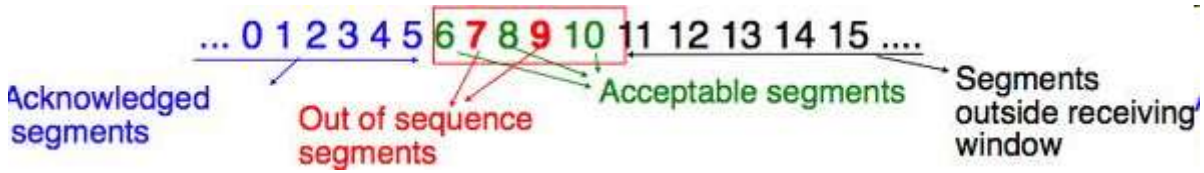
Technische Hochschule Ulm

Rechnernetze

Kapitel 3: Die Sicherungsschicht und lokale Netze

Hochschule Ulm
Prof. Dr. F. Steiper



- *Urheberrechte*
  - *Die Vorlesungsmaterialien und Vorlesungsaufzeichnungen zum Kurs „Rechnernetze (INF2)" dürfen nur für private Zwecke im Rahmen Ihres Studiums an der Technischen Hochschule Ulm genutzt werden.*
  - *Eine Vervielfältigung und Weitergabe dieser Materialien in jeglicher Form an andere Personen ist untersagt.*
  - *© Copyright. Frank Steiper. 2022. All rights reserved*

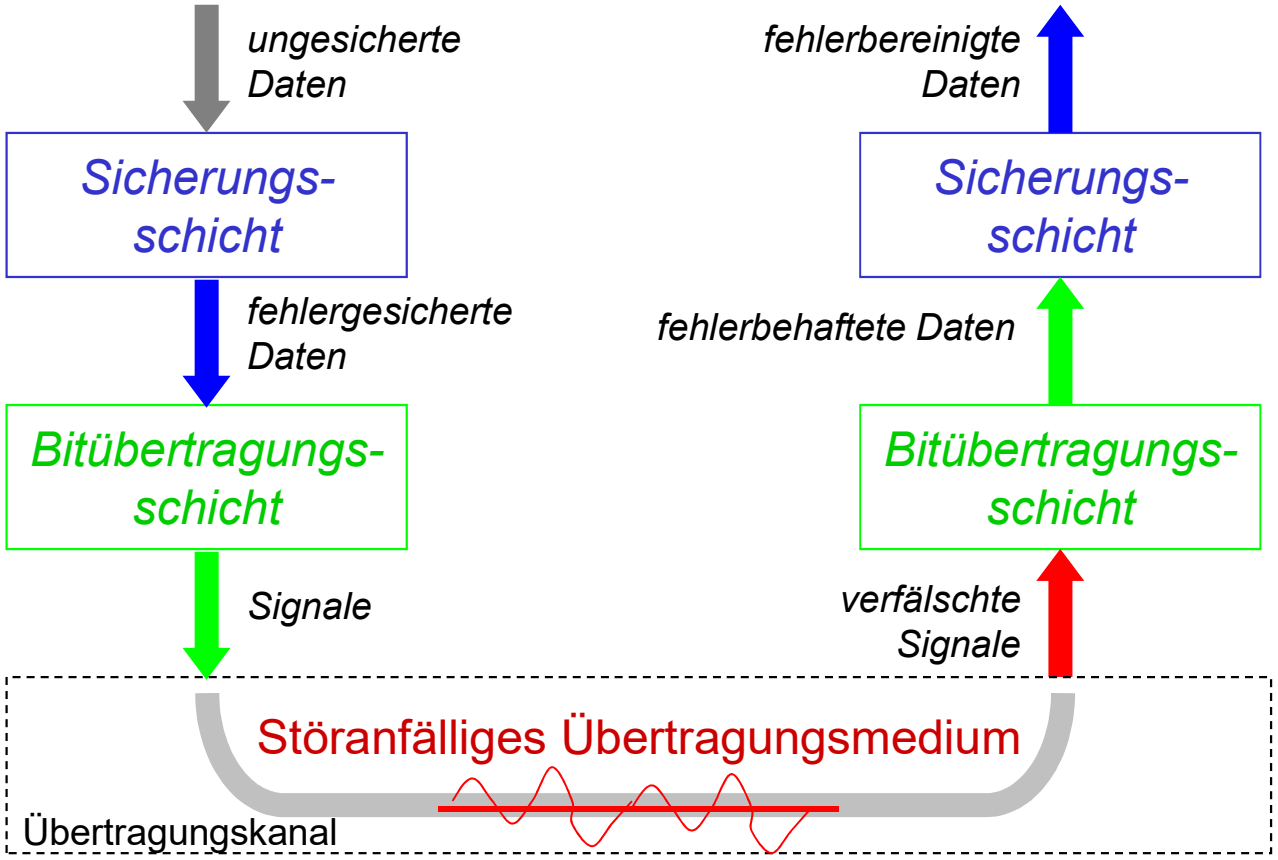
3. Die Sicherungsschicht

- *Aufgaben der Sicherungsschicht (engl.: Data Link Layer)*
  - *Bereitstellung einer logischen Verbindung zwischen direkt verbundenen Kommunikationssystemen*
    - *D.h. zwischen zwei direkt über den gleichen physikalischen Übertragungskanal kommunizierenden Systemen*
  - *Zuverlässige Zustellung von Daten für die Vermittlungsschicht*
    - *Bereitstellung einer definierten Dienstschnittstelle, z.B.*
      1. *Unbestätigte, verbindungslose Dienste*
      2. *Bestätigte, verbindungslose Dienste*
      3. *Verbindungsorientierte Dienste*
    - *Sicherung der Daten vor Verfälschung bei der Übertragung*
      1. *Rahmenbildung und –erkennung*
      2. *Fehlerkontrolle über Fehlererkennung und –behebung*
      3. *Flusskontrolle und Vermeidung von Datenverlusten*

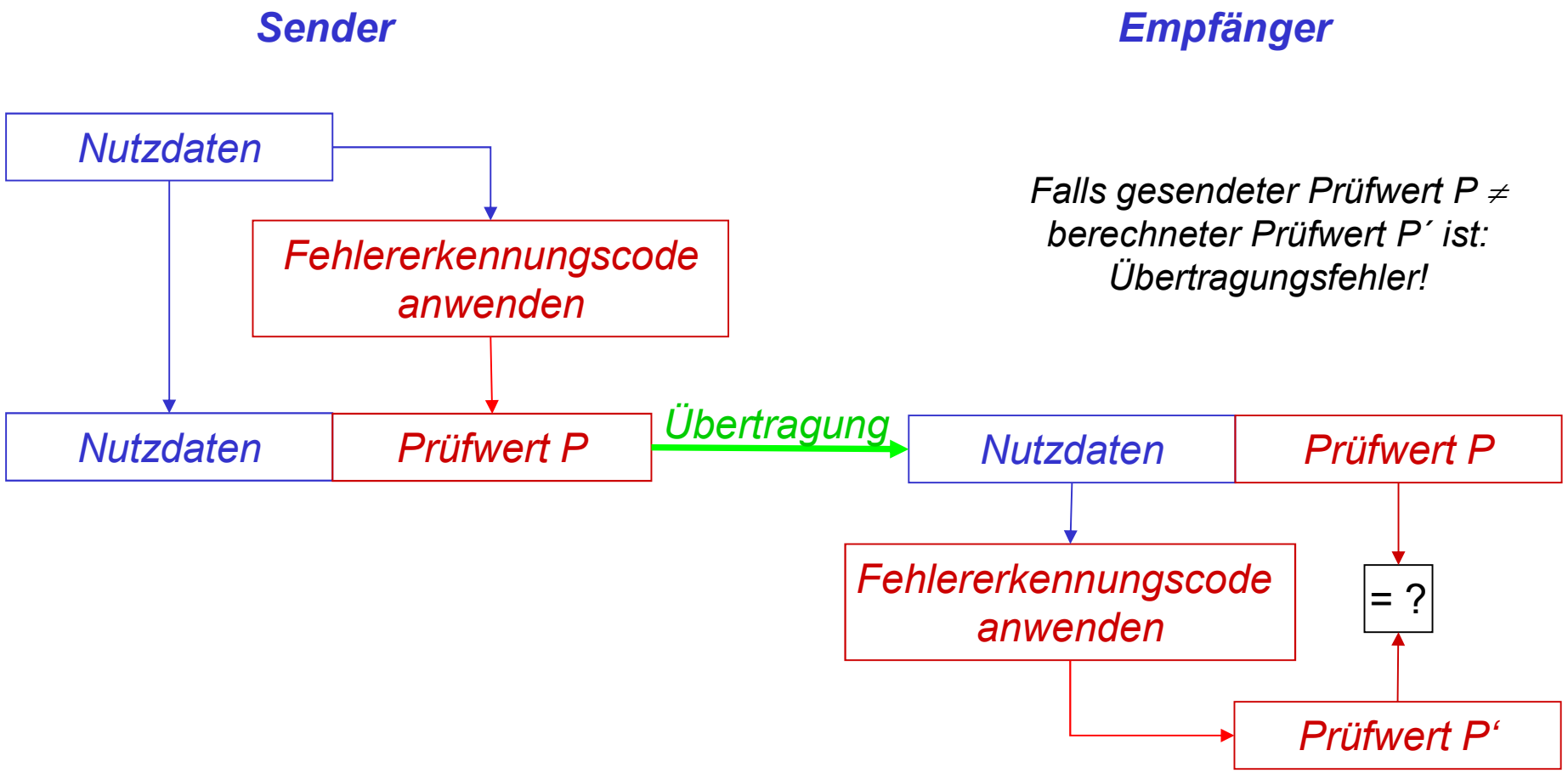
[Ref 1] Kapitel 5, Seite 475-481
[Ref 3] Kapitel E, Seite 111

3. Die Sicherungsschicht

- *Funktion der Sicherungsschicht (Schema)*



- *Allgemeiner Ablauf der Fehlererkennung* [Ref 1] Kapitel 5, Seite 481-483

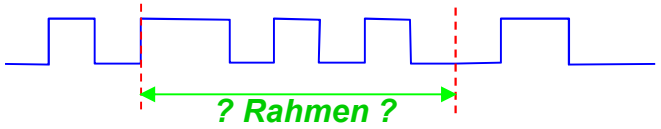


3.1 Rahmenbildung und Fehlererkennung

- *Rahmenbildung (engl.: "framing")*

[Ref 2] Kapitel 3, Seite 238-242
[Ref 3] Kapitel E, Seite 112-115

  - *Problem*
    - *Zur Fehlererkennung werden Bitströme in kleine Dateneinheiten, sogenannte Rahmen aufgeteilt*
    - *Wie erkennt ein Empfänger Anfang und Ende der vom Sender generierten Rahmen?*



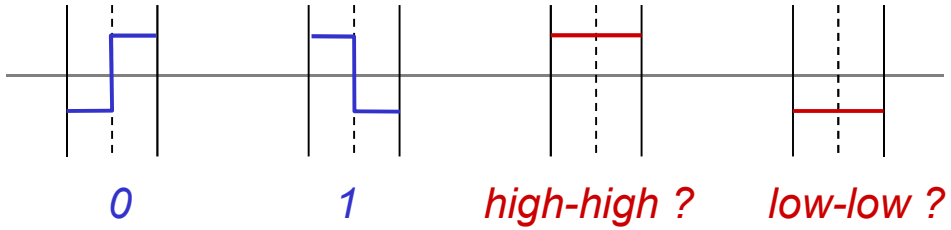
  - *Format von Rahmen*
    - *Hängt von der eingesetzten Netztechnologie ab*
      - → *Ethernet, Token Ring, ATM,…*
    - *Typischer Aufbau*
      - → *Start-/End-Begrenzer, Header (H/W-Adressen, Sequenzen,…), Nutzdaten („Payload") und Prüfsumme (zur Fehlererkennung- und evtl. –korrektur)*

Start	Header	Nutzdaten	Prüfsumme	End

3.1 Rahmenbildung und Fehlererkennung

- *Alternativen zur Erkennung von Rahmengrenzen*

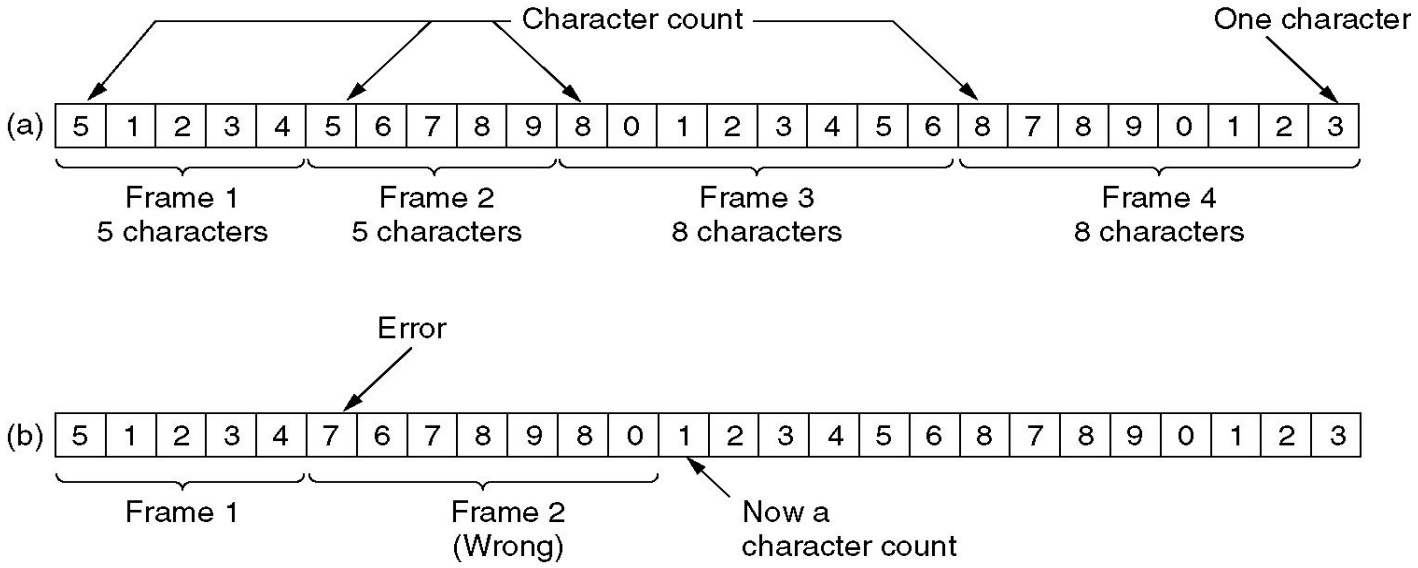
  1. *Verwendung illegaler Codezeichen auf Bitübertragungsebene*
     - *Z.B. bei Manchester-Codierung: kein Signalübergang in der Mitte des Intervalls*



- *Alternativen zur Erkennung von Rahmengrenzen*

2. *Längenangabe im Rahmen-Header: Byte-Zählmethode*
   - *Sobald der Empfänger das Header-Feld liest, ist bekannt, wie viele Bytes mit Nutzdaten im Rahmen folgen*
   - *Problem: Verfälschung des Headers während der Übertragung*

     → *Neusynchronisation ohne weitere Mechanismen nicht mehr möglich!*



- *Alternativen zur Erkennung von Rahmengrenzen*

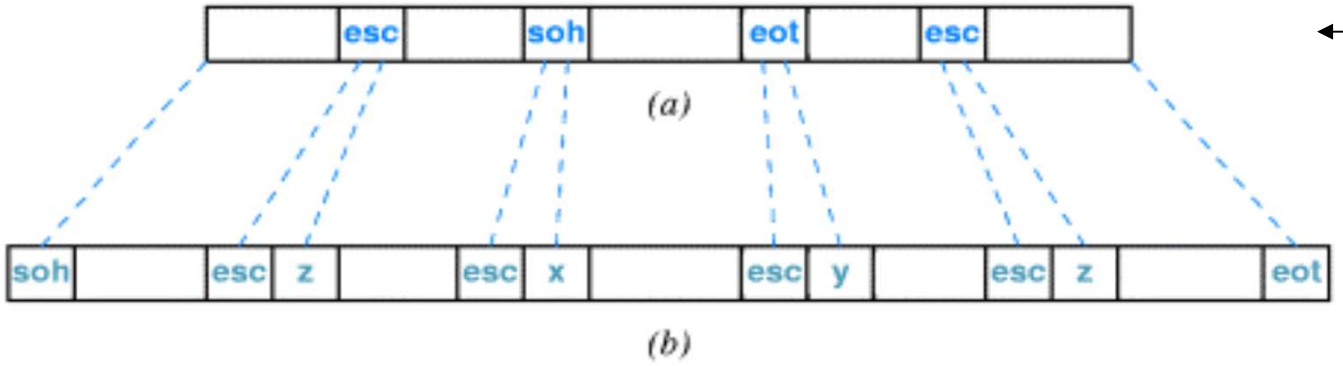
3. *Verwendung von speziellen Steuerzeichen: Byte-Stopfen*
   - *Beispiel: Ein Datenrahmen enthält eine Anzahl von Zeichen*
     - → *Die Zeichen sind z.B. ASCII-codiert*
     - → *Spezielle ASCII-Zeichen werden als Steuerzeichen genutzt, z.B.:*
       SOH → „Start of Header“
       EOT → „End of Transmission“

   - *Problematik:*
     - → *In Nutzdaten können zufällig Steuerzeichen auftreten (z.B. in einer Binärdatei)*
     - → *Wie können zufällig vorkommende SOH- und EOT-Steuerzeichen in den Nutzdaten von wirklichen Rahmenbegrenzern unterschieden werden?*

- *Alternativen zur Erkennung von Rahmengrenzen*

3. *Verwendung von speziellen Steuerzeichen: Byte-Stopfen…*
   - *Verfahren: Vor der Versendung zufällige Steuerzeichen umcodieren*

Fehlinterpretierbare Nutzdaten (a)		*Gesendete Daten (b)*
soh	↔	*esc x*
eot	↔	*esc y*
esc	↔	*esc z*



← Nutzdaten

← Übertragene Daten

→ *Empfänger wandelt umkodierte Byte-Sequenzen wieder in ursprüngliche Form zurück*

3.1 Rahmenbildung und Fehlererkennung

- *Fehlererkennung*

[Ref 1] Kapitel 5, Seite 483-488
[Ref 2] Kapitel 3, Seite 244-258
[Ref 3] Kapitel E, Seite 116-124

  - *Fehlererkennung*
    - *Aufteilung der Daten in einzelne Rahmen durch den Sender*
    - *Pro Rahmen wird redundante Zusatzinfo mitgeschickt*
      - *→ Zusatzinfo wird gemäß vereinbartem Fehlererkennungscode bestimmt*
    - *Ermöglicht Empfänger, Übertragungsfehler aufzudecken*
  - *Zulässige Wörter einer Codierung*
    - *Mit den m Daten- und Header-Bits eines Rahmens kann man prinzipiell 2^m unterschiedliche Bitfolgen bilden.*
    - *Werden r zusätzliche Prüfsummenbits angehängt, können also maximal $2^{(m+r)}$ unterschiedliche Bitfolgen gebildet werden*
    - *Der Fehlererkennungscode schränkt die Zahl auf weniger als $2^{(m+r)}$ zulässige Wörter ein, - wir nehmen an, es seien 2^X Wörter*
    - *Die 2^X Wörter nennt man die zulässigen Wörter des „Wortschatzes“ bzw. der Codierung*

- *Die Hamming-Distanz*
  - *Erlaubt die Bewertung von Fehlererkennungscodes*
    - *Wie viele unabhängige Bitfehler können in einem Datenwort mit gegebener Bitlänge erkannt bzw. behoben werden?*
  - *Definitionen*
    - *Distanz zwischen 2 zulässigen Worten*
      - → *ist die Anzahl unterschiedlicher Bitpositionen*
      - → *ist bestimmbar durch die XOR-Verknüpfung zweier Worte; Anzahl der 1en im Ergebnis ist die Distanz zwischen den Worten*
    - *Hamming-Distanz einer Codierung*
      - → *ist die minimale Distanz zweier beliebiger, zulässiger Worte einer Codierung*
  - *Es gelten folgende Regeln:*
    - *Für die Erkennung von d Bitfehlern muss die Codierung eine Hamming-Distanz von d+1 besitzen*
      - → *d Bitfehler können kein zulässiges Wort in ein anderes zulässiges Wort wandeln*
    - *Für die Behebung von d Bitfehlern muss die Codierung eine Hamming-Distanz von (2d+1) besitzen*
      - → *bei maximal d Bitfehlern hat das gültige Codewort die kleinste Distanz*

- *Fehlererkennungscodes*
  - *Eindimensionale Parität*
    - *Übertragung eines zusätzlichen Bits zu jedem Wort der Länge d Bit*

      → *Ungerade Parität (engl.: "Odd Parity", OP)*
      (d+1)tes Bit wird auf 1 gesetzt, falls Anzahl der 1en im d-Bit-Wort gerade

      → *Gerade Parität (engl.: "Even Parity", EP)*
      (d+1)tes Bit wird auf 1 gesetzt, falls Anzahl der 1en in d-Bit-Wort ungerade

Zeichen	Paritätsbit für EP	Paritätsbit für OP
0010111	0	1
1011101	1	0
0000001	1	0
1100110	0	1

    - *Wie groß ist hier die Hamming-Distanz?*

- *Fehlererkennungscodes…*
  - *Zweidimensionale Parität*
    - *Zusätzliche Paritätsberechnung für jeweilige Bit-Position quer über jedes im Rahmen enthaltene Byte*
      - → *Ergibt zusätzliches Paritätsbyte für den gesamten Rahmen*
      - → *Die Hamming-Distanz wächst!*
    - *Beispiel:*
      - → *Rahmen mit sechs 7 Bit-Zeichen; 2-dimensionale gerade Parität*

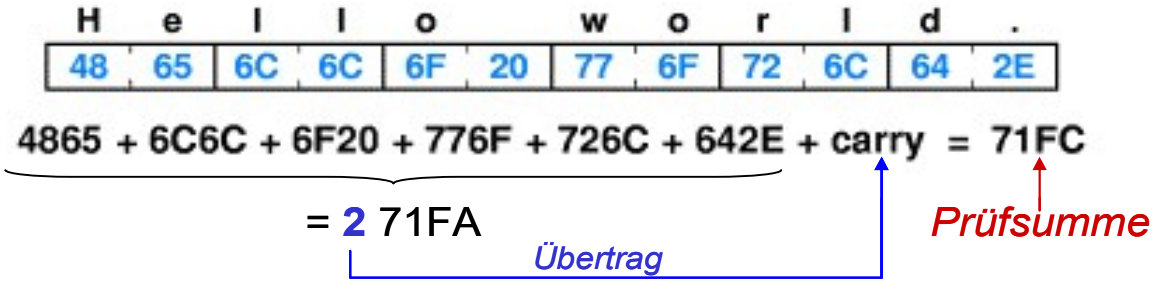
Daten (6 x 7-Bit Zeichen)							EP
0	1	0	1	0	0	1	**1**
1	1	0	1	0	0	1	**0**
1	0	1	1	1	1	0	**1**
0	0	0	1	1	1	0	**1**
0	1	1	0	1	0	0	**1**
1	0	1	1	1	1	1	**0**
1	**1**	**1**	**1**	**0**	**1**	**1**	**0**

zusätzliches Paritätsbyte pro Spalte

Wie groß ist die Hamming-Distanz dieser Codierung?

- *Fehlererkennungscodes…*
  - *Internet-Prüfsumme*
    - *Sender interpretiert Nutzdaten als Folge von Ganzzahlen und berechnet deren Summe*
      - → *Nutzdaten selbst können Zeichen, Gleitkommazahlen, Bilder oder Sonstiges beinhalten*
      - → *Methoden unterscheiden sich durch die Länge der Ganzzahlen zur Summenbildung, z.B. 16-Bit- oder 32-Bit-Prüfsumme*
    - *Beispiel: 16-Bit Prüfsumme, Übertragung von „Hello World“*
      1. *Berechnung der Summe der Ganzzahlen (hexadezimal)*
      2. *Eventuell vorhandene Übertragsbits abschneiden und zum Ergebnis addieren*

- *Fehlererkennungscodes…*
  - *Internet-Prüfsumme…*
    - *Beispiel einer 16-Bit Prüfsumme: Übertragung von „Hello World“*



    - *Fehlererkennungswahrscheinlichkeit des Verfahrens*

      → Besser als Paritätsprüfung; problematisch sind systematische Fehler!

Daten	Dezimal	Daten	Dezimal
00001	1	00011	3
00010	2	00000	0
00011	3	00001	1
00001	1	00011	3
Prüfsumme	7	Prüfsumme	7

- *Fehlererkennungscodes…*
  - *Cyclic Redundancy Check (CRC)*

1. *Nachricht (Nutzdaten) habe Länge von (n+1) Bits, also z.B. 8-Bit Nachricht 10011010 mit n=7*
   - Darstellung der Nachricht als Polynom n-ten Grades M(x):
   $M(X) = 1 \cdot x^7 + 0 \cdot x^6 + 0 \cdot x^5 + 1 \cdot x^4 + 1 \cdot x^3 + 0 \cdot x^2 + 1 \cdot x^1 + 0 \cdot x^0$
2. *Sender und Empfänger einigen sich vor Übertragung auf ein Divisor-Polynom C(x), auch Generator-Polynom genannt, vom Grad k.*
   - z.B. k=3: $C(x) = 1 \cdot x^3 + 1 \cdot x^2 + 0 \cdot x^1 + 1 \cdot x^0$ (entspricht 1101)
3. *Statt M(x) wird eine Nachricht P(X) vom Grad n+k gesendet (entspricht also n+k+1 zu übertragenden Bits)*
   - die zusätzlichen k Bits sind die Fehlererkennungsbits
   - die k Bits werden so gewählt, dass das korrespondierende Polynom P(x) durch C(x) ohne Rest teilbar ist
4. *Empfänger dividiert empfangene Nachricht P(x) durch C(X)*
   - verschwindet Divisionsrest → Datenübertragung erfolgte korrekt
   - die Nachricht besteht aus den höchstwertigen n+1 Bits von P(x)

- *Fehlererkennungscodes…*
  - *Cyclic Redundancy Check (CRC)…*

Beispiel: CRC-Berechnung bei Sender

```
10011010000      /      1101      = 11111001
1101
---------
 1001
 1101
 ---------
  1000
  1101
  ---------
   1011
   1101
   ----------
    1100
    1101
    ---------
     1000
     1101
     ---------------
      101
```

Multiplikation mit x^3

Generatorpolynom $x^3 + x^2 + 1$

101→ Divisionsrest = CRC-Prüfsumme

Divisionsrest ist „101“

Was wird gesendet:

XOR (⇔ Subtraktion)

1 0 0 1 1 0 1 0 0 0 0 ← M(x) • x^3 = T(X)

*0 0 0 0 0 0 0 0 **1 0 1** ← Divisionsrest = R(X)*

1 0 0 1 1 0 1 0 1 0 1 ← zu sendende Nachricht

M(x) R(X)

3.1 Rahmenbildung und Fehlererkennung

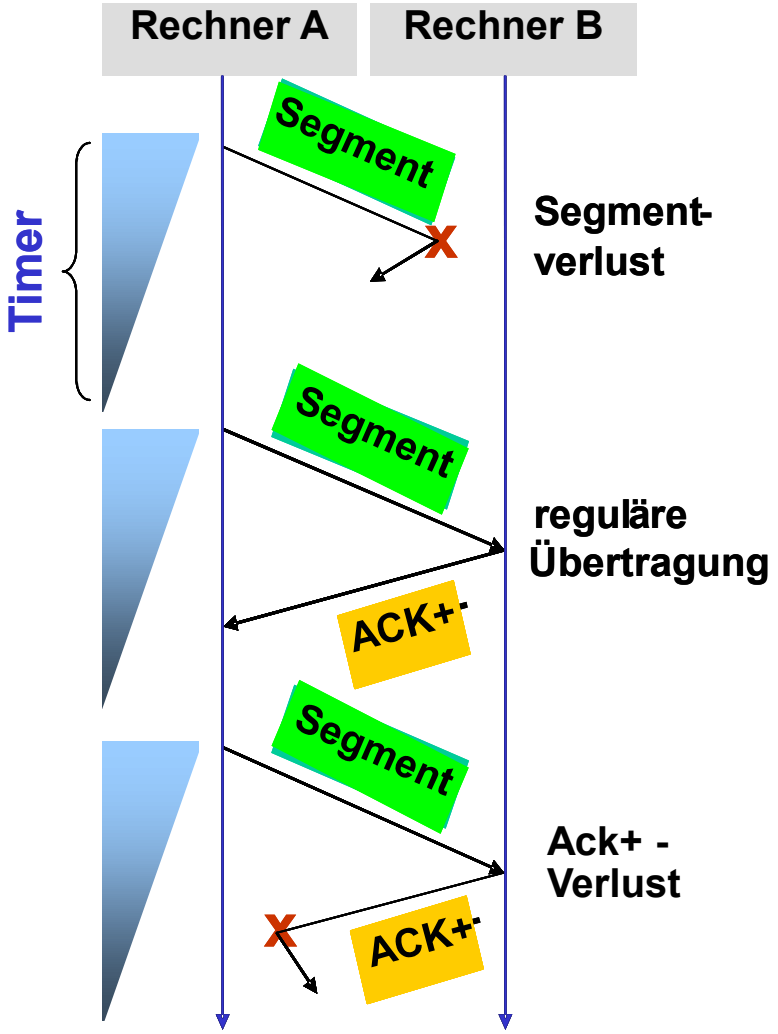
- *Fehlererkennungscodes…*
  - *Cyclic Redundancy Check (CRC)…*
    - *International genormte Generator-Polynome, z.B.*
      - → *CRC-16:* $x^{16} + x^{12} + x^2 + 1$
      - → *CRC-CCITT:* $x^{16} + x^{12} + x^5 + 1$
    - *Beispiel: CRC-16 entdeckt*
      - → *alle Einzel- und Doppelfehler, alle Fehler ungerader Zahl , alle Fehlerbursts mit der Länge <= 16*
      - → *99,997% aller Fehlerbursts mit Länge 17*
      - → *99,998% aller Fehlerbursts mit Länge 18 und mehr*

3.2 Prinzipien der gesicherten Datenübertragung

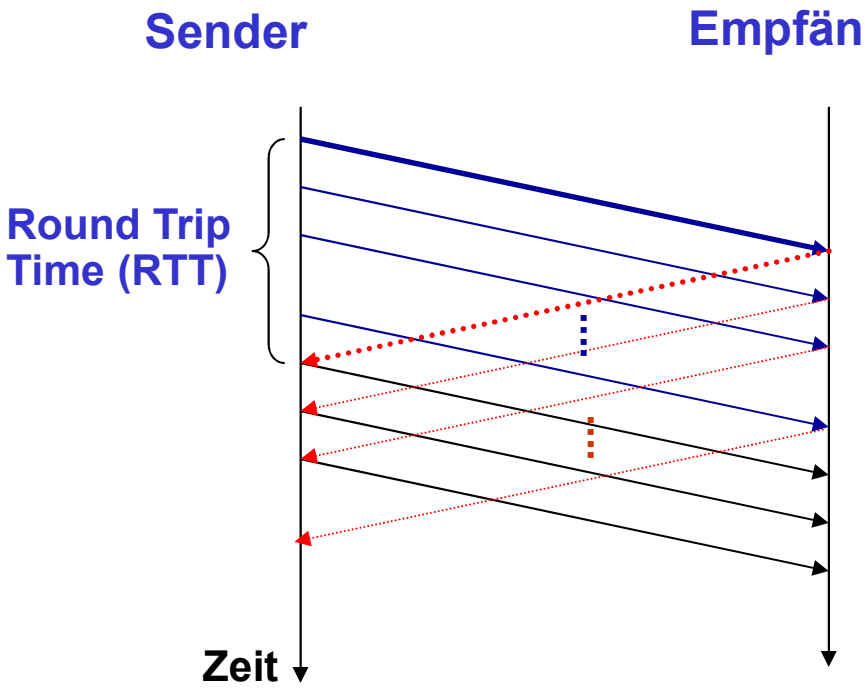
- *Gesicherten Übertragung: Grundprinzip*

[Ref 1] Kapitel 3, Seite 258-271

  - *Prinzip der positiven Bestätigung (ACK+):*
    - *Der erfolgreiche Erhalt eines Datensegments wird mit einem „ACK+“-Paket bestätigt*
    - *Nach Versand eines Datensegments wird eine gewisse Zeit auf das zugeh. ACK+ gewartet*
    - *Falls die Wartezeit überschritten ist, erfolgt eine Sendewiederholung*
  - *Sendepuffer:*
    - *Datensegmente/ACK+ können verzögert werden/verloren gehen!*
    - *Sender muss eine Kopie versandter Daten halten*
  - *Sequenz- und Bestätigungsnummern:*
    - *Datensegmente/ACK+ können verdoppelt werden!*
  - *Problem:*
    - *Je nach Ausbreitungsverzögerung sehr geringe effektive Übertragungsrate*



- *Funktionsweise von Sliding-Window-Protokollen*
  - *Stop-and-Wait: Schlechte Ausnutzung der Übertragungskapazität*
    - *Nur ein Datenrahmen pro ACK wird gesendet*
  - *Jetzt: Sender schickt mehrere Rahmen, ohne ACKs abzuwarten*



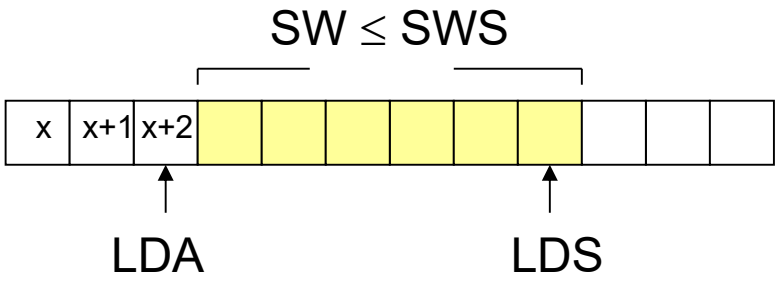
- *Sliding-Window-Protokolle*
  - *Sender verwaltet*
    - *einen Sendepuffer der Größe SWS (Send Window Size)*
    - *eine Variable SW*

- Es muss gelten:

$$SW = LDS - LDA \leq SWS$$

Last Datasegment Sent

Last Datasegment Acknowledged



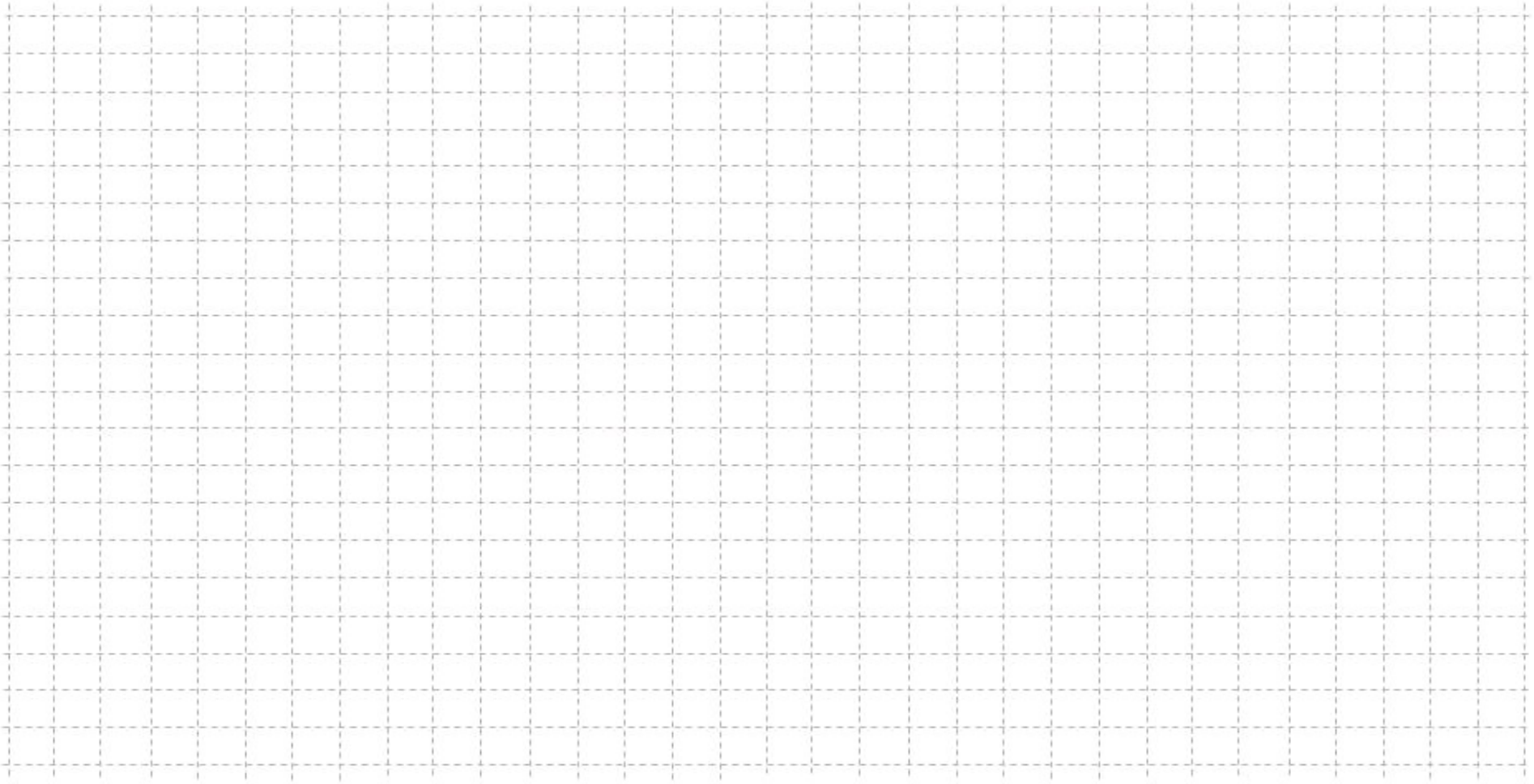
Bezeichnungen:

LDA: Sequenznummer des letzten bestätigten Datensegments

LDS: Sequenznummer des zuletzt gesendeten Datensegments (also noch nicht bestätigt)

3.2 Prinzipien der gesicherten Datenübertragung

- *Sliding-Window-Protokolle…*
  - *Ablaufprinzip*



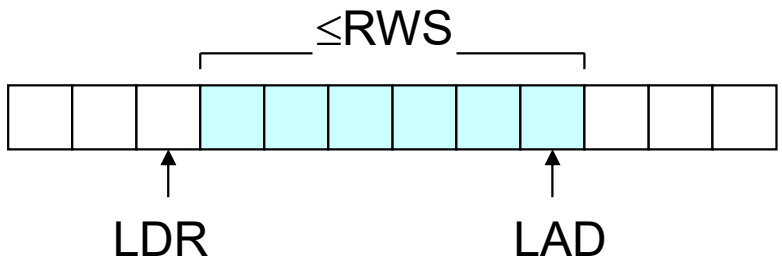
3.2 Prinzipien der gesicherten Datenübertragung

- *Sliding-Window-Protokolle…*
  - *Empfänger verwaltet*
    - *einen Empfangspuffer der Größe RWS (Receive Window Size)*

Es muss gelten:

$$LAD - LDR \leq RWS$$

Largest Acceptable Datasegment — **Last Datasegment Received**



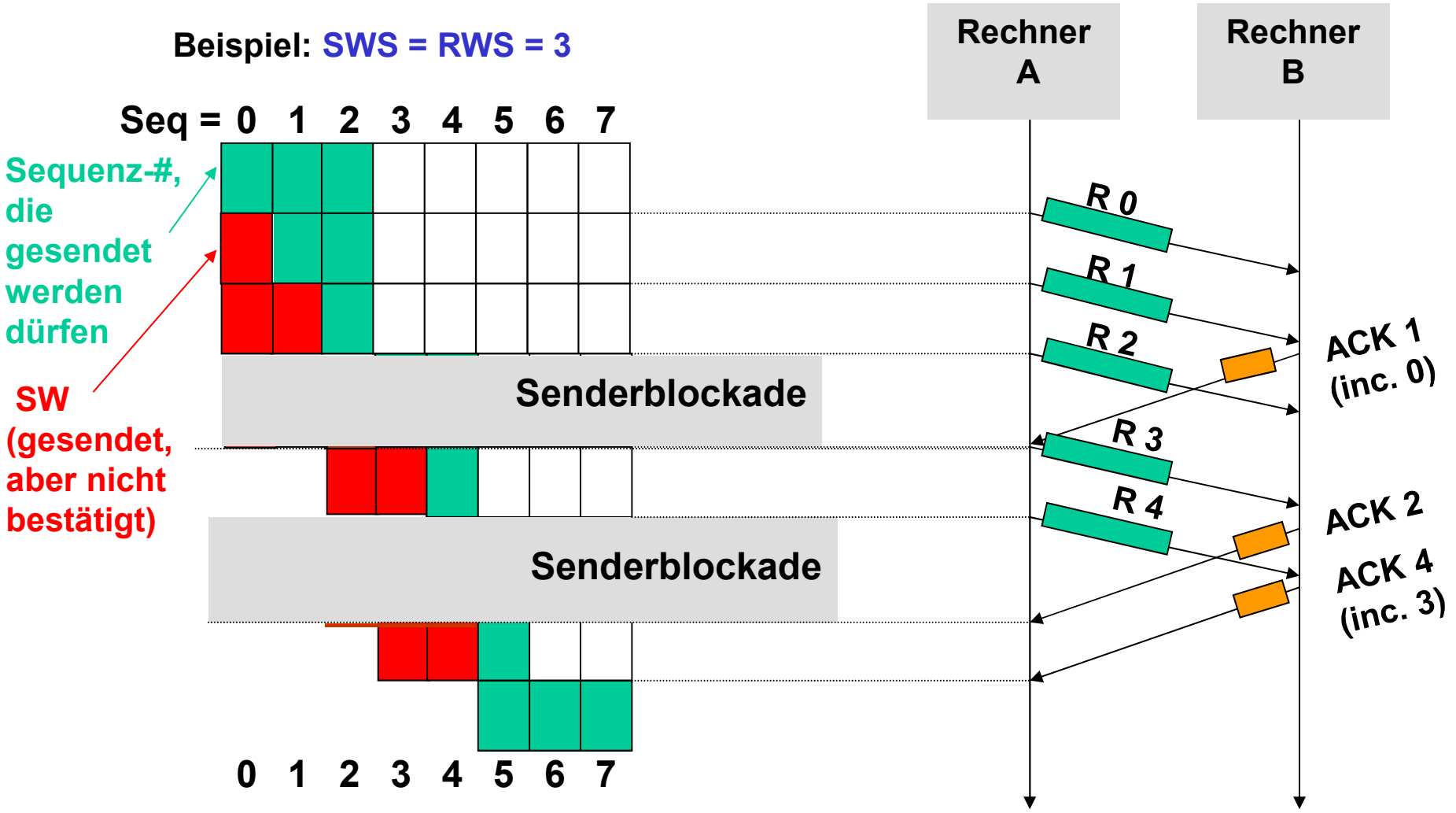
Bezeichnungen:

LAD: größte akzeptierbare Sequenznummer eines Datensegments, die noch im Empfangspuffer aufgenommen werden kann

LDR: Sequenznummer des zuletzt empfangenen und bestätigten Datensegments

3.2 Prinzipien der gesicherten Datenübertragung

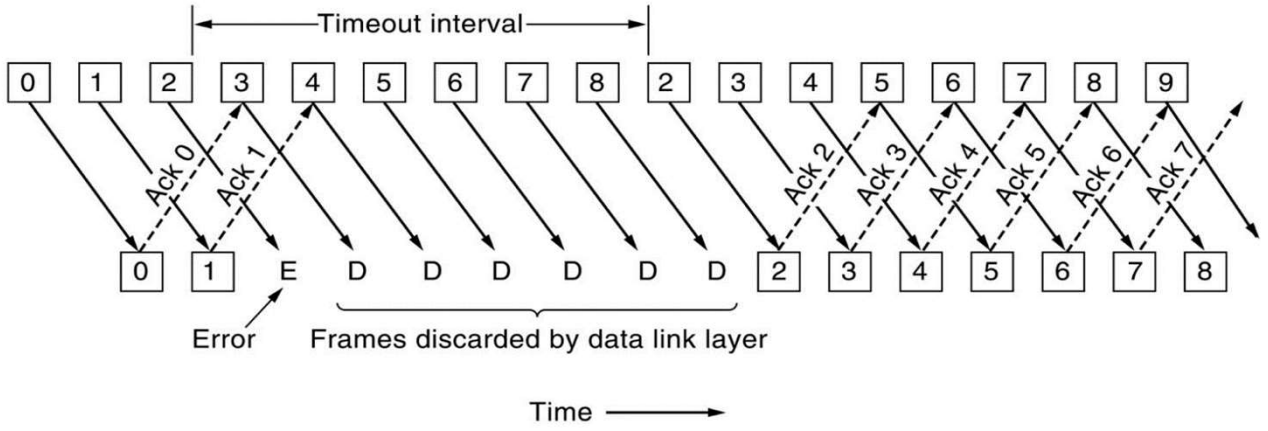
- *Beispiel: Ablauf eines „Sliding Window“-Protokolls*



- *Sliding-Window-Protokolle…*
  - *Quittierung von Datensegmenten*
    - *Kumulatives Acknowledgement*
      - → *ACK für die Sequenznummer n gilt auch als ACK für alle Sequenznummern < n*
    - *Zusätzlich negative Acknowledgements (NACKs) möglich*
      - → *Wenn ein Datensegment mit der Sequenz n empfangen wird, aber noch die Sequenznummer m mit m<n aussteht, wird für die Sequenznummer m ein NACK geschickt*

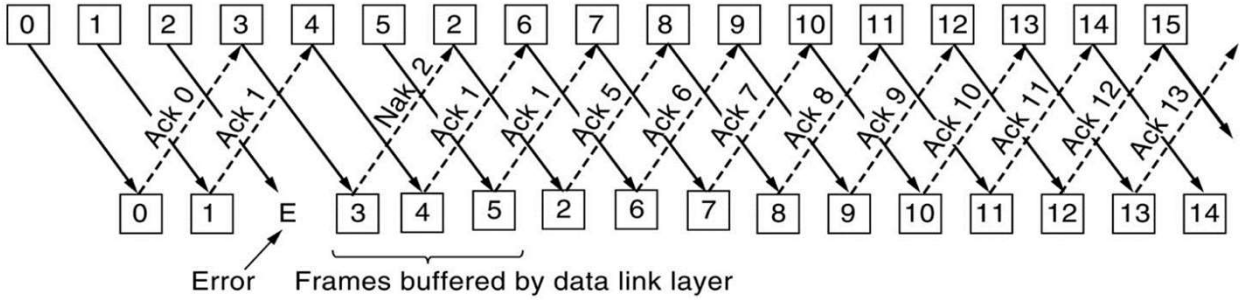
- *Varianten von Sliding-Window-Protokollen*
  - *„Go-Back-n“-Strategie: RWS = 1*
    - *Empfangspuffer des Empfängers kann genau ein Datensegment zwischen puffern*
      - → *Sender versendet nach einer Neuübertragung des Datensegments m auch alle Datenrahmen m+1, m+2, … erneut*
      - → *Empfänger verwirft auch alle Datensegmente m+1, m+2, … nach einem verlorenen oder verfälschten Datensegment m*
  - *„Selective Repeat“-Strategie: RWS > 1*
    - *Empfangspuffer des Empfängers kann mehrere Datenrahmen zwischen puffern*
      - → *Nur ein verlorener/verfälschter Datenrahmen wird neu übertragen*
      - → *Neuübertragung wird beim Sender angestoßen durch ein Timeout-Ereignis oder den Empfang eines NACKs*
      - → *Empfänger puffert alle Datenrahmen, die in sein Empfangsfenster passen. Er bestätigt immer die letzte erfolgreiche Übertragung vor dem Fehler.*

- *Auswirkung eines Fehlers beim „Go-Back-n“-Verfahren*



(a)

- *Auswirkung eines Fehlers beim „Selective-Repeat“-Verfahren*



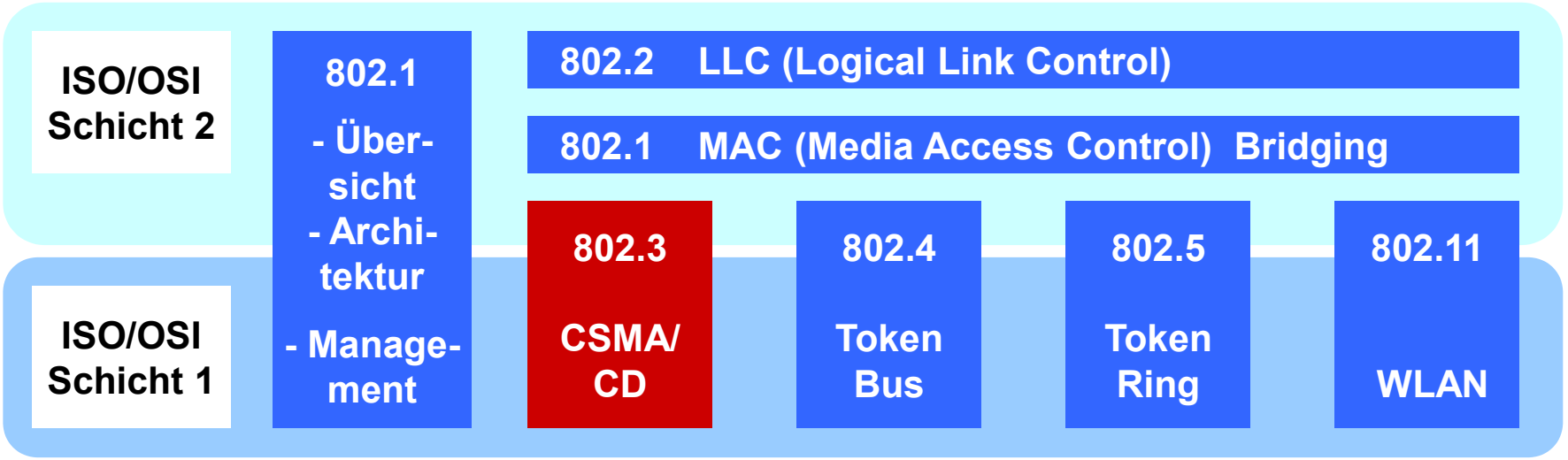
(b)

- *Sliding-Window-Protokolle…*
  - *Wie viele Sequenznummern werden gebraucht?*
    - *Problem: Datensegment kann nur eine begrenzte Anzahl von Bits für die Sequenznummer speichern*
      - → *Z.B. bei 3 Bits sind nur die Sequenznummern 0,1,…,7 möglich*
      - → *MaxSeqNum steht für die Anzahl der verfügbaren Sequenznummern: MaxSeqNum = 8*
  - *Reicht ein endlicher Bereich an Sequenznummern aus?*
    - *Ja, wenn SWS entsprechend gewählt wird:*

Falls RWS = 1:	$SWS \leq MaxSeqNum - 1$
Falls RWS = SWS:	$SWS \leq (MaxSeqNum + 1) / 2$

3.3 Ethernet (IEEE 802.3)

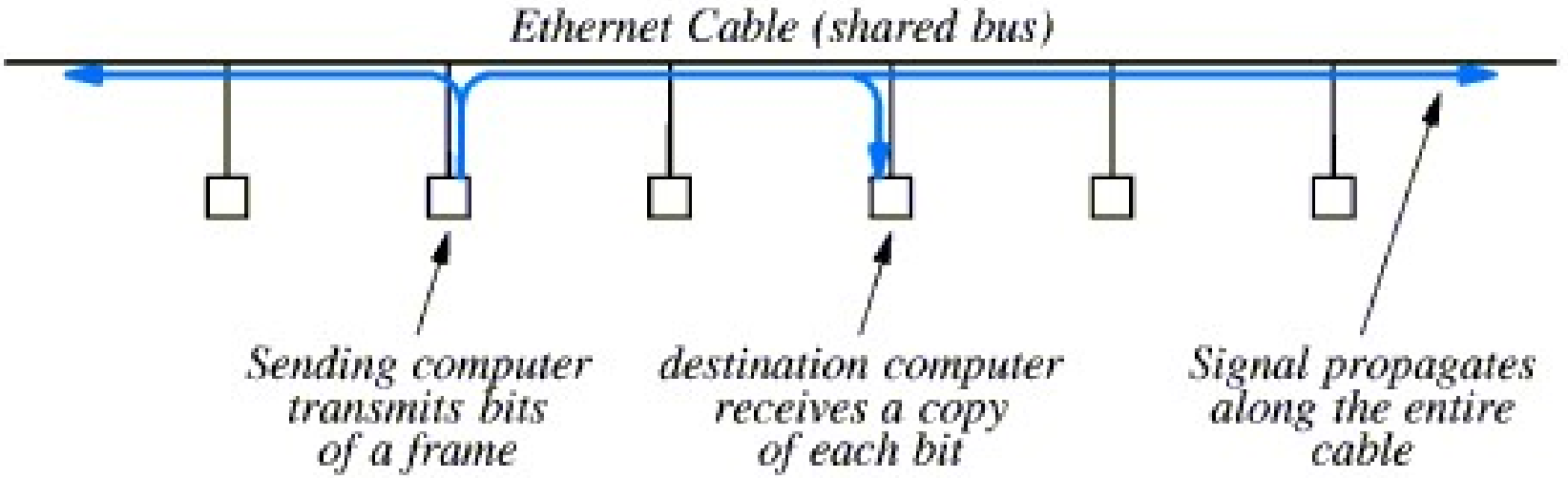
- *Standardisierung nach IEEE 802.x*



- *802.1 : Zusammenhang der Standards und MAC Bridging*
- *802.2 : Logical Link Control-Dienste und -Protokolle*
- ***802.3 : CSMA/CD-Protokoll für Bus-Topologie (→ Ethernet)***
- *802.4 : Token Bus-Protokoll auf Bus-Topologie*
- *802.5 : Token Ring-Protokoll auf Ring-Topologie*
- *802.11: Wireless LAN*
- *802.15(.4): Wireless Personal Area Networks (Zigbee)*

3.3 Ethernet (IEEE 802.3)

- *Ethernet-Funktionsprinzip*
  - *Alle Teilnehmer eines LANs teilen sich die Übertragungskapazität des physikalischen Übertragungswegs: → **„shared network“***
  - *Alle Stationen „sehen“ alle Daten-Rahmen im LAN*
    - *Jeder Rahmen erreicht prinzipiell jede Station!*
    - *Der physikalische Übertragungsweg ist ein **„Broadcast-Kanal“**!*



3.3 Ethernet (IEEE 802.3)

- *CSMA/CD: Medienzugriffsprotokoll für Ethernet*

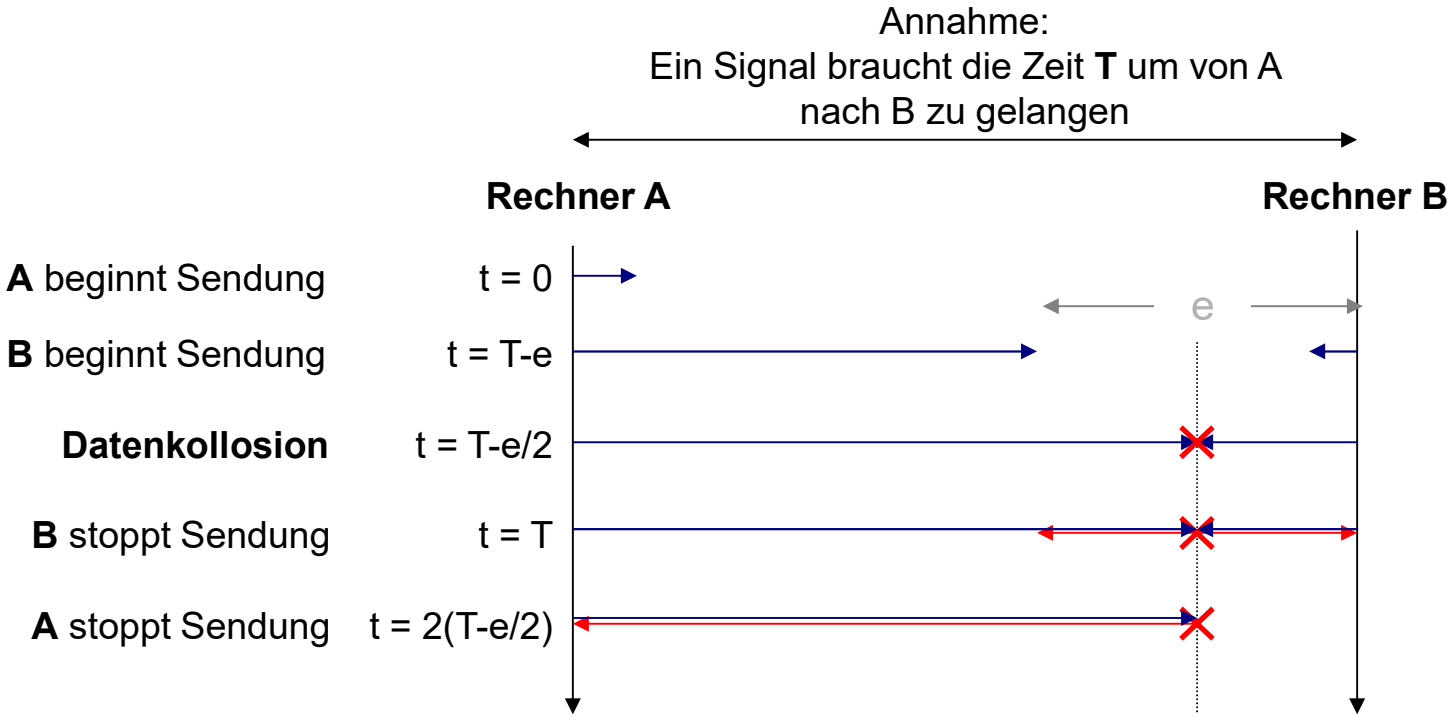
[Ref 1] Abschnitt 5.5.2, Seite 492-499
[Ref 2] Abschnitt 4.3, Seite 328-336
[Ref 3] Kapitel E, Seite 140-149

  - *Fragestellung:*
    - *Welche Station darf wann auf einem Ethernet-Segment senden?*
  - *Lösung: das CSMA/CD-Protokoll*
    - *CSMA/CD = Carrier Sense Multiple Access/Collision Detect*
  - *CSMA/CD-Ablauf*
    - *Sendewillige Station hört Leitung ab*
    - *Bei freier Leitung wird gesendet*
    - *Während der Sendung wird weiter überwacht, ob Datenkollision auftritt*
      - *→ Problem: Laufzeitverzögerung der Signale*
    - *Falls Kollision auftritt:*
      - *→ Beteiligte Stationen brechen Sendung ab und senden ein JAM-Signal*
      - *→ Danach warten alle sendewilligen Stationen eine zufällige Zeitspanne lang für den nächsten Sendeversuch ab → „Exponential Backoff“-Algorithmus*

3.3 Ethernet (IEEE 802.3)

- *Der ”Binary Exponential Back-off” -Algorithmus* [Ref 3] Kapitel E, Seite 160

- *CSMA/CD: Zeitlicher Ablauf bei einer Datenkollision…*



  - *Betrachten Grenzfall e→0*
    - *A muss mindestens 2·T lange senden damit A die Datenkollision sicher erkennen und auf die eigene Datensendung zurückführen kann!*

- *CSMA/CD: Konfliktparameter K*

$$K = \frac{\text{doppelte max. Signallaufzeit}}{\text{min. Nachrichtenübertragungsverzögerung}}$$

entspricht RTT_{max}!

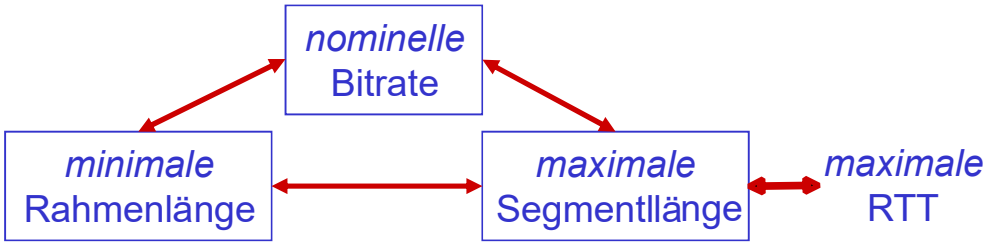
$$\text{min. Nachrichtenübertragungsverzögerung} = \frac{\text{min. Nachrichtenlänge [bit]}}{\text{Übertragungsrate [bps]}}$$

  - *K>1*
    - *Komplette Nachricht kann gesendet werden, bevor Kollision erkannt wird*
    - *Für CSMA/CD nicht praktikabel !*
  - *K≤1*
    - *Für CSMA/CD praktikabel*
    - *Daraus resultiert Limitierung der Kabel-Länge*

3.3 Ethernet (IEEE 802.3)

- *Beispiel: CSMA/CD: Konfliktparameter K*

- *Definitionen*
  - *„Kollisionsdomäne“*
    - *Menge von Rechnern, die um den Zugriff auf das Übertragungsmedium (Netzsegment) konkurrieren*
      - → *Für diesen Bereich gelten die Einschränkungen des CSMA/CD-Protokolls*
      - → *Bridges & Switches (Geräte der Sicherungsschicht, OSI-Ebene 2) entkoppeln Kollisionsdomänen*



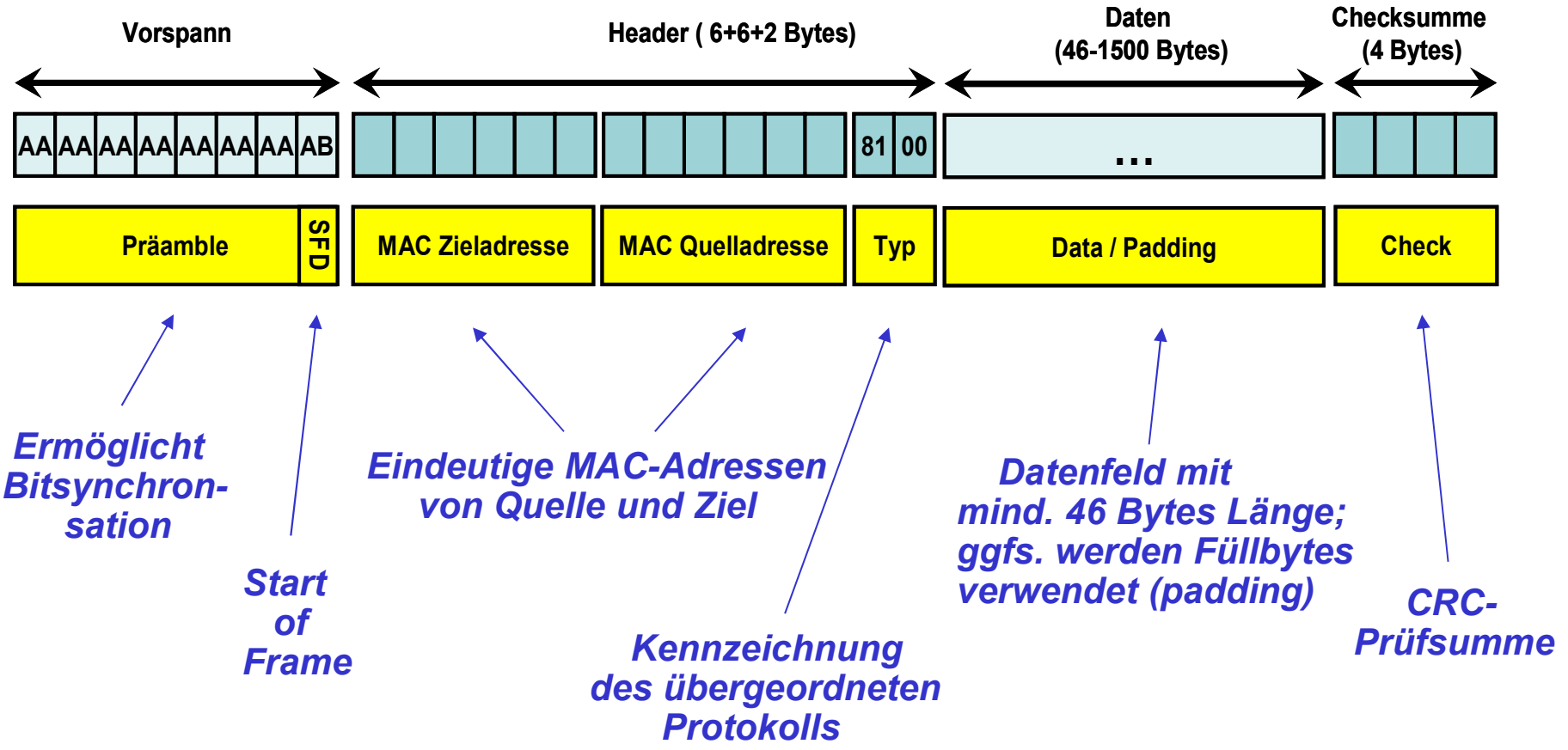
  - *„Broadcast-Domäne“*
    - *„Sichtbarkeitshorizont“ von Ethernet-Rahmen, die an die Ziel-MAC-Adresse ff:ff:ff:ff:ff:ff (Broadcast-Adresse) geschickt werden*
      - → *Router (Geräte der Vermittlungsschicht, OSI-Schicht 3) entkoppeln Broadcast-Domänen*

- *MAC(Ethernet)-Adressen*
  - *Länge: 6 Bytes bzw. 48 Bits*
    - *Übliche Notation:*
      - → *Folge von 6 durch „:“ getrennte Zahlen*
      - → *Jedes Byte wird durch zwei hexadezimale Zahlen dargestellt*
      - → *Führende Nullen werden weggelassen*
  - *Beispiel:*
    0000 1000 0000 0000 0010 1011 1110 0100 1011 0001 0000 0010
    entspricht 8:0:2b:e4:b1:2
    - *Die ersten 3 Bytes werden „Präfix“ genannt*
      - → *Hersteller kauft Adressblock (durch IEEE verwaltet) mit jew. 2^{24}=16.777.216 möglichen Adressen*
      - → *Hersteller muss innerhalb des Blockes Eindeutigkeit sicher stellen*
  - *Broadcast-Adresse*
    - *Alle Bits der LAN-Adresse auf sind 1 gesetzt: ff:ff:ff:ff:ff:ff*
      - → *Alle Ethernet-Adapter im Netz akzeptieren Rahmen mit dieser Ziel-MAC-Adresse und leiten sie weiter*

3.3 Ethernet (IEEE 802.3)

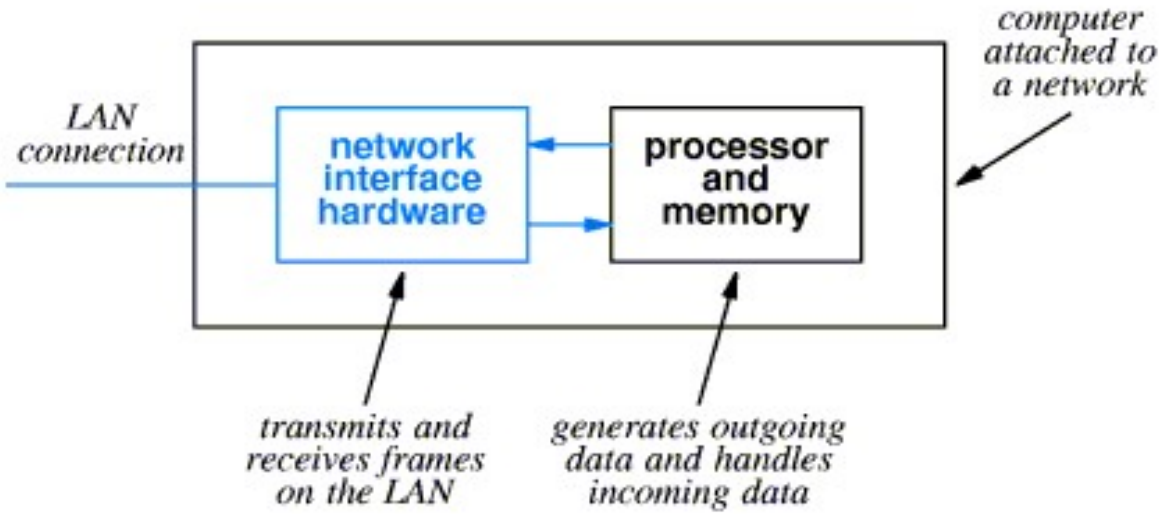
- *Das Ethernet-Rahmenformat*

[Ref 1] Abschnitt 5.5, Seite 509- 513
[Ref 3] Kapitel E, Seite 152-156



- *Funktion eines Ethernet-Adapters* [Ref 3] Kapitel E, Seite 156-157

  - *Der Ethernet-Adapter überprüft jeden gesendeten Rahmen*
    - *Diese Funktion übernimmt die Hardware des Adapters*
      - → *das Betriebssystem des Rechners merkt nichts davon!*
    - *Falls Ziel-Mac-Adr. eines Rahmens = lok. MAC-Adr. des Adapters*
      - → *Rahmen wird an das Betriebssystem weitergeleitet*
      - → *Ausnahmen: Promiscuous Mode und MAC-Broadcast*



3.3 Ethernet (IEEE 802.3): Netzwerkkomponenten

- *Netzkomponenten: Einordnung in das ISO/OSI-Modell*

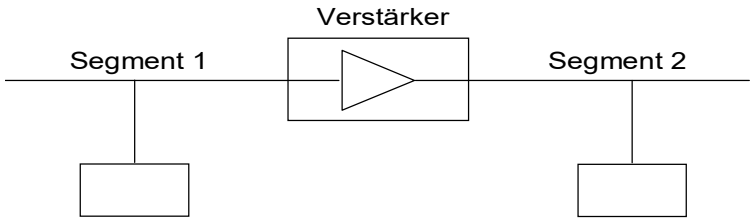
- *Application-Level-Gateway*
- *Circuit-Level-Gateway*
- *Paket-Filter*
- *Router*
- ***Bridge/Switch***
- ***Repeater/Hub***

3.3 Ethernet (IEEE 802.3): Netzwerkkomponenten

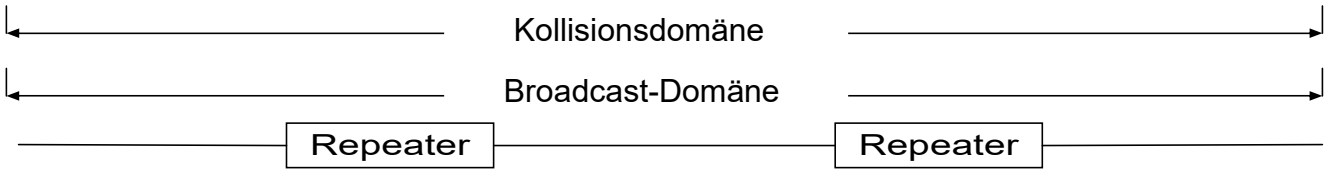
- *Ethernet-Repeater*

[Ref 1] Abschnitt 5.5, Seite 518
[Ref 3] Kapitel E, Seite 161-162

  - *Zweck*
    - *Längenbeschränkung von Ethernet-Segmenten auf Grund der Signaldämpfung aufheben: „Cheapernet“: 185m; TP-Kabel: 100m*
    - *Kopplung gleichartiger Netzsegmente*
  - *Funktion*
    - *Arbeitet auf der Bitübertragungsschicht*
    - *Ersatzschaltbild:*

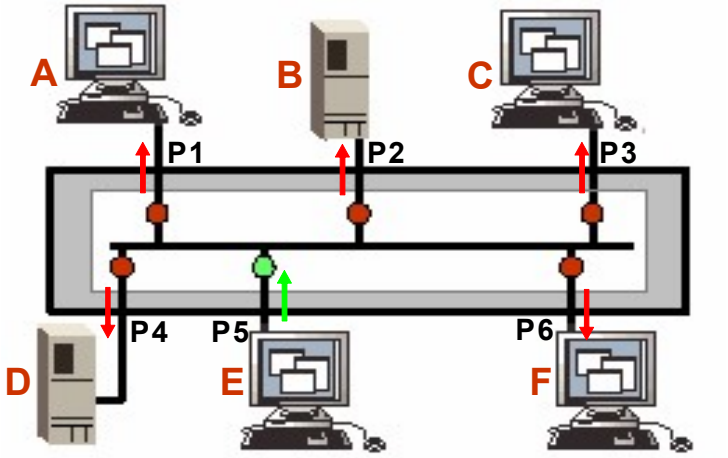


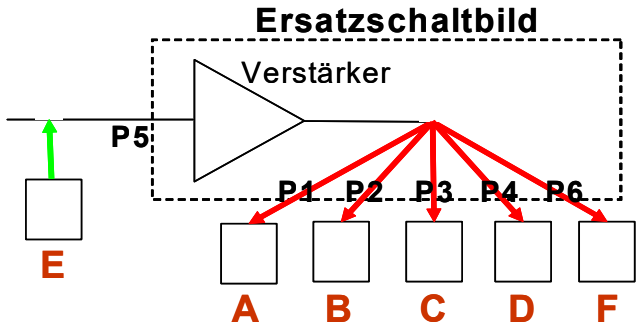
    - *Entstehendes Gesamtnetz bildet wieder eine Kollisions-Domäne!*



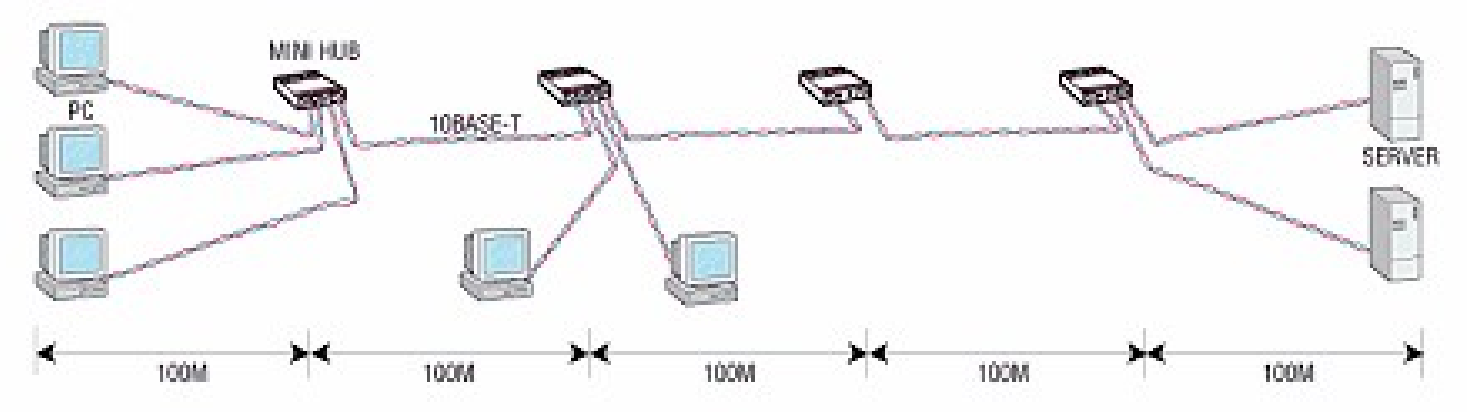
- *Ethernet-Hubs*
  - *Zweck ähnlich wie der von Repeatern*
    - *Abgrenzung: Hubs bilden den zentralen Bus eines Ethernet-Segments, Repeater koppeln Segmente*
  - *Funktion*
    - *Hubs arbeiten (wie Repeater) auf der Bitübertragungsschicht*
    - *Entstehendes Gesamtnetz bildet eine Kollisionsdomäne!*

      → *Auch Gesamtdurchsatz des Netzes bleibt 10 Mbps oder 100 Mbps*





- *„Repeater“-Regel*
  - *Anzahl der kaskadierbaren Hubs/Repeater ist begrenzt*
    - *Grund: Addition der Laufzeiten vs. CSMA/CD-Limitierungen*
    - *Maximal können 5 Segmente durch 4 Repeater/Hubs hintereinander geschaltet werden (nur 3 Segmente mit Anbindung von Rechnern)*

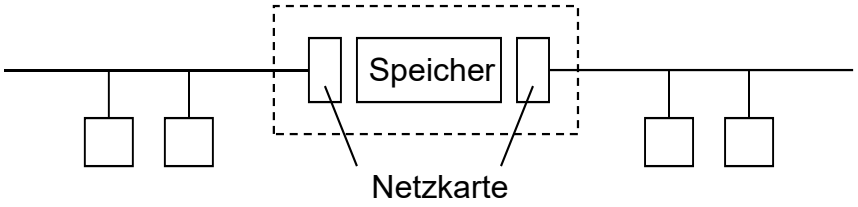


- *Ethernet-Bridges*

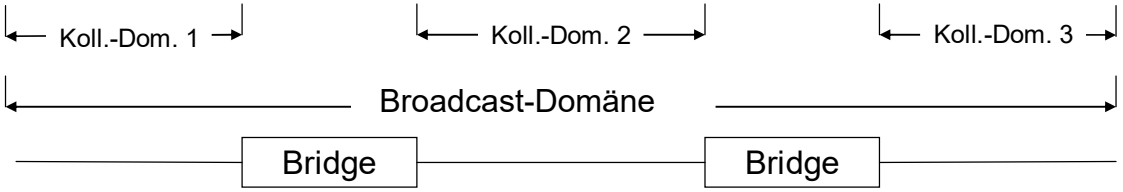
[Ref 1] Abschnitt 5.6, Seite 520-531
[Ref 3] Kapitel E, Seite 163-170

  - *Kopplung zweier Ethernet-Segmente mit folgenden Eigenschaften*
    - *Geschwindigkeitskonversion*
    - *Aufhebung der Repeater-Regel*
  - *Funktion*
    - *Bridges sind Geräte der ISO/OSI-Schicht 2*

      → *Implementieren Funktionen der Bit- & Sicherungsschicht*
    - *Ersatzschaltbild:*



    - *Bridges entkoppeln Kollisionsdomänen*



3.3 Ethernet (IEEE 802.3): Netzwerkkomponenten

- *Beispiel: Kollisions- und Broadcast-Domänen*

3.3 Ethernet (IEEE 802.3): Netzwerkkomponenten

- *„Multiport“-Bridge*
  - *Bridge mit mehr als zwei LAN-Schnittstellen*
    - *Vorläufer der Ethernet-Switches*
  - *Bridge-Tabelle*
    - *Enthält Informationen zur Filterung und Weiterleitung von Rahmen*
    - *Pro Eintrag wird gespeichert:*
      - → *Die MAC-Adresse eines angeschlossenen Rechners*
      - → *Die Schnittstelle (bzw. die Port-Nr.) des angeschlossenen Rechners*
      - → *Der Zeitstempel, wann ein Eintrag zum letzten mal genutzt wurde*

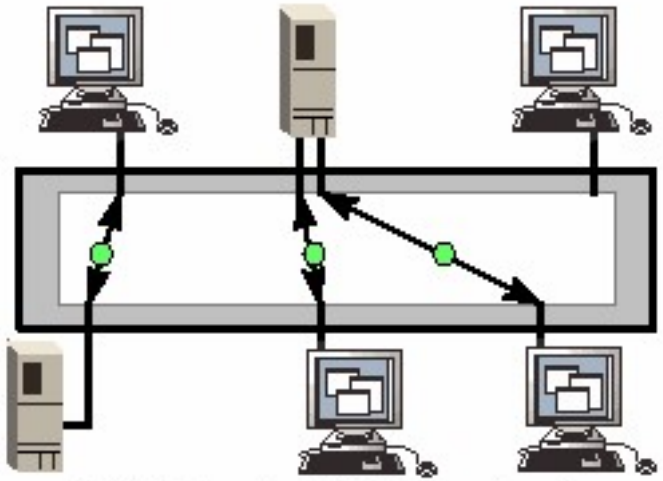
MAC-Adresse	Schnittstelle	Zeitstempel
7C-BA-B2-B4-91-10	1	9:32
62-FE-F7-11-89-A3	2	9:36
.....	.....	.....

- *Selbstlernende Bridges*
  - *Bridge „lernt“ die Bridge-Tabelleneinträge selbstständig*
  - *Lernverfahren:*
    - *1) Bridge wird eingeschaltet: Tabelle ist leer*
    - *2) Die Quell-MAC-Adresse und Eingangsschnittstelle für jeden einkommenden Rahmen wird in Tabelle aufgenommen*
    - *3) Falls die Ziel-MAC-Adresse noch nicht bekannt ist: Rahmen wird an alle anderen Schnittstellen weitergeleitet: „Fluten“*
    - *4) Einträge altern aus (Zeitstempel)*
      - *→ Falls Eintrag für eine bestimmte Zeitspanne ungenutzt bleibt, wird Eintrag gelöscht*
      - *→ Falls Tabelle voll ist, wird der älteste Eintrag gegen die aktuelle Neuinformation ersetzt*

3.3 Ethernet (IEEE 802.3): Netzwerkkomponenten

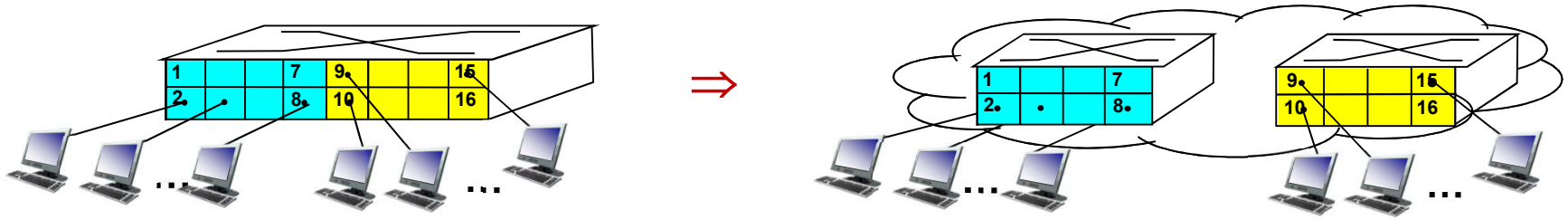
- *Beispiel: Selbstlernende Bridges*

- *Ethernet-Switches*
  - *Funktion*
    - *Ähnlich wie die der Multiport-Bridge, jedoch geringere Durchlaufverzögerung, mehr Ports und höherer Gesamtdurchsatz*
    - *Ebenfalls Geräte der ISO/OSI-Schicht 2*
      - → *Filter- und Weiterleitungsprinzip wie bei Bridges*
      - → *Die an die Switch-Ports angeschlossenen Ethernet-Segmente bilden entkoppelte Kollisionsdomänen*
      - → *Das Gesamtnetz bildet jedoch immer noch eine Broadcast-Domäne!*



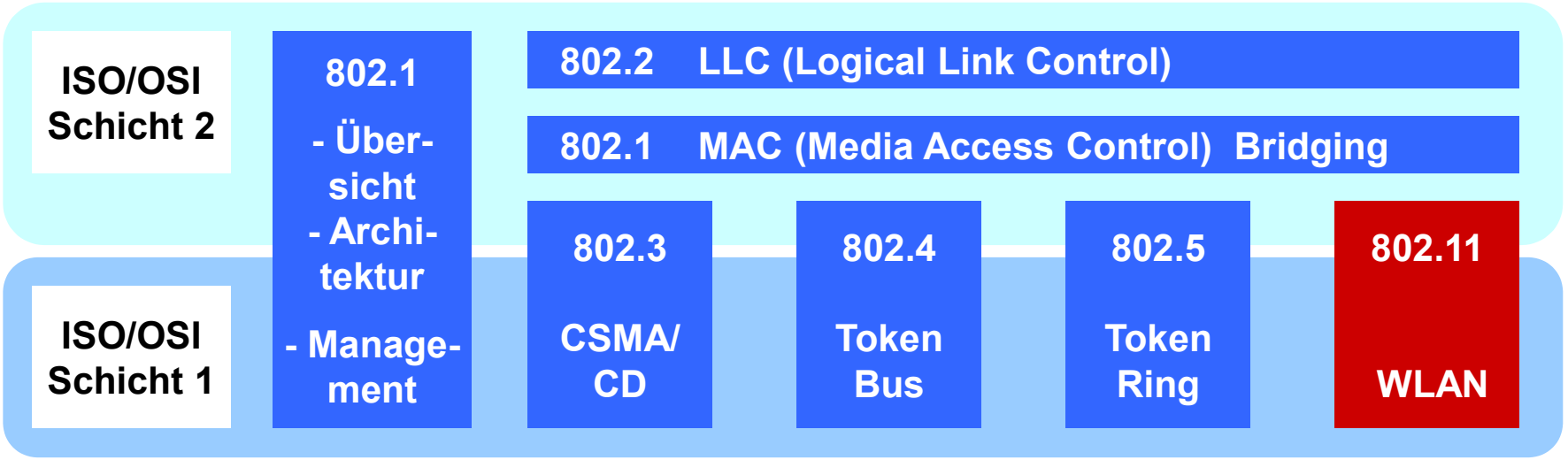
Ein Switch trennt alle Rechner voneinander (Microsegmentierung) und verhindert dadurch das Auftreten von Kollisionen.

- *Virtual LANs (VLANs)*
  - *Ziel*
    - *Ein Switch verwaltet mehrere unabhängige Broadcast-Domänen*
      - → *Angeschlossene Rechner werden durch Konfiguration in verschiedene Gruppen, sogenannte VLANs eingeteilt*
      - → *Eine Kommunikation zwischen zwei unterschiedlichen VLANs ist nur über einen zwischengeschalteten Router möglich*
      - → *Die Organisationsstruktur einer Firma kann leicht abgebildet werden*
  - *Beispiel: Port basierte VLANs*
    - *Switch-Ports werden durch Konfiguration in Gruppen eingeteilt, die dann wie separierte logische Switches arbeiten*



3.4 Wireless LAN (IEEE 802.11)

- *Standardisierung nach IEEE 802.x*



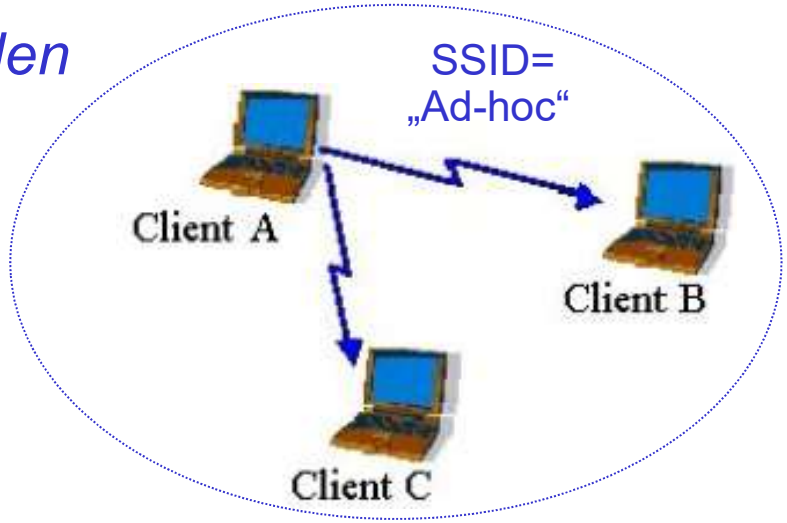
- *802.1 : Zusammenhang der Standards und MAC Bridging*
- *802.2 : Logical Link Control-Dienste und -Protokolle*
- *802.3 : CSMA/CD-Protokoll für Bus-Topologie (→ Ethernet)*
- *802.4 : Token Bus-Protokoll auf Bus-Topologie*
- *802.5 : Token Ring-Protokoll auf Ring-Topologie*
- ***802.11: Wireless LAN***
- *802.15(.4): Wireless Personal Area Networks (Zigbee)*

3.4 Wireless LAN (IEEE 802.11)

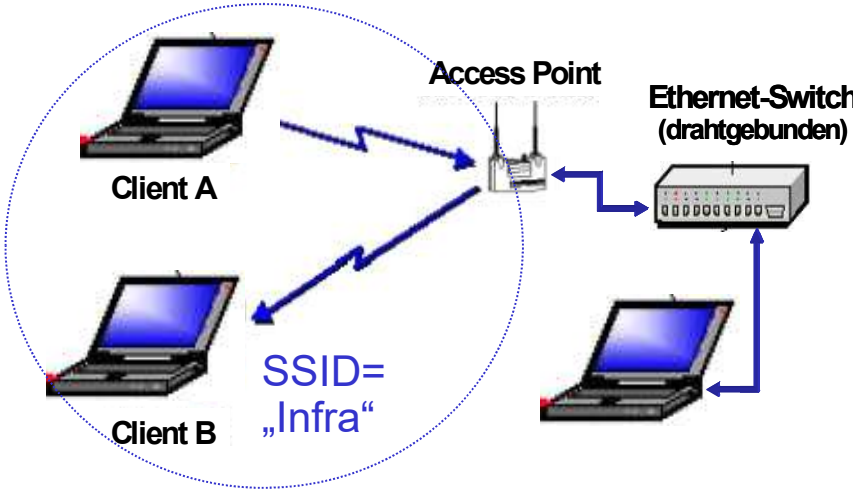
- *„Wireless LAN“: Datenfunk nach den IEEE 802.11-Standard*
  - *Nutzt lizenzfreies 2,4 GHz*) bzw. 5 GHz**) –Band*
  - *Erreichbare Datenraten*
    - *IEEE 802.11b*) → bis 11Mbps (eff. 5-6Mbps)*
    - *IEEE 802.11g*) → bis 54Mbps (eff. 16-18Mbps)*
    - *IEEE 802.11a**) → bis 54Mbps (eff. 16-18Mbps)*
      - *→ Hauptsächlich in den USA eingesetzter Standard*
    - *IEEE 802.11n*,**) → bis 600Mbps (eff. ca. 250 Mbps)*
    - *IEEE 802.11ac **) → bis 1.700Mbps (eff. ca. 800 Mbps)*
  - *Erreichbare Reichweiten*
    - *Ca. 30-50m innerhalb von Gebäuden, bis ca.1km außerhalb*
      - *→ Mit entsprechenden Antennen jedoch erweiterbar*
  - *Einsatzgebiet*
    - *„Wireless LANs“ → drahtlose Netze*
    - *„Wireless Bridging“ → Kopplung drahtgebundener Netze über Funk*

3.4 Wireless LAN (IEEE 802.11)

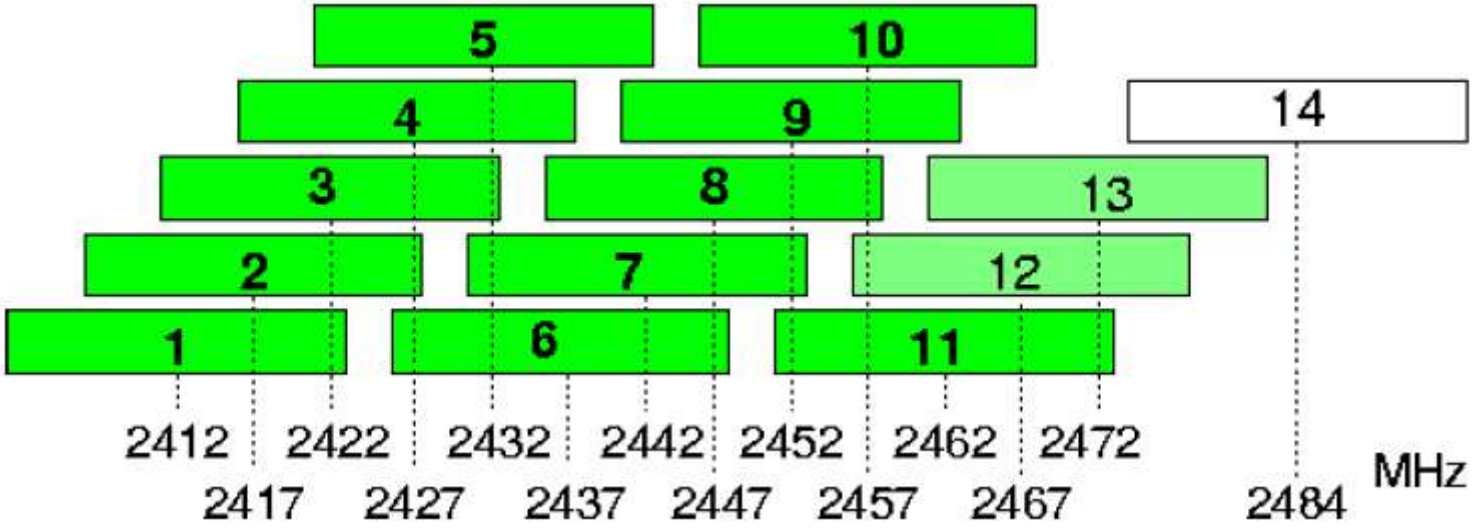
- *802.11-Funknetze: WLAN-Betriebsmoden*
  - *Ad hoc-Modus*
    - *Direkter Verbindungsaufbau zwischen WLAN-Knoten*
    - *Knoten müssen die gleiche Übertragungskanal-Nr. und die gleiche SSID (Funknetz-kennung) verwenden*
  - *Infrastruktur-Modus*
    - *WLAN-Clients kommunizieren indirekt über Access Point (AP)*
    - *Access Point wirkt wie eine Bridge zwischen Funknetz und drahtgebundenem Netz*
    - *Über Broadcast(Beacon)-Meldungen gibt AP Funknetz-Parameter bekannt (z.B. Kanal-Nr, AP-MAC-Adr., optional SSID…)*



[Ref 1] Abschnitt 6.1, Seite 560-563

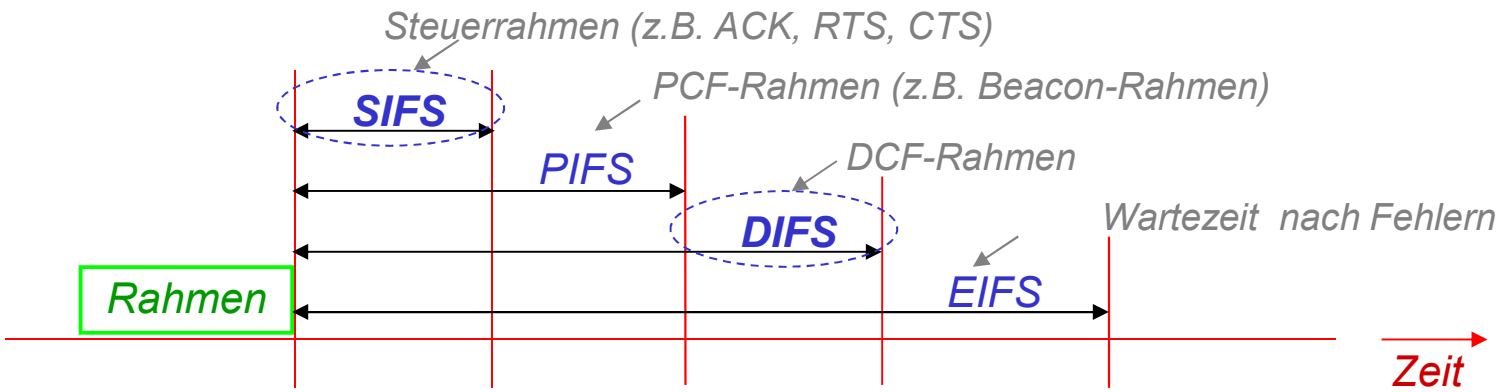


- *Nutzbare Frequenzbänder für IEEE 802.11b/g*
  - *13 überlappende Frequenzbänder (in Europa)*
    - *Maximal 3 Kanäle können überlappungsfrei genutzt werden!*
    - *Standard-Kanal-Nummern: 1,6,11*
  - *Überlappungen führen zu Störungen und Bitrateneinbußen*
    - *Nie benachbarte APs mit benachbarten Kanalnummern betreiben!*



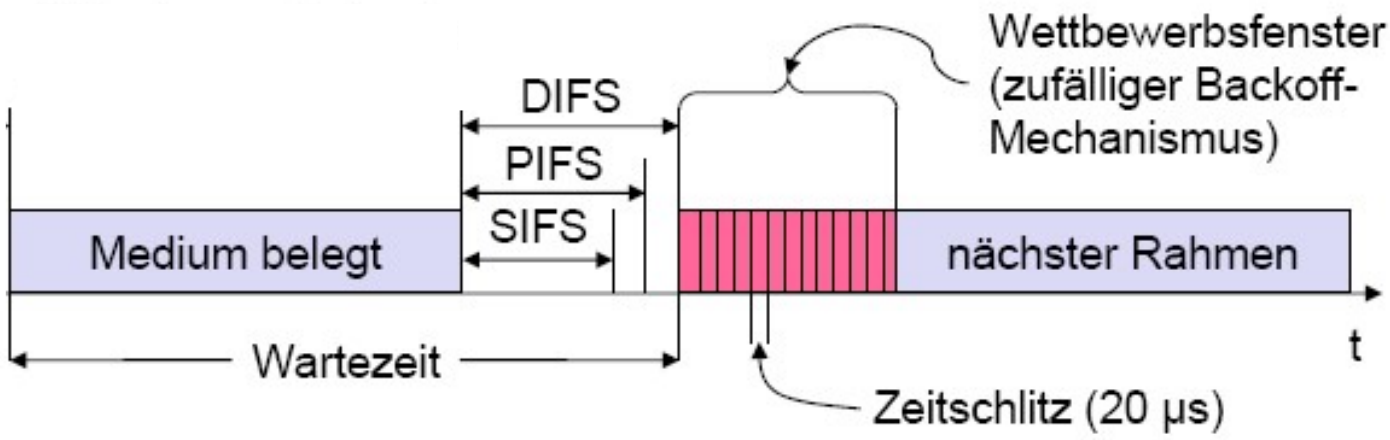
- *CSMA/CA: Medienzugriffsprotokoll für „Wireless LAN“*
  - *Problem: CSMA/CD (wie bei Ethernet) nicht nutzbar*
    - *Besondere Charakteristiken von Funknetzen*
    - *Funkschnittstellen arbeiten im Half-Duplex-Mode!*
      - → *Während des Sendens ist kein Mithören möglich*
      - → *Nur der Empfänger kann Kollisionen erkennen (z.B. durch Prüfsummen)*
  - *CSMA/CA = Carrier Sense Multiple Access / Collision Avoidance*
    - *Methode, um Kollisionen möglichst zu vermeiden*
      - → *Kollisionen sind aber auch hier immer noch möglich*
    - *Unterschiede zu CSMA/CD*
      - → *Keine Kollisionserkennung beim Senden:*
        Empfänger muss jeden empfangenen Daten-Rahmen mit Acknowledgement(ACK)-Rahmen bestätigen (nach erfolgtem CRC-Check)
      - → *Immer vor dem Senden muss der Übertragungskanal für eine bestimmte Mindestzeit abgehört und als frei erkannt werden*
        → Interframe Spacing (IFS)-Zeiten

- *CSMA/CA: Medienzugriffsprotokoll für „Wireless LAN“…*
  - *Interframe Spacing (IFS)-Zeiten*
    - *Mindestzeit, die ein Sender den Übertragungskanal abhören und als frei erkannt haben muss*
    - *Erst danach darf gesendet werden*
    - *Unterschiedliche IFS-Zeiten wurden für unterschiedliche Rahmen-Typen festgelegt → Priorisierung bestimmter Rahmentypen*

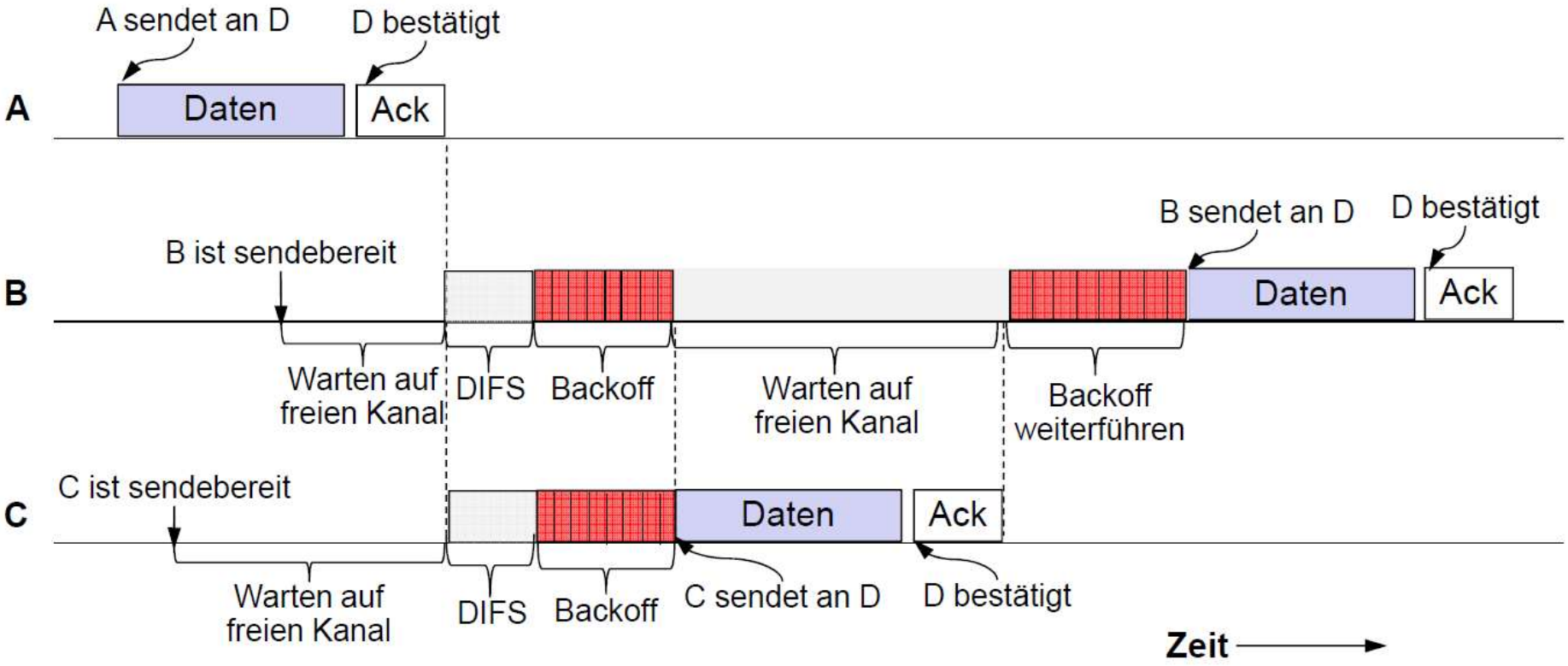


IEEE 802.11b	10µs	30µs	50µs	364µs
IEEE 802.11g	16µs	25µs	34µs	186µs

- *CSMA/CA: Medienzugriffsprotokoll für „Wireless LAN“…*
  - *Sendewillige Station hört Medium ab*
    - *Ist das Medium für die Dauer einer „Interframe Spacing (IFS)“-Zeit frei, darf gesendet werden*
    - *Ansonsten wird auf eine freie IFS-Zeit gewartet und zusätzlich um eine zufällige Backoff-Zeit verzögert*



- *CSMA/CA: Medienzugriffsprotokoll für „Wireless LAN“…*
  - *Beispiel mit drei sendebereiten Stationen A-C*

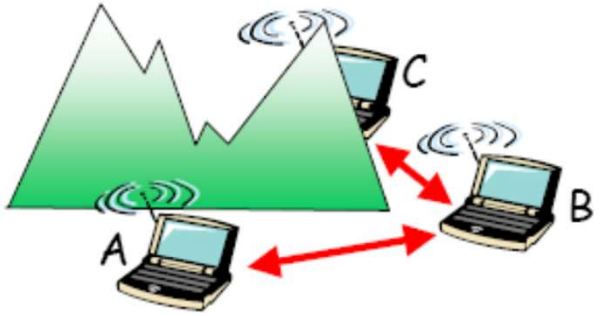


- *Besondere Charakteristiken von Funknetzen*

[Ref 1] Abschnitt 6.3, Seite 571 - 582

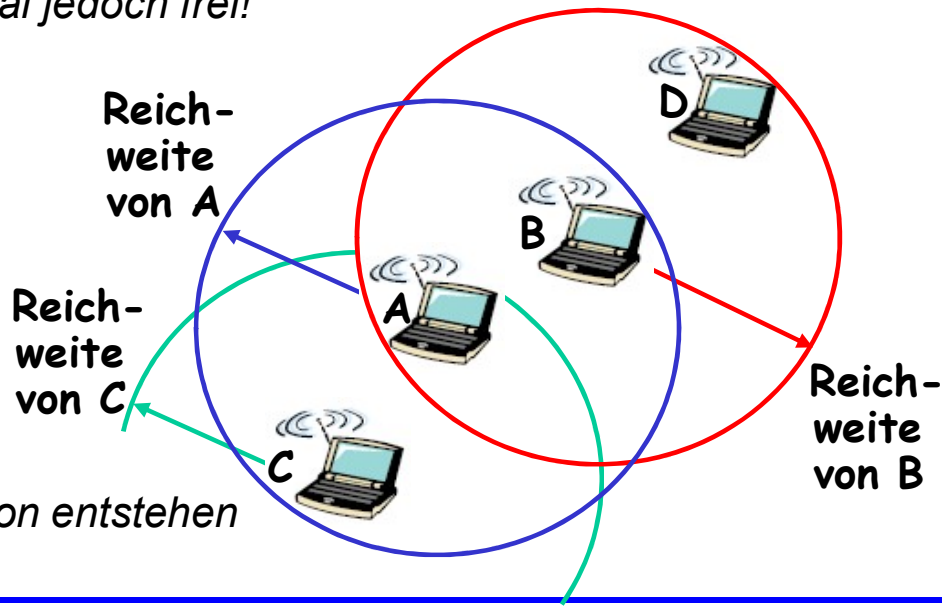
  - *Hidden-Terminal-Problem*
    - *A und B als auch B und C können sich gegenseitig hören*
    - *C sendet Daten*
    - *Für B kommt es zur Datenkollision, falls A anfängt, an B zu senden*

      → *Für A ist der Übertragungskanal jedoch frei!*

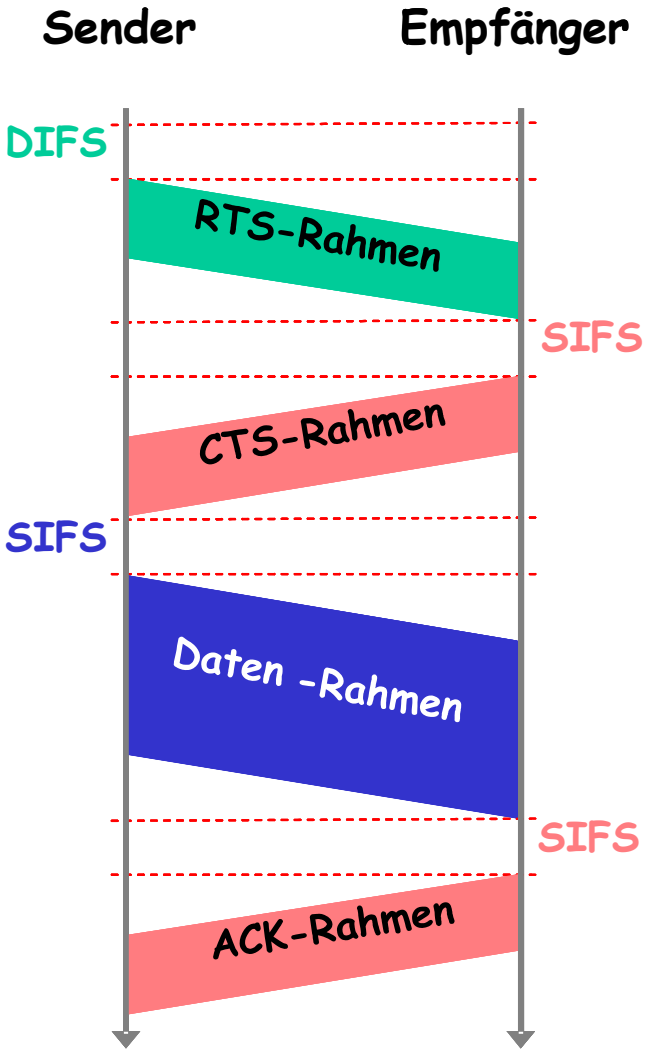


  - *Exposed-Terminal-Problem*
    - *A sendet an C*
    - *Auch B hört diese Daten-Sendung*
    - *B will an D senden, glaubt aber, dass dies eine Kollision verursachen könnte*

      → *Bei C würde aber keine Kollision entstehen*



- *Erweiterung des CSMA/CA-Protokolls: RTS-/CTS-Rahmen*
  - *Ermöglicht schnelle Neuübertragung durch die Sicherungsschicht selbst*
  - *RTS-/CTS-Rahmen*
    - *Mit RTS-Rahmen wird Übertragungskanal reserviert*
    - *Mit CTS-Rahmen wird der Erhalt des RTS-Rahmens bestätigt*
    - *RTS/CTS-Rahmen enthalten Längeninfo des zu übertragenden Datenrahmens*



- *CSMA/CA: Kollisionsvermeidung durch RTS/CTS-Rahmen*
  - *Grundsätzlicher Ablauf*
    1. *Sender sendet einen RTS (Request To Send)-Rahmen an Empfänger*
       - *RTS-Rahmen enthält die Dauer der geplanten Datenübertragung (d.h. Länge)*
       - *Andere Stationen, die RTS hören, warten bis CTS übertragen ist*
         - Zeit ergibt sich aus CTS-Rahmenlänge und der Signallaufzeit
    2. *Empfänger antwortet mit CTS (Clear to Send)-Rahmen*
       - *CTS-Rahmen enthält auch die Dauer der Datenübertragung*
       - *Andere Stationen, die CTS hören, senden nicht vor Ablauf der Übertragungsdauer*
         - Damit ist das Hidden-Terminal-Problem weitgehend gelöst
       - *Andere Stationen, die CTS nicht hören, können gleichzeitig senden*
         - Damit ist das Exposed-Terminal-Problem gelöst
    3. *Der Sender beginnt mit der Sendung*
  - *Bei RTS-Kollisionen*
    - *Falls zwei RTS-Rahmen z.B. bei einem Access-Point kollidieren, kommt kein CTS-Rahmen bei den Sendern an*

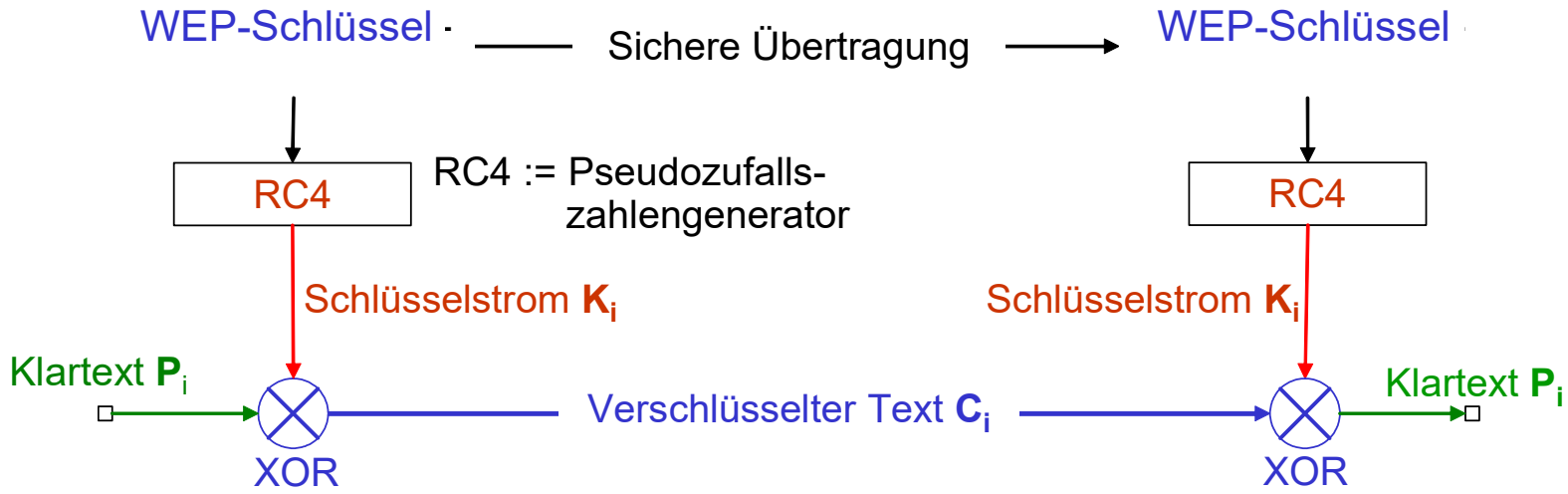
      → *Sender warten IFS-Zeit + „zufällige“ Zeitspanne (gem. Exp. Backoff-Alg.) für nächsten RTS-Sendeversuch ab*

- *Eingesetzte Sicherheitsmechanismen für 802.11-Netze*
  - *Alt: Sicherheitsmechanismen der WLAN-Standards 802.11 b/g/a*
    - *(E)SSID; (Extended) Service Set Identity: Kennung des Netzes*
      - → *WLAN-Client braucht Kennung, um sich bei einem Access Point (AP) anzumelden*
      - → *Wird oft durch Beacon-Frames vom AP selbst bekannt gemacht*
    - *MAC-ACLs (Media Access Control-Access Lists)*
      - → *AP führt Liste von erlaubten Client-MAC-Adressen*
      - → *MAC-Adressen können heute in WLAN-Karten oft modifiziert werden*
    - *WEP (Wired Equivalent Privacy)-Verschlüsselung*
      - → *Weist einige Schwachstellen auf (siehe nachfolgende Seiten)*
  - *Neu: Sicherheitsmechanismen nach IEEE 802.11i*
    - *Aktueller Sicherheitsstandard in WLAN-Netzen*
      - → *Voraus gegangenen sind die WPA und WPA2-Industriestandards*
    - *Bietet deutlich bessere Sicherheit als WEP-Standard*

3.4 Wireless LAN (IEEE 802.11)

- *Sicherheitsmechanismen der WLAN-Standards 802.11 b/g/a*
  - *Stromchiffren*

- *Sicherheitsmechanismen der WLAN-Standards 802.11 b/g/a*
  - *WEP (Wired Equivalent Privacy)-Verschlüsselung*
    - *Basiert auf dem RC4-Stromverschlüsselungsverfahren*
    - *Schema der Stromverschlüsselung:*



→ *Verschlüsselung:* $C_i = P_i$ *xor* K_i
(→ C_i *wird übertragen)*

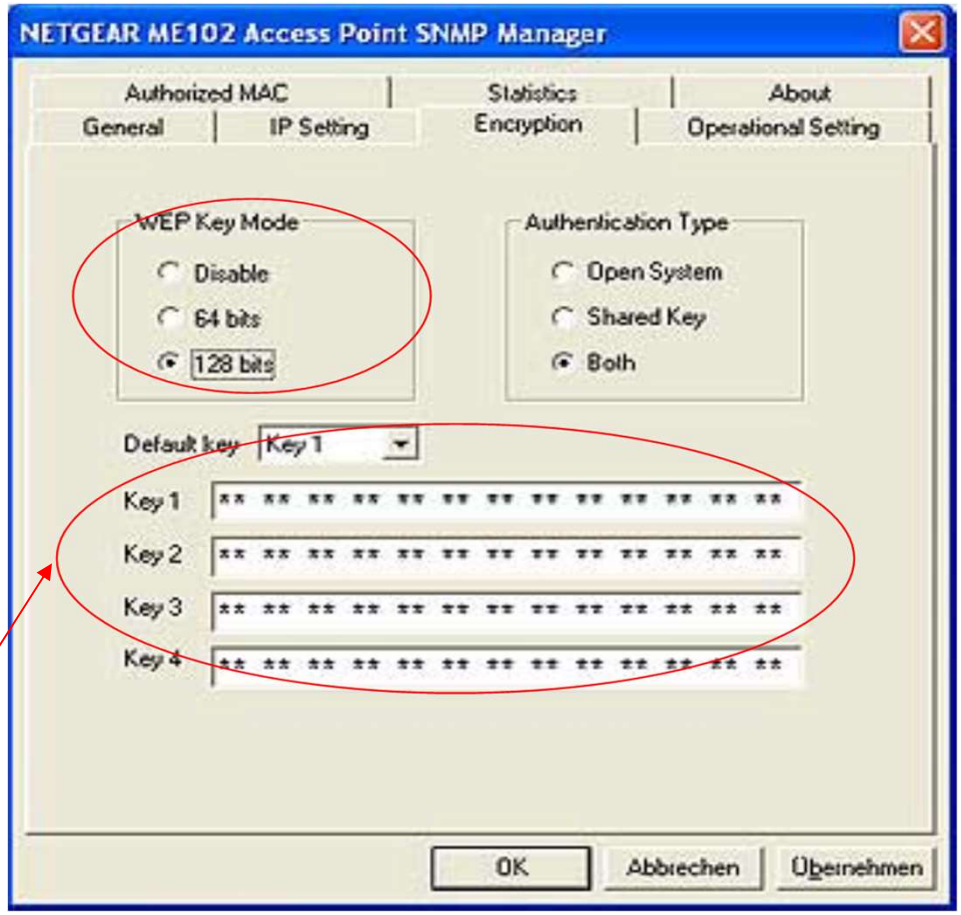
→ *Entschlüsselung:* C_i *xor* $K_i = P_i$ *xor* K_i *xor* $K_i = P_i$
(→ K_i *muss von Empfänger generiert werden können)*

- *Sicherheitsmechanismen der WLAN-Standards 802.11 b/g/a*
  - *Der WEP-Schlüssel besteht aus zwei Teilen*
    1. *„Geheimer“ Benutzerschlüssel:*
       Auf allen berechtigten Endgeräten einzutragen!
       - → *„WEP64“: enthält 40Bit-Benutzerschlüssel*
       - → *In Praxis oft „WEP128“: enthält 104Bit-Benutzerschlüssel*
    2. *24-Bit-Initialisierungsvektor (IV):*
       Wird für „jedes“ verschickte Datenpaket geändert
       - → *Grund: niemals zwei Datenpakete P1 und P2 mit gleichem WEP-Schlüssel übertragen!*
       - → *P1 xor P2 = C1 xor C2 falls C1 u. C2 Chiffretexte sind, die mit gleichem WEP-Schlüssel aus P1/P2 erstellt wurden;*
         Mithören von C1 und C2 bei (teilweise) bekanntem P1 ermöglicht Ermittlung von P2

3.4 Wireless LAN (IEEE 802.11)

- *Sicherheitsmechanismen der WLAN-Standards 802.11 b/g/a*
  - *WEP-Schlüssel:*
    - *Geheimer Benutzerschlüssel (40/104 Bit)*
      +
      automatisch generierter Initialisierungsvektor (IV: 24 Bit)
      =
      WEP-Schlüssel

Geheime Benutzerschlüssel (Key1-4): auf jedem Client einzutragen, der Teil des WLANs werden will!



- *Sicherheitsmechanismen der WLAN-Standards 802.11 b/g/a*
  - *Schwächen der WEP-Verschlüsselung*
    - *IV ist zu kurz: nur 24 Bit*
      - → *Ca. alle 16 Mio. Datenpakete (=2^{24}) wiederholt sich Initialisierungsvektor*
      - → *Da der IV im Klartext übertragen wird, ist eine „Geburtstagsattacke“ möglich*
      - → *Schon ca. alle 5000 Pakete wiederholt sich ein beliebiger IV*
      - → *Wenn zwei Klartexte P_1 / P_2 mit gleichem IV übertragen werden, gilt für die Schlüsseltexte C_1 / C_2: P_1 XOR P_2 = C_1 XOR C_2*
    - *Keine Schlüsselverwaltung*
      - → *Speicherung der geheimen Benutzerschlüssel in jedem Client des WLANs*
        - » *Sind die Benutzerschlüssel dann noch geheim?*
      - → *Invalidierung eines Benutzerschlüssels ist manuell für alle Teilnehmer durchzuführen!*

- *Sicherheit von 802.11-Funknetzen*
  - *Sicherheit des aktuellen IEEE 802.11i-Standards*
    - *Ermöglicht den Einsatz fortgeschrittener Authentifizierungsverfahren*
      - *→ Nutzung von digitalen Zertifikaten*
    - *Enthält eine dynamische Schlüsselverwaltung*
      - *→ Generierung von temporären Schlüsseln zwischen Client und Access Point aus einem „Master Secret“-Key*
      - *→ „Master Secret“-Key ist entweder*
        - eine vorher festgelegte Zeichenkette: Pre-Shared Key (PSK)-Verfahren
        - oder eine von einem Authentifizierungsserver generierte Zeichenkette
      - *→ Daher unterschiedliche Schlüssel für jeden Client*
    - *Verwendung eines besser geeigneten Verschlüsselungsverfahrens*
      - *→ Nutzung der Block-Chiffre AES (Advanced Encryption Standard)*
      - *→ IV-Problematik wurde eleminiert*
      - *→ Verlangt jedoch neue Hardware auf WLAN-Karten und Access Points!*

- *IEEE 802.11i: Authentifizierung über zentralen Server*
  - *Geeignet in großen Umgebungen mit zentraler Benutzer-Verwaltung*
  - *Nutzt den 802.1X-Standard in Kombination mit dem EAP-Protocol und RADIUS*
    - *802.1X*
      - → *Spezifiziert ein auf der Sicherungsschicht definiertes Protokoll für den Transport von Authentifizierungsnachrichten, ohne ein höheres Protokoll wie IP zu nutzen.*
      - → *Ermöglicht eine Authentifizierung eines Clients (Supplicant), bevor die Verbindung zwischen Client und Access Point (Authenticator) aktiv geschaltet wird.*
    - *EAP (Extensible Authentification Protocol)*
      - → *Ursprünglich für Remote-Access-Authentifizierung für Clients, die über PPP angebunden sind, gedacht.*
      - → *Ermöglicht Authentifizierung über verschiedenste Authentifizierungs-methoden. Für den Austausch von EAP-Paketen über Ethernet-ähnliche Netze (z.B. WLANs) werden EAP-Pakete in 802.11-Rahmen gekapselt. Der zugehörige Standard nennt sich EAPOL (EAP Over LAN).*

- *IEEE 802.11i: Authentifizierung über zentralen Server…*
  - *RADIUS (Remote Authentication Dial-In User Service)*
    - → *Der RADIUS-Server ist ein zentraler Authentifizierungsserver, der Authentifizierungsanfragen vom Authenticator entgegen nimmt und dann eine Benutzer/Passwort-Überprüfung durchführt.*
    - → *Dies kann in seiner lokalen Benutzerdatenbank erfolgen oder auch durch Weiterleitung an andere zentrale Benutzerverwaltungen (z.B. Active Directory Server).*