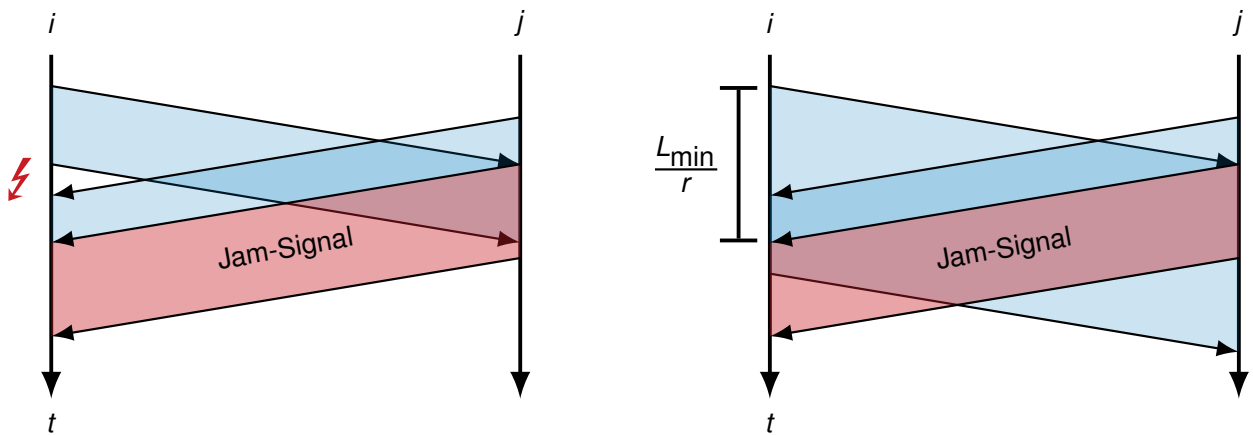
- Erkenne Kollisionen und wiederhole die Übertragung, wenn eine Kollision erkannt wird
- Verzichte auf das Senden von Bestätigungen
- Wird keine Kollision erkannt, gilt die Übertragung als erfolgreich

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

Voraussetzung für CSMA/CD [2]

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:

$$L_{\text{min}} = \frac{2d}{\nu c_0} r$$



CSMA, CSMA/CD, CSMA/CA

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

Beim k-ten Sendeversuch einer Nachricht

- wählt der Sender zufällig $n \in \left\{0, \ldots, \min\{2^{k-1} - 1, 1023\}\right\}$ aus und
- wartet n Slotzeiten vor einem erneuten Sendeversuch.

Die maximale Wartezeit ergibt sich bei $k = 11$ (also bei 12 Sendeversuchen) und beträgt 1023 Slotzeiten.

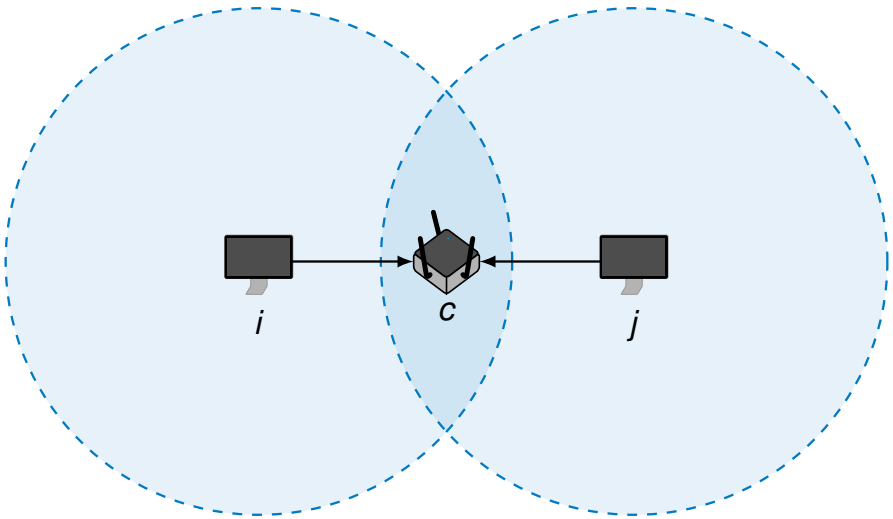
Durch die Wartezeiten, die

- 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 feste Zeit dann 1.
2. Wenn Medium belegt, warte bis frei, dann 1.

Fallbeispiel: IEEE 802.11 DCF (Distributed Coordination Function)

- Festes Zeitintervall zwischen Rahmen: DIFS (DCF Interframe Spacing).
- Wenn Medium mind. für DIFS unbelegt 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.

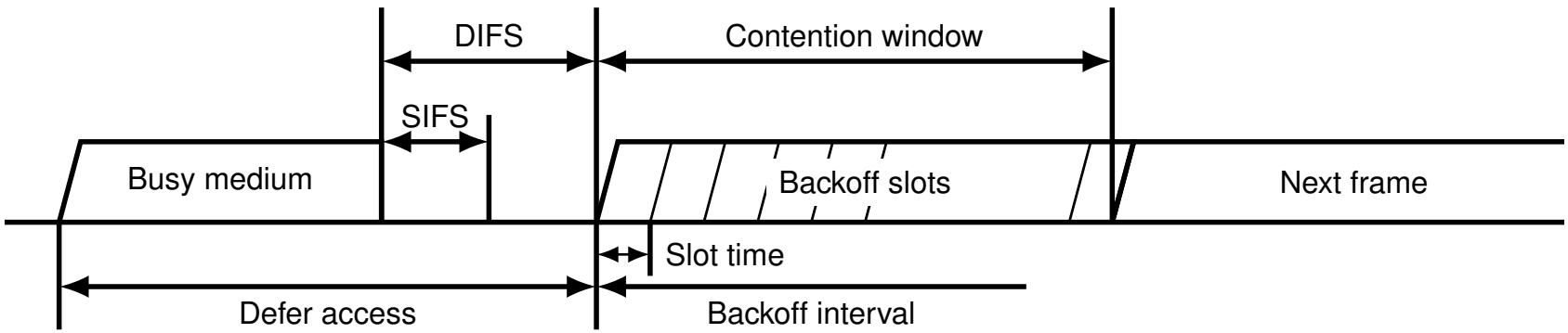


Abbildung 3: IEEE 802.11 DCF

- Medienzugriff hat durch festes c 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)

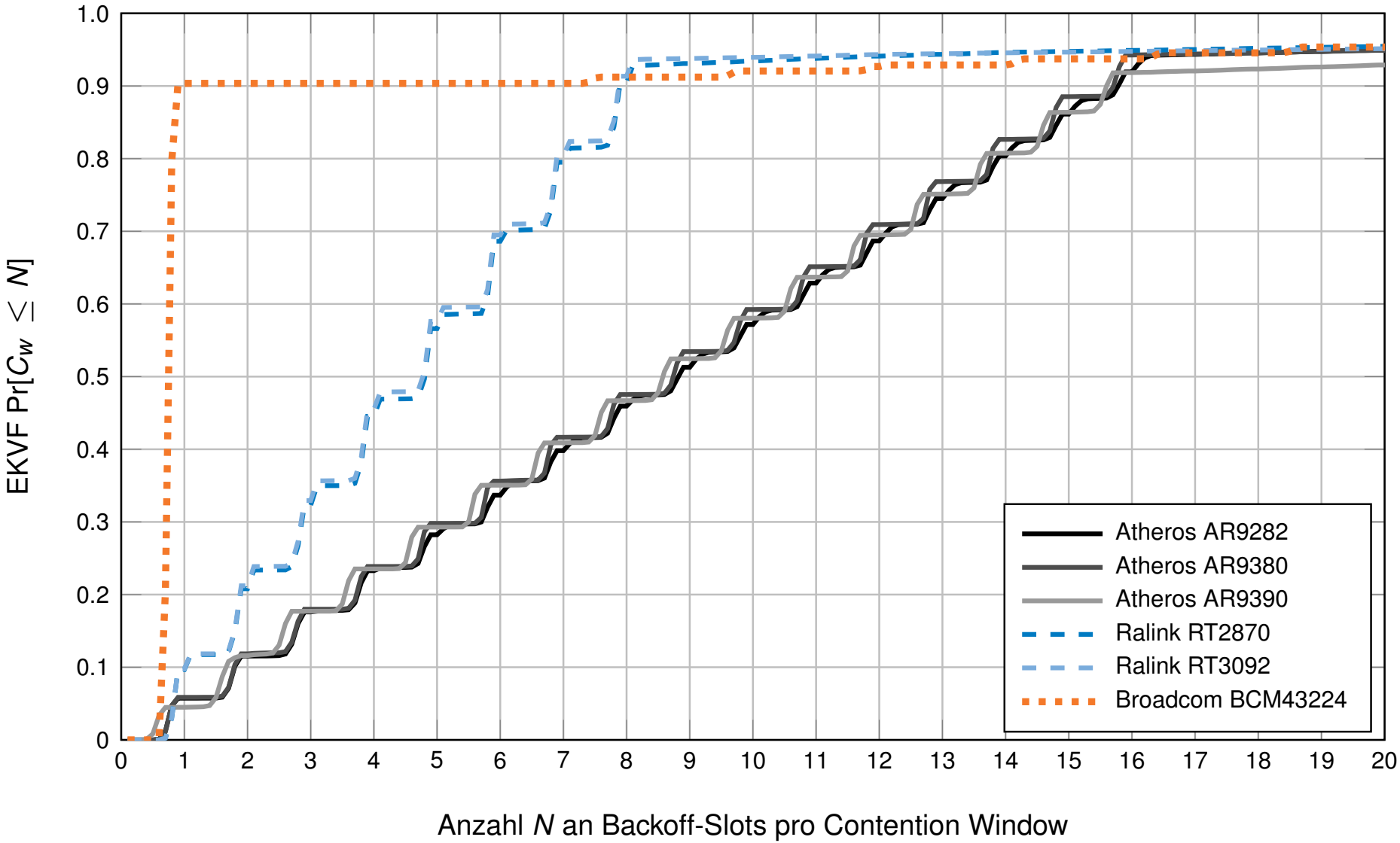
Was passiert in der Praxis? Beispiel anhand handelsüblicher Hardware im Monitor Mode[3]:

- 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 (IEEE 802.11 macht kein Media Sensing während eine Übertragung läuft).
- Das Contention Window sollte aber $\{0,1,2,\ldots,15\}$ betragen und Backoff-Slots unabhängig und gleichverteilt daraus gezogen werden.
  $\Rightarrow$ Ein zu sendender Frame sollte (bei freiem Medium) im Mittel um 7,5 Slotzeiten verzögert werden.

Was wir nun genau tun:

- Wir können die Verzögerung zwischen aufeinander folgenden Frames einer Station (Interframe Times) mittels einer zweiten Station relativ genau messen.
- Aus den gemessenen Zeiten erstellen wir eine (empirische) kummulative Verteilungsfunktion (KVF).
- Diese EKVF gibt $\Pr[X \leq N]$ an, also die Wahrscheinlichkeit, dass die Anzahl der gewarteten Slotzeiten (die Größe des Contention Windows) kleiner oder gleich N Slotzeiten ist.
- Diese sollte einer Treppenfunktion zwischen 0 und 15 mit äquidistanten Inkrementen um jeweils 1 Slotzeit folgen.

[3] Monitor Mode bezeichnet einen Operationsmodus von IEEE 802.11 Hardware, in dem die Netzwerkkarten alle eingehenden Frames vollständig unverarbeitet zugänglich machen, unabhängig davon, ob es sich um Daten-, Management- oder Control-Frames handelt und unabhängig davon, ob das Frame überhaupt an die jeweilige Station adressiert war. Umgekehrt können in diesem Modus auch beliebige Link-Layer Frames „von Hand gebaut“ und unverändert verschickt werden.



EKVF der zeitl. Abstände zwischen Frames einer Station gemessen in Vielfachen von Slotzeiten.

Fallbeispiel: IEEE 802.11 DCF (Distributed Coordination Function)

Was bedeutet das nun?

- Während eine dieser Broadcom-Karten sendet, haben alle anderen Sendepause.
- Hält sich ein Gerät nicht an die Vorgaben, kann es sich beim Senden auf Kosten anderer „Vorteile“ verschaffen, da kleinere Contention Phases
  - die Wahrscheinlichkeit erhöhen, die Contention Phase zu gewinnen und
  - natürlich die Idle-Time des Medium reduzieren.
- Gerade Letzteres (Idle-Times) ist bei IEEE 802.11 der limitierende Faktor, da die Zeit zwischen Frames im Verhältnis zur Serialisierungszeit sehr hoch ist.

Anmerkungen:

- Das Beispiel diskutiert IEEE 802.11 Hardware im Monitor Mode und ist daher für den (praxisrelevanten) Infrastructure Mode nur wenig aussagekräftig.
- In einer Masterarbeit haben wir aber kürzlich gezeigt, dass es dort aber nicht viel besser aussieht...
- Das Verhalten hängt nicht nur von der Hardware/Firmware, sondern auch vom Treiber ab.
- Nein, wir werden nicht von Atheros/Qualcomm bezahlt. :)
- Das dazu passende Paper gibts auf `grnvs.net`.

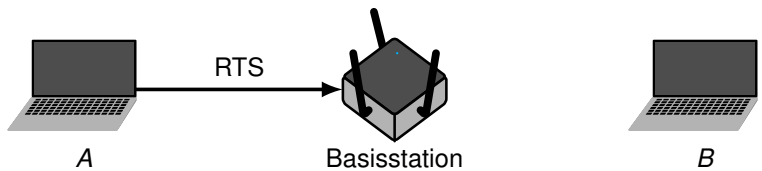
CSMA, CSMA/CD, CSMA/CA

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

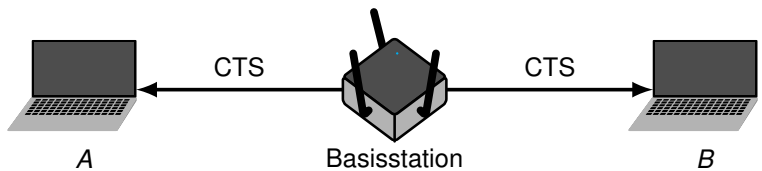
- Übertragungen werden i.d.R. von einer Basisstation gesteuert
- Bevor ein Knoten eine Nachricht überträgt, wird ein RTS an die Basisstation geschickt
- Nur wenn die Basisstation mit einem CTS antwortet, darf die Übertragung beginnen

Beispiel:

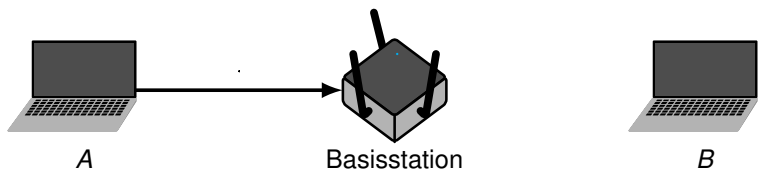
1. *A* sendet RTS, welches von *B* aufgrund der Distanz nicht empfangen wird.



2. Basistation antwortet mit CTS, welches von *A* und *B* empfangen wird.



3. *A* darf senden, *B* muss eine im CTS definierte Zeitspanne abwarten, bevor überhaupt ein RTS gesendet werden darf.



CSMA, CSMA/CD, CSMA/CA

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

Vorteile:

- Kollisionen mit Hidden Stations werden vermieden, aber nicht gänzlich verhindert.
- Insgesamt weniger Kollisionen, auch ohne Hidden Stations.

Nachteile:

- Es können noch immer Kollisionen auftreten, z.B. wenn *B* das CTS nicht empfängt.
- RTS/CTS nimmt vorab Zeit in Anspruch, was die maximal erzielbare Datenrate reduziert.

Anmerkungen:

- RTS/CTS ist Bestandteil des sog. Virtual Carrier Sensing, da mit dem CTS das Medium für eine bestimmte Zeitspanne für eine Übertragung reserviert wird.
- Um die Verlustwahrscheinlichkeit von RTS/CTS-Nachrichten zu minimieren, werden diese mit der robustesten Kodierung übertragen, was i.d.R. der niedrigsten unterstützten Datenrate entspricht. Im Gegenzug sind RTS/CTS-Nachrichten sehr klein.
- Es ist streng genommen für RTS/CTS nicht notwendig, dass ein Netzwerk durch eine Basisstation kontrolliert wird. Es funktioniert auch im ad-hoc Modus [4] oder (mit Einschränkungen) in Mesh-Netzwerken.
- Alle Geräte, unabhängig davon ob sie zum selben Service Set[5] gehören oder nicht, sollten CTS-Nachrichten verarbeiten.

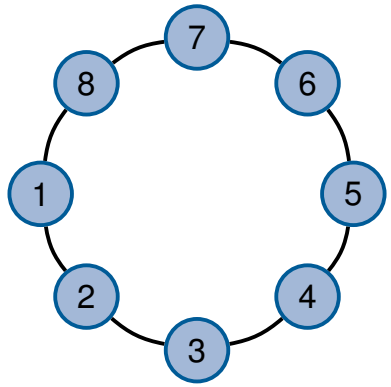
[4] Bezeichnet eine Gruppe IEEE 802.11-fähiger Geräte, welche ohne Basistation direkt miteinander kommunizieren

[5] Bezeichnung für eine Gruppe miteinander kommunizierender IEEE 802.11-fähiger Geräte

Token Passing

Idee: Kollisionsfreie Übertragung durch Weitergabe eines Tokens

- Stationen werden zu einem logischen Ring zusammen geschaltet
- 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 Kilometern. 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
- CSMA/CA (Kollisionsvermeidung)
  IEEE 802.11 WLAN

Geregelter Zugriff:

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

Kapitel 2: Sicherungsschicht

Darstellung von Netzwerken als Graphen

Verbindungscharakterisierung, Mehrfachzugriff, Medienzugriffskontrolle

Rahmenbildung, Adressierung und Fehlererkennung

- Erkennung von Rahmengrenzen und Codetransparenz
- Adressierung und Fehlererkennung

Verbindung auf Schicht 1 und 2

Zusammenfassung

Literaturangaben

Motivation

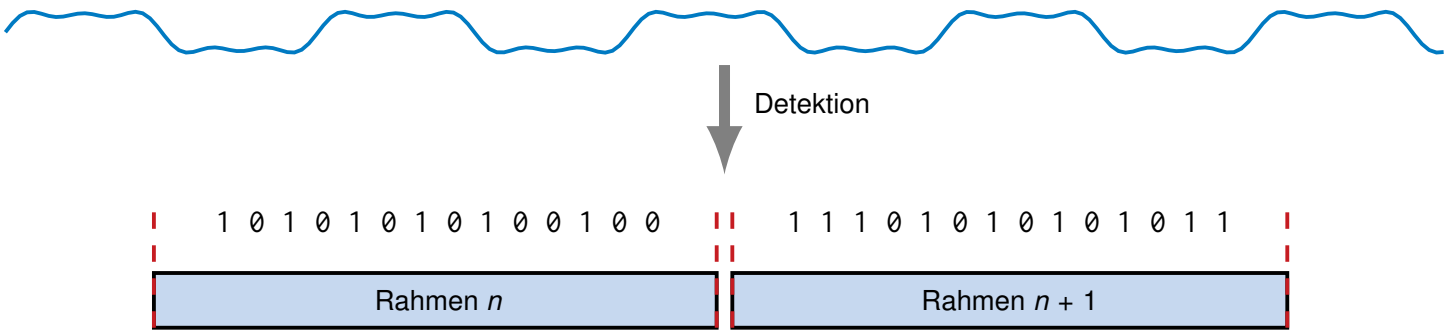
Bislang haben wir nur von Nachrichten gesprochen, ohne uns Gedanken über deren Format zu machen. Aus Sicht der physikalischen Schicht ist eine Nachricht lediglich eine Folge von Bits. Für eine Betrachtung der Sicherungsschicht reicht diese Vorstellung aber nicht mehr aus.

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).

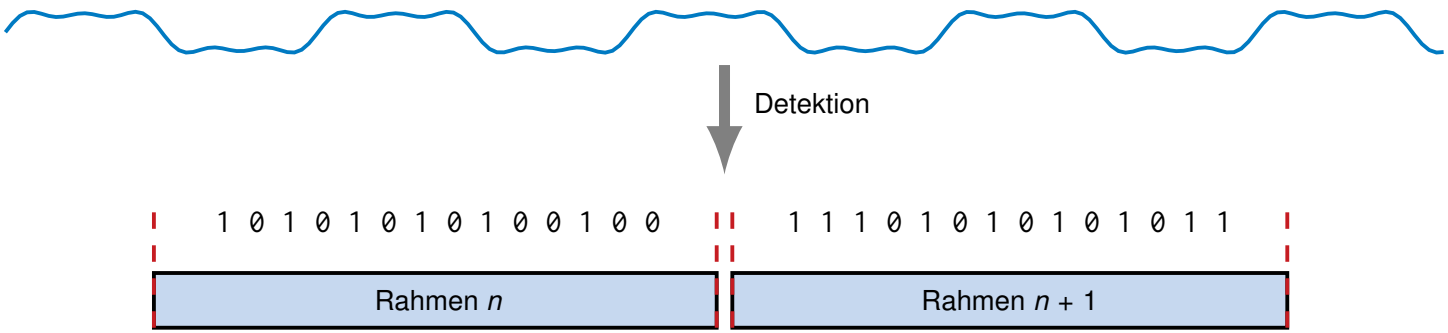
Erkennung von Rahmengrenzen und Codetransparenz



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 und Codetransparenz



Wie kann der Empfänger Rahmen erkennen, insbesondere wenn

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

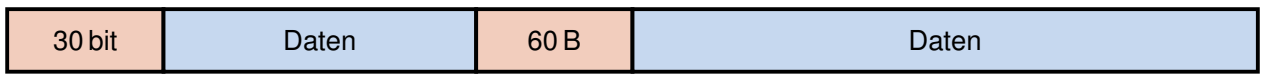
Es gibt viele Möglichkeiten:

- 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

- 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

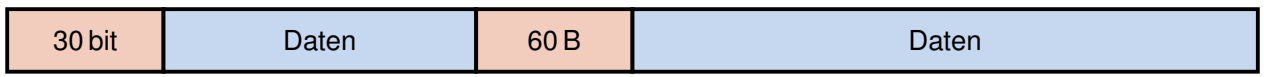


Wie kann der Beginn eines Rahmens erkannt werden?

Erkennung von Rahmengrenzen und Codetransparenz

Längenangabe der Nutzdaten

- 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, s. Kapitel 1)

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

Erkennung von Rahmengrenzen und Codetransparenz

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
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Erkennung von Rahmengrenzen und Codetransparenz

Was ist, wenn Steuerzeichen zufällig in den Nutzdaten vorkommen?

1. Im Fall des 4B5B-Codes 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

Meist wird automatisch für Codetransparenz gesorgt, so dass sich der Benutzer nicht darum kümmern muss. Das trifft nicht auf Programmiersprachen zu:

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

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

Weitere Beispiele:

- Bash (`Ctrl+C`)
- Texteditoren (Emacs)

Begrenzungsfelder und Bit-Stopfen

- 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

- 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 in Nutzdaten nach fünf aufeinanderfolgenden `1`-en eine `0` ein

Eingabe:		`1100101111110111111`	
Ausgabe:	`01111110`	`110010111110101111101`	`01111110`

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

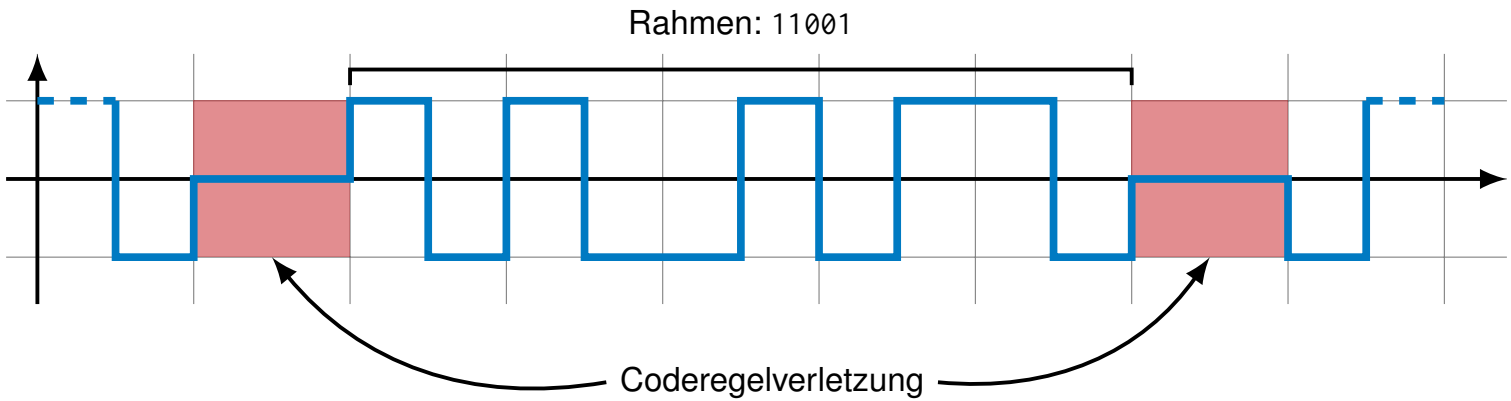
Coderegelverletzung

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

Beispiel: Manchester-Code



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.

IEEE 802.3z (Gigabit Ethernet over Fiber): 1000 Mbit/s

- Als Leitungscode wird NRZ in Kombination mit dem 8B10B-Code verwendet.
- Start und Ende von Rahmen werden durch Steuerzeichen des 8B10B-Codes markiert.
- IEEE 802.3ab (Gigabit Ethernet over Copper) verwendet andere Leitungscodes, da die Dämpfung andernfalls zu groß wäre.

Zusätzlich wird bei all diesen Beispielen jedem Rahmen noch eine Präambel (s. Kapitel 1) vorangestellt. Diese dient allerdings nur der Taktsynchronisierung zwischen Sender und Empfänger.

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.

Anmerkung: Alle folgenden Konzepte werden anhand der IEEE 802-Standards erklärt. Die wesentlichen Punkte sind mit kleinen Modifikationen auf andere Verfahren übertragbar.

Adressierung in Direktverbindungsnetzen:

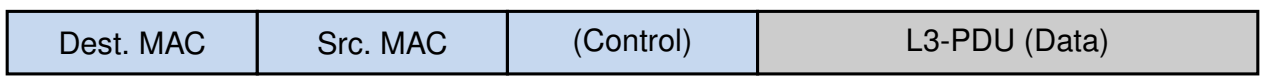
- sind angeschlossene Knoten direkt erreichbar,
- es findet also keine Vermittlung (engl. Routing) zwischen Knoten statt.

Anforderungen an Adressen auf Schicht 2:

- Eindeutige Identifizierung der Knoten innerhalb des Direktverbindungsnetzes.
- Zumeist existiert eine Broadcast-Adresse, welche alle Knoten im Direktverbindungsnetz anspricht.
- Zusätzlich kann es Multicast-Adressen[6] geben, die bestimmte Gruppen von Knoten ansprechen.

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

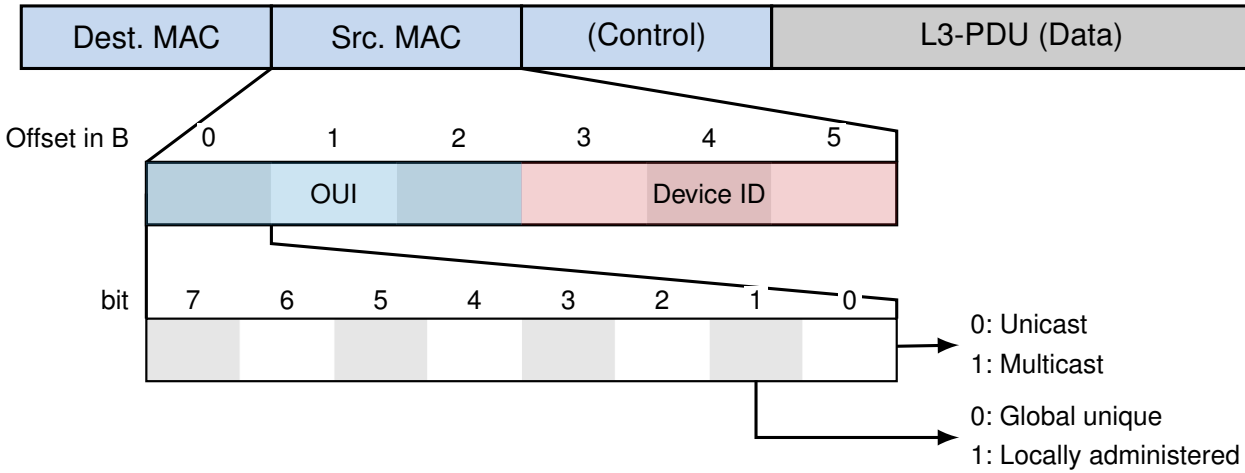
Beispiel:



[6] Mutlicast-Adressen auf Schicht 2 werden häufig wie Broadcasts behandelt. Speziell im Einsatz mit IPv6 in geswitchten Netzwerken sind sie aber von großer Bedeutung.

Adressierung und Fehlererkennung

MAC-Adressen aller IEEE 802-Standards (z.B. Ethernet, WLAN, Bluetooth) haben den folgenden Aufbau:



- Netzwerkkarten besitzen eine ab Werk im ROM (Read Only Memory) hinterlegte MAC-Adresse
- Auftrennung in OUI und Device ID ermöglicht es den Herstellern von Netzwerkkarten, eindeutige MAC-Adressen zu vergeben
- Vergeben werden die OUIs von der IANA (Internet Assigned Numbers Authority) [1]
- Der Hersteller einer Netzwerkkarte kann folglich anhand deren MAC-Adresse identifiziert werden (z. B. `7c:6d:62` ≙ Apple)
- Als Broadcast-Adresse ist `ff:ff:ff:ff:ff:ff` („all ones“) definiert
- Ob es sich bei einer Adresse um eine Unicast- oder Multicast-Adresse handelt, bestimmt das lowest order Bit des ersten Oktetts.

Anmerkung: Für bestimmte Anwendungen ist es sinnvoll, auf die herstellerübergreifende Eindeutigkeit zu verzichten, z.B. bei virtualisierten Netzwerkadaptern. Hierfür sind die sog. lokal-administrierten Adressen (zweites Bit des ersten Oktetts) vorgesehen.

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, engl. Checksums):

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

Adressierung und Fehlererkennung

Cyclic Redundancy Check (CRC) [3]

Im Gegensatz zu fehlerkorrigierenden Codes (Kanalcodes, Kapitel 1), handelt sich sich bei CRC um eine Familie fehlererkennender Codes. Mit ihrem Einsatz werden folgende Ziele verfolgt:

- Eine grosse Anzahl von Fehlern (Einbit-, Mehrbit-, Burstfehler) sollen erkannt werden.
- Die zugefügte Redundanz soll gering sein.
- Fehler sollen lediglich erkannt aber nicht korrigiert werden können.

Adressierung und Fehlererkennung

Cyclic Redundancy Check (CRC) [3]

Im Gegensatz zu fehlerkorrigierenden Codes (Kanalcodes, Kapitel 1), handelt sich sich bei CRC um eine Familie fehlererkennender Codes. Mit ihrem Einsatz werden folgende Ziele verfolgt:

- Eine grosse Anzahl von Fehlern (Einbit-, Mehrbit-, Burstfehler) sollen erkannt werden.
- Die zugefügte Redundanz soll gering sein.
- Fehler sollen lediglich erkannt aber nicht korrigiert werden können.

Grundlagen:

- Ein Datenwort der Länge n bit lässt sich darstellen als Polynom

$$a(x) = \sum_{i=0}^{n-1} a_i x^i \text{ mit } a_i \in \mathbb{F}_2 \text{ mit } \mathbb{F}_2 = \{0,1\}.$$

- Alle Datenworte der Länge genau n bit bilden die Menge

$$F_q[x] = \left\{ a \;\middle|\; a(x) = \sum_{i=0}^{n-1} a_i x^i,\ a_i \in \mathbb{F}_2 \right\}.$$

- Zusammen mit passend definierter Addition und Multiplikation entsteht ein sog. endlicher Körper (finite extension field) $\langle F_q[x], +, \cdot \rangle$ mit $q = 2^n$ Elementen, auf dem die üblichen Regeln zur Addition und Multiplikation gelten.

Was heißt „passend definiert“?

Summe: Für die Summe zweier beliebiger $a,b \in F_q[x]$ erhalten wir

$$c(x) = a(x) + b(x) = \sum_{i=0}^{n-1} (a_i + b_i)\, x^i,$$

wobei für die Summe der Koeffizienten die Addition des GF(2) gilt[7], d. h. die Summe zweier Datenwörter entspricht einer bitweisen XOR-Verknüpfung.

[7] Galois Feld, endlicher Körper (s. Diskrete Strukturen)

Was heißt „passend definiert“?

Summe: Für die Summe zweier beliebiger $a,b \in F_q[x]$ erhalten wir

$$c(x) = a(x) + b(x) = \sum_{i=0}^{n-1} (a_i + b_i)\, x^i,$$

wobei für die Summe der Koeffizienten die Addition des GF(2) gilt[7], d. h. die Summe zweier Datenwörter entspricht einer bitweisen XOR-Verknüpfung.

Produkt: Das Produkt ist komplizierter, da für $d(x) = a(x) \cdot b(x)$ der Grad von $d(x)$ im Allgemeinen größer als $n-1$ ist und damit $d(x) \notin F_q[x]$. Daher wählt man ein Reduktionspolynom r(x) mit $\text{grad}(r(x)) = n$ und definiert das Produkt von $a,b \in F_q[x]$ als

$$d(x) = (a(x) \cdot b(x)) \bmod r(x).$$

- Dies entspricht einer normalen Polynommultiplikation (wobei die Addition einer XOR-Verknüpfung entspricht) mit anschließender Modulo-Operation über $r(x)$.
- Die Modulo-Operation entspricht einer normalen Polynomdivision mit dem Divisionsrest als Ergebnis.
- Dies sorgt dafür, dass $\text{grad}(d(x)) < n$ ist.

[7] Galois Feld, endlicher Körper (s. Diskrete Strukturen)

Beispiel: $F_4[x] = \{0,1,x,x+1\}$ mit $r(x) = x^2 + x + 1$

Ist $r(x)$ irreduzibel?

Beispiel: $F_4[x] = \{0,1,x,x+1\}$ mit $r(x) = x^2 + x + 1$

Ist $r(x)$ irreduzibel?

$\cdot$	0	1	x	$x+1$
0	0			
1	0	1		
x	0	x	x^2	
$x+1$	0	$x+1$	x^2+x	„$x^2 + 2x + 1$“ $= x^2 + 1$

Die Addition über $F_4[x]$ entspricht der Addition über $\mathbb{F}_2$ zwischen Monomen gleichen Grades.

$\Rightarrow$ Ja, $r(x)$ ist irreduzibel, da es sich nicht als Produkt $a \cdot b$ mit $a,b \in F_4[x]$ darstellen lässt.

Beispiel: $F_4[x] = \{0,1,x,x+1\}$ mit $r(x) = x^2 + x + 1$

Ist $r(x)$ irreduzibel?

$\cdot$	0	1	x	$x+1$
0	0			
1	0	1		
x	0	x	x^2	
$x+1$	0	$x+1$	x^2+x	„x^2+2x+1“ $= x^2+1$

Die Addition über $F_4[x]$ entspricht der Addition über $\mathbb{F}_2$ zwischen Monomen gleichen Grades.

$\Rightarrow$ Ja, $r(x)$ ist irreduzibel, da es sich nicht als Produkt $a \cdot b$ mit $a,b \in F_4[x]$ darstellen lässt.

Für die Multiplikation benötigen wir $r(x)$, um die farbigen Ergebnisse in obiger Tabelle zu reduzieren:

$+$	0	1	x	$x+1$
0	0			
1	1	0		
x	x	$x+1$	0	
$x+1$	$x+1$	x	1	0

$\cdot$	0	1	x	$x+1$
0	0			
1	0	1		
x	0	x	$x+1$	
$x+1$	0	$x+1$	1	x

$$x^2 : (x^2 + x + 1) = 1, \text{ Rest: } x + 1$$

$$(x^2 + x) : (x^2 + x + 1) = 1, \text{ Rest: } 1$$

$$(x^2 + 1) : (x^2 + x + 1) = 1, \text{ Rest: } x$$

Anmerkungen:

- Wählt man für $r(x)$ ein irreduzibles Polynom, d. h. heißt $r(x)$ kann nicht als Produkt zweier $a,b \in F_q[x]$ dargestellt werden, so erhält man einen endlichen Körper mit $q = 2^n$ Elementen.
- Für CRC wählt man häufig $r(x) = p(x)(x+1)$ mit $p \in F_q[x]$ als Reduktionspolynom von Grad $n-1$:
  - Sowohl $p(x)$ als auch $x+1$ sind Elemente von $F_q[x]$ und $r(x)$ ist als Produkt zweier solcher Elemente offensichtlich nicht irreduzibel.
  - Mit dieser Wahl von $r(x)$ ist $\langle F_q[x], +, \cdot \rangle$ kein endlicher Körper.
  - Diese Wahl von $r(x)$ ermöglicht es jedoch, alle ungeradzahligen Fehler zu erkennen.
- Die Wahl von $r(x)$ bestimmt also nicht nur die Länge der Prüfsumme, sondern auch maßgeblich die Fehlererkennungseigenschaften.

Zurück zu CRC

- CRC berechnet zu einem gegebenen Datenblock (z. B. L2-PDU) eine Checksumme fester Länge.
- Codewörter sind Polynome $a \in F_q[x]$.
- Der Grad n des Reduktionspolynoms $r(x)$ bestimmt
  - den maximalen Grad $n-1$ aller möglichen Codewörter $a \in F_q[x]$ sowie
  - welche Arten von Bitfehlern (Einbit-, Mehrbit-, Burstfehler) erkannt werden können.
- Ethernet verwendet CRC32 mit dem Reduktionspolynom

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

Wie funktioniert CRC?

Angenommen wir haben ein Reduktionspolynom $r(x)$ des Grads n und eine Nachricht $m(x)$ des Grads k (d. h. die Nachricht besteht aus $k + 1$ bit), die mittels CRC gesichert werden soll:

1. Hänge n Nullen an $m(x)$ an: $m'(x) = m(x) \cdot x^n$.
2. Bestimme den Divisionsrest $c(x) = m'(x) \bmod r(x)$, welcher der Checksumme entspricht.
3. Die zu sendende Nachricht besteht aus der Summe $s(x) = m'(x) + c(x)$.

Wie funktioniert CRC?

Angenommen wir haben ein Reduktionspolynom $r(x)$ des Grads n und eine Nachricht $m(x)$ des Grads k (d. h. die Nachricht besteht aus $k + 1$ bit), die mittels CRC gesichert werden soll:

1. Hänge n Nullen an $m(x)$ an: $m'(x) = m(x) \cdot x^n$.
2. Bestimme den Divisionsrest $c(x) = m'(x) \bmod r(x)$, welcher der Checksumme entspricht.
3. Die zu sendende Nachricht besteht aus der Summe $s(x) = m'(x) + c(x)$.

Der Empfänger prüft die eingehende Nachricht $s'(x) = s(x) + e(x)$, welche möglicherweise einen Übertragungsfehler $e(x) \neq 0$ enthält:

1. Er bestimmt den Divionsrest

$$c'(x) = s'(x) \bmod r(x) = (s(x) + e(x)) \bmod r(x).$$

2. Ist $c'(x) = 0$, so ist mit hoher Wahrscheinlichkeit kein Übertragungsfehler aufgetreten. Ist $c'(x) \neq 0$, so ist sicher ein Fehler aufgetreten.

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

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

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

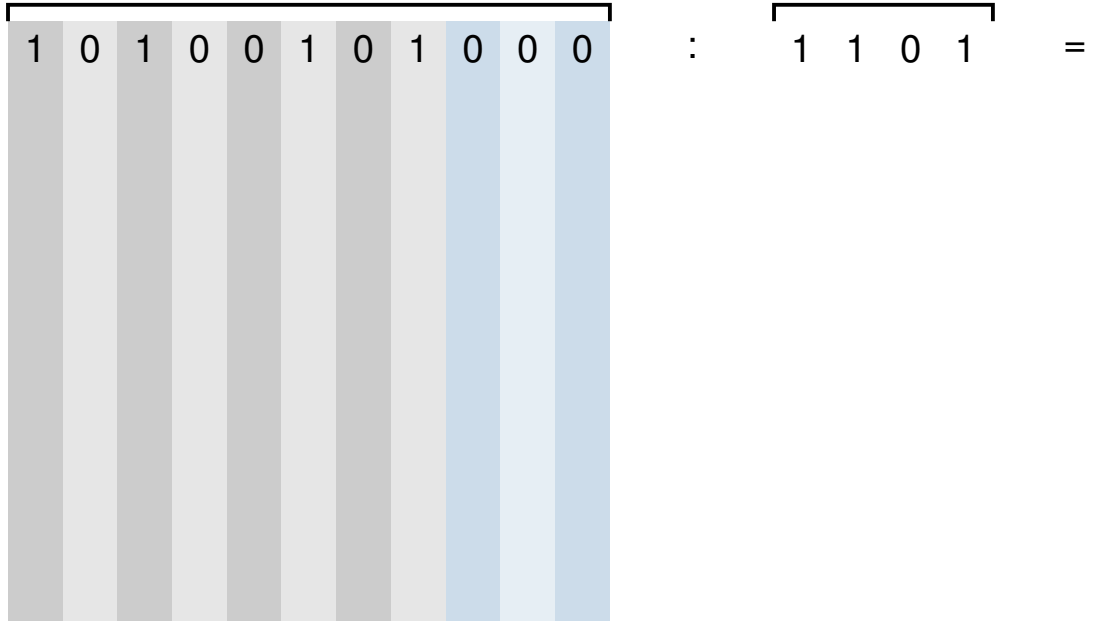
1. Koeffizienten bestimmen: $r(x) \mathrel{\hat{=}} 1101$ und $m(x) \mathrel{\hat{=}} 10100101$
2. $\text{grad}(r(x)) = 3 \Rightarrow$ Daten mit x^3 multiplizieren. Dies entspricht dem „Anhängen“ von 3 Nullen: $m'(x) = m(x) \cdot x^3 \mathrel{\hat{=}} 10100101\mathbf{000}$

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

1. Koeffizienten bestimmen: $r(x) \mathrel{\hat{=}} 1101$ und $m(x) \mathrel{\hat{=}} 10100101$
2. $\text{grad}(r(x)) = 3 \Rightarrow$ Daten mit x^3 multiplizieren. Dies entspricht dem „Anhängen“ von 3 Nullen: $m'(x) = m(x) \cdot x^3 \mathrel{\hat{=}} 10100101\mathbf{000}$
3. Polynomdivision $m'(x)/r(x)$ ausführen und den Rest (Checksumme) $c(x)$ bestimmen.

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

$m'(x)$: $r(x)$ =



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

$m'(x)$: $r(x)$ = 1

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

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

$m'(x)$: $r(x)$ = 1 1

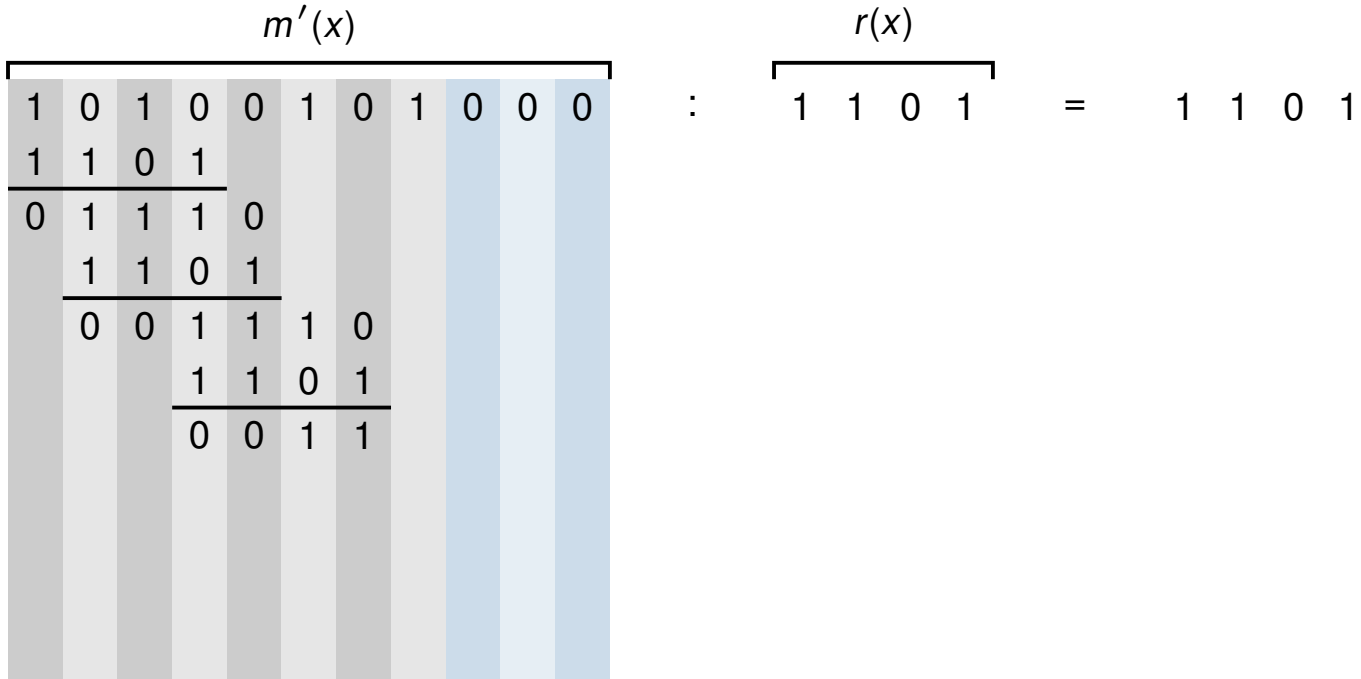
1	0	1	0	0	1	0	1	0	0	0
1	1	0	1							
0	1	1	1	0						
	1	1	0	1						
	0	0	1	1						

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

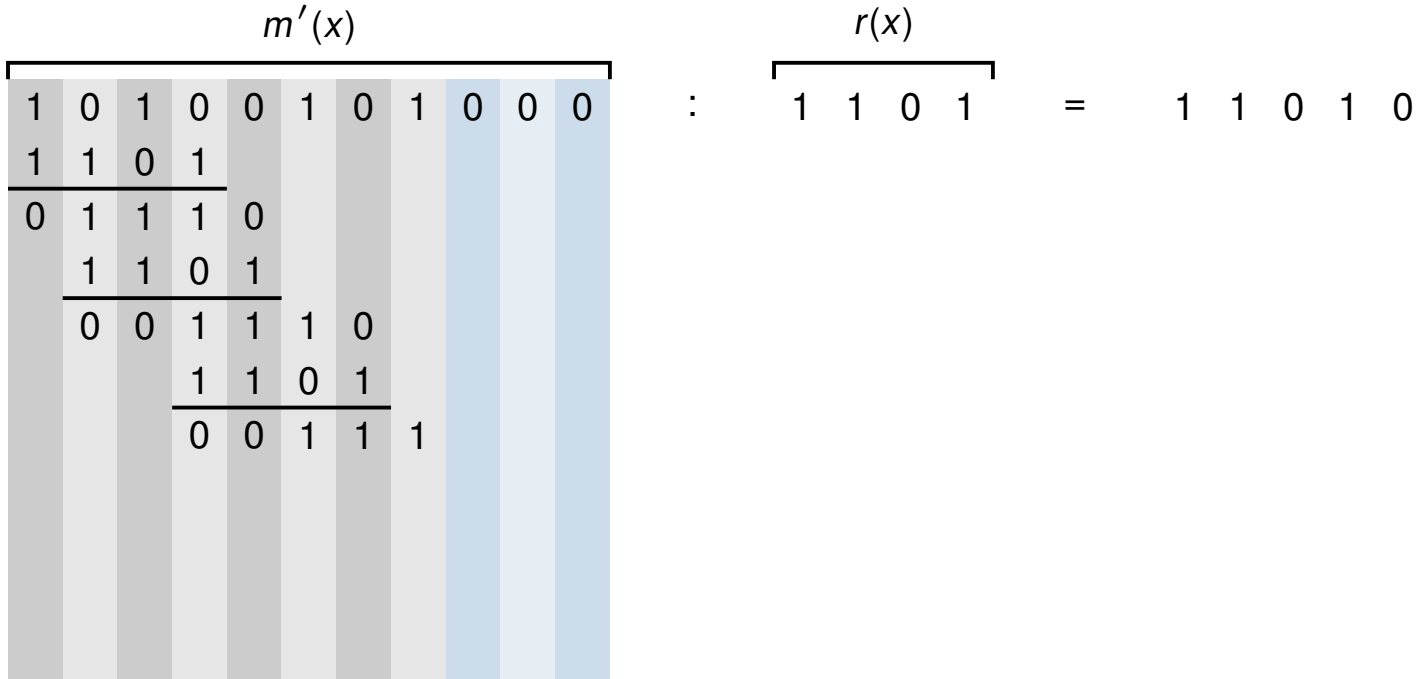
$m'(x)$: $r(x)$ = 1 1 0

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					

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



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



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

$m'(x)$: $r(x)$ = $q(x)$

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																			
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														

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

$m'(x)$: $r(x)$ = Ergebnis

1	0	1	0	0	1	0	1	0	0	0	:	1	1	0	1	=	1	1	0	1	0	1	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														

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

$m'(x)$: $r(x)$ = $c(x)$

1	0	1	0	0	1	0	1	0	0	0	:	1	1	0	1	=	1	1	0	1	0	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	0																
					1	1	0	1																
					0	0	1	1	0	0														
							1	1	0	1														
							0	0	0	1														

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

1. Koeffizienten bestimmen: $r(x) \mathrel{\hat{=}} 1101$ und $m(x) \mathrel{\hat{=}} 10100101$
2. $\text{grad}(r(x)) = 3 \Rightarrow$ Daten mit x^3 multiplizieren. Dies entspricht dem „Anhängen“ von 3 Nullen: $m'(x) = m(x) \cdot x^3 \mathrel{\hat{=}} 10100101\mathbf{000}$
3. Polynomdivision $m'(x)/r(x)$ ausführen und den Rest (Checksumme) $c(x)$ bestimmen.
4. Die zu sendende Nachricht ist $s(x) = m'(x) + c(x)$. Die Addition reduziert sich auf ein XOR, da wir auf GF(2) arbeiten.

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

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

1. Koeffizienten bestimmen: $r(x) \mathrel{\hat{=}} 1101$ und $m(x) \mathrel{\hat{=}} 10100101$
2. $\text{grad}(r(x)) = 3 \Rightarrow$ Daten mit x^3 multiplizieren. Dies entspricht dem „Anhängen“ von 3 Nullen: $m'(x) = m(x) \cdot x^3 \mathrel{\hat{=}} 10100101\mathbf{000}$
3. Polynomdivision $m'(x)/r(x)$ ausführen und den Rest (Checksumme) $c(x)$ bestimmen.
4. Die zu sendende Nachricht ist $s(x) = m'(x) + c(x)$. Die Addition reduziert sich auf ein XOR, da wir auf GF(2) arbeiten.

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

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

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

Welche Fehler erkennt CRC?

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

- Alle 1 bit-Fehler
- Isolierte 2 bit-Fehler, d. h. Fehler an den Bitstellen i und j wobei $i > j$ so dass $i - j > n$
- Einige Burst-Fehler, die länger sind als n

Abhängig von der konkreten Wahl des Reduktionspolynoms können auch entweder

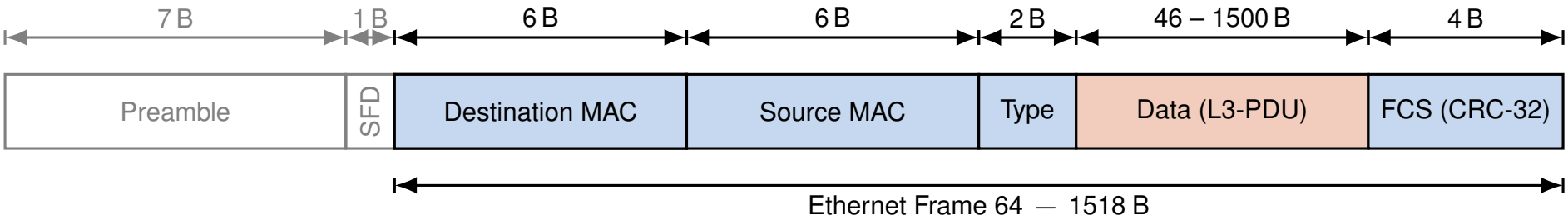
- alle Burst-Fehler, deren Länge kleiner ist als n oder
- alle Fehlermuster, deren Anzahl der Fehlerbits ungerade ist

erkannt werden.

Welche Fehler erkennt CRC nicht zuverlässig oder gar nicht?

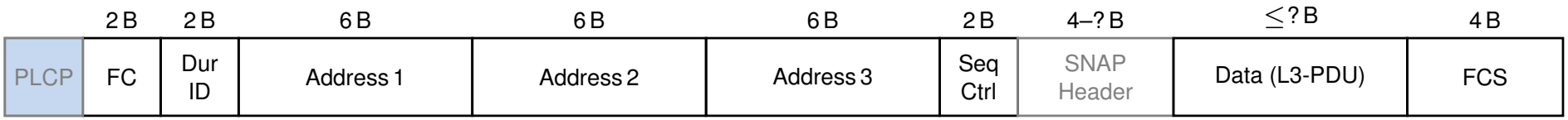
- Fehler, die länger sind als n
- Fehler, die aus mehreren Bursts bestehen
- Alle Fehler, die ein Vielfaches des Reduktionspolynoms sind

Frame vor der 4B5B-Kodierung:



- Präambel und Start Frame Delimiter (SFD) dienen der Taktsynchronisation.
- Ein Byte der Präambel wird durch das J/K-Symbol des 4B5B-Codes ersetzt (Start Frame Delimiter).
- 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` ≙ IPv4 Payload, `0x0806` ≙ ARP).
- Das Datenfeld muss (vor der Kodierung) mind. 46 B lang sein – andernfalls wird es bis zu diesem Wert gepadded.

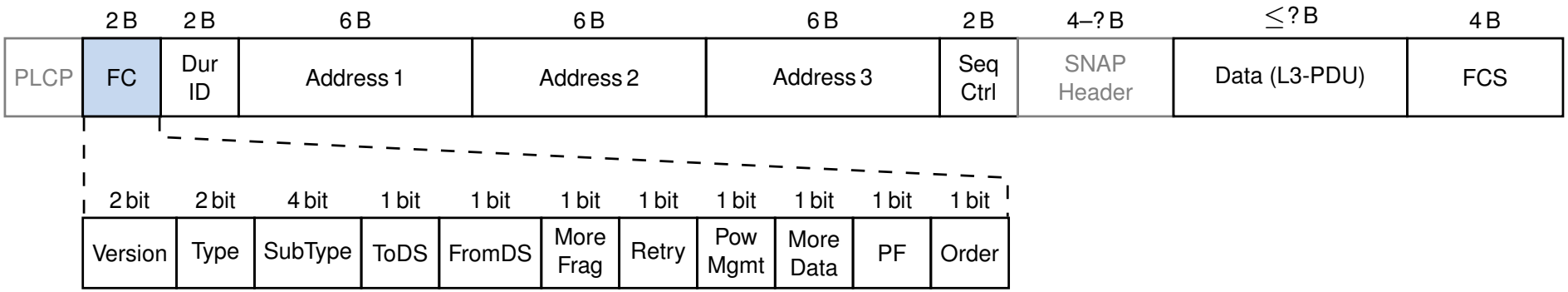
Daten-Frame im Infrastructure Mode (d. h. mit Access Point) ohne Verschlüsselung:



Physical Layer Convergence Procedure (PLCP)

- Header des Physical Layers
- Dient der Synchronisation sowie der Mitteilung von Übertragungsparametern (Datenrate, Modulation, Coderate, etc.)
- Nicht Bestandteil des L2-Headers

Daten-Frame im Infrastructure Mode (d. h. mit Access Point) ohne Verschlüsselung:



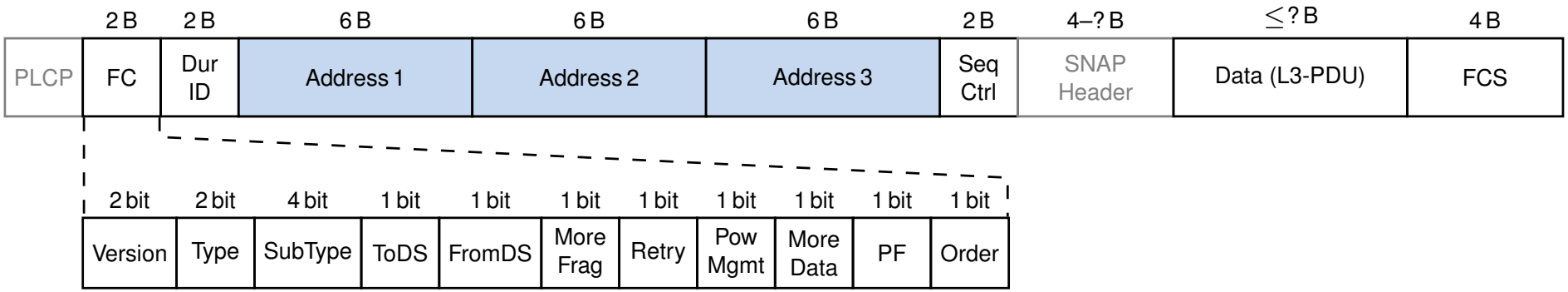
Frame Control (FC)

- Gibt den Typ des Rahmens an (Data, Management oder Control)
- Definiert, wie die im Rahmen enthaltenen Adressen zu interpretiern sind (ToDS / FromDS Bits)
- Verschieden weitere Parmeter:
  - Folgen weitere Fragmente, die zum selben Rahmen gehören?
  - Handelt es sich um einen Retransmit (Wiederholung)?
  - Liegen am Sender noch weitere Rahmen vor?
  - . . .

Adressierung und Fehlererkennung

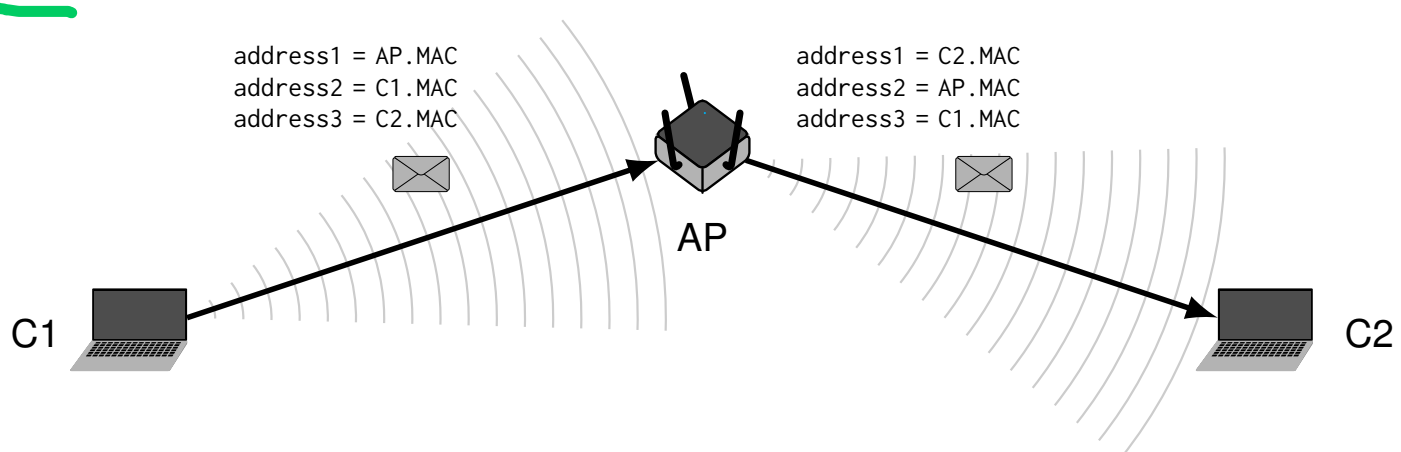
Fallbeispiel: IEEE 802.11a/g (WLAN)

Daten-Frame im Infrastructure Mode (d. h. mit Access Point) ohne Verschlüsselung:



MAC-Adressen (variable Anzahl, nachfolgend typische Nutzung der Felder)

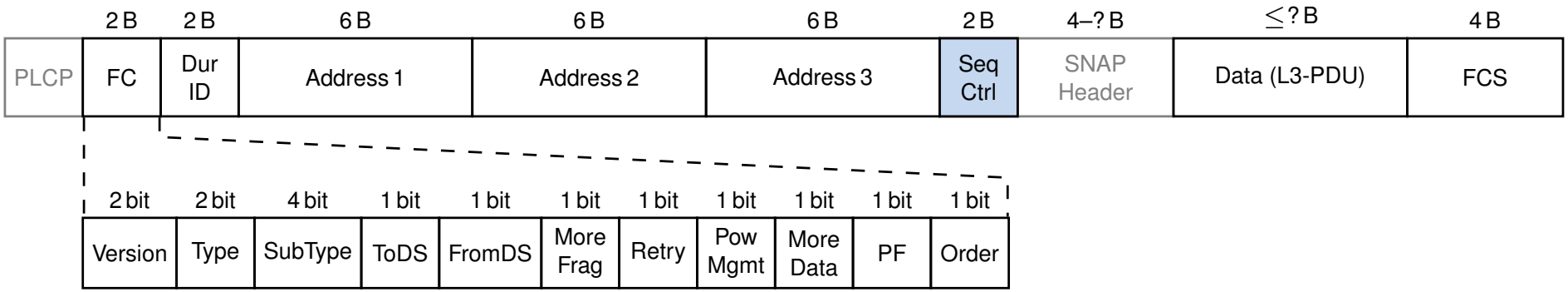
- Address 1 gibt den direkten Empfänger (Receiver Address, RA) an
- Address 2 gibt die Adresse der übertragenden Station (Transmitter Address, TA) an
- Address 3 gibt den Sender (Source Address, SA) bzw. das Ziel (Destination Address, DA) an



Adressierung und Fehlererkennung

Fallbeispiel: IEEE 802.11a/g (WLAN)

Daten-Frame im Infrastructure Mode (d. h. mit Access Point) ohne Verschlüsselung:



Sequence Control

- Sequenznummer des Rahmens
- Dient der Erkennung von fehlenden Rahmen und der Sortierung empfangener Rahmen

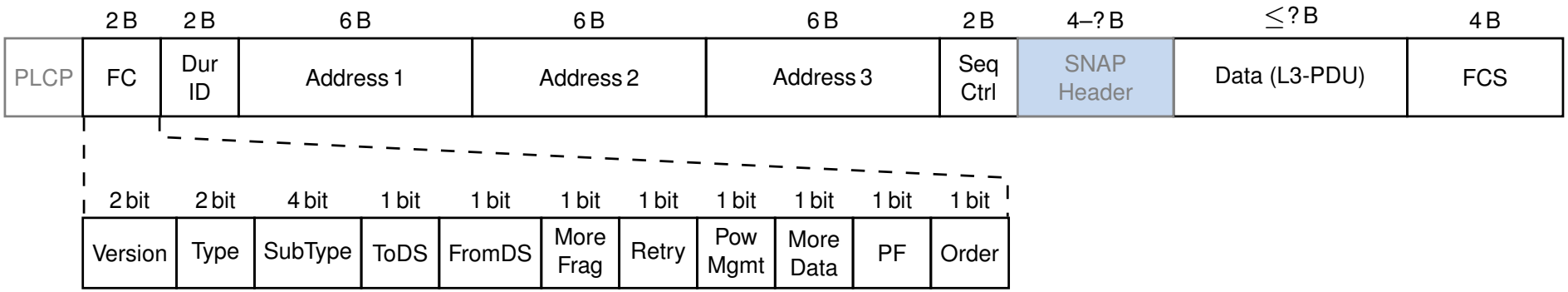
Hinweis: Im Gegensatz zu Ethernet werden im WLAN sog. Quittungsverfahren auf Schicht 2 eingesetzt, da das Übertragungsmedium selbst zu unzuverässig ist.

Dies ist kein Ersatz für Bestätigungen höhrer Schichten, sondern eine Notwendigkeit, damit Protokolle höhrer Schichten überhaupt funktionieren.

Adressierung und Fehlererkennung

Fallbeispiel: IEEE 802.11a/g (WLAN)

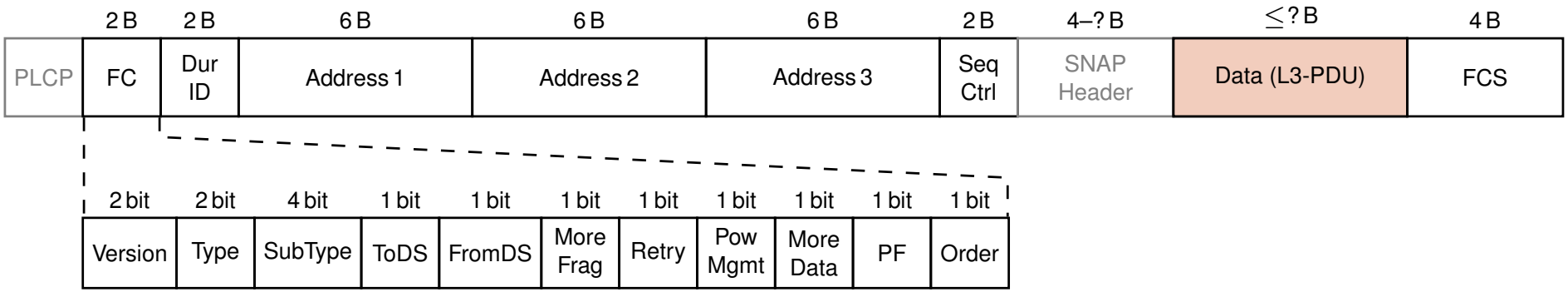
Daten-Frame im Infrastructure Mode (d. h. mit Access Point) ohne Verschlüsselung:



Subnetwork Access Protocol (SNAP)

- Header variabler Länge zur Angabe des Typs der L3-PDU
- Entfernt vergleichbar mit dem Ethertype (aber um vieles flexibler)

Daten-Frame im Infrastructure Mode (d. h. mit Access Point) ohne Verschlüsselung:



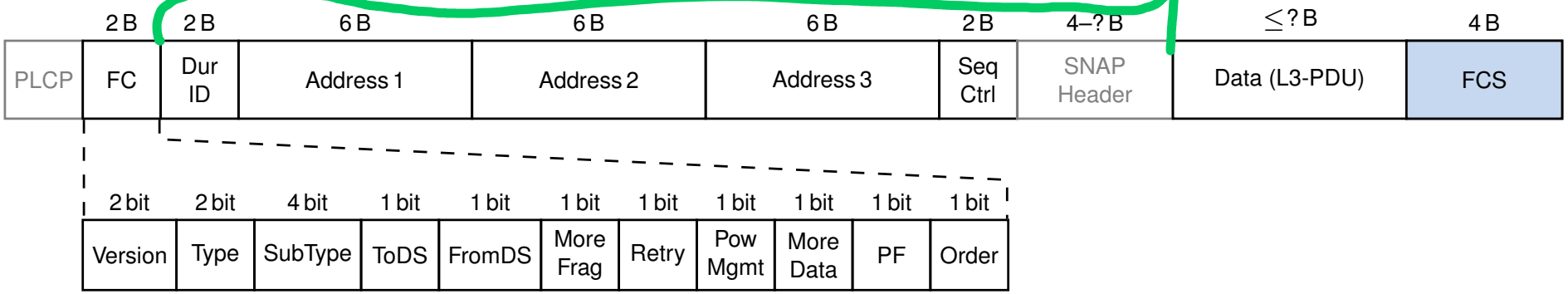
Daten (L3-PDU)

- Daten variabler Länge
- Die maximale Rahmengröße in IEEE 802.11-Netzen ist um ein Vielfaches größer als bei Ethernet
  - Der Medienzugriff benötigt hier sehr viel Zeit
  - Je kleiner die einzelnen Rahmen, desto mehr Zeit geht durch den Medianzugriff verloren
    $\Rightarrow$ Tendenz zu größeren Rahmen trotz höherer Bitfehlerwahrscheinlichkeit

Adressierung und Fehlererkennung

Fallbeispiel: IEEE 802.11a/g (WLAN)

Daten-Frame im Infrastructure Mode (d. h. mit Access Point) ohne Verschlüsselung:



Frame Check Sequence (FCS)

- CRC32-Checksumme über den gesamten L2-Rahmen (alles außer PLCP und die FCS selbst)
- Bis auf Implementierungsdetails identisch zu Ethernet

Kapitel 2: Sicherungsschicht

Darstellung von Netzwerken als Graphen

Verbindungscharakterisierung, Mehrfachzugriff, Medienzugriffskontrolle

Rahmenbildung, Adressierung und Fehlererkennung

Verbindung auf Schicht 1 und 2

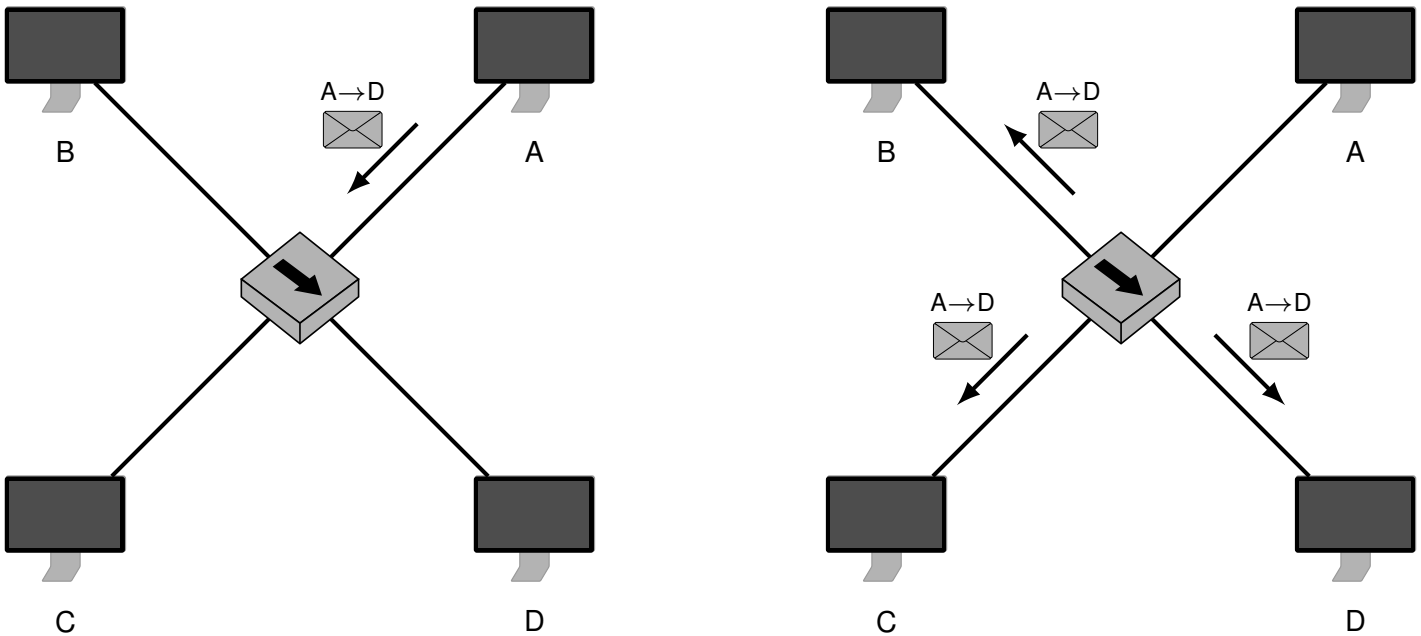
- Hubs, Bridges und Switches
- WLAN Access Points

Zusammenfassung

Literaturangaben

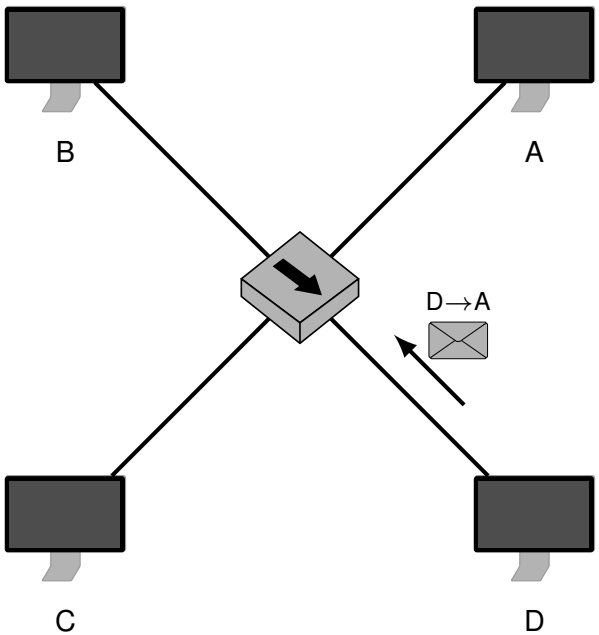
Verbindung auf Schicht 1: Hub [4]

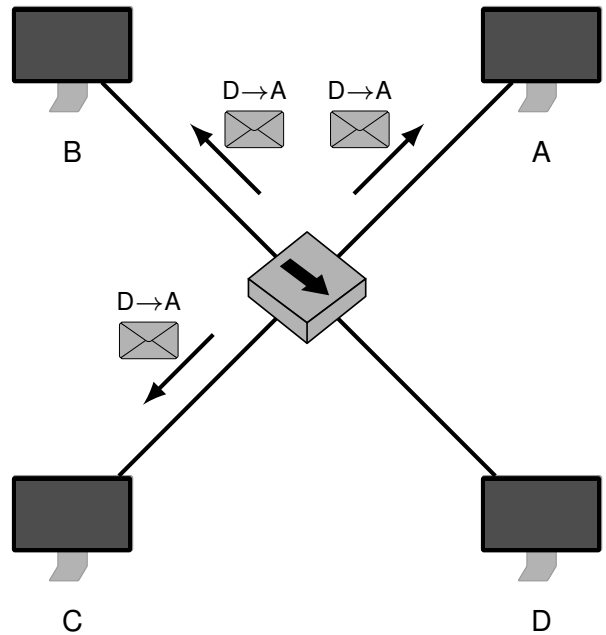
- Knoten A sendet einen Rahmen an Knoten D
- Der Hub verbindet die einzelnen Links zu einem gemeinsamen Bus
- Der Rahmen erreicht alle Knoten
- Es darf folglich zu jedem Zeitpunkt nur ein Knoten senden, andernfalls treten Kollisionen auf



Wichtig: Bis auf wenige Ausnahmen arbeitet Schicht 2 verbindungslos, d. h. es wird keine logische Verbindung zwischen den Kommunikationspartnern aufgebaut.

- 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:

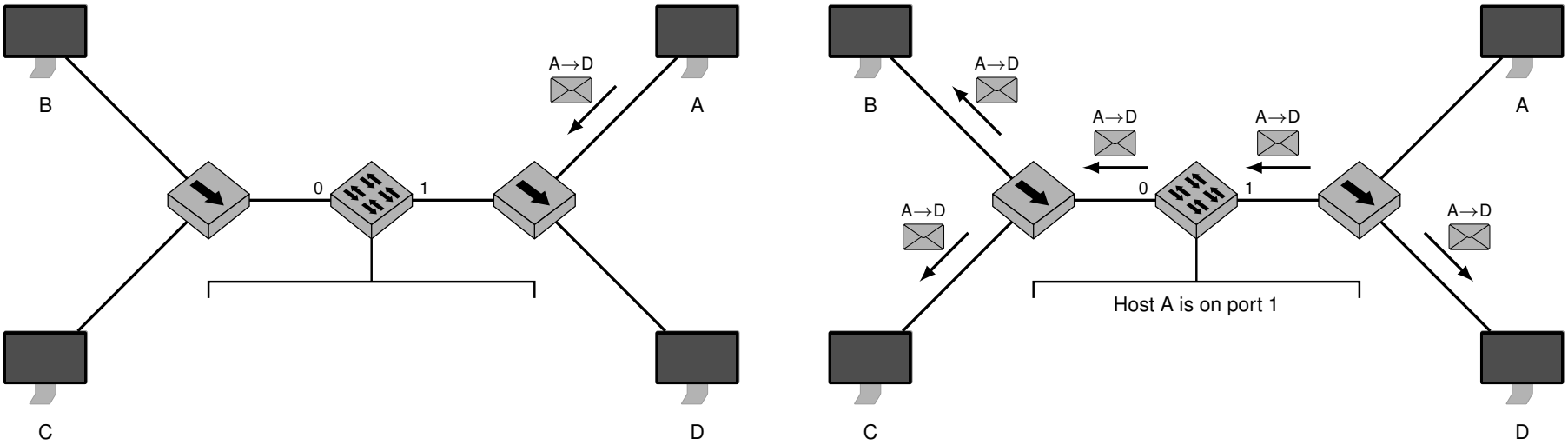
- Nicht mehr als 5 Abschnitte,
- verbunden durch 4 Repeater,
- wobei nur in 3 Abschnitten aktive Endgeräte enthalten sein dürfen.

Anmerkung: 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?

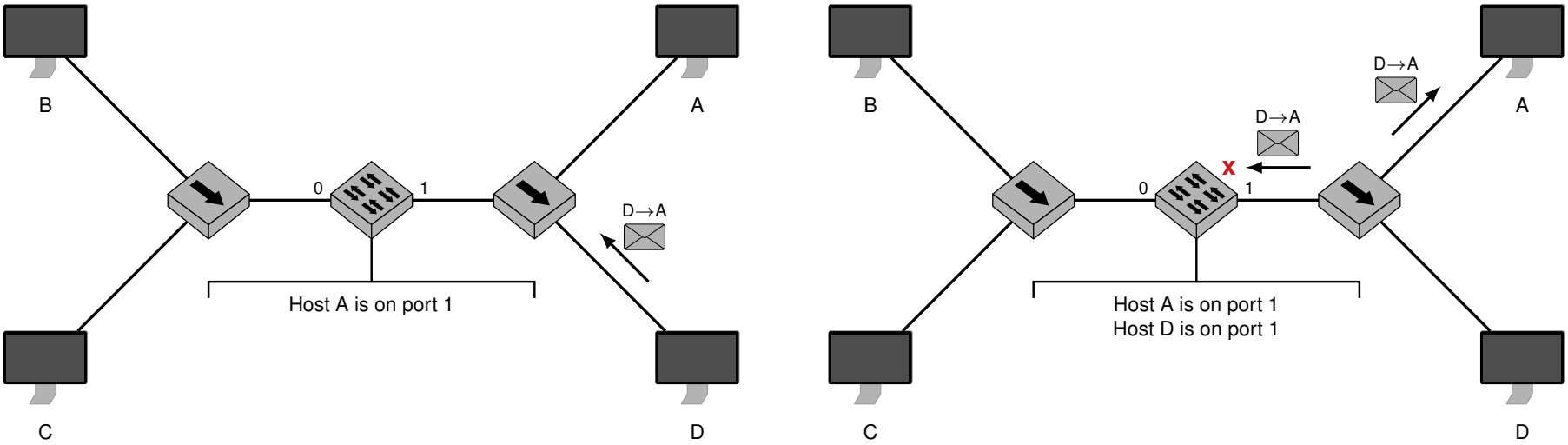
- Ja, wenn auf allen Abschnitten dasselbe Medienzugriffsverfahren genutzt wird (beispielsweise Verbindung Ethernet über BNC- und Patch-Kabel mit jeweils gleicher Datenrate).
- Unterschiedliche Zugriffsverfahren können nicht gekoppelt werden.

Verbindung auf Schicht 2: Switch [4]



- Zwei Gruppen von Hosts, die jeweils über Hubs verbunden sind, werden im obigen Beispiel durch einen Switch gekoppelt.
- Der Switch arbeitet zunächst wie ein Hub mit 2 Ports (Learning-Phase).
- Dabei merkt sich der Switch, über welchen Port ein Rahmen empfangen wurde.
- So ordnet er den Ports 0 und 1 die MAC-Adressen der Knoten zu, die an den jeweiligen Port angeschlossen sind.
- Ein Switch mit nur zwei Ports (was es heute im Kontext von Virtualisierung wieder häufiger gibt), nennt man auch Bridge.

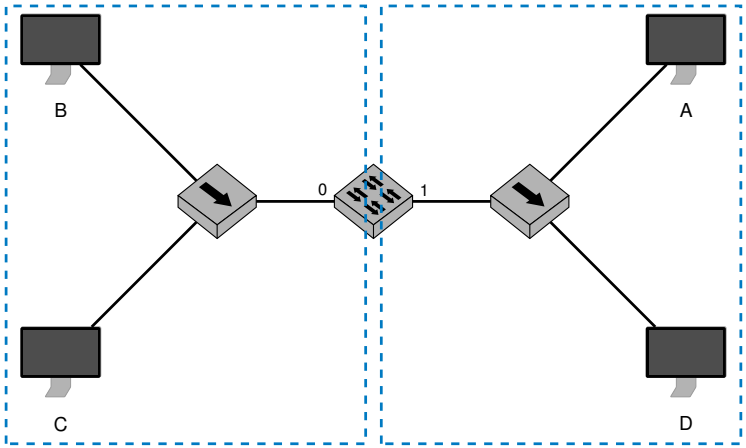
Verbindung auf Schicht 2: Switch



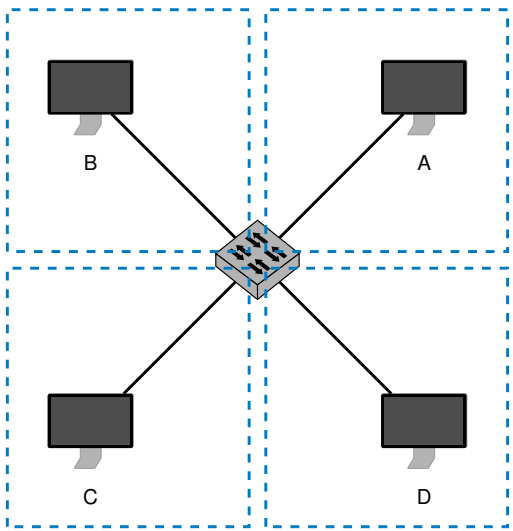
- Die Ziel-Adresse eingehender Rahmen wird mit den Einträgen in der Switching-Table verglichen.
- Ist ein Eintrag vorhanden, wir der Rahmen nur an den betreffenden Ziel-Port weitergeleitet.
- Ist kein Eintrag vorhanden, so wird der Rahmen an alle Ports weitergeleitet.
- Einträge erhalten einen Zeitstempel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

Verbindung auf Schicht 2: Switch

- Ein Switch bzw. eine Bridge unterbricht Kollisionsdomänen (auch als Segmentierung bezeichnet).
- Wenn ein Switch alle angeschlossenen Geräte kennt, darf in jedem der beiden Segmente jeweils ein Knoten zur selben Zeit senden.



- Ist pro Switchport genau ein Host angeschlossen, spricht man von Microsegmentation oder einem vollständig geswitchtem Netz (heute der Regelfall).
- In diesem Fall können jeweils zwei beliebige Hosts gleichzeitig miteineinander kommunizieren.



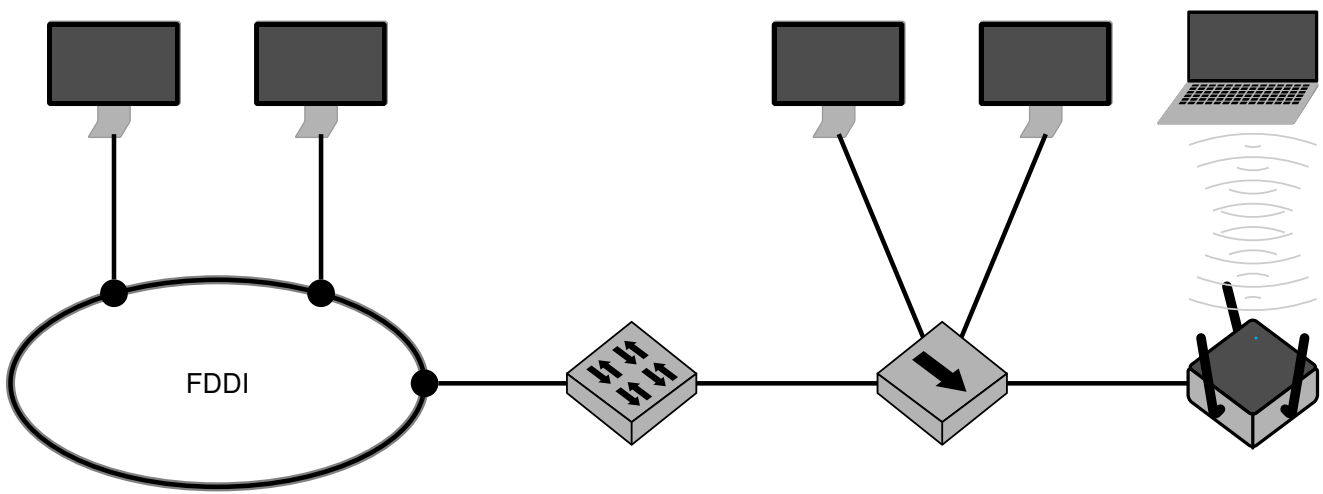
Hubs, Bridges und Switches

Switches können auch genutzt werden, um Netzsegmente mit unterschiedlichen Zugriffsverfahren zu koppeln:

- FDDI-Ethernet-Switch 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 ein Switch verwendet wird und
- im normalen Betrieb wird ein Host niemals direkt mit einem Switch kommunizieren.



Voraussetzung: Die MAC-Adressen müssen „kompatibel“ sein, um den jeweiligen Empfänger über seine MAC-Adressen identifizieren zu können.

Anmerkungen

- Switches sind für Hosts transparent, d. h. ein Host weiß nicht, dass er über einen Switch mit anderen Hosts kommuniziert.
- Sender- und Empfänger-Adresse werden von Switches nicht verändert.
- Switches 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).
- Ein Switch benötigt zur Erfüllung seiner grundlegenden Aufgaben keine eigene MAC-Adresse.
- Weiterleitungsentscheidungen werden 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

- 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.
- Um robuste lokale Netze aufzubauen werden Topologien mit redundanten Pfaden verwendet. Fällt eine Verbindung oder ein Knoten aus, kann der Verkehr umgeleitet werden. Aus redundanten Pfaden können Schleifen entstehen.

Wie werden Schleifen vermieden?

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

WLAN Access Points

WLAN Access Points sind im wesentlichen Brücken zwischen Twisted Pair und Funkübertragung:

- Ein RJ45-Interface in Richtung des kabelgebundenen Netzwerks
- Ein Wireless Transceiver in Richtung des Funknetzwerks

Allerdings besteht ein wesentlicher Unterschied zu Brücken bzw. Switches:

- WLAN Access Points sind nicht transparent auf Schicht 2!
  - Clients sind sich der Anwesenheit eines Access Points bewusst.
  - Zur Komminukation untereinander wird der Access Point direkt adressiert.

Gemeinsam mit Switches haben Access Points aber:

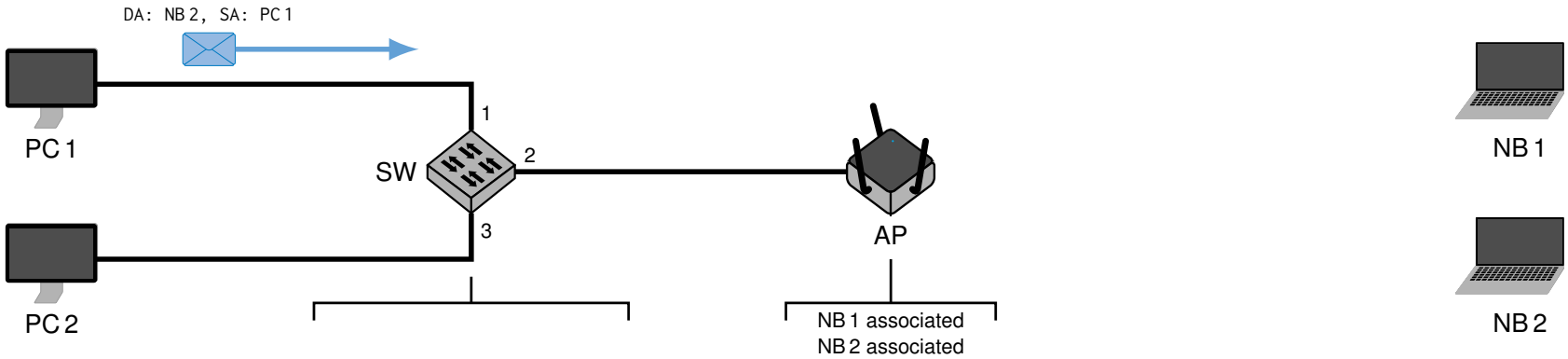
- Sie treffen Weiterleitungsentscheidungen auf Basis von MAC-Adressen.
- Sie unterbrechen Kollisionsdomänen auf logischer Ebene, d. h. ein Rahmen würde nicht weitergeleitet, sofern der betreffende Empfänger nicht mit dem jeweiligen AP assoziiert (verbunden) ist.

Wichtig: Da es ich hier um ein Broadcast-Medium handelt, kann jeweils nur eine Transmission zur selben Zeit stattfinden, da Rahmen andernfalls auf Schicht 1 kollidieren.

WLAN Access Points

Beispiel: PC 1 will mit NB 2 kommunizieren, wobei wir annehmen, dass

- NB 1 und NB 2 mit dem AP assoziiert (verbunden) sind und
- sonst noch keine Kommunikation im Netzwerk stattgefunden hat.

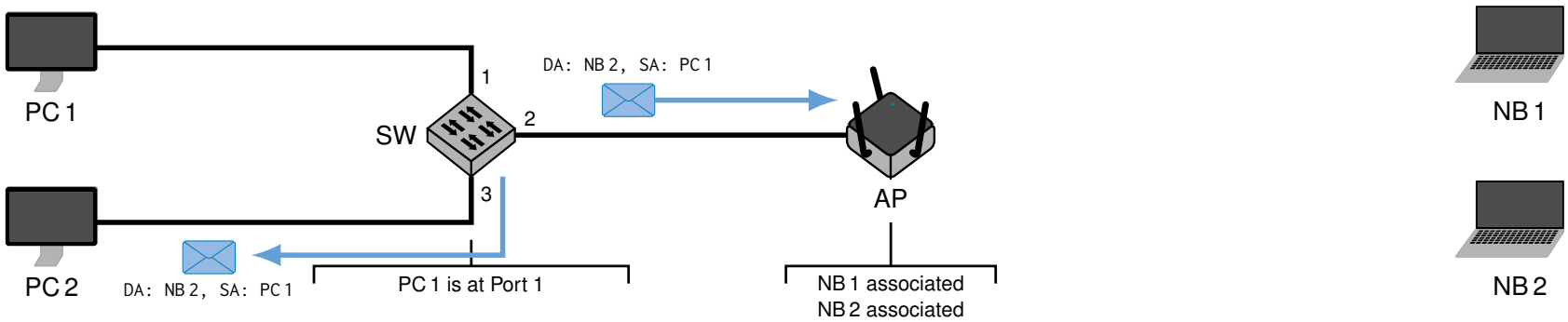


- PC 1 sendet einen Rahmen an NB 2.
- Source Address (SA) und Destination Address (DA) sind damit zunächst festgelegt.

WLAN Access Points

Beispiel: PC 1 will mit NB 2 kommunizieren, wobei wir annehmen, dass

- NB 1 und NB 2 mit dem AP assoziiert (verbunden) sind und
- sonst noch keine Kommunikation im Netzwerk stattgefunden hat.

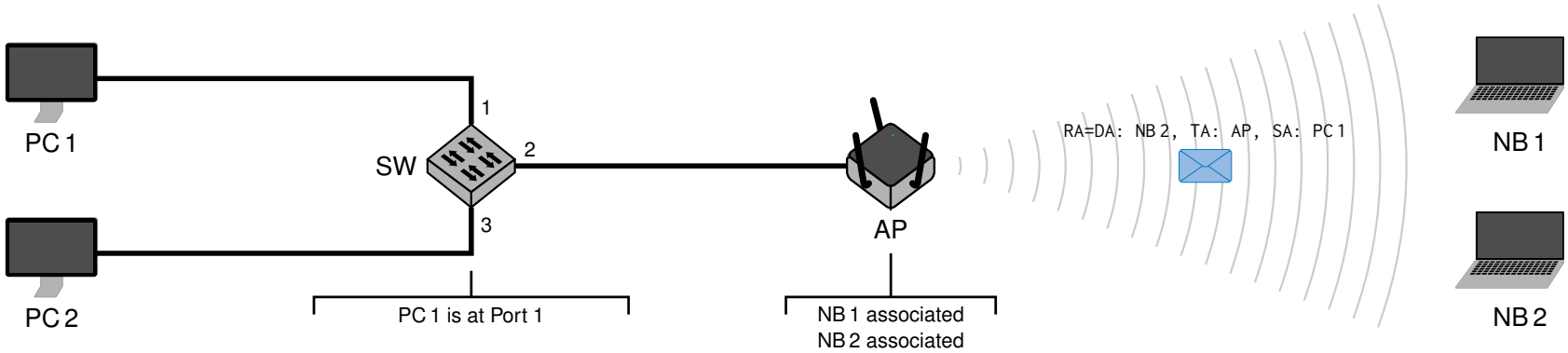


- Der Switch SW lernt, dass PC 1 and Port 1 angeschlossen ist.
- Der Empfänger NB 2 ist aber noch unbekannt, weswegen der Rahmen über alle Ports (außer dem, von dem er empfangen wurde) gesendet wird.

WLAN Access Points

Beispiel: PC 1 will mit NB 2 kommunizieren, wobei wir annehmen, dass

- NB 1 und NB 2 mit dem AP assoziiert (verbunden) sind und
- sonst noch keine Kommunikation im Netzwerk stattgefunden hat.

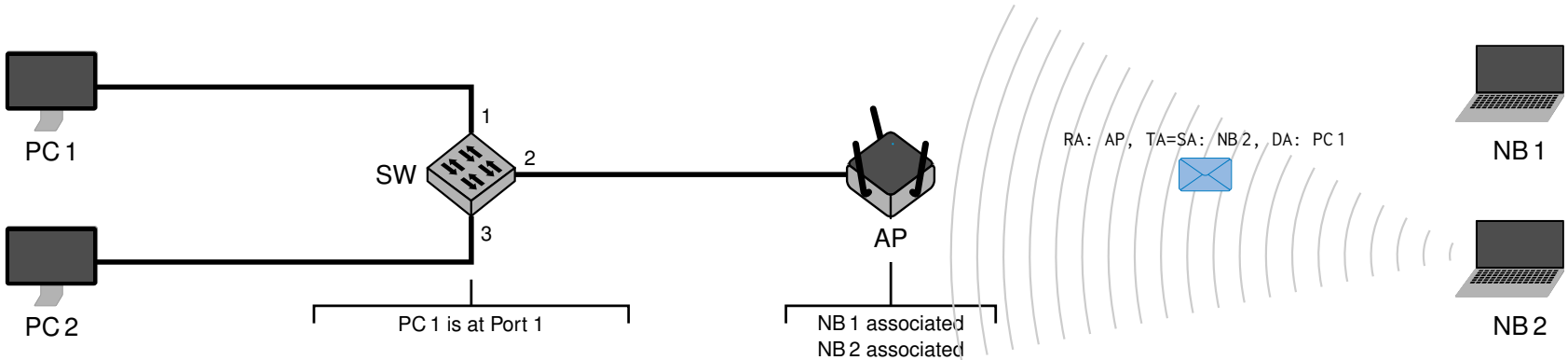


- Der Access Point AP empfängt den Rahmen und weiß, dass NB 2 eine assoziierte (verbundene) Station ist.
- Er akzeptiert daher den Rahmen und wandelt das Format von IEEE 802.3 zu IEEE 802.11 um.
- Die Receiver Address (RA) entspricht der Destination Address (DA).
- Transmitter Address (TA) ist die MAC-Adresse des AP.
- Source Address (SA) bleibt die Adresse von PC 1.
- NB 2 wird den Rahmen akzeptieren, NB 1 wird ihn ignorieren.

WLAN Access Points

Beispiel: PC 1 will mit NB 2 kommunizieren, wobei wir annehmen, dass

- NB 1 und NB 2 mit dem AP assoziiert (verbunden) sind und
- sonst noch keine Kommunikation im Netzwerk stattgefunden hat.

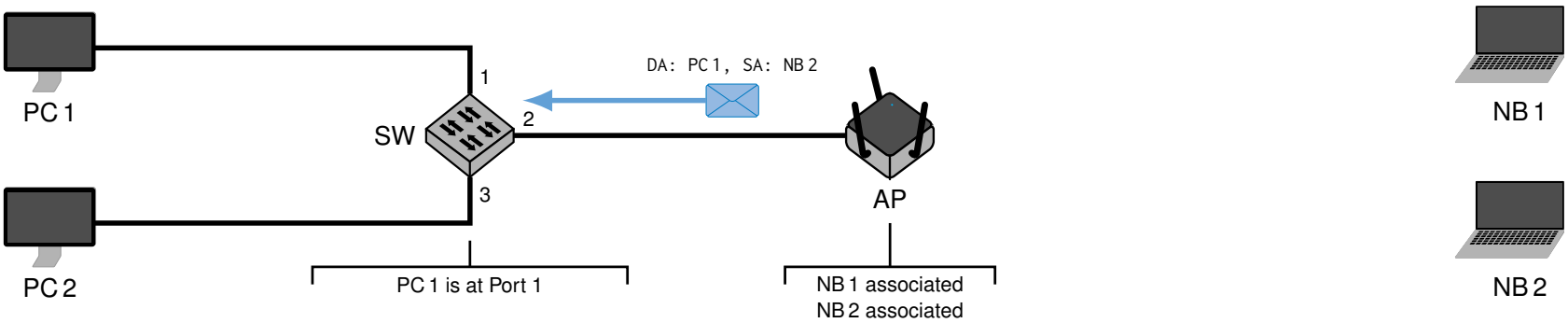


- NB 2 antwortet mit einem neuen Rahmen.
- Receiver Address (RA) ist der AP.
- Transmitter Address (TA) entspricht der Source Address (SA).
- Destination Address (DA) ist PC 1.
- Der AP empfängt den Rahmen und akzeptiert ihn, da er an ihn gerichtet ist (RA).

WLAN Access Points

Beispiel: PC 1 will mit NB 2 kommunizieren, wobei wir annehmen, dass

- NB 1 und NB 2 mit dem AP assoziiert (verbunden) sind und
- sonst noch keine Kommunikation im Netzwerk stattgefunden hat.

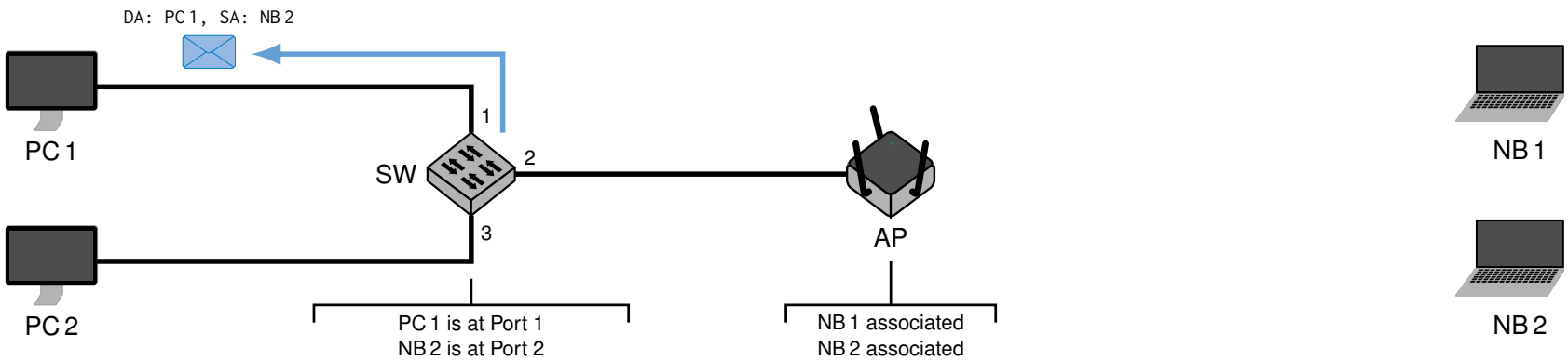


- Der AP weiß, dass PC 1 keine assoziierte Station ist – sich also nicht im WLAN befindet (tatsächlich könnte man die „Switching-Tabelle“ des APs um einen Eintrag von PC 1 erweitern, welcher an das kabelgebundene Interface angeschlossen ist).
- Der AP wird daher den Rahmen von IEEE 802.11 zu IEEE 802.3 zurückübersetzen.
- Source Address (SA) ist NB 2.
- Destination Address (DA) ist PC 1.

WLAN Access Points

Beispiel: PC 1 will mit NB 2 kommunizieren, wobei wir annehmen, dass

- NB 1 und NB 2 mit dem AP assoziiert (verbunden) sind und
- sonst noch keine Kommunikation im Netzwerk stattgefunden hat.

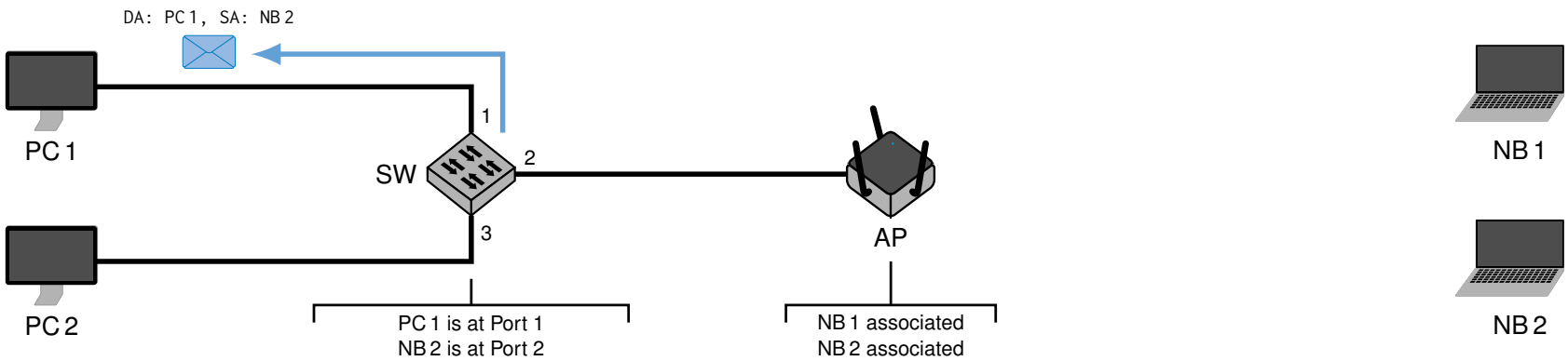


- SW 1 lernt, dass NB 2 an über Port 2 erreichbar ist.
- Da PC 1 bekanntlich an Port 1 angeschlossen ist, wird der Rahmen auch nur über diesen Port weitergeleitet.
- PC 1 akzeptiert den Rahmen.
- Weder PC 1 noch NB 2 haben bemerkt, dass der jeweils andere Kommunikationspartner über ein vollkommen anderes Medienzugriffsverfahren ans lokale Netzwerk angeschlossen ist.

WLAN Access Points

Beispiel: PC 1 will mit NB 2 kommunizieren, wobei wir annehmen, dass

- NB 1 und NB 2 mit dem AP assoziiert (verbunden) sind und
- sonst noch keine Kommunikation im Netzwerk stattgefunden hat.

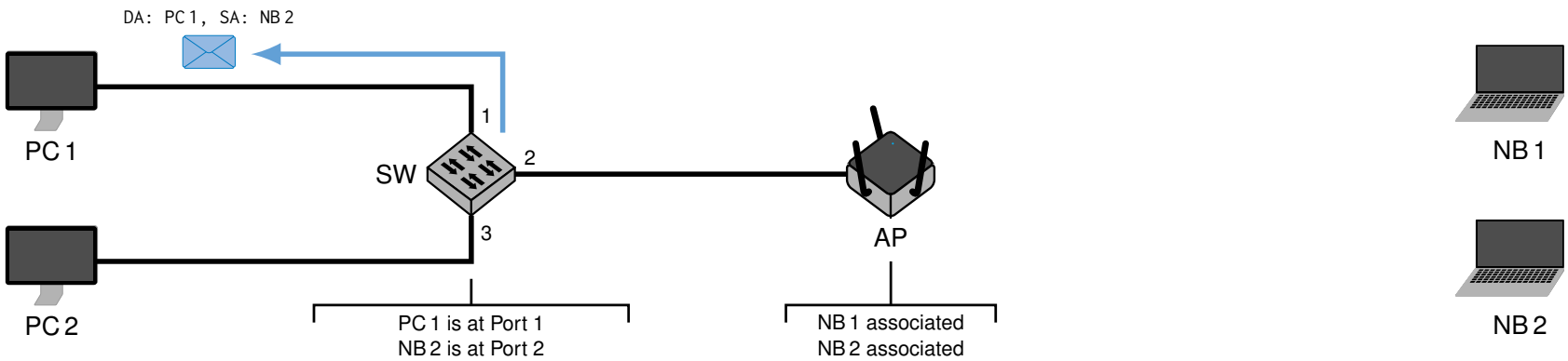


Wichtig: Im Gegensatz zum Switch wird der Access Point explizit adressiert. Innerhalb eines kabellosen Netzes ist ein AP daher nicht transparent.

WLAN Access Points

Beispiel: PC 1 will mit NB 2 kommunizieren, wobei wir annehmen, dass

- NB 1 und NB 2 mit dem AP assoziiert (verbunden) sind und
- sonst noch keine Kommunikation im Netzwerk stattgefunden hat.



Wichtig: Im Gegensatz zum Switch wird der Access Point explizit adressiert. Innerhalb eines kabellosen Netzes ist ein AP daher nicht transparent.

Frage: Woher kennt PC 1 eigentlich eine kryptische MAC-Adresse wie `de:ad:be:ef:00:01`, die zu NB 2 gehört?

WLAN Access Points

Der Unsinn des „WLAN Routers“

Der Begriff „WLAN Router“ ist technisch falsch:

- Hersteller verkaufen hier Geräte, welche gleichzeitig
  - Ethernet Switch,
  - Router (Ethernet $\leftrightarrow$ DSL/Cable/etc.), und
  - WLAN Access Point sind.
- Tatsächlich sind WLAN Access Points nicht mehr als Switches mit integrierten Medienkonvertern, wobei meistens gleich noch ein Router integriert wird.

Routing (siehe Kapitel 3) findet innerhalb kabelloser Netzwerke im Infrastructure Mode nicht statt.

Machen Sie sich bewusst, dass unter diesem Begriff heutzutage Geräte mit mind. drei[8] unterschiedlichen Funktionen vermarketet werden.

[8] Eigentlich sind es sogar mind. vier Funktionen, da für den Übergang von Ethernet auf WAN-Verbindung (Wide Area Network) i. d. R. ein Modem (Modulator / Demodulator) notwendig ist, welcher in das betreffende Gerät meist ebenfalls integriert ist. (vgl. „DSL-Model“).

Kapitel 2: Sicherungsschicht

Darstellung von Netzwerken als Graphen

Verbindungscharakterisierung, Mehrfachzugriff, Medienzugriffskontrolle

Rahmenbildung, Adressierung und Fehlererkennung

Verbindung auf Schicht 1 und 2

Zusammenfassung

Literaturangaben

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.

Kapitel 2: Sicherungsschicht

Darstellung von Netzwerken als Graphen

Verbindungscharakterisierung, Mehrfachzugriff, Medienzugriffskontrolle

Rahmenbildung, Adressierung und Fehlererkennung

Verbindung auf Schicht 1 und 2

Zusammenfassung

Literaturangaben

Literaturangaben

[1] D. Eastlake and J. Abley.
IANA Considerations and IETF Protocol and Documentation Usage for IEEE 802 Parameters, 2013.
http://tools.ietf.org/html/rfc7042.

[2] E. Stein.
Taschenbuch Rechnernetze und Internet, chapter Konzepte: Lokale Netzwerke, pages 191–218.
Fachbuchverlag Leipzig, 2. edition, 2004.

[3] E. Stein.
Taschenbuch Rechnernetze und Internet, chapter Fehlererkennung durch CRC, pages 86–87.
Fachbuchverlag Leipzig, 2. edition, 2004.

[4] E. Stein.
Taschenbuch Rechnernetze und Internet, chapter Netzaufbau, pages 200–203.
Fachbuchverlag Leipzig, 2. edition, 2004.