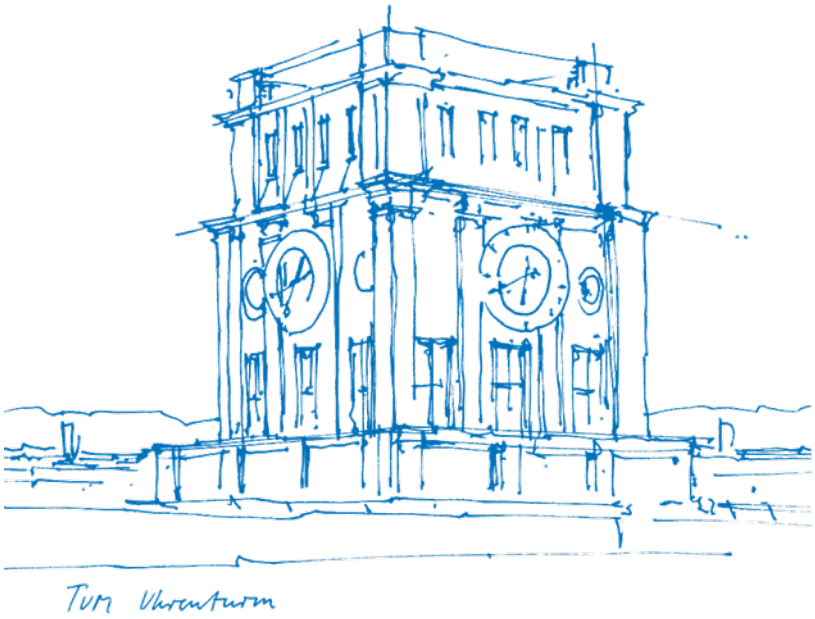
GRNVS Tutorium 05 - SS21

Fabian Sauter
Technische Universität München
Grundlagen: Rechnernetze und Verteilte Systeme (IN0010)
Lehrstuhl für Netzarchitekturen und Netzdienste
Garching, 17.05.2021

Slides & Notes: http://grnvs.uwpx.org



Formales - 1:

- Name: Fabian Sauter
- Studium: 3. Semester M.Sc. Informatik
- E-Mail: fabian.sauter@in.tum.de "[GRNVS]" im Betreff einfügen!
- Meistgenutzte Programmiersprachen
  - C++
  - C#
  - C
  - Python
- Sonstiges
  - XMPP Dev
  - Web: `https://uwpx.org`
- Werde versuchen **alles** hochzuladen:
  - `https://home.in.tum.de/~sauterf/grnvs`
  - `http://grnvs.uwpx.org`

Formales - 2:

- Aufzeichnung des Tutorium - **Nein**
- Dürft ihr mein Tutorium aufzeichnen?
  - Privatkopie: Ja, kann ich eh nichts dagegen machen.
  - Veröffentlichung: **Verboten** - ich bekomme raus von wem das kommt (Metadaten).
- Rechtschreibung manchmal etwas interessant :D

Disclaimer:

Bei all meinen Folien handelt es sich um privat erstellte und verwalte Folien für das Tutorium GRNVS im SS21.
Sie erheben keinerlei Anspruch auf vollständigkeit und korrektheit.
Im Zweifel hat IMMER die Übungsleitung bzw. die offiziellen Vorlesungsmaterialien recht!

Organisatorisches:

Wie meine Tutorien ablaufen werden:

- 15-30 min kurze Wiederholung mit Aufschrieb.
- 75-90 min normales Tutorium/bearbeitung Aufgaben.
- 0-30 min sonstige Beispiele (selten).

Organisatorisches:

Wie meine Tutorien ablaufen werden:

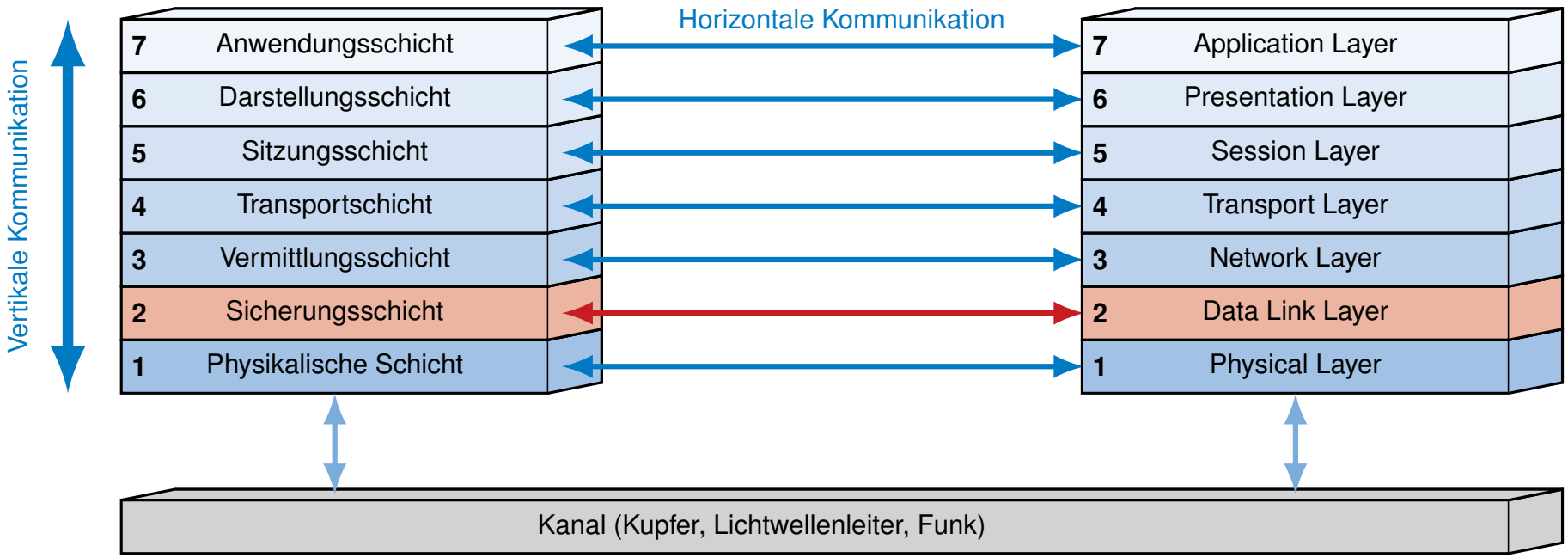
- 15-30 min kurze Wiederholung mit Aufschrieb.
- 75-90 min normales Tutorium/bearbeitung Aufgaben.
- 0-30 min sonstige Beispiele (selten).

Sonstiges:

- Redet mit mir, wenn ihr eine Frage habt! Ich bin genau so wie ihr auch nur ein Student.
- Ich will, dass ihr mitschreibt, da ihr dann nicht nur einfach vor dem Laptop eure Zeit absitzt.
- Stört euch etwas, dann lasst es mich wissen (persönlich, per Mail, ...)

Kapitel 2: Sicherungsschicht

Einordnung im ISO/OSI-Modell



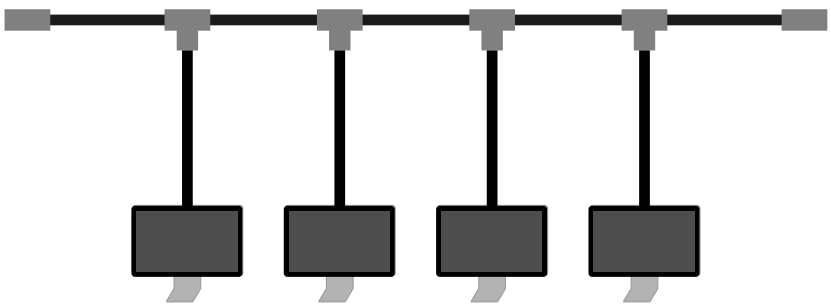
Kapitel 2: Sicherungsschicht

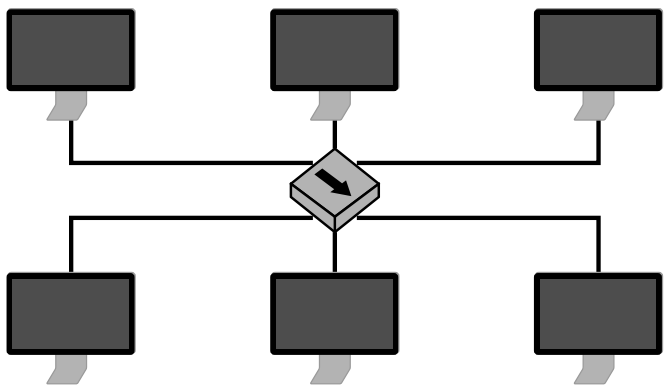
Wir beschäftigen uns zunächst mit sog. Direktverbindungsnetzen, d. h.

- 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)





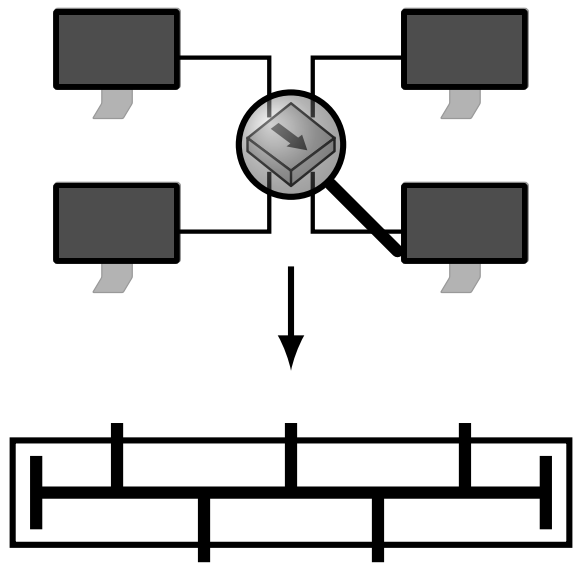
- Verbindung zwischen Basisstation und Mobiltelefon
- Bus-Systeme innerhalb eines Computers, z. B. USB, PCIe etc.

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

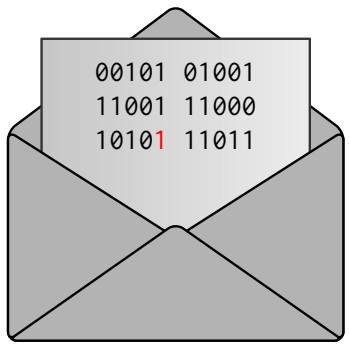


Die wesentlichen Aufgaben der Sicherungsschicht sind

- 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

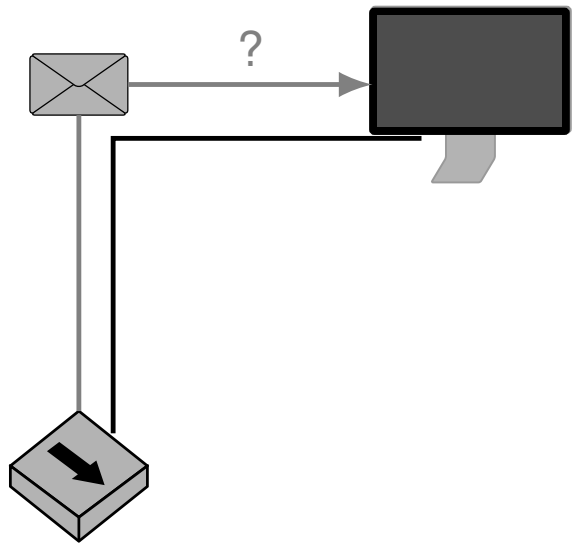


Die wesentlichen Aufgaben der Sicherungsschicht sind

- 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



Übertragungsrate

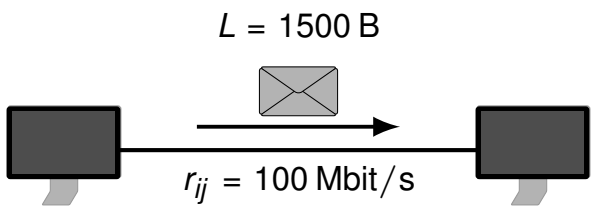
Übertragungsrate und Serialisierungszeit

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:

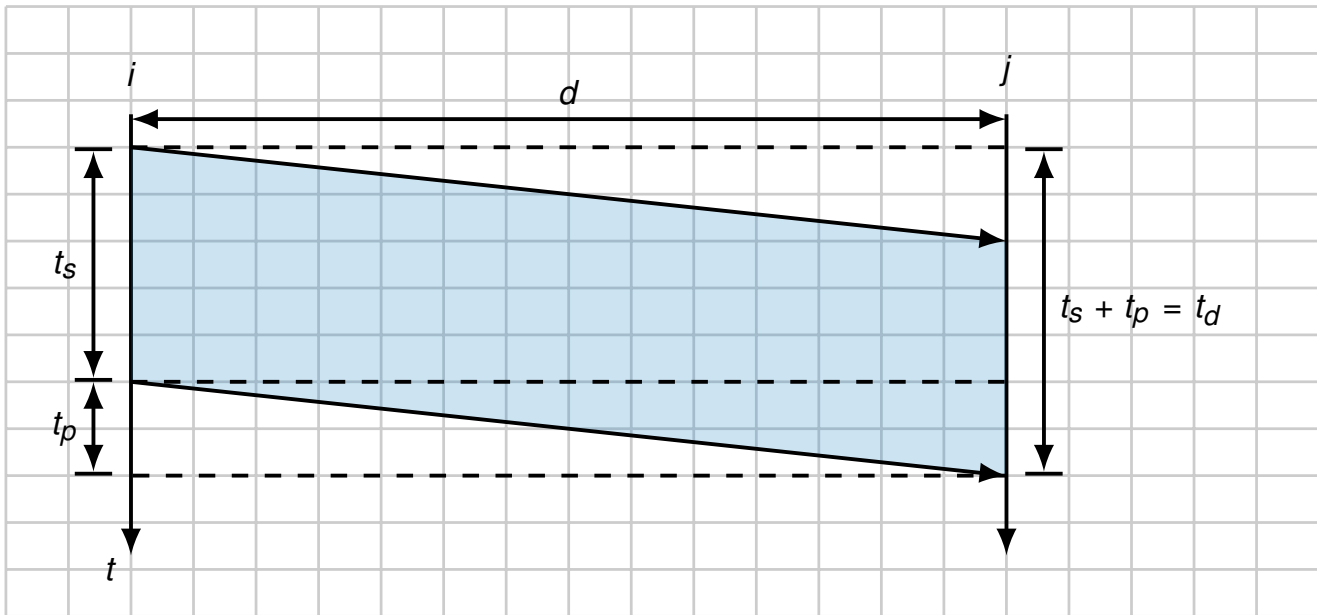


Abbildung 1: Nachrichtenflussdiagramm

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

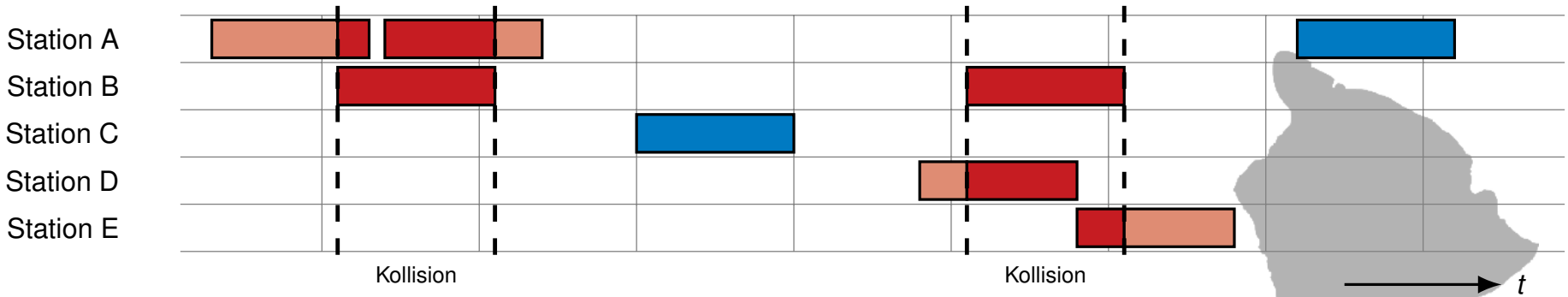
ALOHA und Slotted ALOHA

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

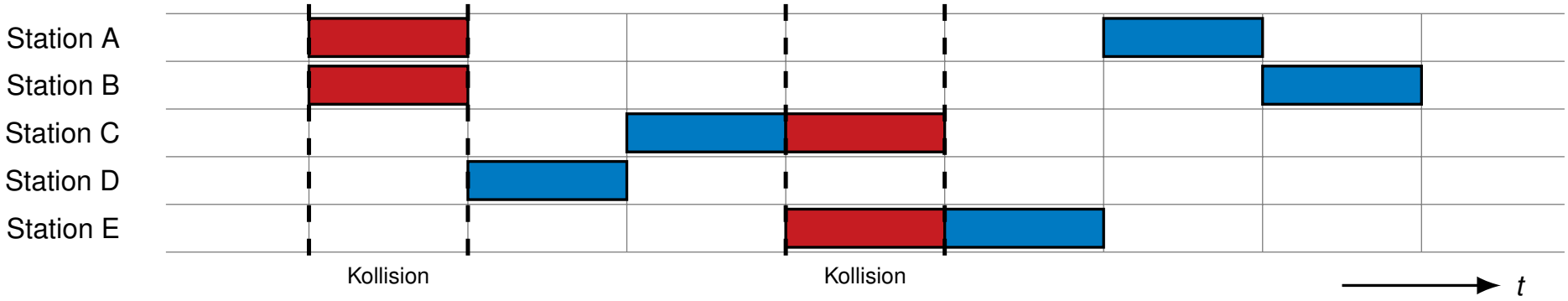
- Jede Station sendet an eine zentrale Station (vgl. „Basisstation“ in WLANs), sobald Daten vorliegen
- Senden zwei Stationen gleichzeitig, kommt es zu Kollisionen
- Erfolgreich übertragene Nachrichten werden vom Empfänger auf anderer Frequenz quittiert („out-of-band“ Bestätigungsverfahren auf Link-Layer,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



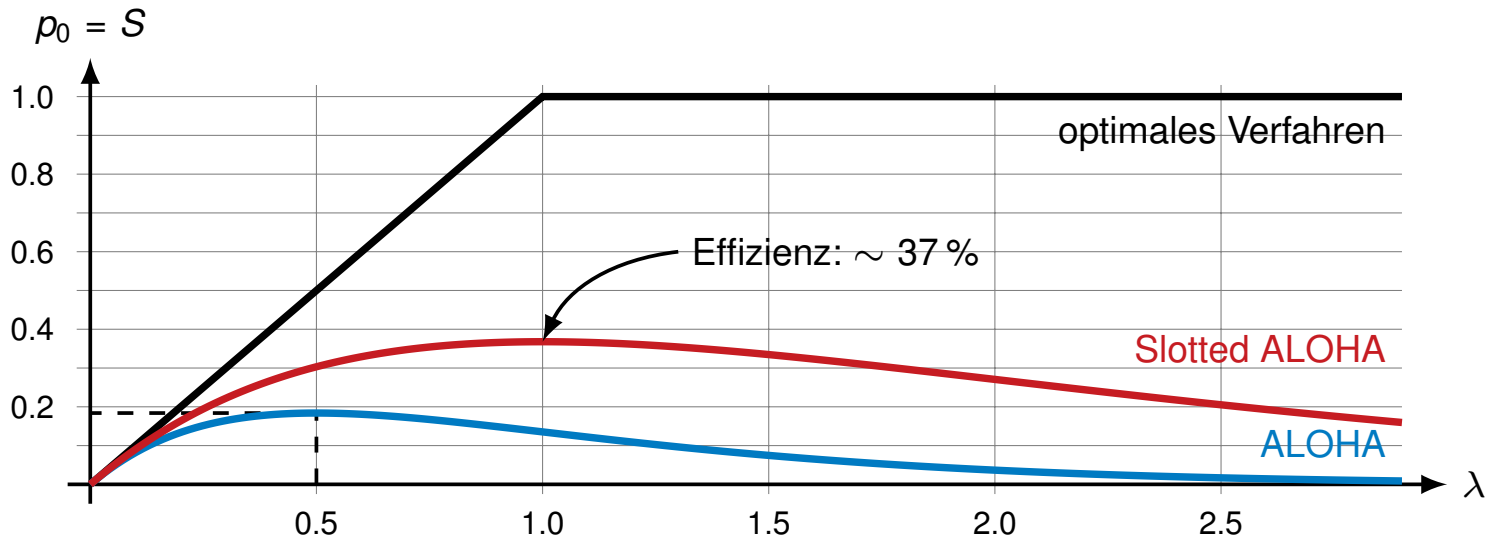
Das Kanalmodell ist vergleichsweise einfach. Es existieren math. Beschreibungen für den sog. ALOHA Random Access Channel.

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/CD (Collision Detection)

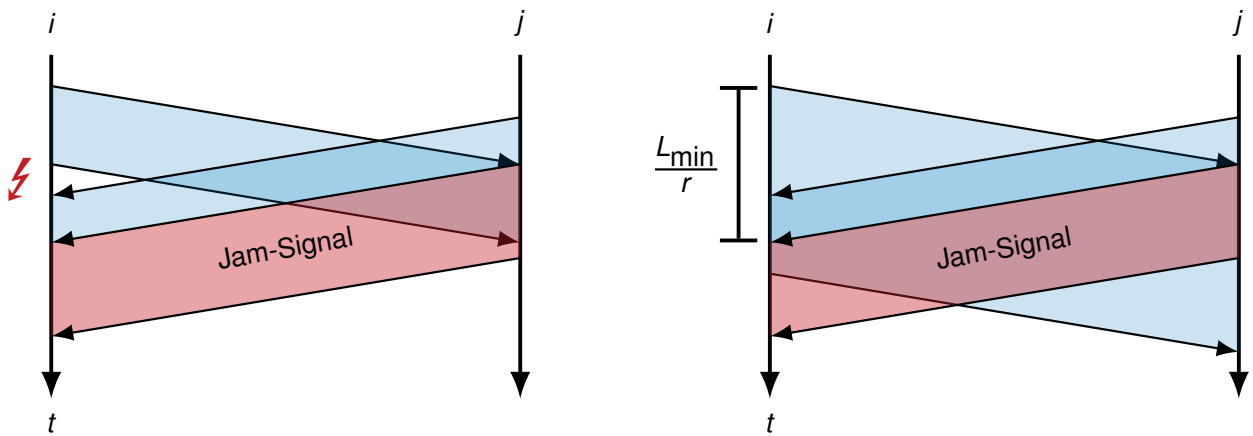
- Erkenne Kollisionen und wiederhole die Übertragung, wenn eine Kollision erkannt wird
- Verzichte auf das Senden von Bestätigungen
- Wird keine Kollision erkannt, gilt die Übertragung als erfolgreich

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

Voraussetzung für CSMA/CD [1]

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

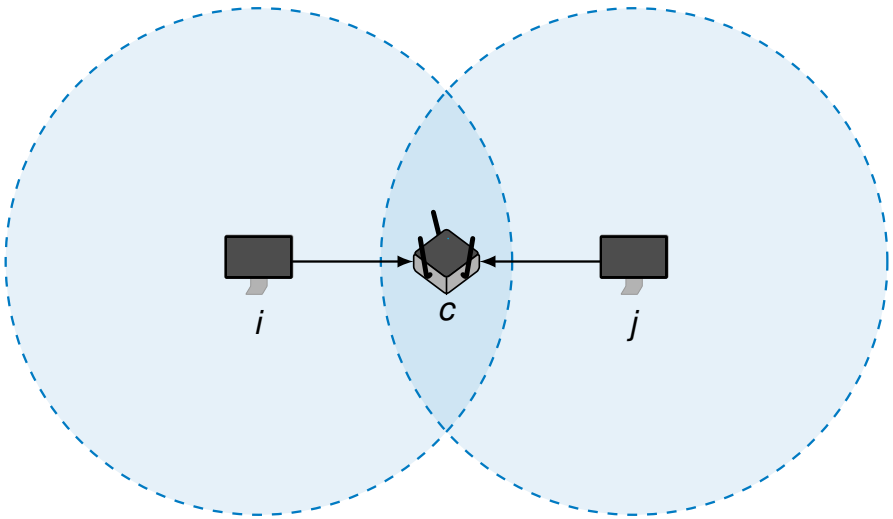
$$L_{\text{min}} = \frac{2d}{\nu c_0} r$$



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.

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														

Organisatorisches:

Wie meine Tutorien ablaufen werden:

- 15-30 min kurze Wiederholung mit Aufschrieb.
- 75-90 min normales Tutorium/bearbeitung Aufgaben.
- 0-30 min sonstige Beispiele (selten).

Sonstiges:

- Redet mit mir, wenn ihr eine Frage habt! Ich bin genau so wie ihr auch nur ein Student.
- Ich will, dass ihr mitschreibt, da ihr dann nicht nur einfach vor dem Laptop eure Zeit absitzt.
- Stört euch etwas, dann lasst es mich wissen (persönlich, per Mail, ...)

Tipp zu den Programmieraufgaben: **Nicht** in Java machen!

Wiederholung

Ablauf:

- Aufgabe 1
- Aufgabe 2
- Aufgabe 3

Studenten zählen

(Nur als Erinnerung für mich.)