**Lehrstuhl für Netzarchitekturen und Netzdienste**
Institut für Informatik
TU München – Prof. Carle

# Grundlagen: Rechnernetze und Verteilte Systeme

## Kapitel 3: Direktverbindungsnetze

HDLC, PPP, Ethernet

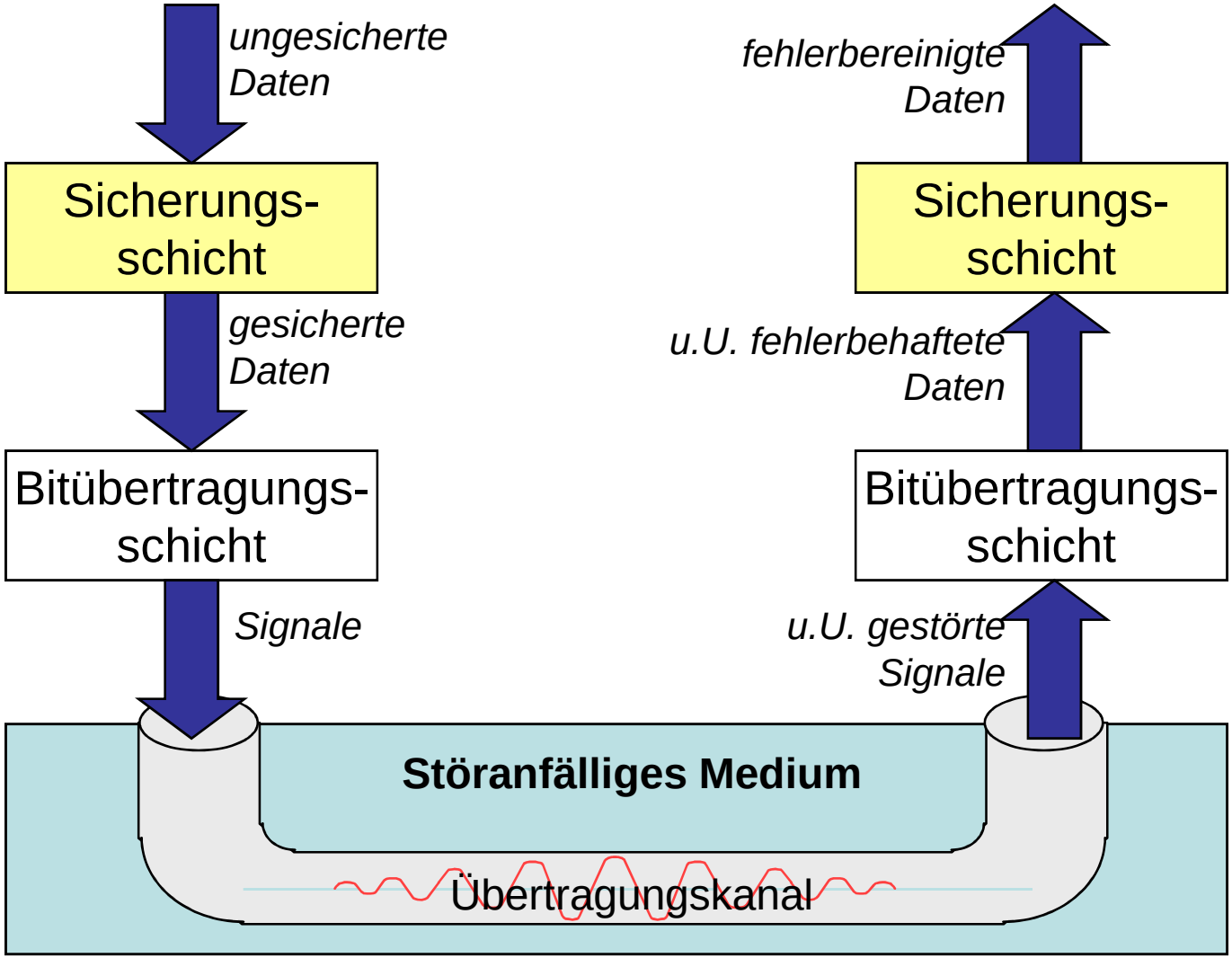
Prof. Dr.-Ing. Georg Carle
Lehrstuhl für Netzarchitekturen und Netzdienste
Technische Universität München
carle@net.in.tum.de
http://www.net.in.tum.de

# Ziele

- In diesem Kapitel wollen wir vermitteln
  - Grundverständnis von Daten- und Signalübermittlung
  - Fehlerursachen und Fehlertypen
  - Fehlerbehandlungen
  - Vorgänge in der Sicherungsschicht
  - Zugriffsverfahren

# Kapitelgliederung

# 3.1. Daten und Signale



# Data Link Control-Protokolle (DLC)

- Ziel von Datenübermittlungsprotokollen
  - *Sicherstellung einer fehlerfreien Übertragung* von Dateneinheiten über einen nicht-speichernden, „durchgehenden“ Übertragungskanal.

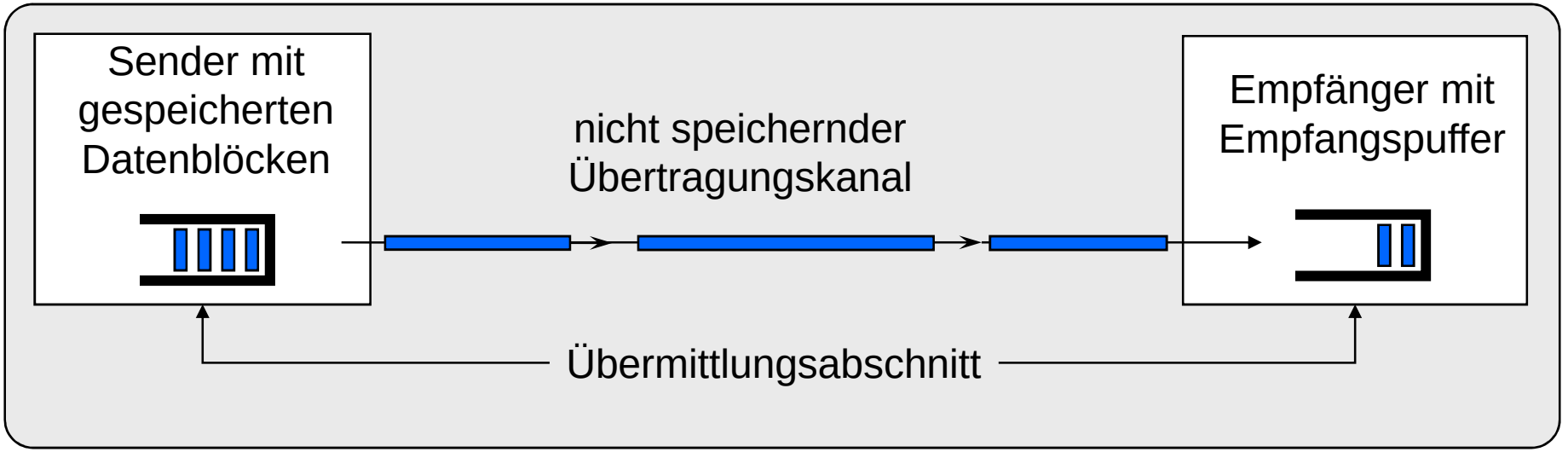
- Aufgaben
  - Aufbau und Unterhaltung einer „logischen“ Verbindung zwischen zwei „benachbarten“, d.h. über eine physikalische (unter Vernachlässigung der endlichen Übertragungsgeschwindigkeit nicht-speichernde) Verbindung direkt kommunizierender Systeme.
  - Gesicherte Übermittlung von Daten.
  - Synchronisation der Datenübertragung.

- Datenübertragungseinheiten
  - einzelne Zeichen
  - Datenblöcke (Übertragungsblöcke, Rahmen, Pakete, Zellen) (englisch: *transmission block*, *frame*, *packet*, *cell*)

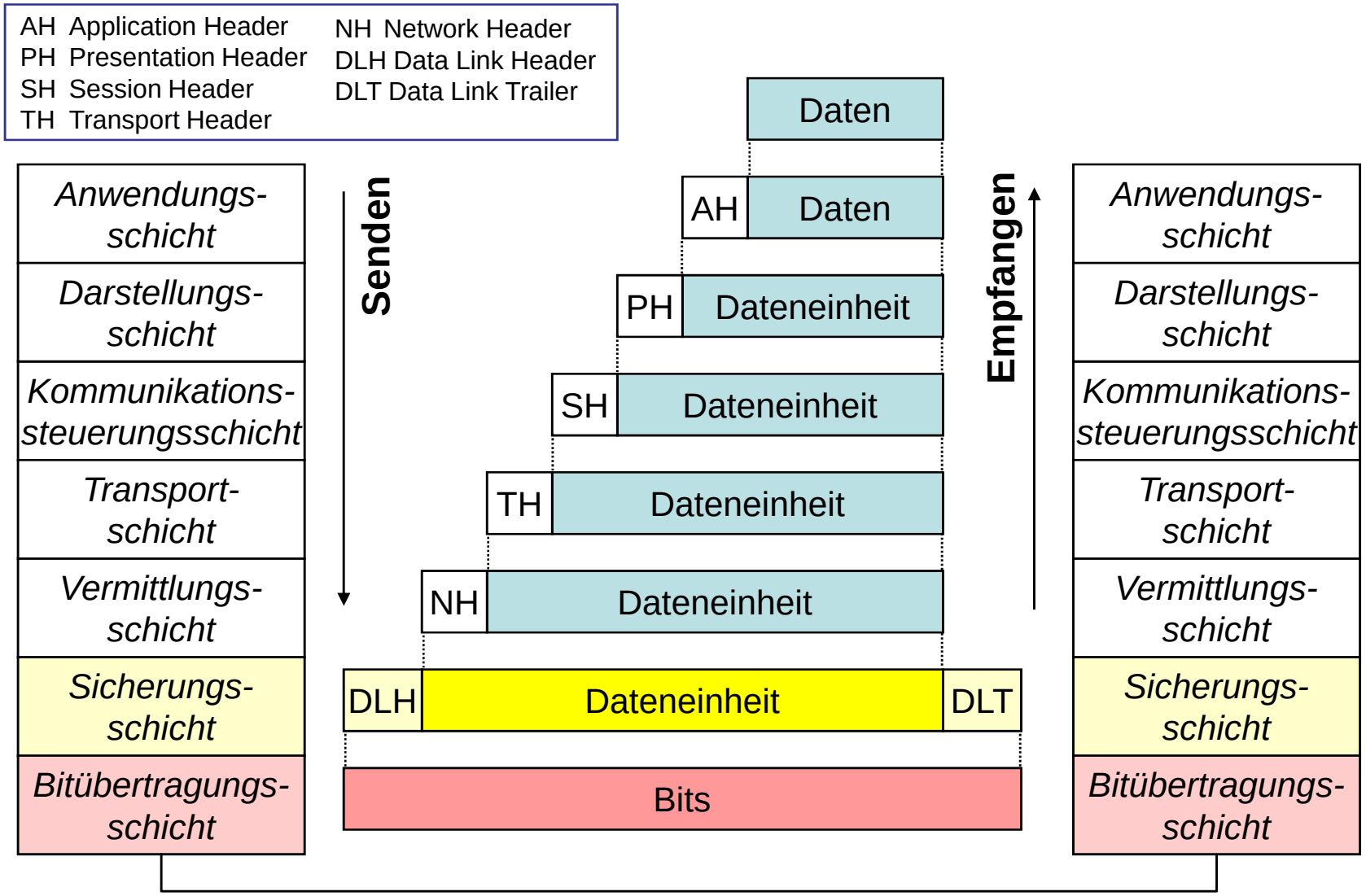
- Realisierung: Datenübermittlungsprotokolle - Data Link Control (DLC)-Protokolle

# Konzept des Übermittlungsabschnitts

- Ausgangspunkt: Puffer-zu-Puffer-Betriebsmodell
- Ein **Übermittlungsabschnitt (data link)** umfasst konzeptionell beide Pufferspeicher für den zu übermittelnden Datenblock und den nicht speichernden Übertragungsweg.
- Eine Übermittlung ist erst abgeschlossen, wenn der zu übermittelnde Datenblock vollständig und fehlergeprüft im Empfangsspeicher des betreffenden Übermittlungsabschnittes abgelegt ist.



# Einkapselung von Daten: Sicherungsschicht



# DLC: Generelles Aufgabenspektrum

- **Zeichen-/Blocksynchronisation**
  - Korrekte positionsrichtige Erkennung von Zeichen bzw. allgemein Bitfolgen für die Interpretation
  - Erkennung von Blockbegrenzungen
- **Medium-/Übermittlungsmanagement (Link Management)**
  - koordinierter Medienzugriff
  - Vergabe von Senderechten/ Übertragungsinitiativen (Arbitrierung)
- **Fehlererkennung und -behandlung**
  - Datenfehlererkennung und -behandlung mittels
    - Zeichenparität
    - Blockparität
    - Kreuzsicherung
    - Zyklische Blocksicherung
  - Protokollfehler, z.B. Verfälschung von Protokollkontrolldaten, Steuerzeichen, Adressen durch Störungen

# DLC: Konkrete Aufgabenstellung

- **Datenblockformate:**
  - Festlegung und Erkennung
- **Übermittlungsprotokolle:**
  - Übermittlungssteuerungsverfahren (z.B. Initialisierung, Terminierung, Identifikation, Halbduplex-/Vollduplexbetrieb)
- **Codetransparenz:**
  - Übertragung jeglicher Kombination von Daten der darüber liegenden Schicht
- **Fehlerbehebung:**
  - Erkennung und Behandlung von Fehlern im Daten- und im Protokollbereich
- **Zugriffsregelung:**
  - Vergabe von Senderechten, Vermeidung von Kollisionen
- **Datenflusskontrolle:**
  - Verhinderung von Überlastsituationen zwischen Sender und Empfänger des Übermittlungsabschnittes
- **Bei zeichenorientierten Protokollen:**
  - Vereinbarung eines standardisierten Übermittlungsalphabets („zeichencode-kompatibel“)

# 3.2. Synchrone Übertragung und Codetransparenz

- **Synchrone Übertragung:**
  - Empfänger muss Anfang und Ende eines Datenblocks erkennen können

- **Codetransparenz:**
  - Übertragung von Nutzdaten ermöglichen, die beliebiger Bit- bzw. Zeichenkombinationen enthalten

- **Lösungsansätze:**
  (1) Längenangabe der Nutzdaten
  (2) Steuerzeichen und Zeichenstopfen (Character Stuffing)
  (3) Begrenzungsfeld und Bitstopfen (Bit Stuffing)
  (4) Coderegelverletzungen

# Synchrone Übertragung und Codetransparenz

## (1) Längenangabe der Nutzdaten:


- Länge des Datenblocks (in Bytes/Zeichen) wird dem Empfänger im Rahmenkopf mitgeteilt

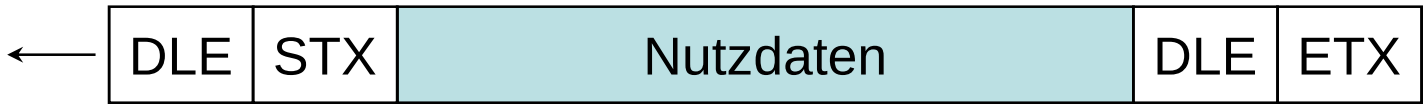
- **Problem:** Längenfeld kann durch Übertragung verfälscht werden
  - kein Erkennen der Rahmengrenze mehr möglich
  - Verlust der Synchronisation zwischen Sender und Empfänger

## (2) Steuerzeichen und Zeichenstopfen (Character Stuffing):


- reservierte Steuerzeichen markieren Anfang und Ende des Datenblocks (z.B. ASCII-Steuerzeichen)

- **Problem:**
  - Steuerzeichen dürfen nicht in den Nutzdaten auftauchen

- **Abhilfe:**
  - zeichengesteuerter Transparenzmodus durch zusätzliches Steuerzeichen DLE (Data Link Escape).




  - falls DLE in Nutzdaten auftaucht: Zeichenstopfen

# Internationales 7-bit-Alphabet (IA5) – Deutsche Referenzversion

| $b_4b_3b_2b_1$ \ $b_7b_6b_5$ | | 0 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1 0 0 | 1 0 1 | 1 1 0 | 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 0 0 0 0 | 0 | NUL | $TC_7$(DLE) | SP | 0 | @ | P | ` | p |
| 0 0 0 1 | 1 | $TC_1$(SOH) | $DC_1$ | ! | 1 | A | Q | a | q |
| 0 0 1 0 | 2 | $TC_2$(STX) | $DC_2$ | " | 2 | B | R | b | r |
| 0 0 1 1 | 3 | $TC_3$(ETX) | $DC_3$ | # | 3 | C | S | c | s |
| 0 1 0 0 | 4 | $TC_4$(EOT) | $DC_4$ | $ | 4 | D | T | d | t |
| 0 1 0 1 | 5 | $TC_5$(ENQ) | $TC_8$(NAK) | % | 5 | E | U | e | u |
| 0 1 1 0 | 6 | $TC_6$(ACK) | $TC_9$(SYN) | & | 6 | F | V | f | v |
| 0 1 1 1 | 7 | BEL | $TC_{10}$(ETB) | ' | 7 | G | W | g | w |
| 1 0 0 0 | 8 | $FE_1$(BS) | CAN | ( | 8 | H | X | h | x |
| 1 0 0 1 | 9 | $FE_2$(HT) | EM | ) | 9 | I | Y | i | y |
| 1 0 1 0 | A | $FE_3$(LF) | SUB | * | : | J | Z | j | z |
| 1 0 1 1 | B | $FE_4$(VT) | ESC | + | ; | K | Ä | k | ä |
| 1 1 0 0 | C | $FE_5$(FF) | $IS_4$(FS) | , | < | L | Ö | l | ö |
| 1 1 0 1 | D | $FE_6$(CR) | $IS_3$(GS) | - | = | M | Ü | m | ü |
| 1 1 1 0 | E | SO | $IS_2$(RS) | . | > | N | ^ | n | ß |
| 1 1 1 1 | F | SI | $IS_1$(US) | / | ? | O | _ | o | DEL |

Ursprung: American Standard Code of Information Interchange (ASCII)

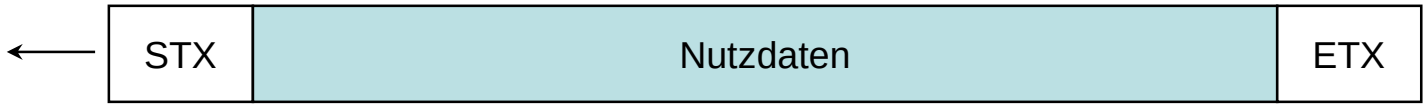
Hexadezimaldarstellung

# IA5: Steuerzeichen

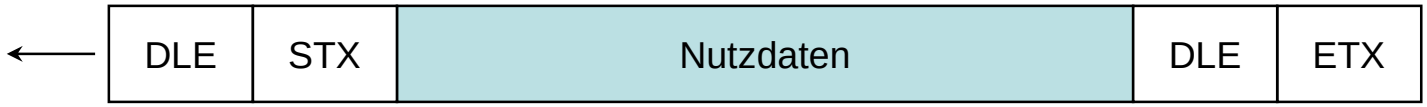
- *10 Übertragungszeichen*
  (Transmit Control TC)
  SOH Start of Header
  **STX Start of Text**
  **ETX End of Text**
  EOT End of Transmission
  ENQ Enquiry
  ACK Acknowledgement
  **DLE Data Link Espace**
  NAK Negative ACK
  SYN Synchronous Idle
  **ETB End of Transmission Block**
- *4 Gerätesteuerzeichen*
  (Device Control DC)
  nicht genormt, sondern frei belegbar.
- *3 Steuerzeichen zur Codeerweiterung*
  SO Shift Out
  SI Shift In
  ESC Escape
- *4 Informationstrennzeichen*
  (Information Separator IS)
- *6 Formatzeichen*
  (Format Encoding FE)
  $FE_1$ Backspace
  $FE_2$ Horizontal Tabulation
  $FE_3$ Line Feed
  $FE_4$ Vertical Tabulation
  $FE_5$ Form Feed
  $FE_6$ Carriage Return
- *7 Sonstige Steuerzeichen*
  NUL Füllzeichen ohne Bedeutung
  BEL Klingelzeichen
  DEL Löschen von Zeichen
  CAN Cancel
  EM End Of Medium
  SUB Substitute
  SP Space — Leerzeichen

# Codetransparenz durch Zeichenstopfen (Character Stuffing)

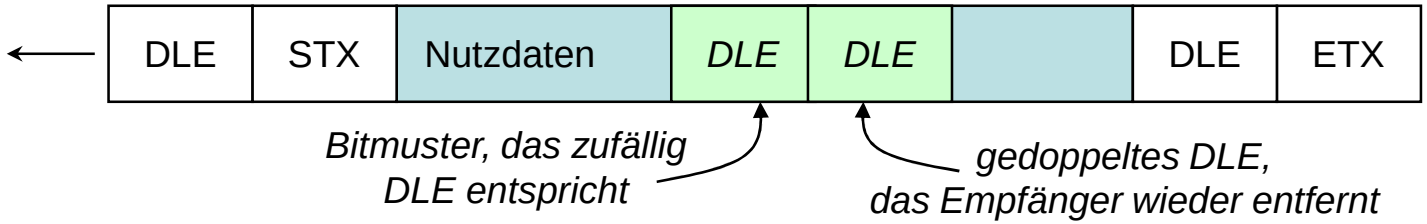
**Ansatz:** Anfang und Ende eines Rahmens werden durch STX bzw. ETX symbolisiert:



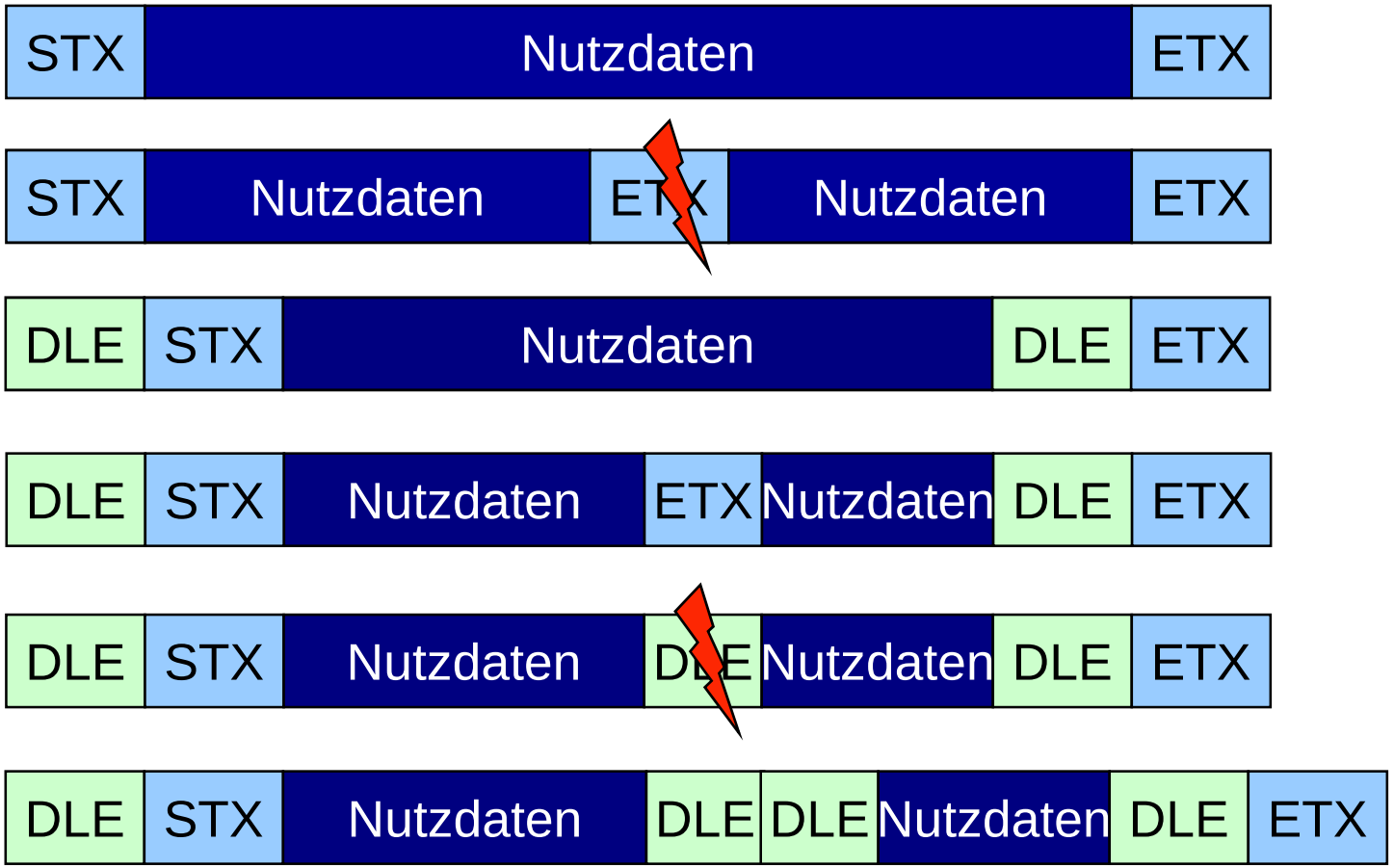

**Problem**: Ein ETX in den Nutzdaten würde ein vorzeitiges Ende des Rahmens signalisieren.
**Lösung**: Mit Hilfe eines speziellen Zeichens (DLE = Data Link Escape) werden die Nutzdaten transparent gemacht. Ein ETX wird daher nur dann als solches behandelt, wenn ein DLE davor steht.




**Problem**: Ein DLE in den Nutzdaten könnte jetzt zu einer Fehlinterpretation führen.
**Lösung**: Erkennt der Sender, dass in den Nutzdaten ein Bitmuster vorkommt, das einem DLE entspricht, so doppelt er dieses Zeichen. Der Empfänger überprüft, ob nach einem von ihm erkannten DLE ein weiteres folgt. Wenn ja, löscht er zweite DLE einfach, wenn nein, muss das folgende Zeichen als Steuerzeichen interpretiert werden.

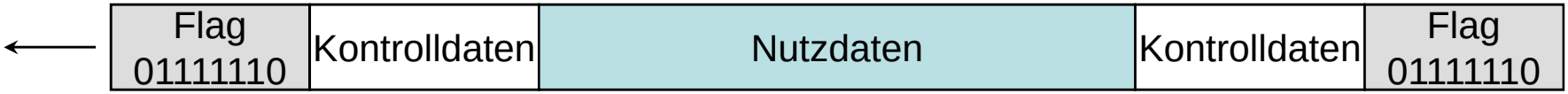


# Character Stuffing – Arbeitsfolie

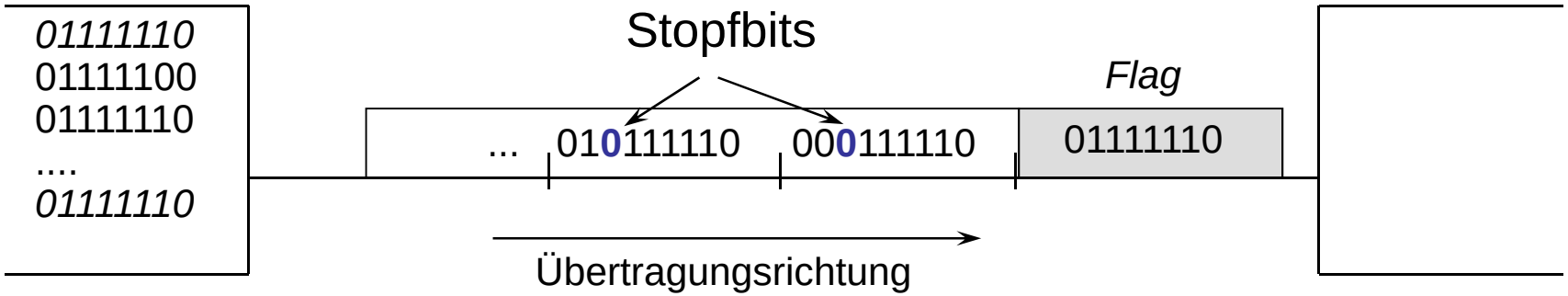


# Synchrone Übertragung und Codetransparenz

## (3) Begrenzungsfeld und Bitstopfen (Bit Stuffing):



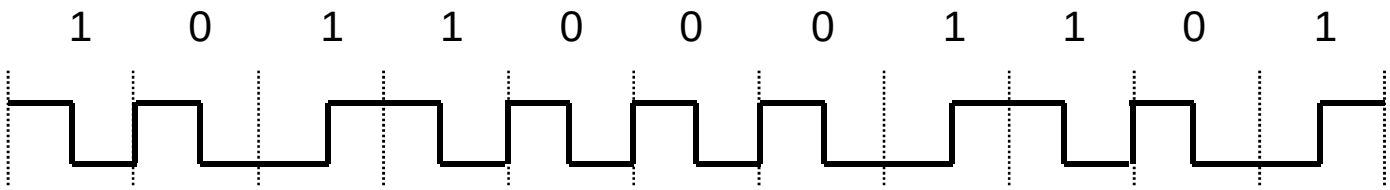

- **Ansatz:** Blockbegrenzung (Flag) ist eine besondere Bitfolge (01111110)
- **Problem:** Zufälliges Auftreten von 01111110 im DÜ-Block möglich
- **Lösung:** Einfügen von Stopfbits in die Nutzdaten
  - Sender fügt nach 5 aufeinander folgenden Binärzeichen „1“ ein Binärzeichen „0“ ein.
  - Empfänger entfernt nach 5 aufeinander folgenden Binärzeichen „1“ ein folgendes Binärzeichen „0“.




- Bemerkung: Blockprüfsumme (siehe später) zur Fehlererkennung wird ***vor*** dem Bitstopfen erstellt.

**(4) Coderegelverletzungen:**

- Blockbegrenzung durch Verwendung ungültiger Codes
- Voraussetzung: Codierung auf Schicht 1 mit Redundanz
- Beispiel: Manchester-Code (siehe Kapitel 11)




- Ungültige Codes/Pegelwerte: (0,0) und (1,1)
- Einsatz bei IEEE 802.3 (Ethernet)

# Fehlerursachen, Fehlertypen

- **Übertragungsfehler** sind hardwareinduzierte Fehler, die vorzugsweise auf dem Übertragungsmedium entstehen, aber auch in den Anschlusselektroniken der kommunizierenden Stationen.
- Art und Häufigkeit signaltechnischer Fehler sind stark vom Übertragungsmedium abhängig.
- In der Funktechnik existieren andere Fehlerursachen, Fehlerhäufigkeiten und Fehlerauswirkungen als in der leitungsgebundenen Übertragungstechnik.
- Bei Übertragung digitaler Daten führen Störeinflüsse (Fehlerquellen) zu falsch detektierten Bits.
- Typen:
  - **Einzel-Bit-Fehler**: Z. B. Rauschspitzen, die die Entscheiderschwelle bei digitaler Signalerfassung überschreiten.
  - **Bündelfehler**: Länger anhaltende Störung durch Überspannung, Starkstromschaltprozesse usw.
  - **Synchronisierfehler**: Alle Bits bzw. Zeichen werden falsch erkannt.
- Auswirkung einer Störung bestimmter Dauer ist abhängig von der Übertragungsgeschwindigkeit ⇨ Einzelbit oder Bündelstörung

# Fehlerwirkungen: Rechenbeispiel

- Eine Störung von 20 ms führt ...

  - bei Telex (50 bit/s, Signaldauer: 20 ms) zu einem Fehler von 1 Bit ⇨ Einzelfehler

  - bei ISDN (64 Kbit/s, Signaldauer: 15,625 µs ) zu einem Fehler von 1280 Bit ⇨ Bündelfehler

  bei B-ISDN / SDH

  - 155 Mbit/s, Signaldauer:6,45 ns: zu einem Fehler von ca. 3,1 Mbit = 387,5 Kbyte
  - 622 Mbit/s, Signaldauer: 1,61 ns: zu einem Fehler von ca. 12,4 Mbit = 1,5 Mbyte
  - 2,4 Gbit/s, Signaldauer: 0,4 ns: zu einem Fehler von ca. 48 Mbit = 6 Mbyte

  ⇨ Bündelfehler großer Länge

# Fehlerhäufigkeiten

- Maß für die Fehlerhäufigkeit:

$$\textbf{Bitfehlerrate} = \frac{\text{Summe gestörte Bits}}{\text{Summe übertragene Bits}}$$

  - Stark vom Übertragungsmedium bzw. Netz abhängig

- Typische Wahrscheinlichkeiten für Bitfehler:
  - Analoges Fernsprechnetz $2*10^{-4}$
  - Funkstrecke $10^{-3}$ - $10^{-4}$
  - Ethernet (10Base2) $10^{-9}$ - $10^{-10}$
  - Glasfaser $10^{-10}$ - $10^{-12}$

# Fehlerwirkungen

- Fehlerwirkungen sind abhängig davon, welche Bits betroffen sind:
  - **(Nutz-)Datenfehler**: Bits innerhalb der Nutzdaten (gesehen z. B. aus Sicht der Sicherungsschicht) werden gestört.


  - **Protokollfehler**: Störungen können Protokollsteuerdaten (z.B. Adressen oder sonstige protokollrelevante Daten) verfälschen.

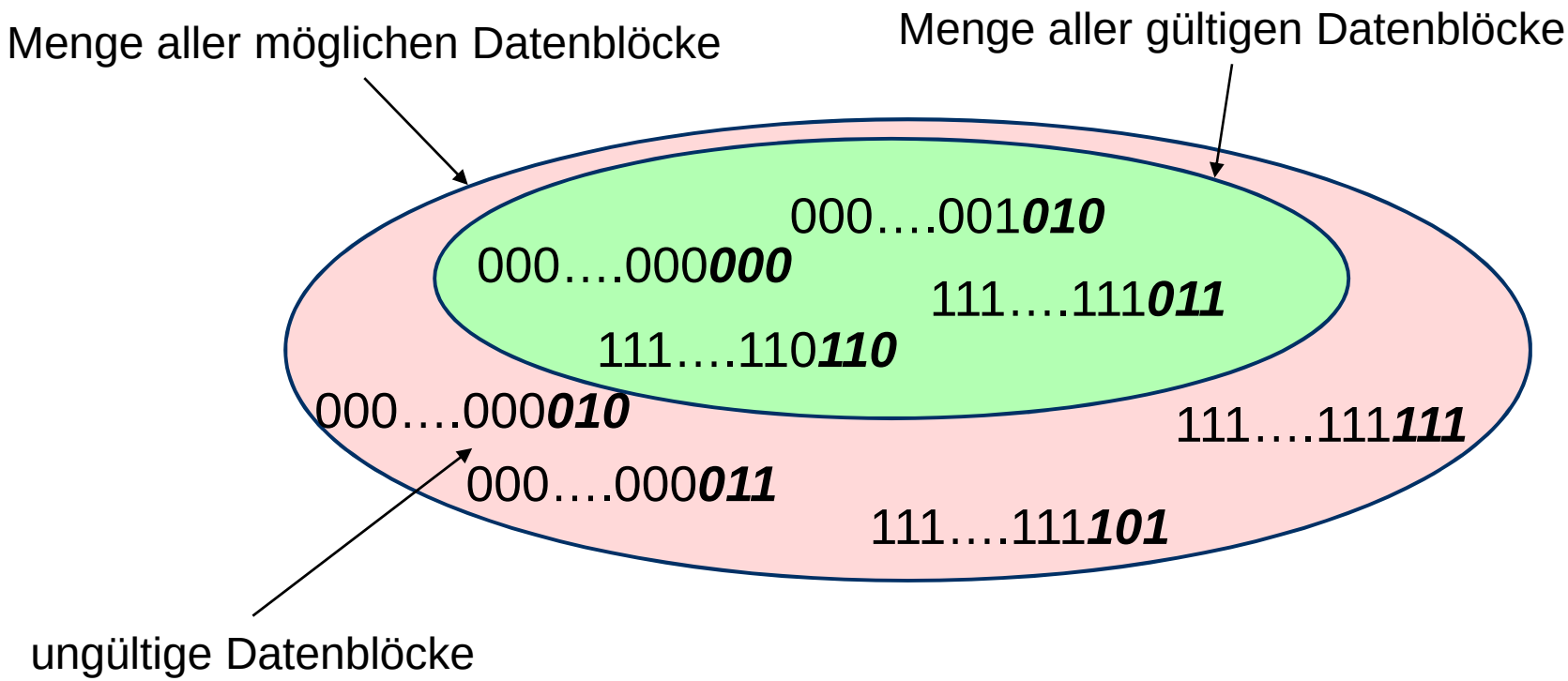

⇨ **Fehlererkennungs- und Behandlungsmaßnahmen** (error detection and recovery) erforderlich.

⇨ Fehlererkennung durch (künstliches) Hinzufügen von Redundanz beim Sender

- error detecting codes
- (Spezialfall: error correcting codes)

# Fehlererkennung: Grundprinzip

- Unterteilung aller möglichen Datenblöcke in gültige und ungültige:




- Wie?
  - durch gezieltes Hinzufügen von Redundanz beim Sender

# 3.2.2. Fehlerbehandlung

- Fehler ignorieren
  - Beispiel: Audio/Video-Stream ➔ kurzzeitiges Fehlsignal oder Lücke
  - nicht möglich bei Fehlern in den Steuerdaten!
- Wiederholung der Übertragung
  - Empfänger verwirft fehlerhaften Datenblock
  - implizite oder explizite Wiederholungsanforderung (siehe später)
- Fehlerkorrektur:
  - Codes, mit denen Bitfehler korrigiert werden können
  - erfordert größere Redundanz als reine Fehlererkennung

# Fehlererkennung: Paritätssicherung — Überblick

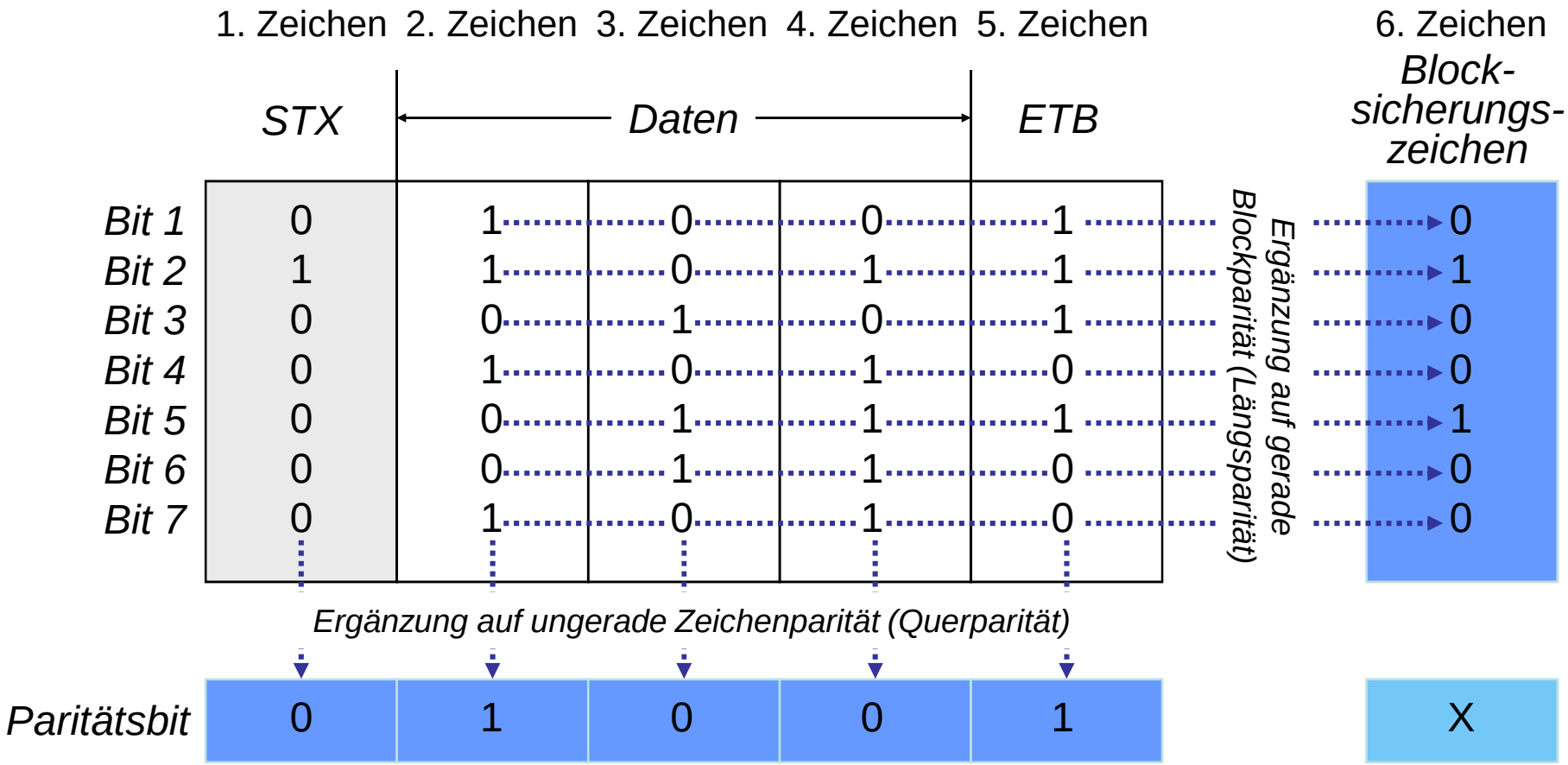
- *Gerade Parität*:
  - Gesamtzahl der „1” einschließlich des Paritätsbits ist gerade.
- *Ungerade Parität*:
  - Gesamtzahl der „1” einschließlich des Paritätsbits ist ungerade.
- *Zeichen- oder Querparität*: (VRC: Vertical Redundancy Check)
  - Sicherung von Einzelzeichen. Einziges Verfahren bei asynchroner Einzelzeichen-Übertragung.
- *Block- oder Längenparität*: (LRC: Longitudinal Redundancy Check)
  - Alle Bits gleicher Wertigkeit innerhalb eines aus Zeichen bestehenden Übertragungsblocks werden durch ein Paritätsbit (gerade oder ungerade) gesichert. Sie bilden ein Blockprüfzeichen (BCC: Block Check Character).
- *Kreuzsicherung*:
  - Gleichzeitige Anwendung von Längs- und Querparität. Festlegung über die Bildung des Paritätsbits des Blockprüfzeichens erforderlich!

- **Hinweis:** STX wird nicht in Block-Paritätsprüfung einbezogen, da der Empfänger erst nach START OF TEXT weiß, dass ein prüfenswerter Übertragungsblock beginnt.

- **Problem:** Fehlererkennungswahrscheinlichkeit bei Paritätsprüfung nicht hoch. Mehrfachfehler (zwei oder eine gerade Zahl in gleicher Zeile oder Spalte liegende Fehler) werden nicht erkannt.

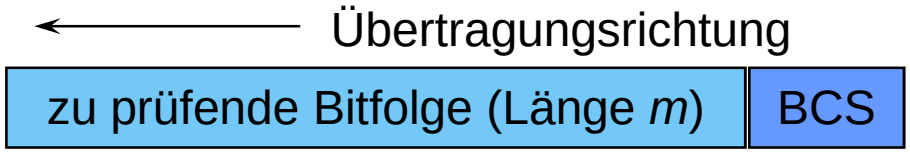
# Fehlererkennung: Paritätsüberprüfung

- Gerade/ungerade Parität
- Querparität, Längsparität, Kreuzparität



# Fehlererkennung: Cyclic Redundancy Check (CRC)

- Zu prüfender Block wird als unstrukturierte Bitfolge aufgefasst.
  - Anzahl der zu prüfenden Bits ist beliebig
- Prüfbitfolge [Block Check Sequence (BCS) bzw. Frame Check Sequence (FCS)] wird an den zu prüfenden Übermittlungsdatenblock angehängt.




Anhängen der Blockprüfdaten an den Übermittlungsblock

- Bildung der Prüfsequenz:
  - Die zu prüfende Bitfolge wird als Polynom $M(x)$ aufgefasst, d.h. jedes der $m$ Bit als Koeffizient eines Polynoms vom Grad ($m$-1) interpretiert.
  - Zur Berechnung der BCS/FCS wird an die zu prüfenden Bitfolge eine Nullfolge der Länge $r$ angehängt (entspricht Multiplikation mit $x^r$ ), wobei $r$ der Grad des Prüfpolynoms (Generatorpolynoms) $G(x)$ ist.
  - Das erhaltene Polynom $M(x) \cdot x^r$ wird durch das Prüfpolynom $G(x)$ geteilt; der Rest $R(x)$ der Division ist die gesuchte BCS/FCS.
  - $M(x) \cdot x^r - R(x)$ wird als Bitfolge an den Empfänger übertragen.
  - Beim Empfänger wird die empfangene Bitfolge durch $G(x)$ dividiert. Bei fehlerfreier Übertragung ist der Rest 0.

# Fehlererkennung: CRC-Beispiel – Senden

- Zu sendende Bitfolge: 110011 ⇨ $M(x) = x^5 + x^4 + x + 1$
- Prüfpolynom: $G(x) = x^4 + x^3 + 1$ ⇨ Divisor in Modulo-2-Binärarithmetik: 11001
  - Addition/Subtraktion Modulo-2 entspricht einer bitweisen XOR-Verknüpfung
  - Dividend ist teilbar durch Divisor, falls der Dividend mindestens so viele Stellen besitzt wie der Divisor (führende Bits müssen beide 1 sein)
- Länge der Sicherungsfolge = Grad des Prüfpolynoms = 4

- Berechnung der Sicherungsfolge:

angehängte Nullen

```
11 0011 0000 ÷ 1 1001 = 10 0001
11 001
00 0001 0000
       1 1001
       0 1001 = Rest
```

- Zu übertragende Bitfolge: 11 0011 1001.

# Fehlererkennung: CRC-Beispiel – Empfangen

- Empfangen einer korrekten Bitfolge:

```
11 0011 1001 ÷ 1 1001 = 10 0001
11 001
-------
00 0001 1001
      1 1001
      -------
      0 0000  = Rest
```

  - Kein Rest, somit sollten Daten fehlerfrei sein.

- Empfangen einer verfälschten Bitfolge:

```
11 1111 1000 ÷ 1 1001 = 10 1001
11 001
-------
00 1101 1
   1100 1
   -------
   0001 0000
      1 1001
      -------
      0 1001  = Rest ≠ 0
```

  - Es bleibt Rest ungleich 0, somit war ein Fehler in der Übertragung.

# Cyclic Redundancy Check: Leistungsfähigkeit

- Folgende Fehler werden durch CRC erkannt:
  - sämtliche Einzelbitfehler;
  - sämtliche Doppelfehler, wenn ($x^k + 1$) nicht durch das Prüfpolynom teilbar ist, für alle $k \leq$ Rahmenlänge;
  - sämtliche Fehler ungerader Anzahl, wenn ($x$+1) Faktor des Prüfpolynoms ist;
  - sämtliche Fehlerbursts der Länge ≤ Grad des Prüfpolynoms.
- International genormt sind folgende Prüfpolynome:
  - CRC-12 $= x^{12} + x^{11} + x^3 + x^2 + x + 1$
  - CRC-16 $= x^{16} + x^{15} + x^2 + 1$
  - CRC-CCITT $= x^{16} + x^{12} + x^5 + 1$
- CRC-16 und CRC-CCITT entdecken
  - alle Einzel- und Doppelfehler,
  - alle Fehler ungerader Anzahl,
  - alle Fehlerbursts mit der Länge ≤ 16
  - 99,997 % aller Fehlerbursts mit der Länge 17
  - 99,998 % aller Fehlerbursts mit der Länge 18 und mehr

# 3.2.3. Vorwärtsfehlerkorrektur für Datenpakete

- Bislang war die Redundanz nur zur Überprüfung der Daten mitgeliefert worden, jetzt soll sie dazu dienen, verloren gegangene Pakete zu rekonstruieren.

- Beispiel:
  - Zu senden sind die Pakete
    0101 - P1
    1111 - P2
    0000 - P3
  - Dazu wird über XOR ein weiteres Paket berechnet: 1010 - P4
  - Diese vier Pakete werden jetzt an den Empfänger geschickt.

# Vorwärtsfehlerkorrektur — Ablauf

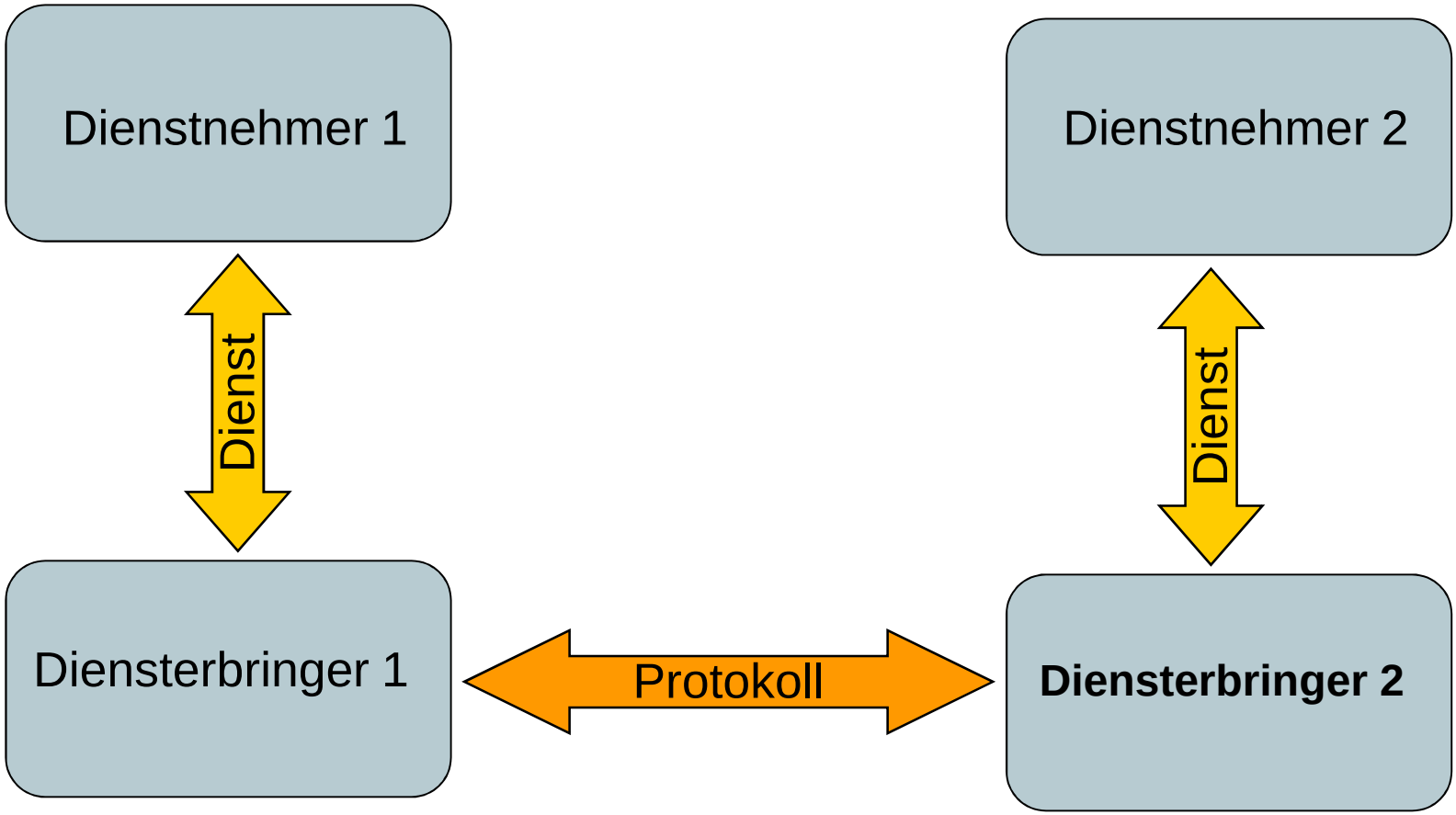
- Der Empfänger braucht nur drei der vier Pakete, um das fehlende zu rekonstruieren. Er verknüpft einfach die korrekten Pakete wieder XOR, und erhält so das fehlende:
  - P1 geht verloren:

    1111 – P2
    0000 – P3
    1010 – P4
    ⇨ 0101 – P1
  - P2 geht verloren:

    0101 – P1
    0000 – P3
    1010 – P4
    ⇨ 1111 – P2
  - P3 geht verloren:

    0101 – P1
    1111 – P2
    1010 – P4
    ⇨ 0000 – P3
- Natürlich muss der Empfänger wissen, welches Paket verloren gegangen ist ...

# Wiederholung: Dienst und Protokoll



# Wiederholung: Bezeichnungskonventionen

- **(N)-Schicht**

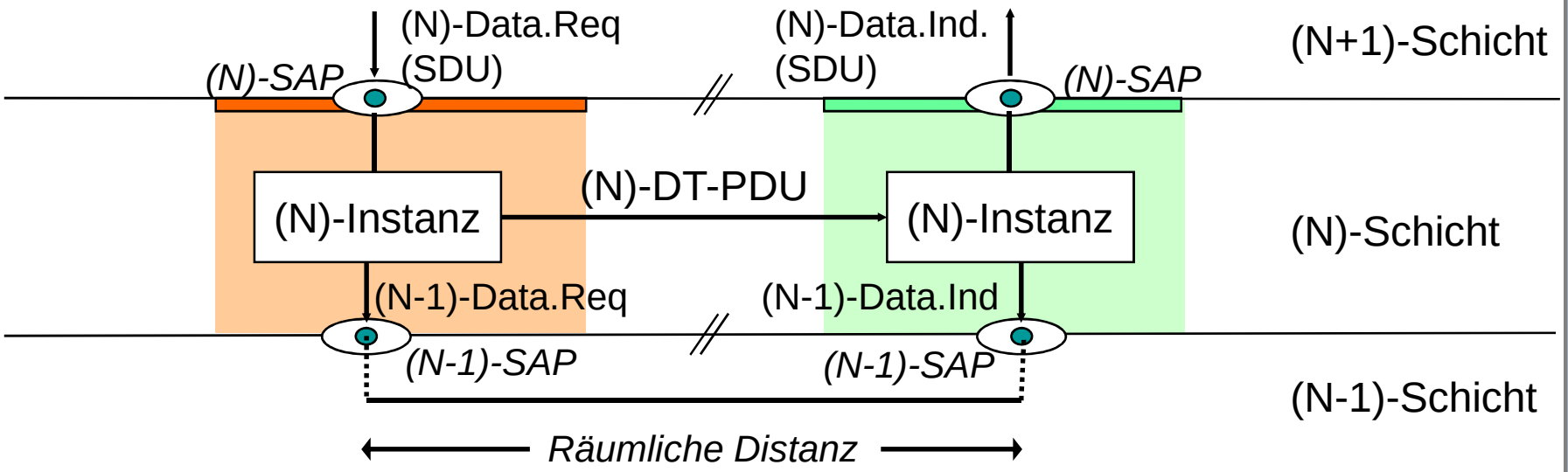
| | | |
|---|---|---|
| A | -Schicht: Anwendungsschicht | (**A**pplication Layer) |
| P | -Schicht: Darstellungsschicht | (**P**resentation Layer) |
| S | -Schicht: Kommunikationssteuerungsschicht | (**S**ession Layer) |
| T | -Schicht: Transportschicht | (**T**ransport Layer) |
| N | -Schicht: Vermittlungsschicht | (**N**etwork Layer) |
| DL | -Schicht: Sicherungsschicht | (**D**ata **L**ink Layer) |
| Ph | -Schicht: Bitübertragungsschicht | (**Ph**ysical Layer) |

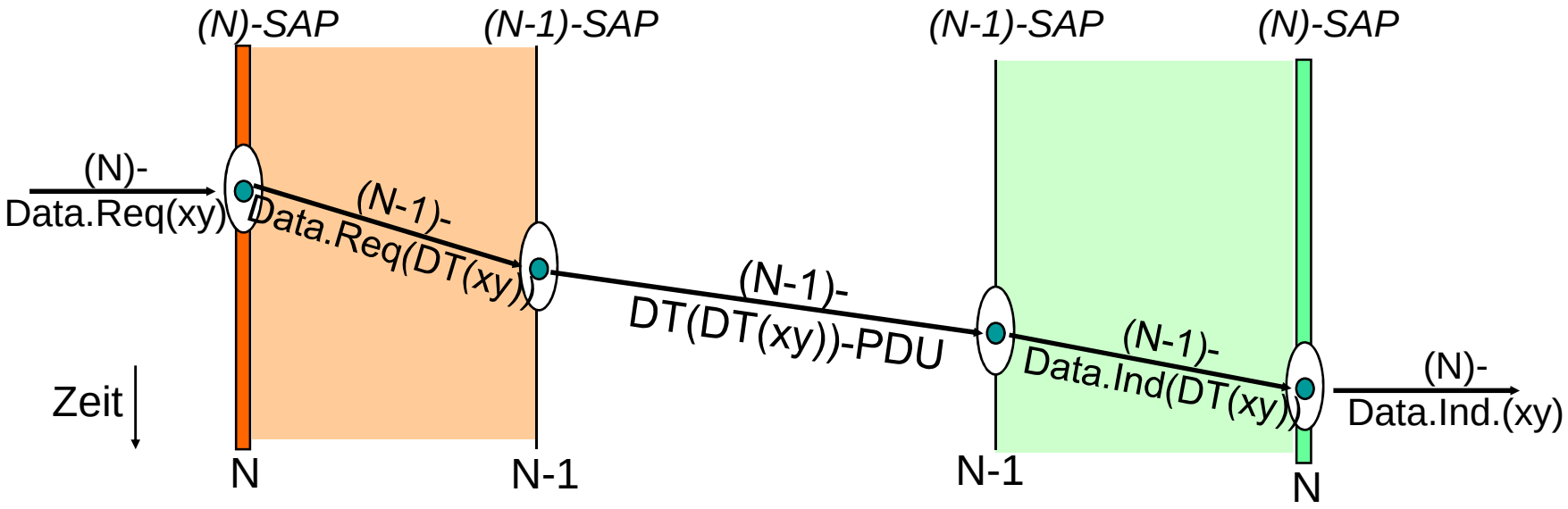
- **(N)-Dienstprimitive**

(N)-*Dienst.typ*

| | | |
|---|---|---|
| Ph | - Connect | .Req (Request, Anforderung) |
| DL | - Data | .Ind (Indication, Anzeige) |
| N | - Disconnect | .Rsp (Response, Antwort) |
| T | ... | .Cnf (Confirmation, Bestätigung) |

# Von der Schichtendarstellung zum Weg-Zeit-Diagramm



# 3.3. Sicherungsschicht mit Fehlerbehandlung

- **Aufgaben:**
  - Kommunikation zwischen Partnern im gleichen Subnetz (über Punkt-zu-Punkt- bzw. Punkt-zu-Mehrpunkt-Verbindung)
  - Dienste der Sicherungsschicht:
    - unbestätigt (unquittiert) oder bestätigt (quittiert)
    - verbindungslos oder verbindungsorientiert (Aufbau, Datenübertragung, Abbau)
    - ungesichert oder gesichert (mit Bezug auf Übertragungsfehler)

- **Funktionalität der Sicherungsschicht:**
  - Bildung von Übertragungsrahmen
  - Fehlerbehandlung (Erkennen und Beheben von Verfälschung, Verlust)
  - Flusssteuerung zur Überlastvermeidung
  - Verbindungsverwaltung

- **Betrachteter Dienst:**
  - Gesicherter Dienst: Reihenfolgetreue Auslieferung, Fehlerbehandlung und Duplikaterkennung

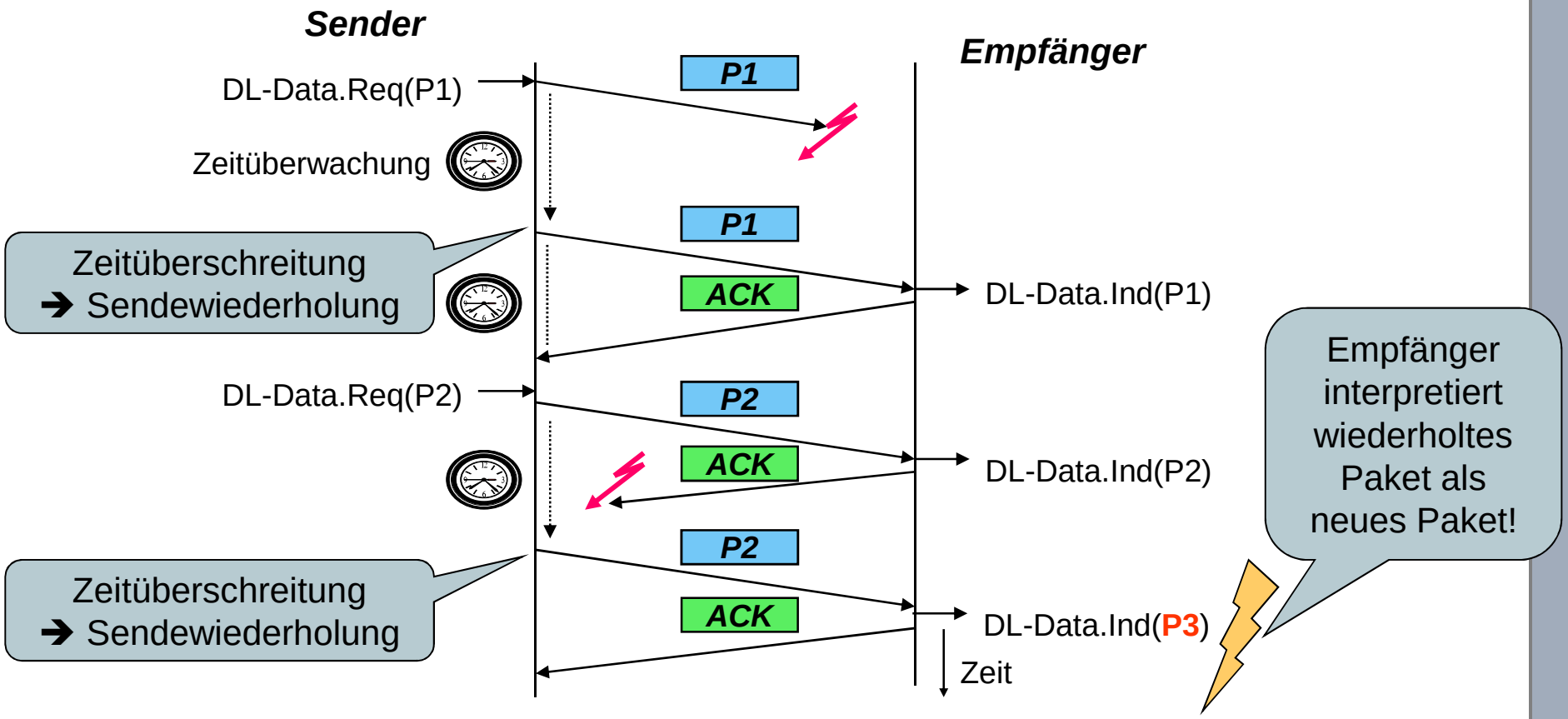
# Protokolle zur Fehlerbehandlung

- Ein zuverlässiger Dienst bei unzuverlässigem Kanal benötigt ein automatisches Sendewiederholungsverfahren (ARQ - Automatic Repeat Request):
  - Einfache Protokolle
    - Protokoll mit impliziter Wiederholungsanforderung (Beispiel: Alternating-Bit-Protokoll)
    - Protokoll mit expliziter Wiederholungsanforderung
  - Schiebefensterprotokolle
    - Go-Back-N-Verfahren (mit impliziter oder expliziter Wiederholungsanforderung)
    - Selektive Wiederholung (mit impliziter oder expliziter Wiederholungsanforderung)

# Fehlerbehandlung durch implizite Wiederholungsanforderung

**Einfaches Bestätigungsprotokoll:**

- Sender sendet ein Paket und wartet auf Empfangsbestätigung
- nach Eingang der Empfangsbestätigung wird der nächste Datenblock gesendet
- geht keine Empfangsbestätigung ein (Zeitüberschreitung), wird derselbe Datenblock erneut gesendet



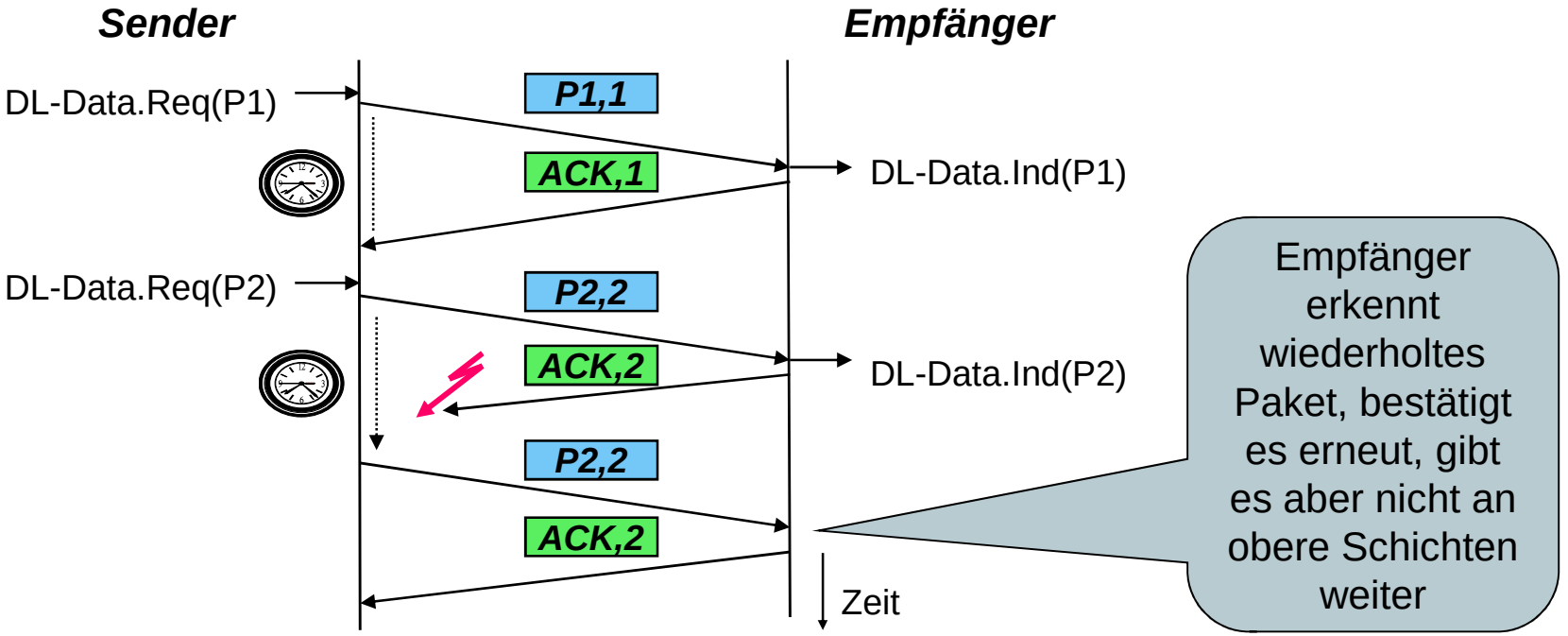
# Fehlerbehandlung durch implizite Wiederholungsanforderung

**Problem:**

- Empfangsbestätigung muss einem Paket zugeordnet werden können

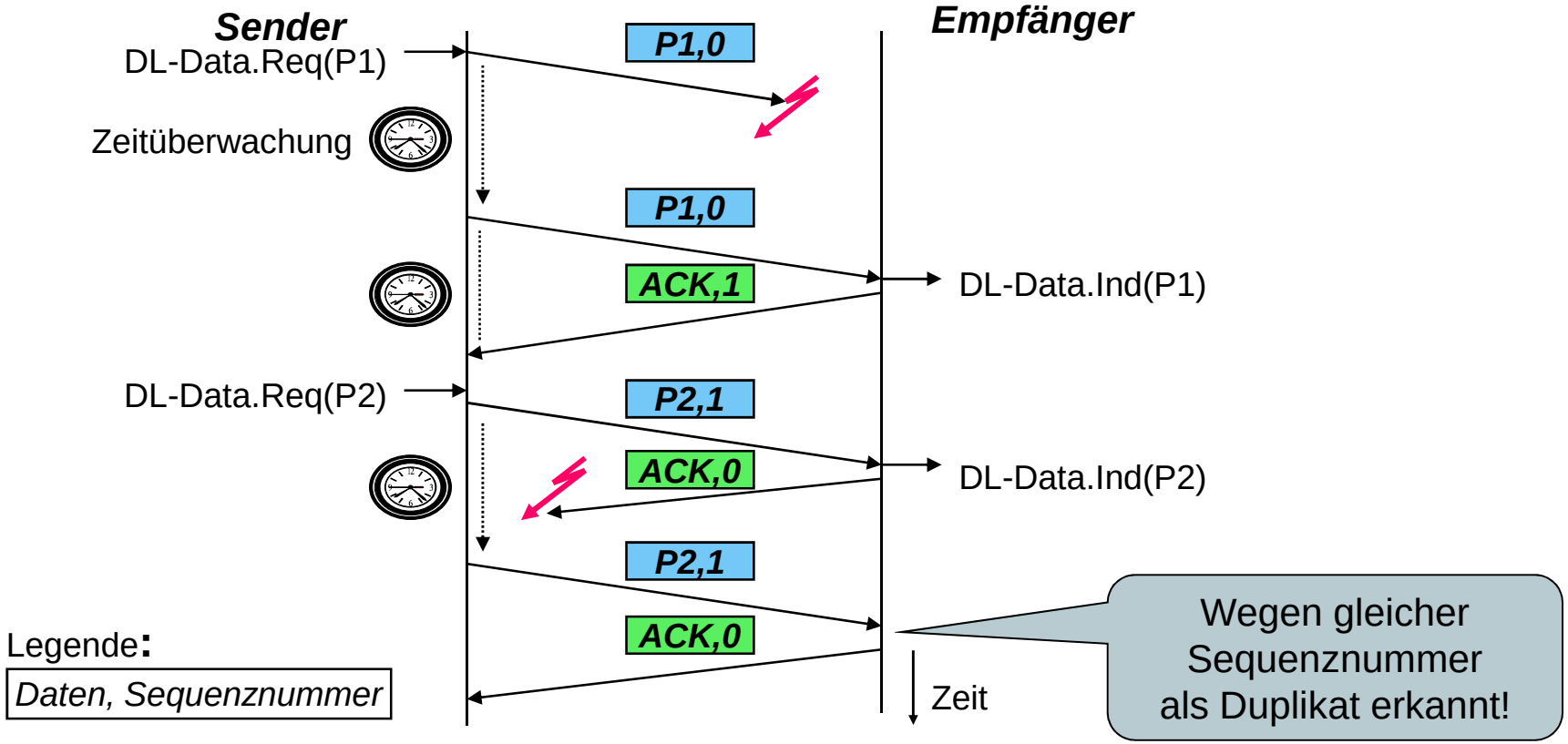
**Lösung:** Sequenznummern

- Sender vergibt aufsteigende Sequenznummern an versendete Pakete, Sequenznummer wird im Paketkopf als Kontrollinformation mitgesendet
- Empfangsbestätigung enthaltet Sequenznummer des bestätigten Pakets



# 3.3.1. Alternating-Bit-Protokoll

- Im Halbduplex-Betrieb reicht ein Sequenznummernraum {0; 1}, der zyklisch durchlaufen wird ⇨ Alternating-Bit-Protokoll
- Unterschiedliche Semantik der im Ack enthaltenen Sequenznummern möglich
  - Ack enthält entweder zuletzt empfangene SN, oder (üblich) nächste erwartete SN




- Wichtig: Zeitüberwachung größer als die RTT (Round-Trip-Time)
  - ansonsten unnötige Sendewiederholungen

# Alternating-Bit-Protokoll

**Programm für Sender:**

```
sender(){
   int frameSent = 0;          /*Sequenznummer des Rahmens*/
   frame s,r;                  /*Sende- und Empfangsrahmen*/
   packet buffer;              /*Puffer für ausgehenden Nutzdaten*/
   eventType event;            /*FrameArrival, Timeout, CRCError*/

   FromNetworkLayer(&buffer); /*Hole erstes Paket*/

   while(true) {
      s.info = buffer;         /*Erstelle Rahmen zur Übertragung*/
      s.seq = frameSent;       /*Sequenznummer in Rahmen einf.*/
      ToPhysicalLayer(&s);     /*Sende Rahmen*/
      StartTimer();            /*Timer für Sendewiederholung*/
      wait(&event);            /*FrameArrival, Timeout, CRCError*/
      if(event == Timeout || event == CRCError) { /* nichts */ }
      if(event == FrameArrival) {          /*Gültiger Rahmen angek.*/
         FromPhysicalLayer (&r);           /*Rahmen einlesen*/
         if (r.seq != frameSent) {         /*Überprüfe Sequenznr.*/
            FromNetworkLayer(&buffer);     /*Hole nächsten R.*/
            invert(frameSent);}}           /*Invertiere Seq.-Bit*/
}
```

⇨ Wiederhole Rahmen nach Ablauf des Zeitgeber, Fehler oder doppelter Quittierung, sonst sende nächsten Rahmen

# Alternating-Bit-Protokoll

**Programm für Empfänger:**

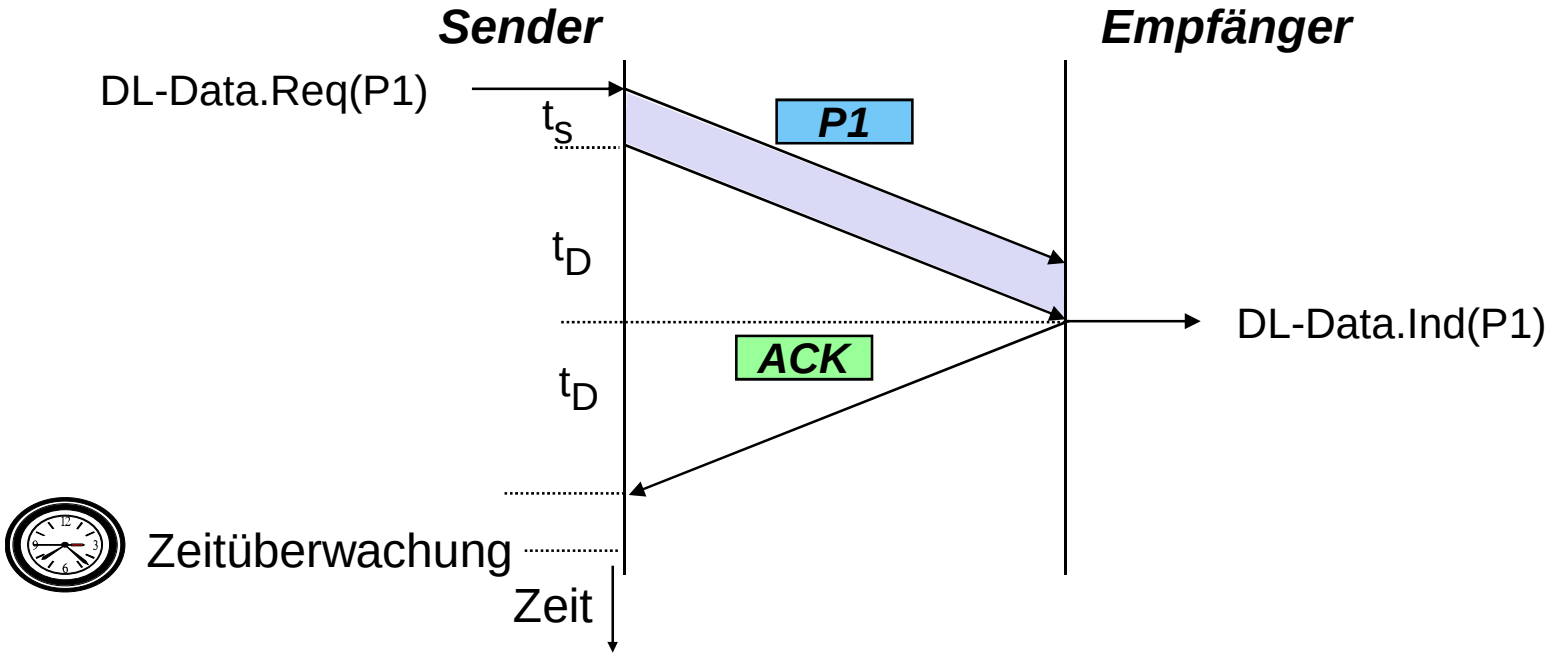
```
receiver(){
   int frameExpected = 0;     /*Erwartete Sequenznummer */
   frame r,s;                 /*r: empfangener Rahmen; s: Quittung*/
   eventType event;
   while (true) {
      wait (&event);          /*FrameArival, CRC Error*/
      if (event == FrameArrival) {  /*Gültiger Rahmen angekommen */
         FromPhysicalLayer (&r);
         if (r.seq == FrameExpected) {
             ToNetworkLayer (&r.info);
             invert (frameExpected);
         }
         s.seq = frameExpected;    /*Folgenummer quittieren*/
         ToPhysicalLayer (&s);
      }
   }
}
```

⇨ liefere nur korrekte Rahmen aus, aber bestätige *alle* Rahmen

Bei Duplex-Betrieb evtl. Anhängen der Bestätigungen an Nutzdaten von Empfänger zu Sender "Huckepack“ ("Piggyback")

# Leistungsbetrachtung

- Die unterschiedlichen Protokolle können je nach Kanal zu großen Leistungsunterschieden führen



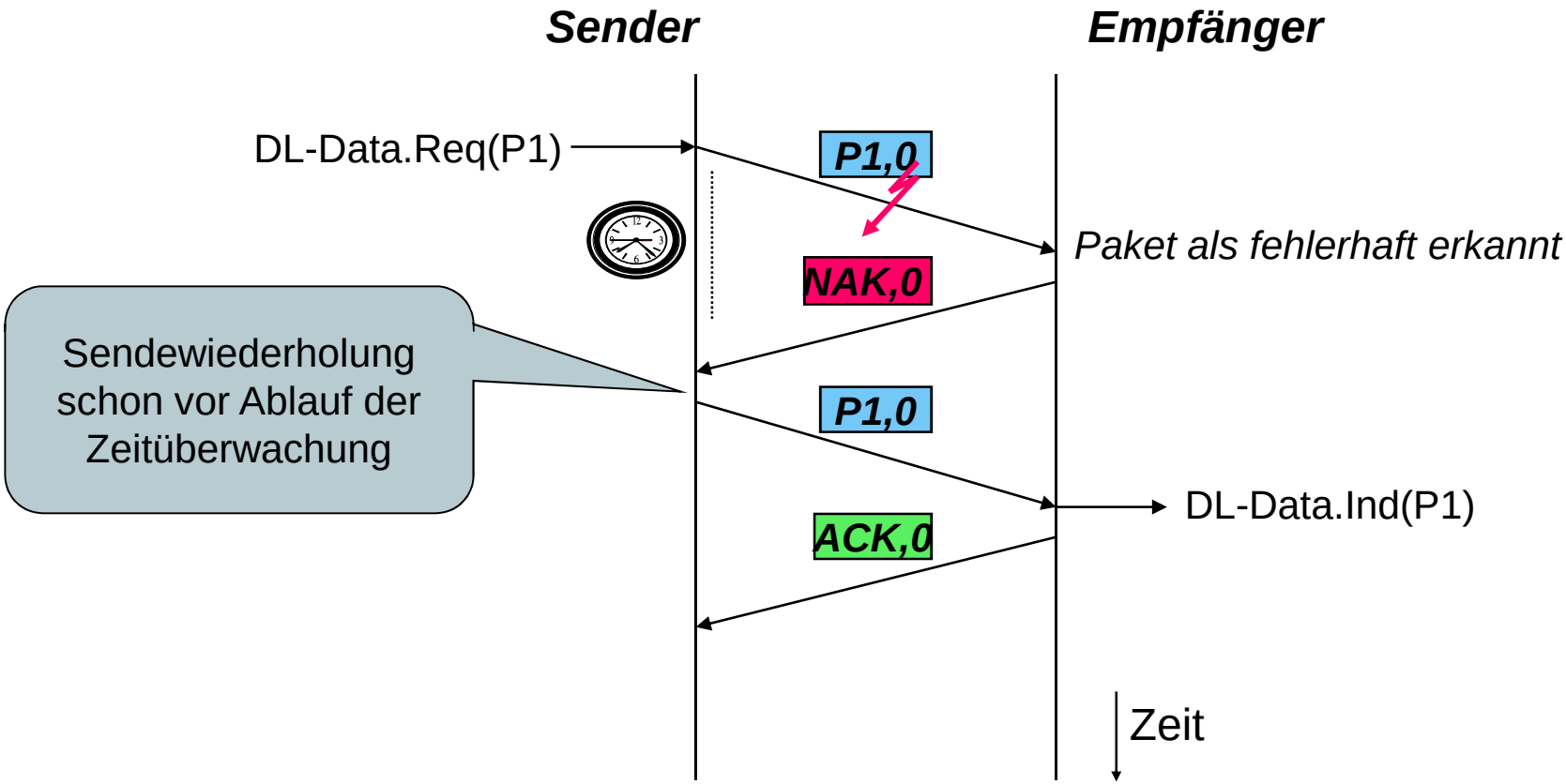

Bsp:

| | | |
|---|---|---|
| $l_R$ | Rahmenlänge [bit] | $l_R$ =1000 bit |
| Ü | Übertragungskapazität [bit/s] | Ü = 500 kbit/s |
| $t_S$ | Sendezeit [s] | $t_S = l_R$ / Ü = 2 ms |
| $t_D$ | Übertragungsverzögerung [s] | $t_D$ = 240 ms |
| η | Kanalausnutzung (Effizienz) | $\eta = t_s /(t_s + 2\, t_D) \approx 0{,}4\%$ |

⇨ Effizienzsteigerung durch Schiebefensterprotokolle

# Fehlerbehandlung durch explizite Wiederholungsanforderung

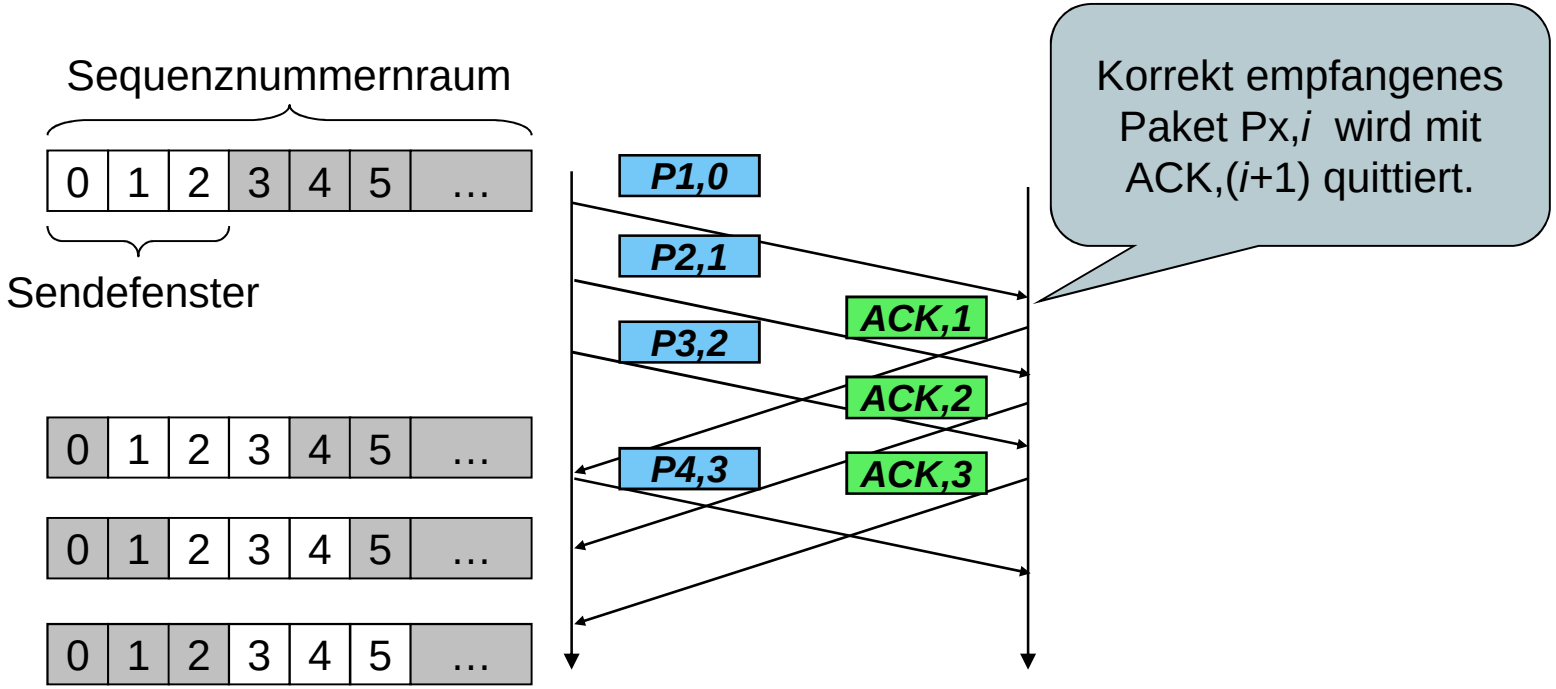
- Um den Ablauf der Übertragungswiederholung zu beschleunigen können fehlerhafte Pakete explizit durch negative Quittungen (NAK - Negative Acknowledgement) angefordert werden.




⇨ Wichtige Optimierung für Kanäle mit großen Verzögerungsschwankungen

# 3.3.2. Schiebefenster (Sliding Window)

- **Ziel:**
  - Höherer Durchsatz durch Zulassung mehrere unbestätigter Pakete
- **Schiebefenster:**
  - Sequenznummernraum {0; 1; …; m-1}
  - Sender darf Sequenznummern aus vorgegebenem Sendefenster verwenden, ohne auf eine Empfangsbestätigung zu warten
  - Empfänger bestätigt Empfang mit der **nächsten erwarteten Sequenznummer**



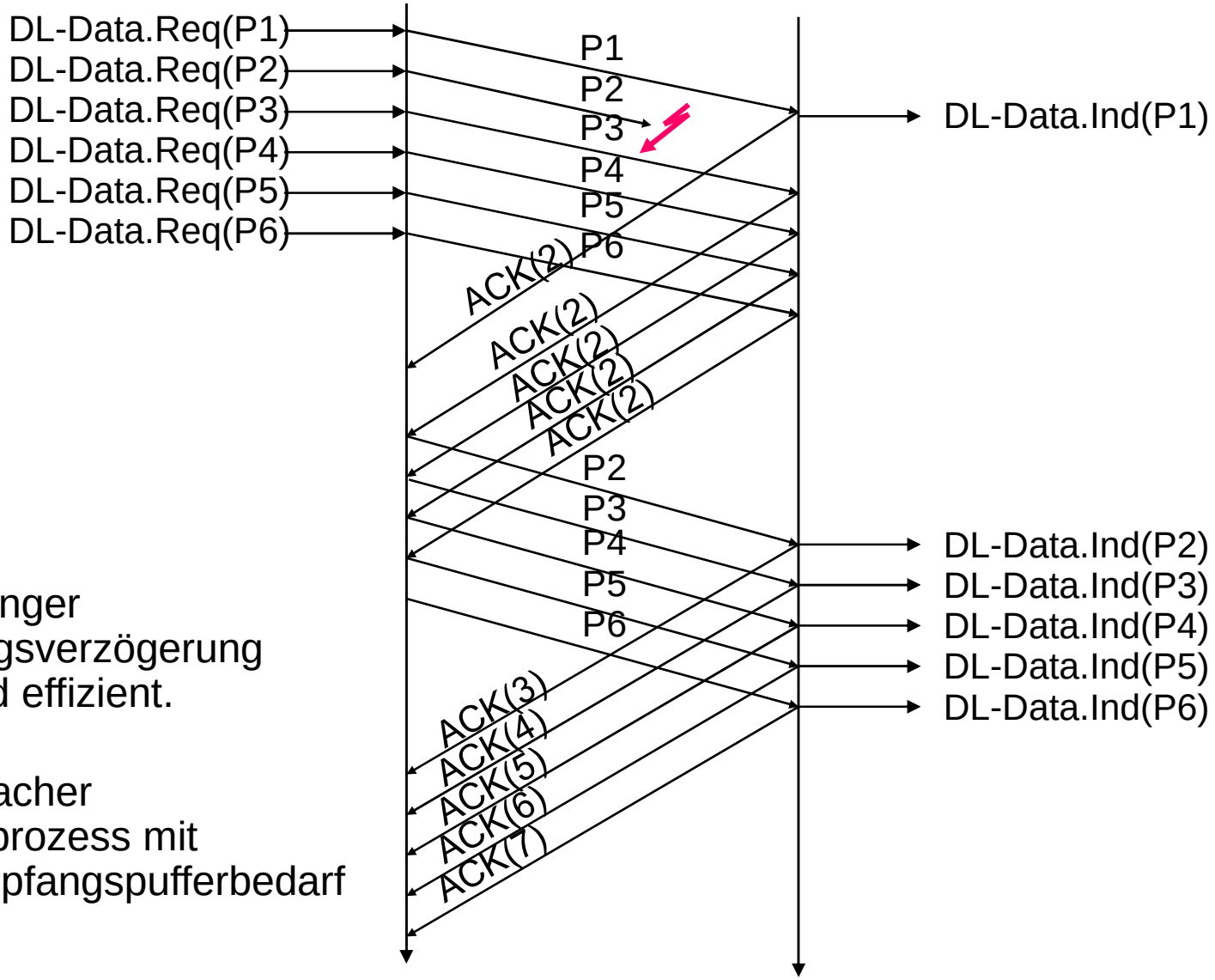
# Fehlerbehandlung bei Schiebefensterprotokolle

- **Implizite Wiederholungsanforderung:**
  - ausbleibende Empfangsbestätigungen (Zeitüberschreitung)
  - wiederholte Empfangsbestätigung für ein vorangegangenes Paket

- **Explizite Wiederholungsanforderung:**
  - Empfänger fordert Wiederholung eines bestimmten Pakets mit negativer Quittung

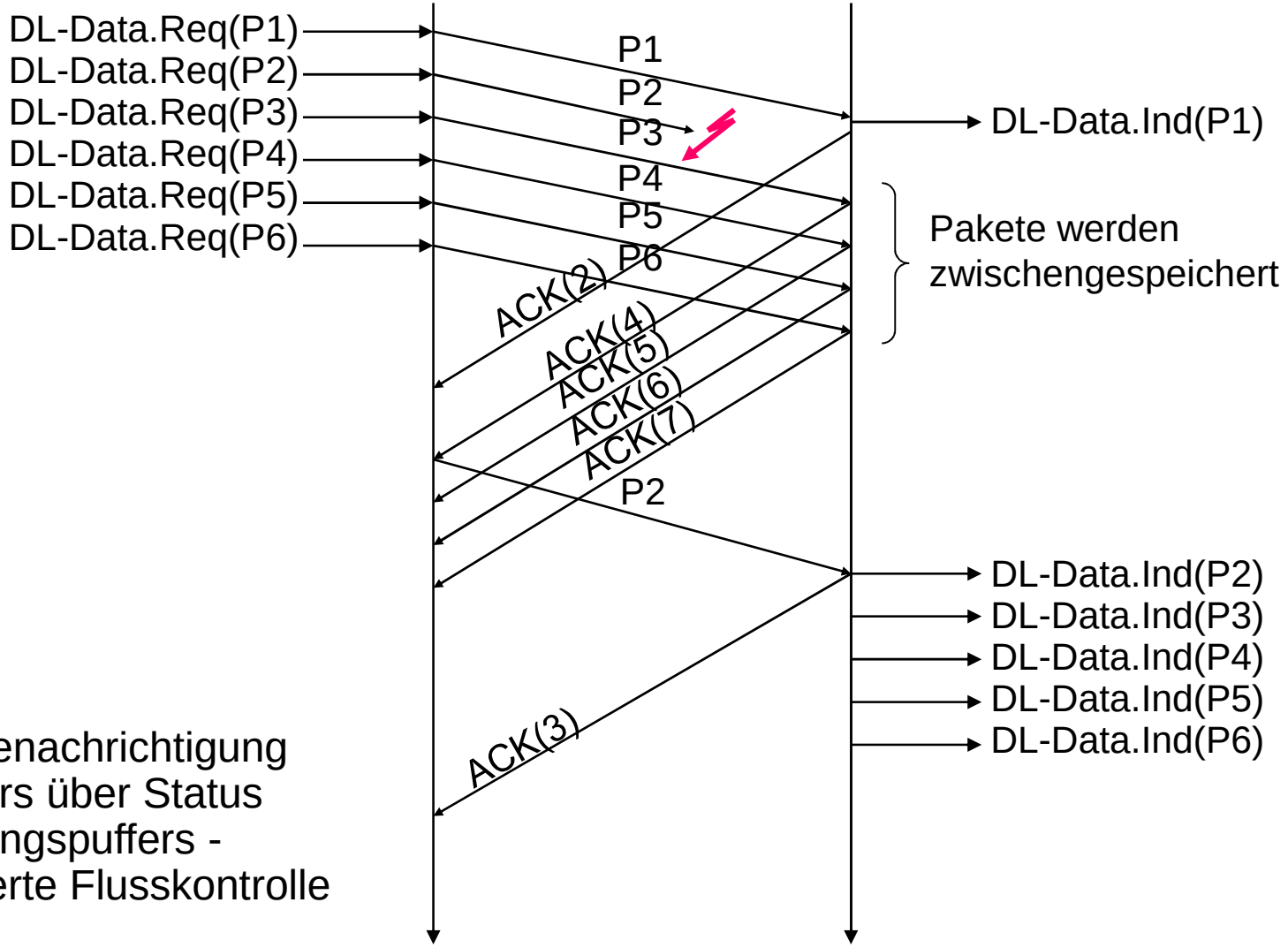
- **Fehlerbehandlungsverfahren:**
  - Go-Back-N:
    - sämtliche Pakete ab der fehlerhaften Sequenznummer werden erneut übertragen
  - Selektive Repeat (selektive Wiederholung):
    - nur das als fehlerhaft angegebene Paket wird erneut übertragen
    - Empfänger muss Pakete, die außer der Reihe ankommen, zwischenspeichern

# Fehlerbehandlung: Go-back-N




- ⇨ Nur bei geringer Übertragungsverzögerung ausreichend effizient.
- ⇨ Vorteil: einfacher Empfängerprozess mit kleinem Empfangspufferbedarf

# Fehlerbehandlung: Selektive Wiederholung

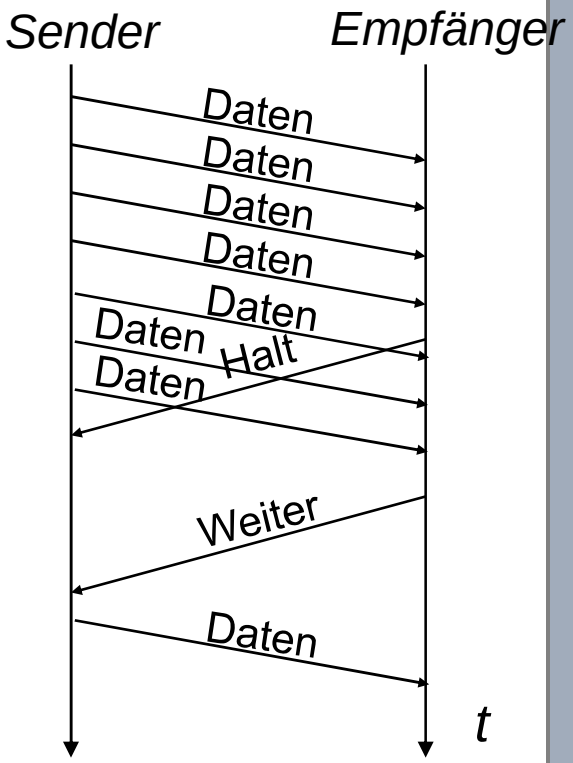



⇨ Wichtig: Benachrichtigung des Senders über Status des Empfangspuffers - Kreditbasierte Flusskontrolle

# Flusssteuerung mit Halt-/Weiter-Meldungen (Stop-and-Wait)

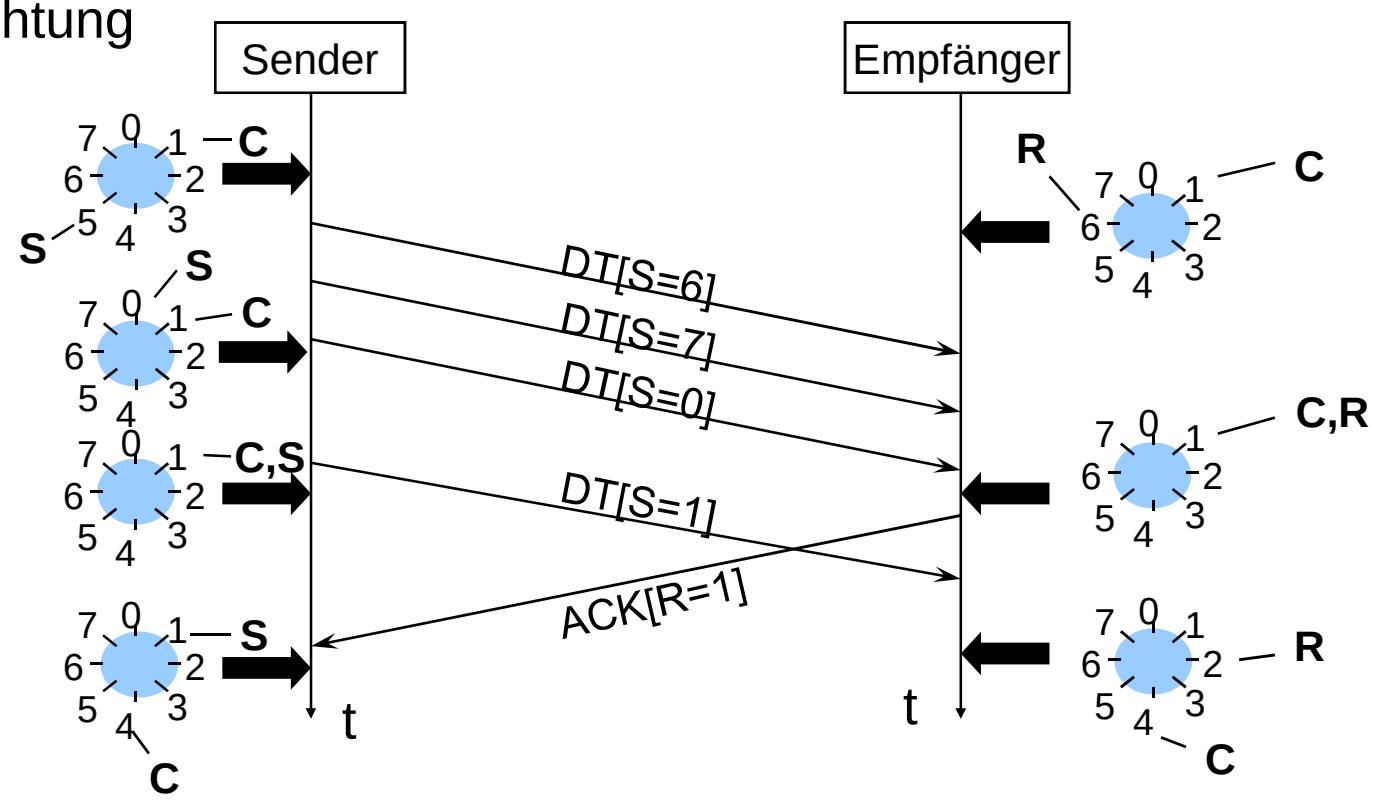
- Einfachste Methode
  - Sender-Empfänger-Flusssteuerung
    - Meldungen
      - Halt
      - Weiter
  - Kann der Empfänger nicht mehr Schritt halten, schickt er dem Sender eine Halt-Meldung.
  - Ist ein Empfang wieder möglich, gibt der Empfänger die Weiter-Meldung.

- Beispiel: Protokoll XON/XOFF
  - Mit ISO 7-Bit-Alphabetzeichen.
  - XON ist DC1 (Device Control 1).
  - XOFF ist DC3 (Device Control 3).
  - Nur auf Vollduplex-Leitungen verwendbar.



# Kreditbasierte Flusssteuerung: Sliding Window

- Darstellung zeigt Fenstermechanismus (Kredit 4) für eine Senderichtung

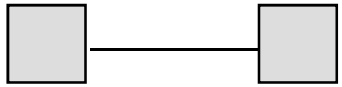



S: Sendefolgenummer (des zuletzt gesendeten Pakets)
R: Nächste erwartete Sendefolgenummer = Quittierung bis Folgenummer R-1
C: Oberer Fensterrand (maximal erlaubte Sendefolgenummer)

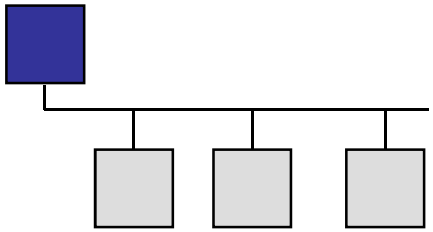
*Nachteil:* Kopplung von Fluss- und Fehlerkontrolle.

# Zugriff geteiltes Medium — Arbitrierung

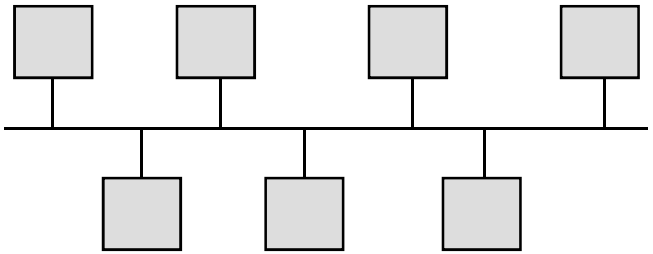
- Medienzugriff: Wer darf wann senden?
  - Zwei Kommunikationspartner, Punkt-zu-Punkt-Verbindung
    - Halbduplex „Richtungskonkurrenz“



  - Mehrere Kommunikationspartner, Mehrpunktverbindung
    - Stark von Topologie der Vernetzung abhängig
    - Gemeinsames Medium (*shared medium*)
    - Wichtiger Fall: Bus (Grenzfall der Baumtopologie)



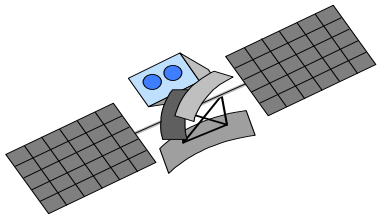
asymmetrisch



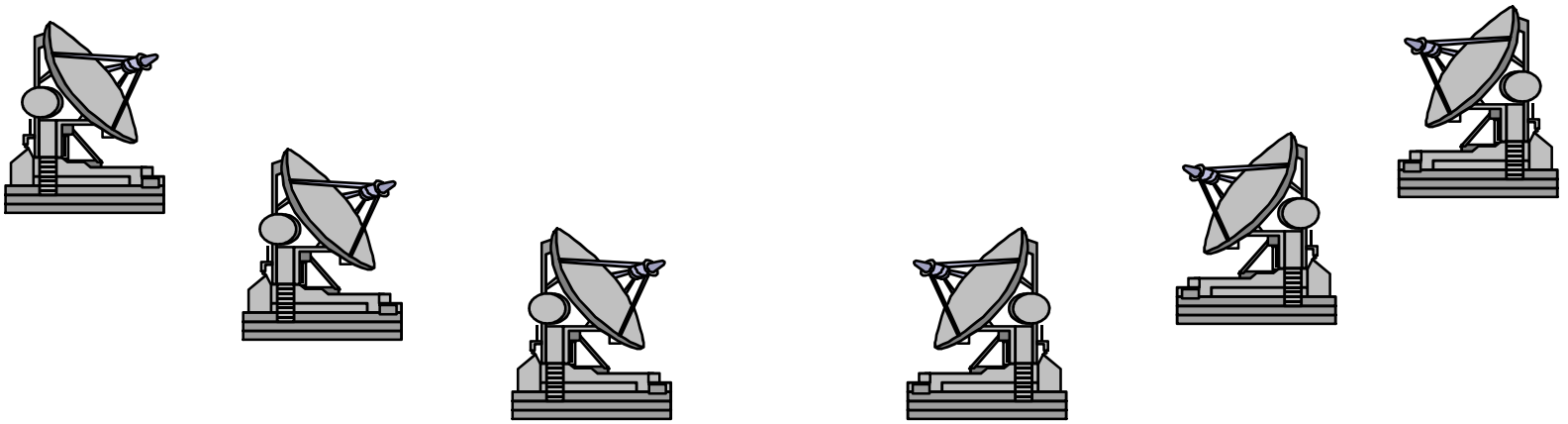
symmetrisch

    - Weiterer Fall: Ring

# Zugriffsverfahren auf ein Medium

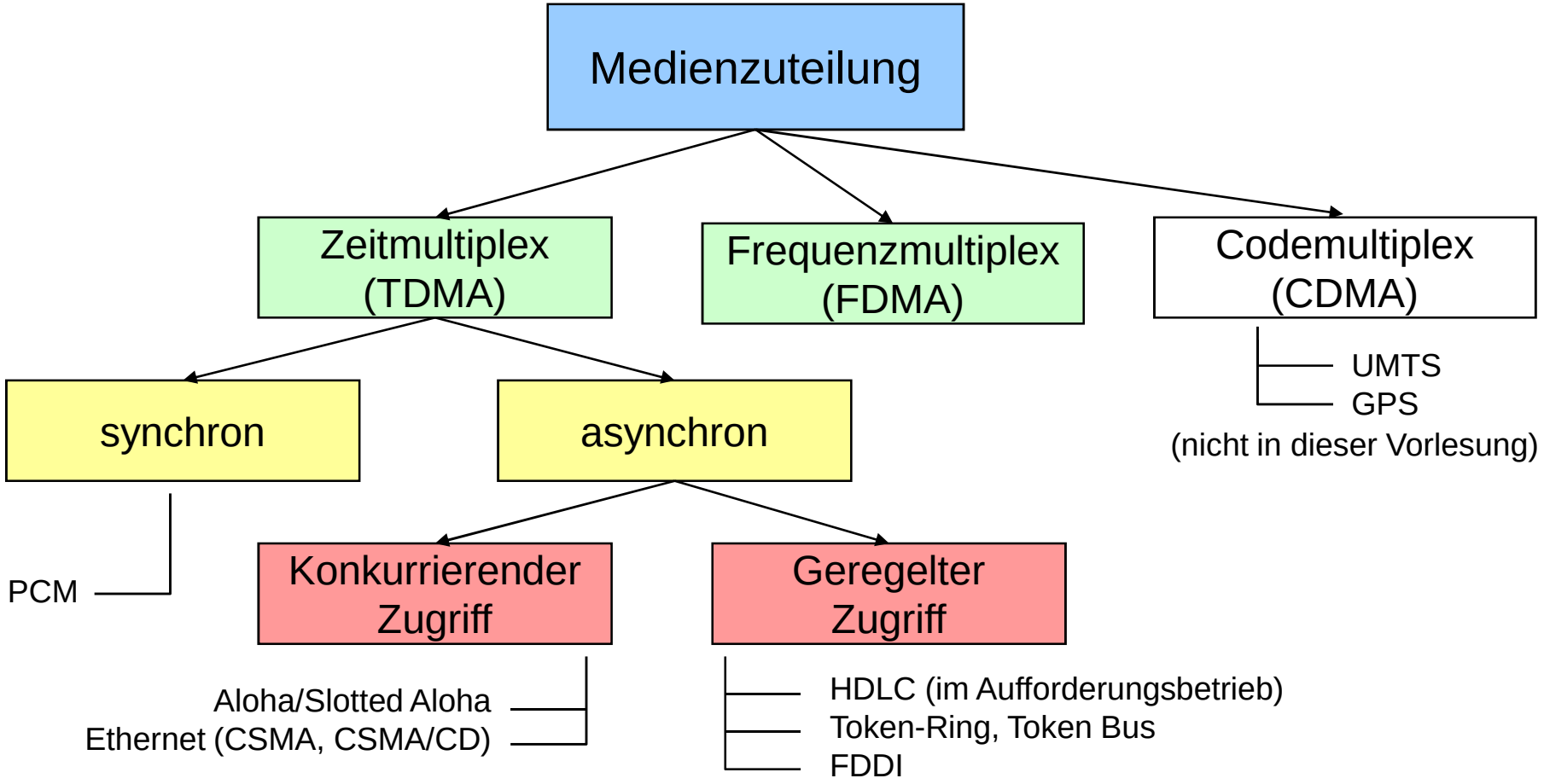


?

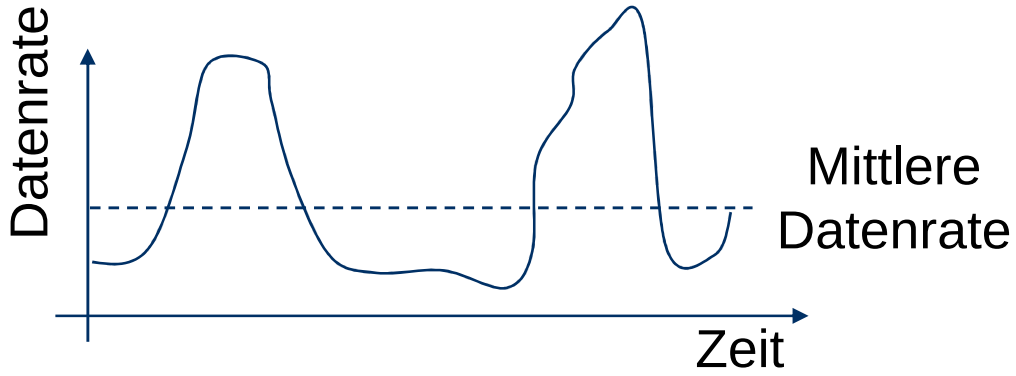


# 3.4. Zugriffsverfahren

- Szenario: Mehrere Stationen treten als Dienstnehmer eines einzigen physikalischen Mediums auf (shared medium)



# Charakteristik des Datenverkehrs




- Datenverkehr ist häufig stoßartig
  - Verhältnis Spitzenlast zu mittlerer Last?

# Durchsatz von Medienzugriffsverfahren

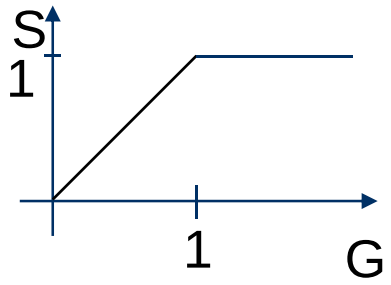
- Bewertung eines Medienzugriffsverfahrens

  Mögliche Ziele:

  - Möglichst viele Rahmen pro Zeiteinheit über den Kanal übertragen können (auch bei zahlreichen Sendeeinheiten, die gleichzeitig senden wollen)
  - Möglichst geringe Verzögerung für einzelne Rahmen (auch bei geringer Last)
  - Fairness (werden alle potentiellen Sender gleich behandelt?)

- Durchsatz in Abhängigkeit der Last

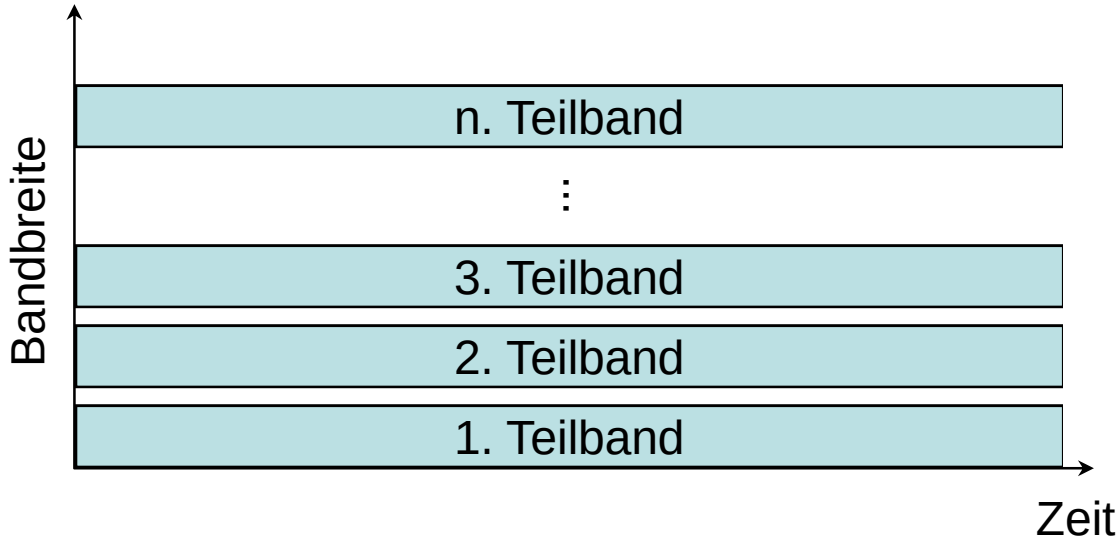
  - Last G: Anzahl Rahmen pro Zeiteinheit, die dem Protokoll an der Dienstschnittstelle zur Übertragung übergeben werden
  - Bei mehr als einem Rahmen pro Zeitabschnitt: Überlast
  - Ideales Medienzugriffsverfahren:
    - Durchsatz S = Last G für G<1
    - Durchsatz S = 1 für G ≥ 1

    - Sowie: gleichbleibend geringe Verzögerung für beliebige Anzahl Sender
  - Eigenschaften eines realen Medienzugriffsverfahrens?

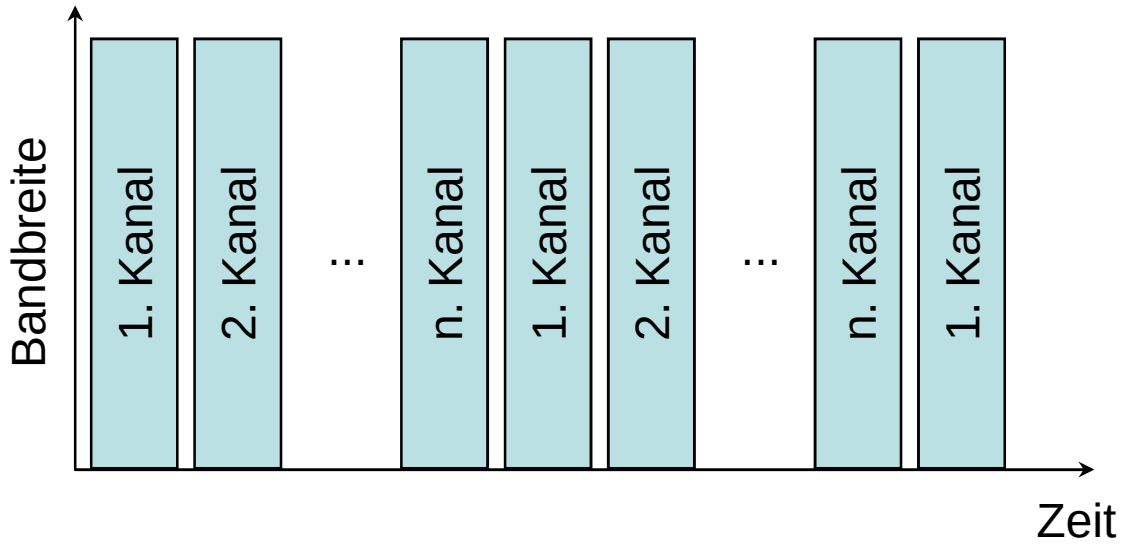
# Zugriffsverfahren FDMA

- Frequency Division Multiple Access
  - Aufteilung des Frequenzspektrums, z.B. eines Satellitenkanals, in Unterkanäle
  - Jeder Teilnehmer erhält einen Unterkanal
  - Übertragung beliebig digital oder analog

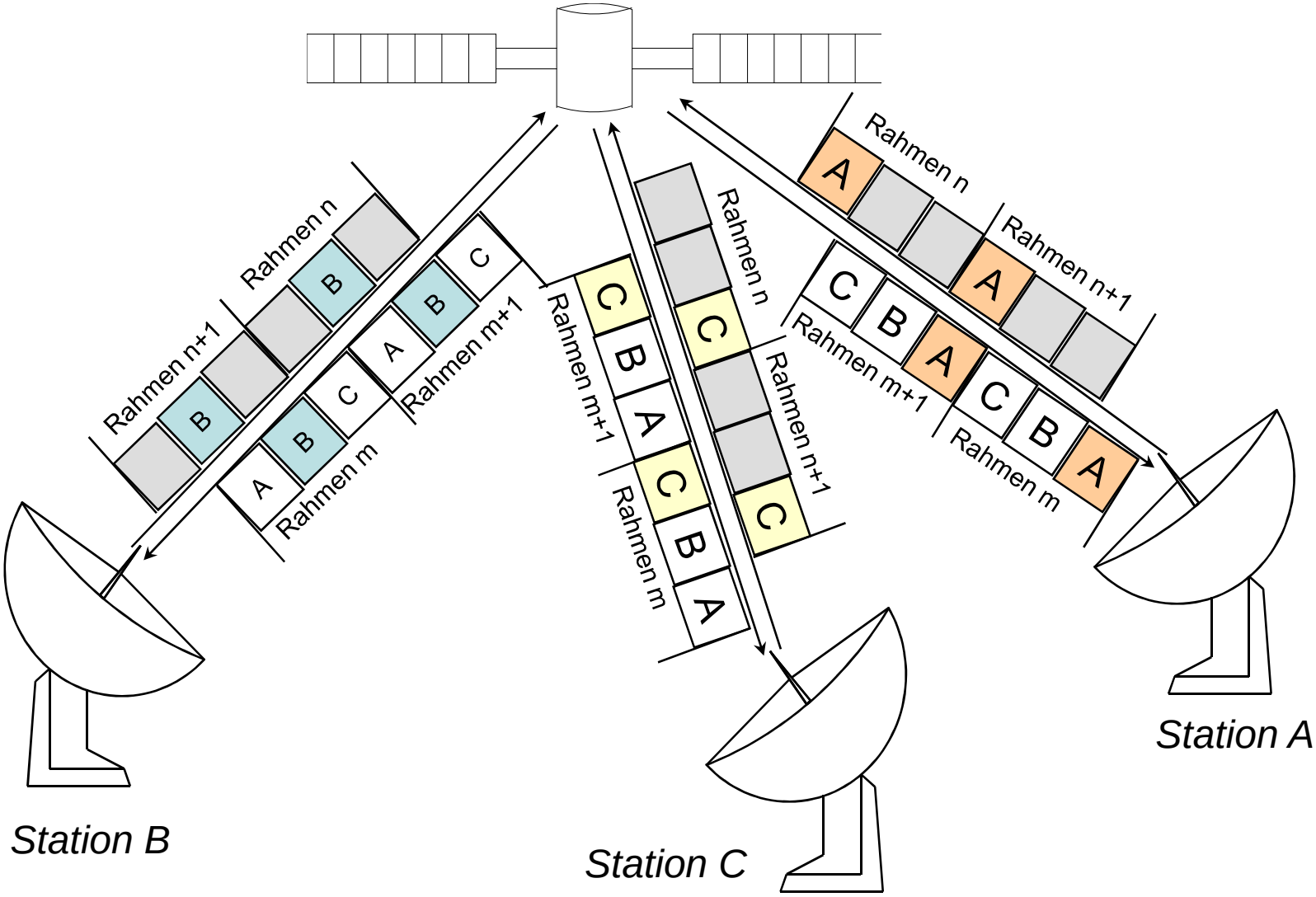


# Zugriffsverfahren TDMA

- Time Division Multiple Access
  - Synchrones Zeitmultiplex, Aufteilung der Kanalkapazität nach festen Intervallen
  - Jeder Sender bekommt zyklisch einen Zeitschlitz zugewiesen
  - TDMA-Systeme arbeiten grundsätzlich digital

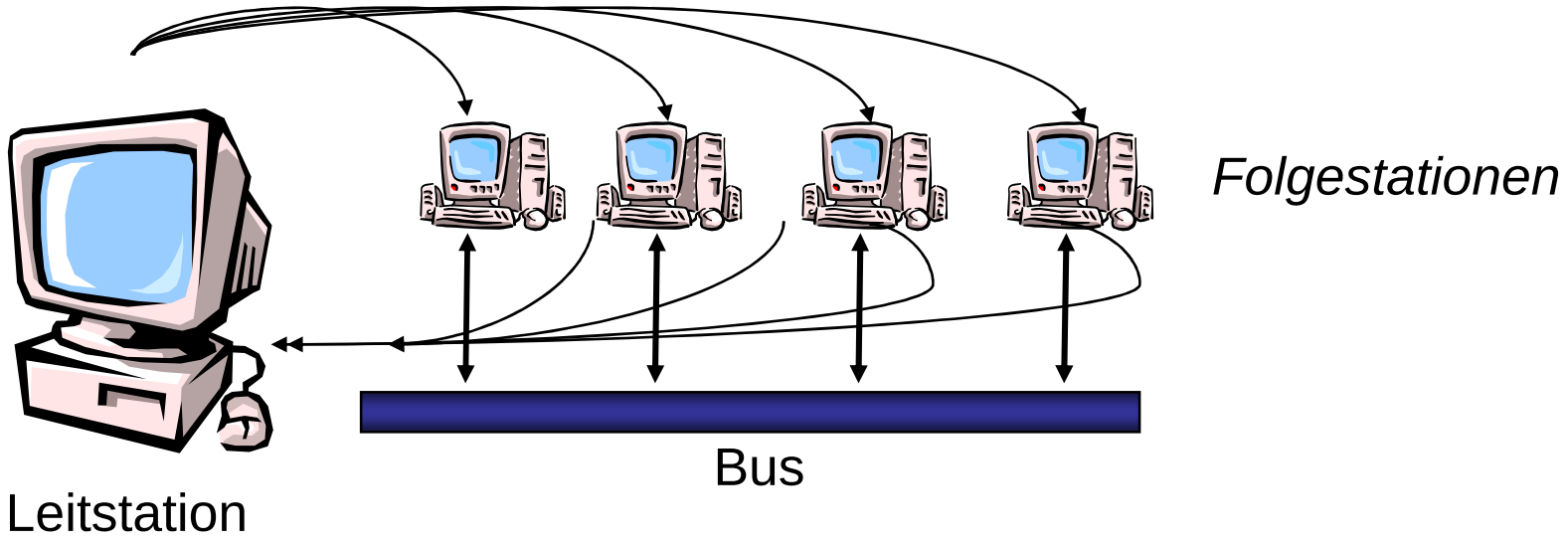


# TDMA-Schema


Rahmen n
Rahmen n+1
Rahmen m
Rahmen m+1
A
B
C
Station A
Station B
Station C

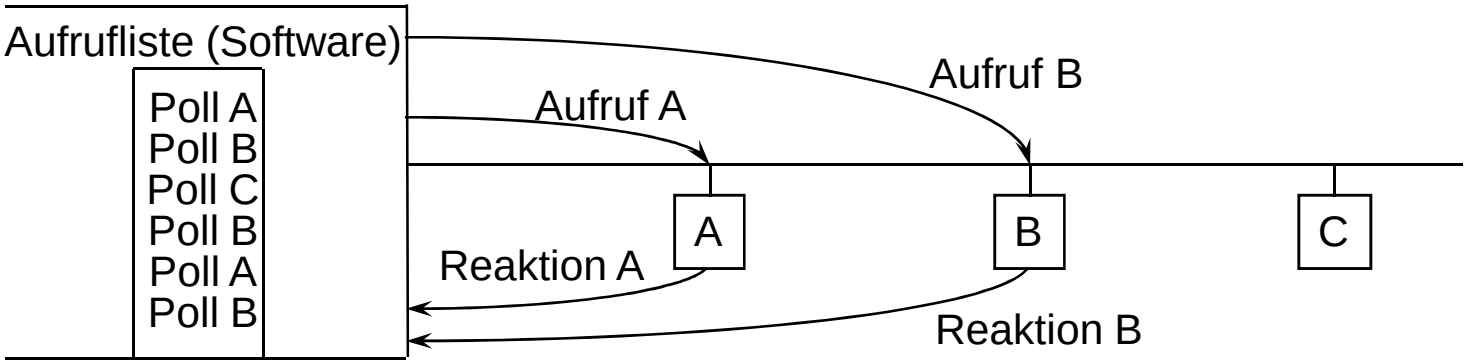
# Geregelter Zugriff: Aufrufbetrieb (Polling)

- asynchroner, geregelter Medienzugriff
- eine dedizierte „intelligente“ Leitstation
- u.U. mehrere „dumme“ Folgestationen
- gekoppelt über Busstruktur
- Leitstation fragt Folgestationen gemäß Abfragetabelle („Polling Table“) ab
- Folgestationen antworten nur nach Aufforderung
- jegliche Kommunikation erfolgt über Leitstation

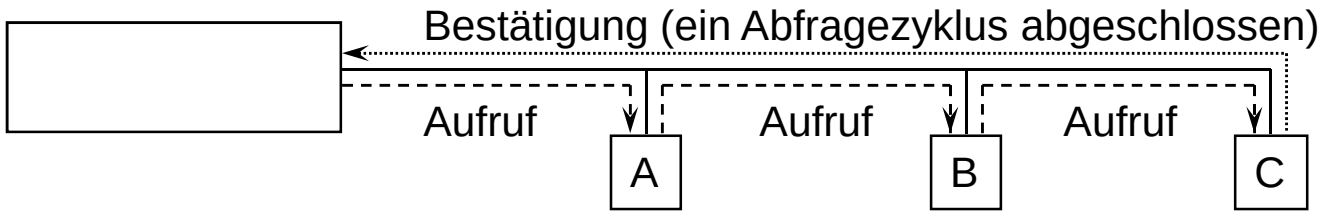


# Zentrale Zuteilungsprotokolle

- Aufrufbetrieb (Poll/Select, Roll call polling), z.B. Universal Serial Bus (USB), Kabelmodems




- Variante: Go-Ahead-Polling
  - Der von der Leitstation initiierte Sendeaufruf wandert von Folgestation zu Folgestation.




- Variante: Polling mit gemeinsamer Busleitung
  - Folgestationen teilen ihren Sendewunsch über eine gemeinsame Sammelleitung (Bus Request) der Leitstation mit.

# Dezentrale Zuteilungsprotokolle

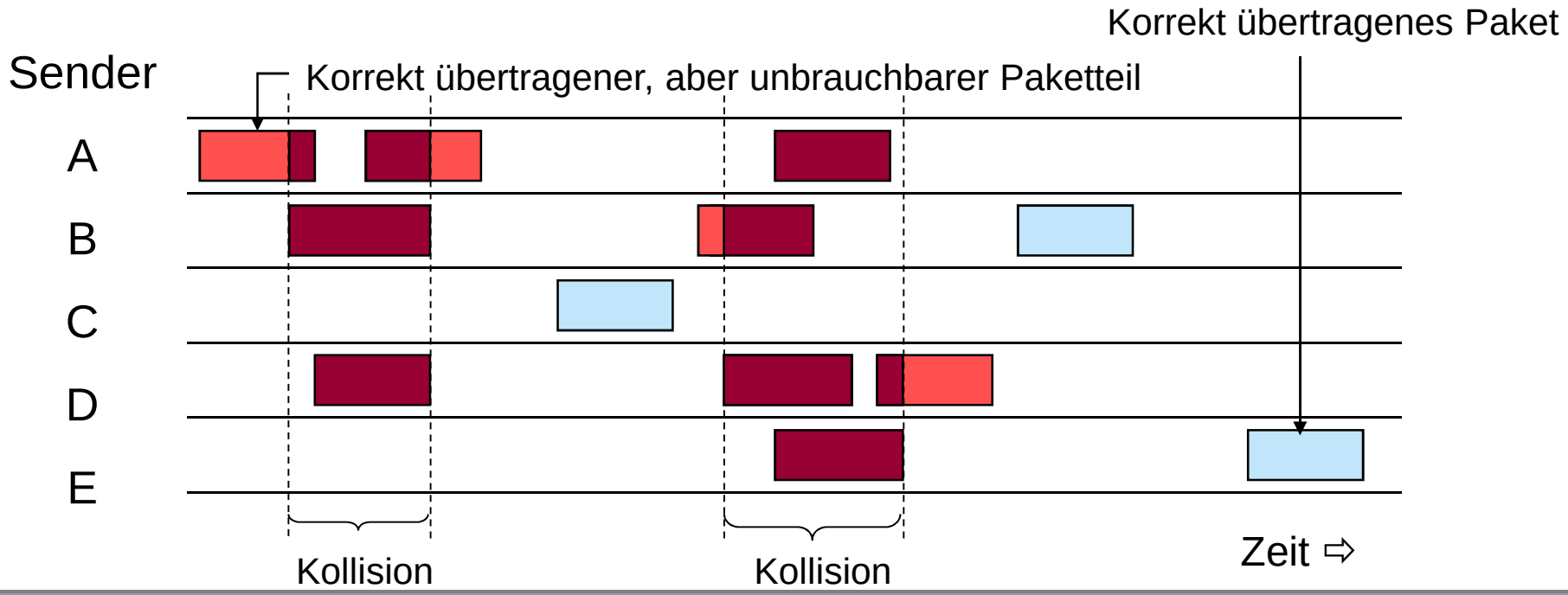
- *Zyklische Buszuteilung* (Token Passing)
  - geregelter Zugriff
  - Senderecht wird zyklisch unter den Stationen durchgereicht
  - Bsp. Token Bus, Token Ring

- *Konkurrenzbetrieb* (contention procedure)
  - konkurrierender Zugriff
  - dezentrales Wettbewerbsverfahren
  - Einsatz bei Punkt-zu-Punkt-Verbindung und Bus-basierten LAN
  - Bsp. Aloha, CSMA, CSMA/CD

- Dezentrale Protokolle werden im Folgenden ausführlicher behandelt.

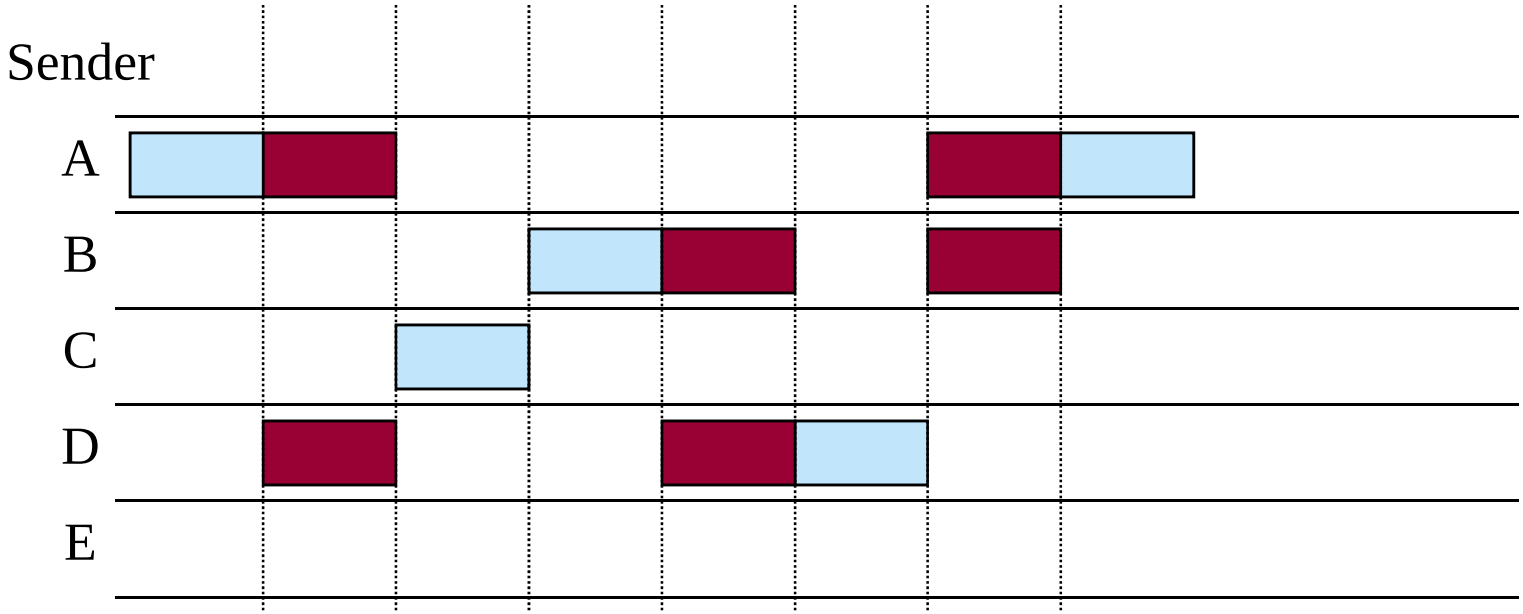
# Konkurrierender Zugriff: Aloha

- Norman Abramson, University of Hawaii (1970)
- Stationen übertragen dann Daten, wenn welche gesendet werden müssen.
- Kollisionen führen zu gestörten Rahmen.
- Empfänger schickt Quittungen, wenn er einen an ihn adressierten Rahmen korrekt empfangen hat. (Auch Quittungen können Kollidieren…)
- Einsatz z.B. bei GSM (Signalisierungskanal)
- Feste Rahmengröße vorgeschrieben, um möglichen Datendurchsatz zu erhöhen
- Maximale Kanalauslastung 18%.



# Konkurrierender Zugriff: Slotted ALOHA

- Larry Roberts, 1972
- Pakete fester Länge werden in festen Zeitabschnitten (Slots) übertragen. Dies erfordert einheitliche Zeitbasis (z.B. durch zentrale Uhr) zur Synchronisation der Stationen
- Paketübertragung nur zu Beginn eines Zeitabschnitts (slot boundary). Es können nur total überlappende Kollisionen auftreten. Damit verkürzt sich die Kollisionszeit von zwei auf eine Paket-Übertragungszeit.
  → Maximale Kanalauslastung auf 36% verbessert!

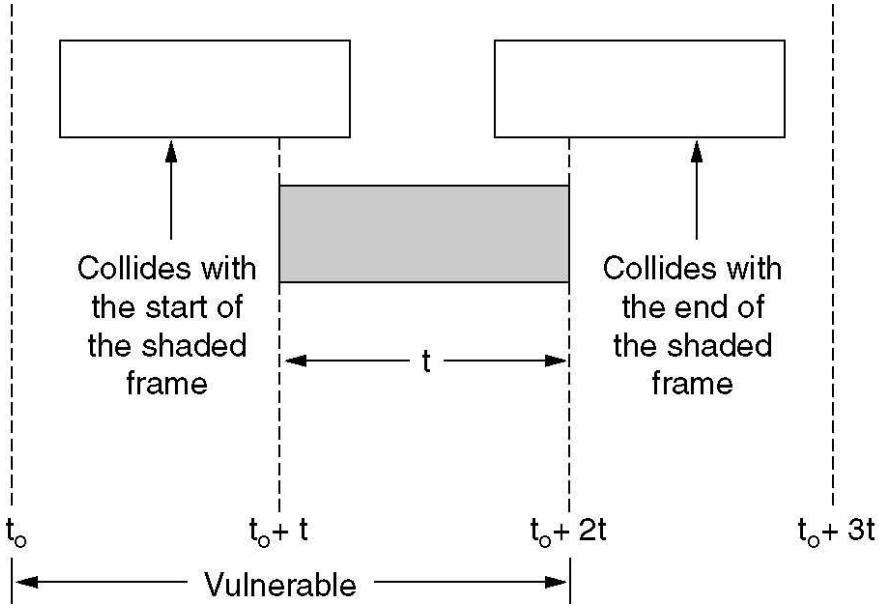


# Leistungsfähigkeit von ALOHA

- Annahme:
  - Unbegrenzte Anzahl von Sendestationen (die sich alle gleich verhalten)
  - Rahmen konstanter Größe
  - Rahmen-Generierungsereignisse haben Eigenschaften eines Poisson-Prozesses (gedächtnislos, Zwischenankunftszeiten exponential-verteilt) mit Mittelwert von N erstellten Rahmen pro Rahmenübertragungszeit
  - Aufgrund Kollisionen werden im Mittel G ≥ N Übertragungsversuche pro Zeiteinheit unternommen
  - Wahrscheinlichkeit $p_0$, dass ein Rahmen keine Kollision erleidet
    ⇨ Durchsatz $S = G\ p_0$
  - Verteilung der Anzahl von Rahmen pro Rahmenübertragungszeit ergibt sich durch Poisson-Verteilung
    - Poisson-Verteilung beschreibt die Wahrscheinlichkeit, dass in einem Zeitintervall eine Anzahl von *k* Ereignissen auftritt.
    - Bedingung: einzelne Ereignisse sind unabhängig voneinander
    - Im Mittel treten pro Zeiteinheit G>0 Ereignisse auf.
    - Wahrscheinlichkeit für das Auftreten von k Rahmen in einer Rahmenübertragungszeit durch Poisson-Verteilung: $\Pr[k] = \frac{G^k e^{-G}}{k!}$

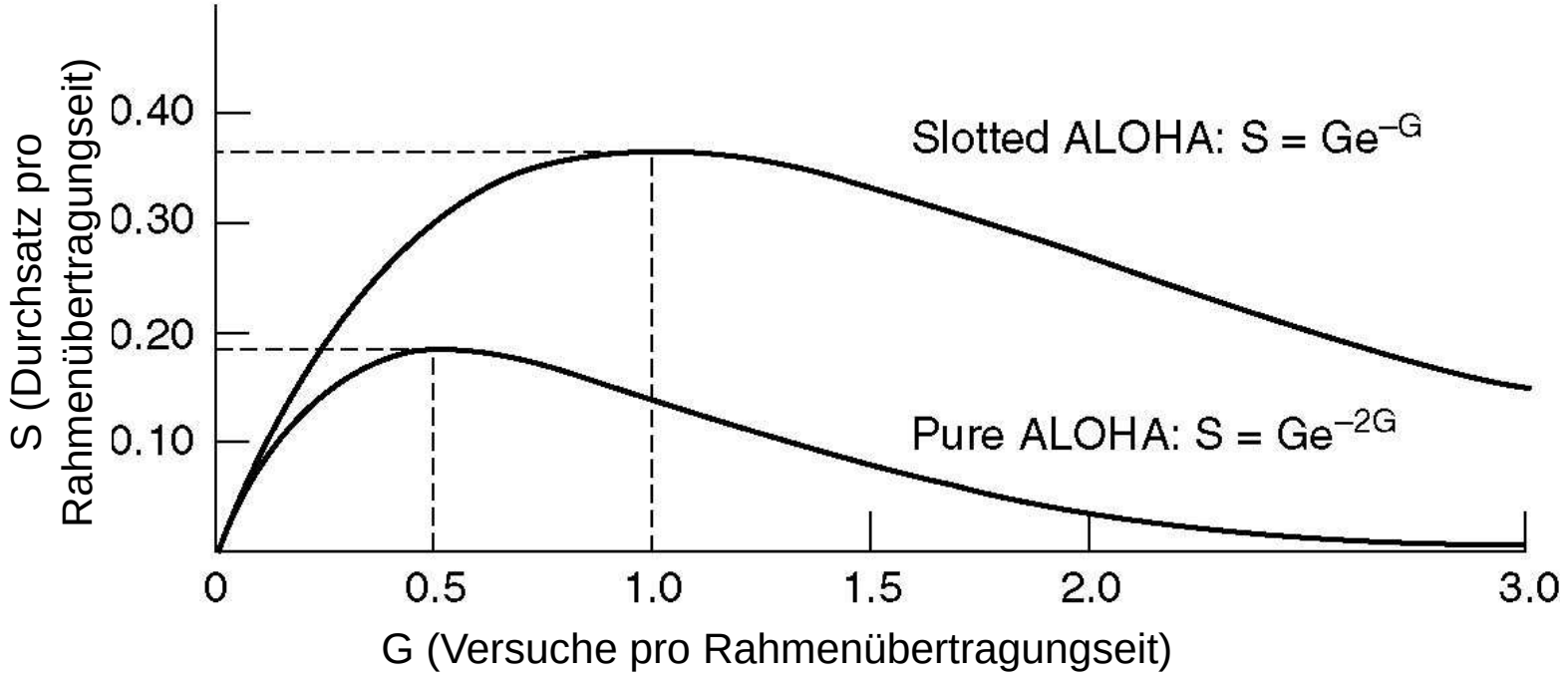
# Leistungsfähigkeit von ALOHA

- In 2 Rahmenübertragungszeiten werden im Mittel 2G Rahmen erzeugt
- Wahrscheinlichkeit, dass keine Kollision erfolgt: $P_0 = e^{-2G}$
- Damit: $S = GP_0 = Ge^{-2G}$

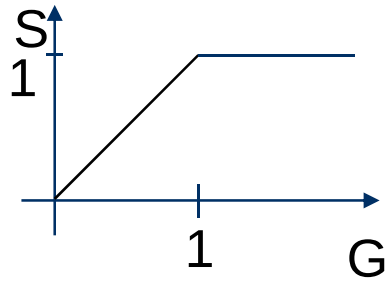


# Durchsatz in Abhängigkeit vom Verkehrsaufkommen

Annahme: Rahmenankünfte Poisson-verteilt, mit mittlerer Ankunftsrate G




Idealer Verlauf wäre:

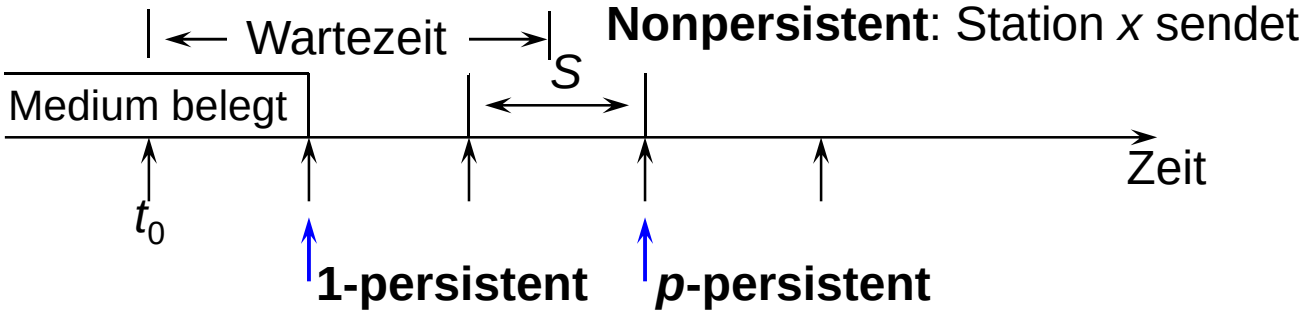


# Konkurrierender Zugriff: CSMA

- ⇨Ethernet - Bob Metcalfe, 1973
- Carrier Sensing Multiple Access:
  - Sender beginnt nur bei freiem Medium zu senden (Listen before talk)
    ➔ verringert Kollisionswahrscheinlichkeit gegenüber ALOHA
- non-persistent CSMA:
  - bei belegtem Medium schaut der Sender nach einer bestimmten Wartezeit wieder nach, ob das Medium frei ist
- 1-persistent CSMA:
  - bei belegtem Medium hört der Sender das Medium weiter ab, bis es frei wird. Danach beginnt er sofort zu senden.
  - Nachteil: sichere Kollision, wenn mehrere Sender senden möchten
- p-persistent CSMA:
  - bei belegtem Medium hört der Sender das Medium weiter ab, bis es frei wird. Danach beginnt er mit Wahrscheinlichkeit p zu senden oder wartet einen weiteren Zeitschlitz (um dann wieder nur mit Wahrscheinlichkeit p zu senden usw.).
  - geringere Kollisionswahrscheinlichkeit als bei 1-persistent
  - Kollisionen sind trotzdem nicht vollständig ausgeschlossen!

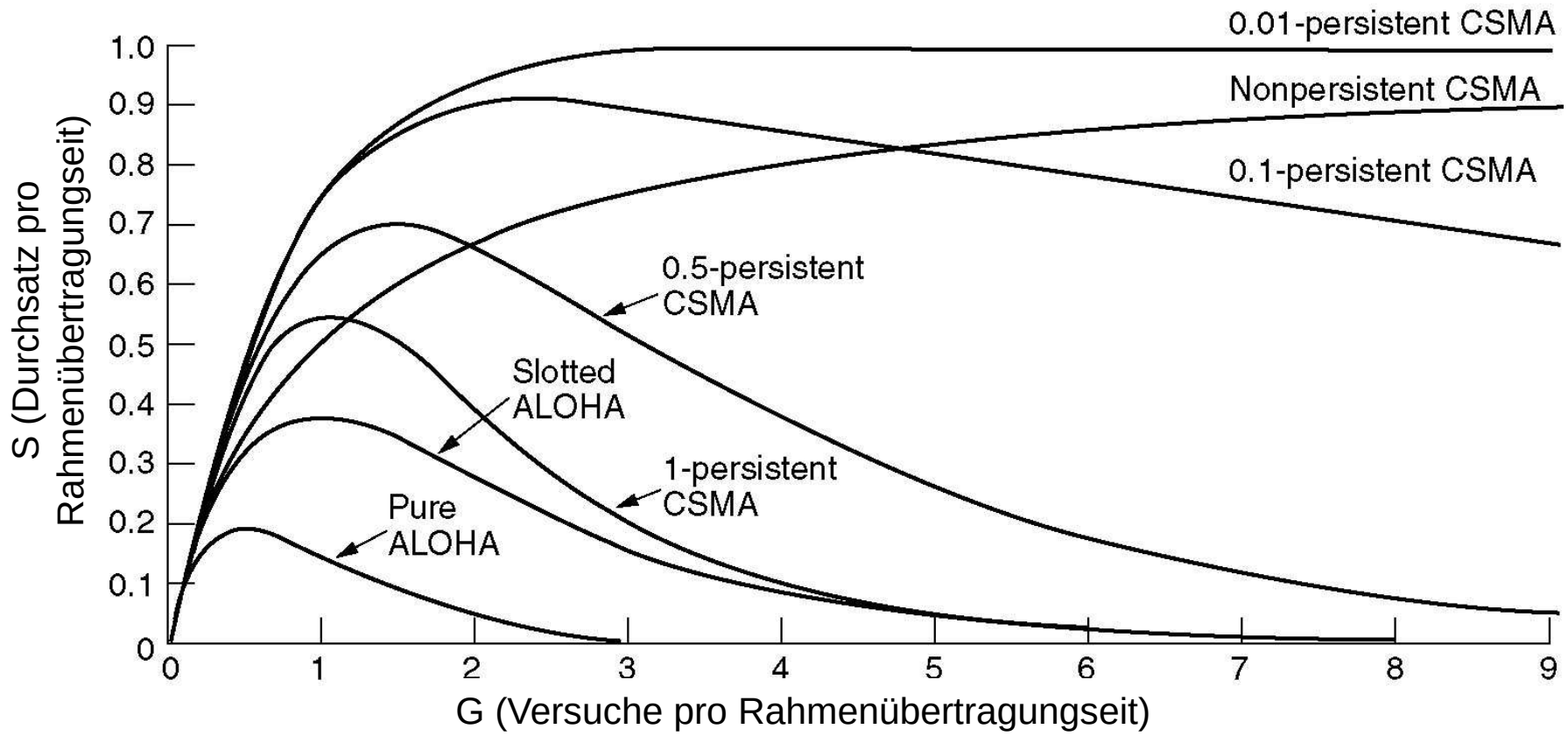
# Konkurrierender Zugriff: CSMA

- **Carrier Sense Multiple Access (CSMA)** bzw. **Listen before Talk (LBT)**
  - Sendewillige Station hört Medium ab und sendet, falls dieses frei ist
  - Erhöhte Kollisionsgefahr nach Ende einer Übertragung

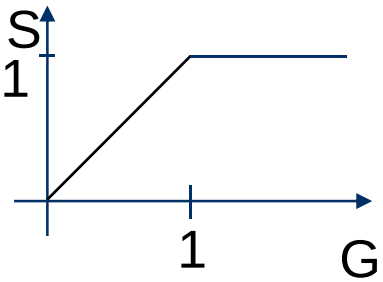



- **Nonpersistent CSMA:**
  (1) Wenn frei, übertrage sofort
  (2) Wenn belegt, warte gewisse (feste, zufällige) Zeit, dann (1)
- **1-persistent CSMA:**
  (1) Wenn frei, übertrage sofort
  (2) Wenn belegt, warte bis frei und übertrage
- ***p*-persistent CSMA:**
  (1) Wenn frei, übertrage mit Wahrscheinlichkeit *p* oder verzögere um einen Zeitslot mit 1 – *p*
  (2) Wenn belegt, warte bis frei, dann (1)
  (3) Wenn um 1 Slot (*S*) verzögert, dann (1)

# Durchsatz in Abhängigkeit vom Verkehrsaufkommen




Idealer Verlauf:



# CSMA: Mögliche Kollisionsbehandlung

- Behandlung auf höheren Protokollschichten:
  - fehlerhafte Übertragungen werden vom Empfänger erkannt bzw. ignoriert
    ➔ ausbleibende Empfangsbestätigungen und Sendewiederholung

- Kollisionserkennung (CSMA/CD, Collision Detection):
  - Sender hört während des Sendens das Medium weiter ab, um Kollisionen zu erkennen (Listen while talk)
  - bei erkannter Kollision sofortiger Abbruch des Sendevorgangs, ggf. Jamming-Signal zur Benachrichtung aller beteiligten Sender (siehe Ethernet)
  - Sendewiederholung nach zufälliger Wartezeit, um erneute Kollision zu vermeiden
  - Binary Exponential Backoff: Verdopplung der mittleren Wartezeit nach jeder erneuten Kollision

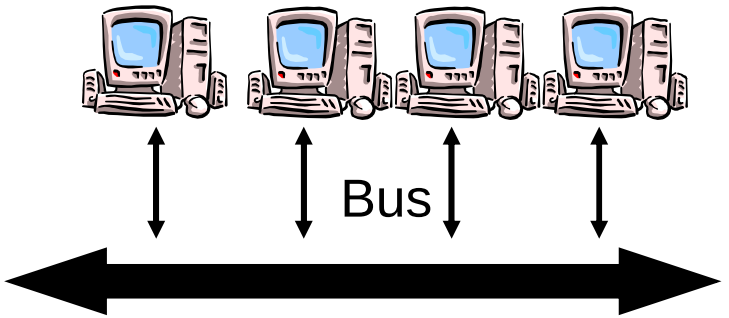
# CSMA/CD – Funktionsprinzip

- **Funktionsprinzip:**
  - Listen before Talk (*Carrier Sense Multiple Access*, CSMA)
  - Listen while Talk (*with Collision Detection*, CD)
  - Bustopologie mit Mehrfach-Zugriffsverfahren
  - Konkurrierendes Zugriffsverfahren (Wettbewerb)
- **Betriebsablauf:**
  - Senden, wenn Medium aktuell frei
  - Bei Kollision: Jamming-Signal; Abbruch der Sendung
  - Kollisiondetektion erfordert Mindestlänge der 802.3-MAC-Blöcke!
    → Sendung darf nach Signallaufzeit von A nach B und zurück noch nicht beendet sein
  - Nach Kollision erneuter Anlauf nach statistisch verteilter *p*-persistent-Verzögerungszeit
  - Blockmindestlänge hängt von Mediumslänge und Übertragungs-geschwindigkeit ab. Sie wird durch Stopffeld (Pad) sichergestellt

# CSMA/CD („Ethernet“)

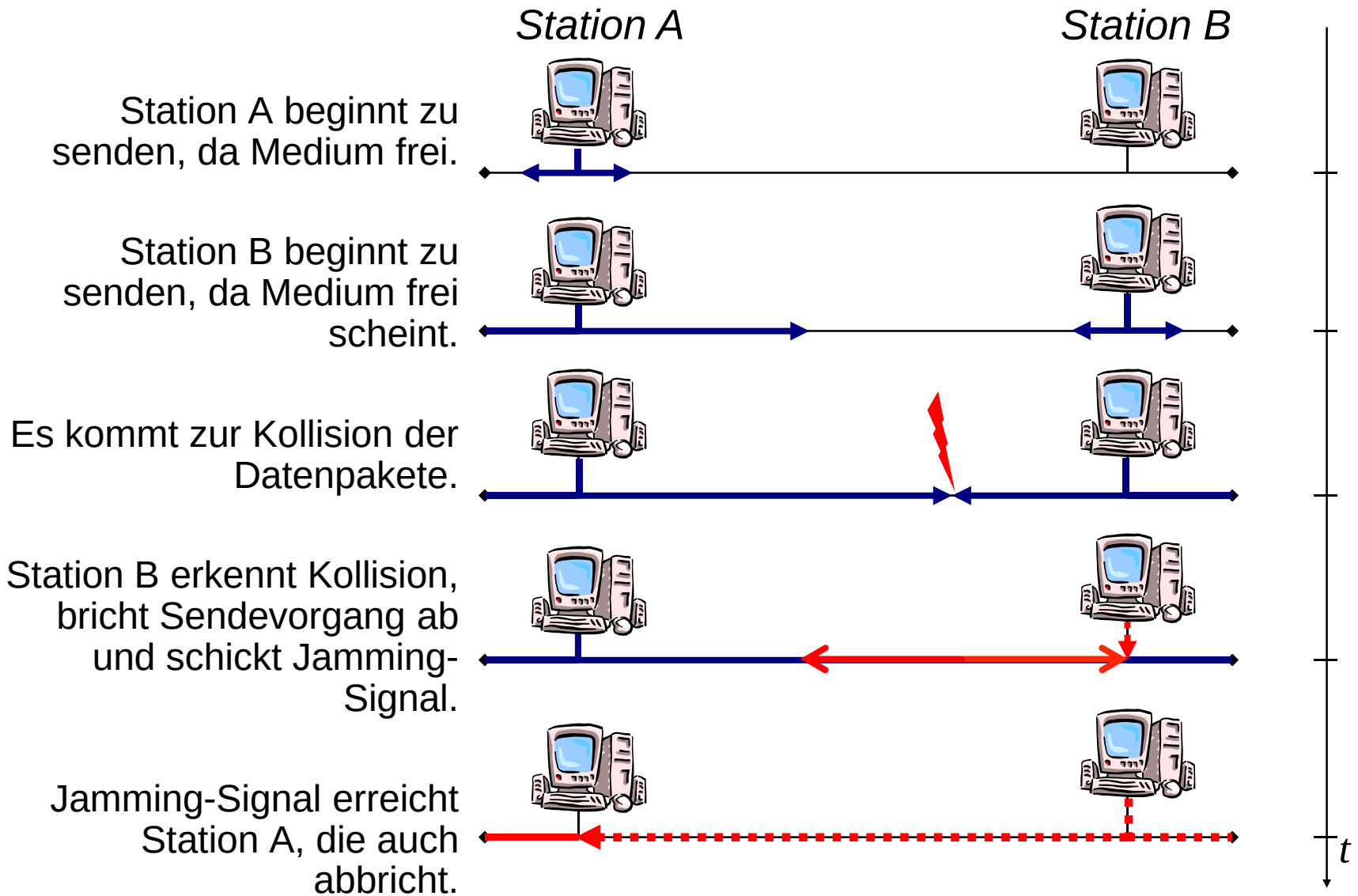
- alle Stationen an einem gemeinsamen Bus angeschlossen.
- keine ausgezeichnete Station.
- jede Station kann zu einem beliebigen Zeitpunkt senden.
  ⇨ **Kollisionen mehrerer Sendungen zerstören übertragene Daten!**

- Vermeidung von Kollisionen:
  *Carrier Sense Multiple Access with Collision Detection (CSMA/CD).*




- Grundlagen von CSMA/CD:
  - vor dem Senden: Abhören des Mediums (*Listen Before Talk*).
  - wenn Medium frei: Beginne mit Senden.
  - während des Sendens: Abhören des Mediums (*Listen While Talk*).
  - wird Kollision erkannt: Breche Sendevorgang ab und benachrichtige die anderen angeschlossenen Stationen.

# Ablaufbeispiel CSMA/CD



# Paketformat CSMA/CD nach IEEE 802.3/Ethernet

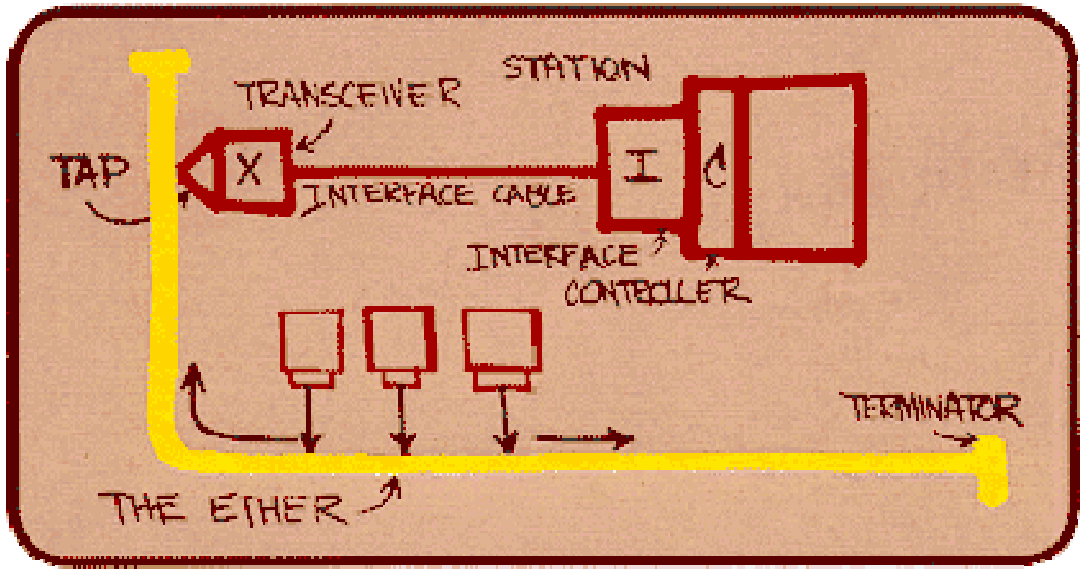
Typ

| PR<br>56 bit | SD<br>(8 bit) | DA<br>(16/48 bit) | SA<br>(16/48 bit) | Länge<br>(16 bit) | Data<br>(≤12.000 bit) | PAD<br>(0-368 bit) | FCS<br>(32 bit) |
|---|---|---|---|---|---|---|---|

PR = Präambel zur Synchronisation (1010101010...)
SD = *Start-of-frame Delimiter* zeigt Blockbeginn an (10101011)
DA = *Destination Address*, Zieladresse
SA = *Source Address*, Herkunftsadresse
Länge = Anzahl der Oktette im Datenfeld
Typ = Protokolltyp der Nutzdaten (z.B. IP, ARP, IPX...)
Data = Datenfeld, das maximal 1.500 Byte umfassen darf
PAD = *Padding*, um zu kurze Datenfelder auf die nötige Länge zu ergänzen
FCS = *Frame Check Sequence*, Polynomdivision mittels CRC32-Polynom zur Fehlererkennung

*Wichtig: Einzelne Realisierungen von CSMA/CD (z.B. Ethernet 1.0, Ethernet 2.0 oder IEEE 802.3) verwenden manche Felder in leicht unterschiedlicher Bedeutung!*

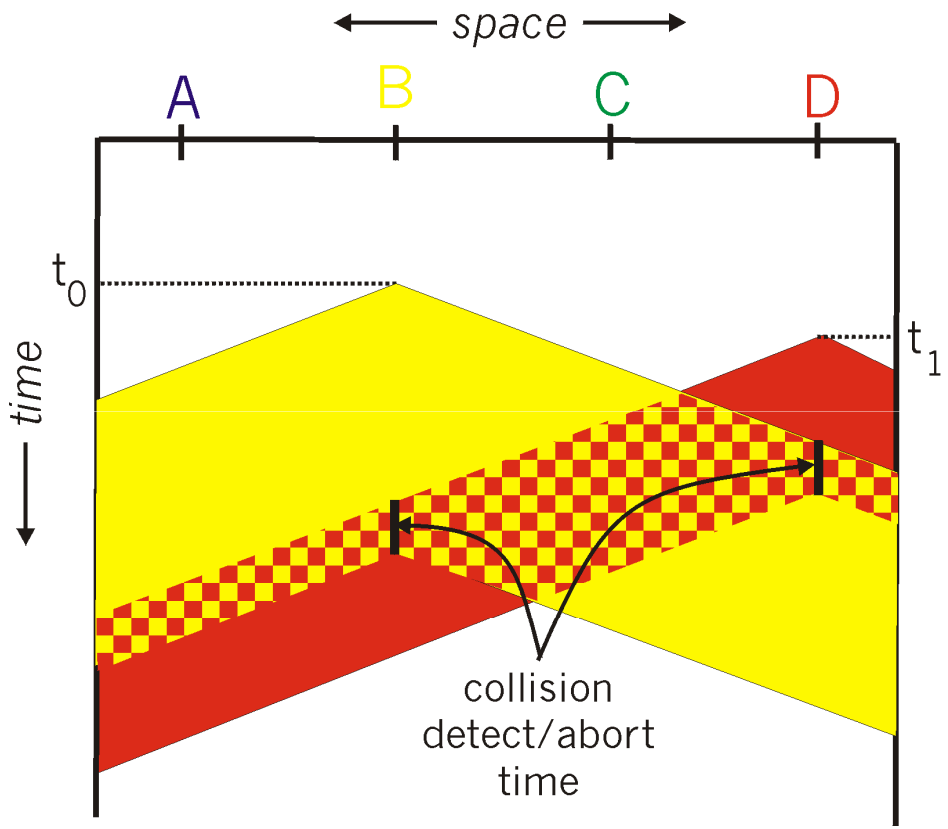
# Ethernet-Entwicklung




Ursprüngliche Skizze von Robert Metcalfe 1973,
Patentanmeldung 1975,
ACM-Veröffentlichung 1976
Doktorarbeit bei Xerox PARC - Xerox Palo Alto Research Center
c.f. parc.com

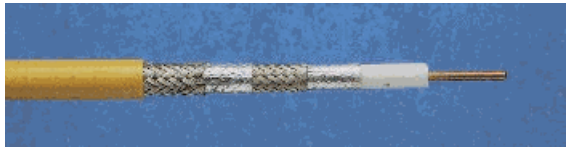
- Metcalfesches Gesetz:
  - Der Nutzen eines Kommunikationssystems wächst mit dem Quadrat der Anzahl der Teilnehmer
    (ebenso wie Anzahl Verbindungen in vollvermaschtem Netz)

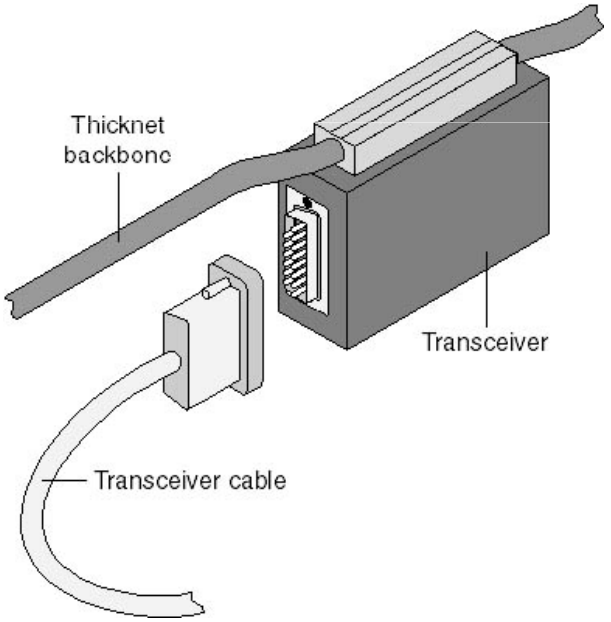
# CSMA/CD - Kollissionserkennung



# LAN: CSMA/CD – Technische Realisierung: 10Base5

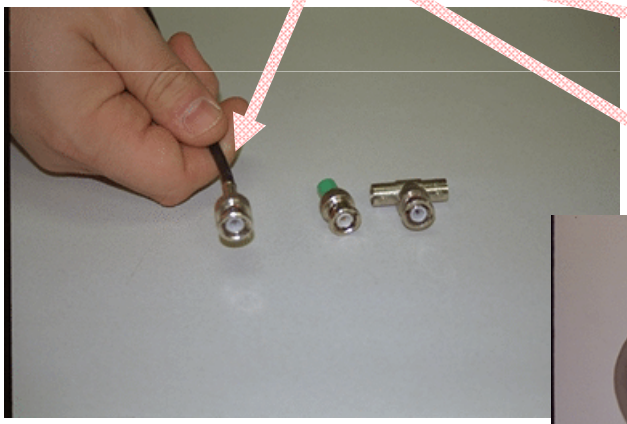
- Technische Realisierung: **Ethernet** (in zwei Versionen 1.0 und 2.0)
  - **10Base5, Thick Ethernet**
    - fingerdickes, gelbes, 4-fach abgeschirmtes Koaxialkabel
    - 10Base5: 10 Mbit/s, Basisbandübertragung, 500 Meter-Segmente
    - Segment-Kopplung über Repeater (max. 5 Segmente)
    - 100 Teilnehmer pro Segment mit mindestens 2,5 m Abstand
    - Teilnehmeranschluss über Transceiver (Transmitter & Receiver), entspricht MAU (Medium Attachement Unit). Transceiver enthält Sende-/Empfangslogik, Kollisionserkennung, „Carrier Sensing“-Funktion.
    - Ein-, Zwei- oder Vierfach-Transceiver
    - Transceiverkabel zum Teilnehmer bis 50 Meter
    - Lösung: robust, teuer, unflexibel

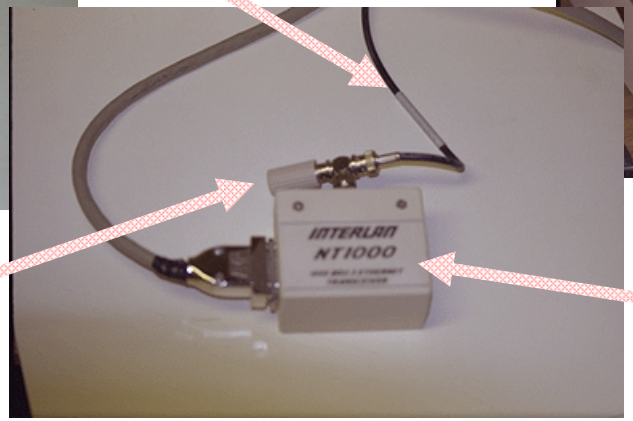


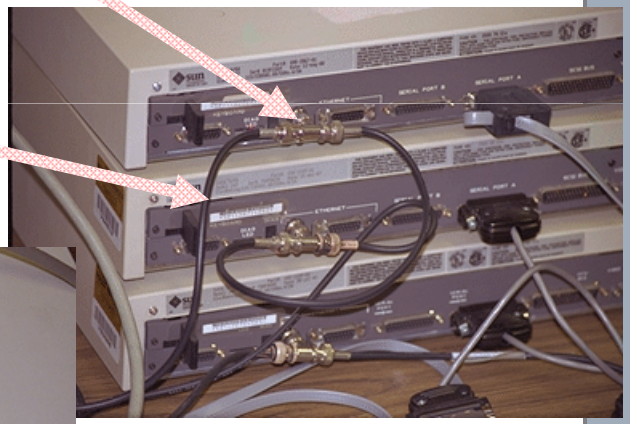


# LAN: CSMA/CD – Technische Realisierung: 10Base2

- **10Base2, Thin Wire Ethernet, Cheapernet**
  - 10Base2: 10 MBit/s, Basisbandübertragung, 185 Meter-Segmente
  - 30 Teilnehmer pro Segment im Abstand von mindestens 0,5 m
  - Transceiver meist direkt auf Ethernet-Adapter im Rechner (BNC-Buchse, T-Stück)
  - Koaxialkabel





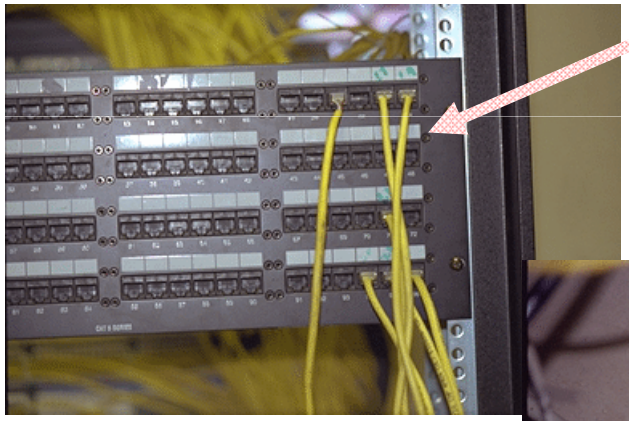



Abschlusswiderstand zur Signalvernichtung

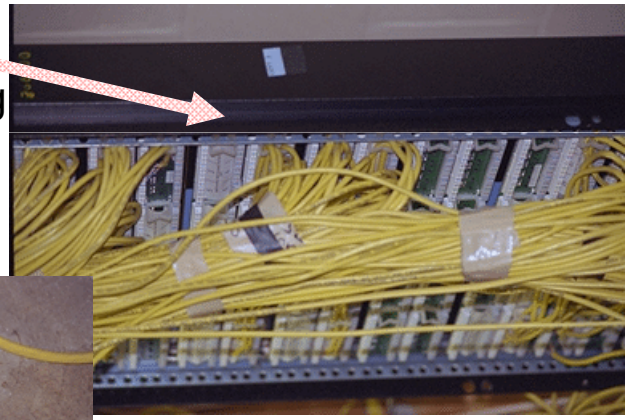
Transceiver

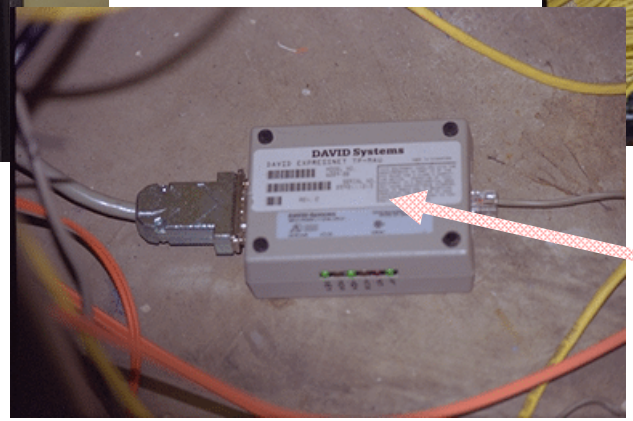
# LAN: CSMA/CD – Technische Realisierung: 10Base-T

- **10Base-T, Twisted Pair**
  - 10 MBit/s, Basisbandübertragung
  - Verdrillte Leitungen
  - jede Station ist über (max. 100 m) Punkt-zu-Punkt-Verbindung an Multiport-Repeater angeschlossen (Hub, Verteilerkasten, Konzentrator)
  - In USA Telefonkabel einsetzbar



„patch-panels“ für die Sternverkabelung






Transceiver

# LAN: 802.3 Ethernet

- Anschluss eines Transceivers an NIC über AUI-Kabel (AUI: Attachment Unit Interface)
- Tranceiver-Kabel: 15-poliger Mini-D-Sub-Stecker
- Funktionalität eines Tranceivers: Weiterleitung codierter Bitserieller Datenströme auf das Hauptkabel, Kollissionserkennung, Carrier-Sense
- Jam Signal: 48 bit lang
- Zeit für Senden eines Bit bei 10 Mbit/s Ethernet : 0.1µs
- IEEE 802.3: minimale Rahmenlänge 64 byte ⇨ 512 bit * 0.1µs = 51,2µs ⇨ bezogen auf Umlaufgeschwindigkeit ergibt sich max. Entfernung $0,5*51,2\mu s*2*10^8 m/s = 5300m$
- IEEE 802.3 10Base5: max. Entfernung zwischen zwei Stationen gemäß IEEE-Spezifikation: 2 Traceiver-Kabel (max. 50 m), drei Segementkabel (max. 500 m), zwei Link-Segmente (je max. 1000 m + 2*50 m), insg. 3800m

# LAN: CSMA/CD – Ausdehnung

Ausdehnung:

- Basisband-Ethernet beschränkt auf max. 1.500-2.500 Meter (inkl. Repeater)
- Größere Distanzen erreichbar mittels:
  - Remote Repeater (Fiber Optic Inter-Repeater Link FOIRL)
    Geteilter Repeater mit bis 1 km Glasfaserübertragungsstrecke
- Weitere Ansätze:
  - 10Base-F
    - Verbindung von Hubs (Sternkoppler) über Glasfaser
    - bis zu 4.000 Meter Glasfaserübertragungsstrecke
  - 10Broad36
    - 10 MBit/s, Breitbandübertragung (Ethernet über CATV-Leitungen), max. Entfernung: 3.600 Meter

# Fast-Ethernet-Standard 100Base-T (1)

- IEEE 802.3u: 100Base-T-Technologie
  - 1995 standardisiert („Fast Ethernet“)
  - Vertreter: Grand Junction Networks, Digital, Intel, SUN, Synoptics, 3Com,...
  - Charakteristika
    - Erhaltung des CSMA/CD-Frame-Formats und Medienzugriffverfahrens
    - Übertragungsraten von 10 und 100 Mbit/s
    - Flexibles Verkabelungskonzept (Hierarchie von Hubs)
    - Kompatibilität zum existierenden Ethernet-Standard ⇨ einfache Migration
    - „Autonegotiation“: Protokoll zur automatischen Festlegung der Übertragungsrate
  - Realisierung

    (a) 100Base-T4
    - Basierend auf Twisted-Pair-Verkabelung der Kategorie 3 (bis 16 MHz)
      - Maximal 100 Meter Kabellänge (zwischen Netzwerkkarte und Hub)
      - Daten werden bei der Übertragung auf 4 Adernpaare aufgeteilt (drei Adernpaare zur Informationsübertragung mit je 33 Mbit/s mit ternärem Code ⇨ Kapitel 11, viertes Adernpaar zur Kollisionsanzeige).
    - Nur Halbduplex-Verkehr möglich

# Fast-Ethernet-Standard 100Base-T (2)

(b) 100Base-Tx

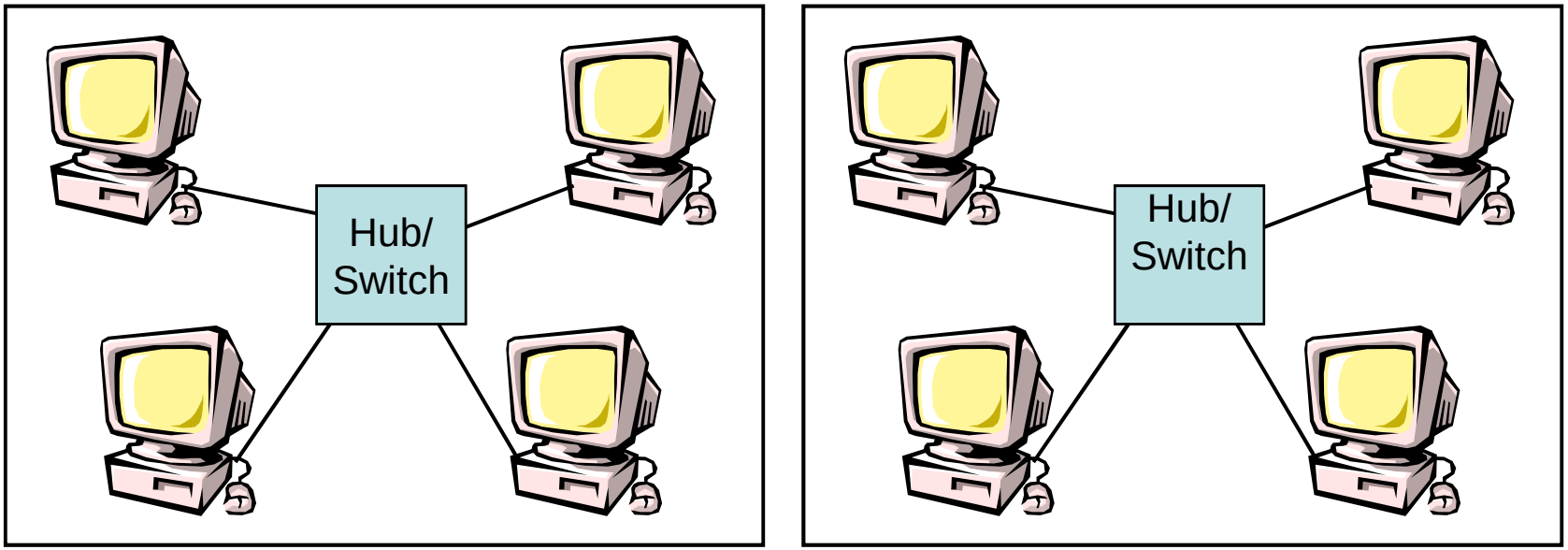
- Basiert auf Twisted-Pair-Verkabelung der Kategorie 5 (bis 100 MHz)
  - Max. 100 Meter Kabellänge (zwischen Netzwerkkarte und Hub)
  - Verwendung von 2 Adernpaaren: Je eines für Senden und Empfangen
- Datenverkehr vollduplex
- Signale nach FDDI-Standard (ANSI X3T9.5):
  Datencodierung mittels 4B5B-Verfahren

(c) 100Base-Fx

- Multimode- oder Monomode-Fasern (benötigt zwei Lichtwellenleitungen)
- Sternförmige Verkabelung
- Datenverkehr vollduplex
- Max. 400 Meter

# Leitbeispiel: Strukturierte Verkabelung

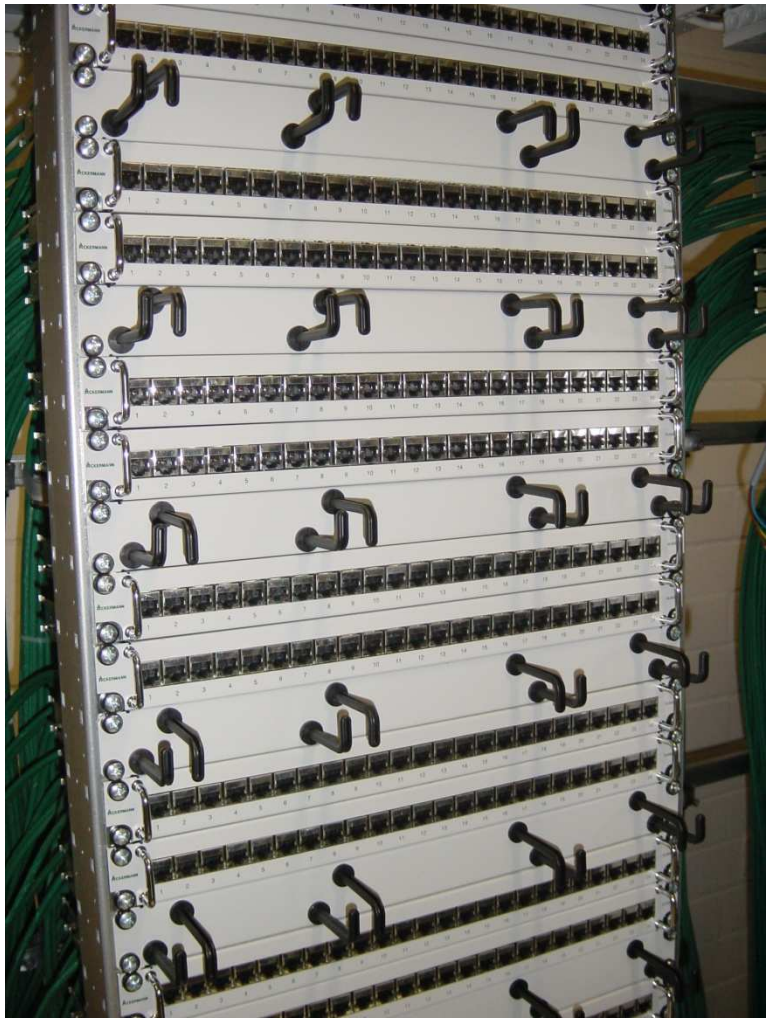
- **Strukturierte Verkabelung:** Aufteilung eines Netzes in mehrere Kabelstrecken, die über ein Backbone oder einen zentralen Hub/Switch zusammengefasst sind.
  - Vernetzung der einzelnen Räume: zentraler Hub/Switch der die einzelnen Rechner miteinander koppelt
  - Anschluss mittels Twisted-Pair (Kategorie 5) an den Hub/Switch verbunden



# Beispiele

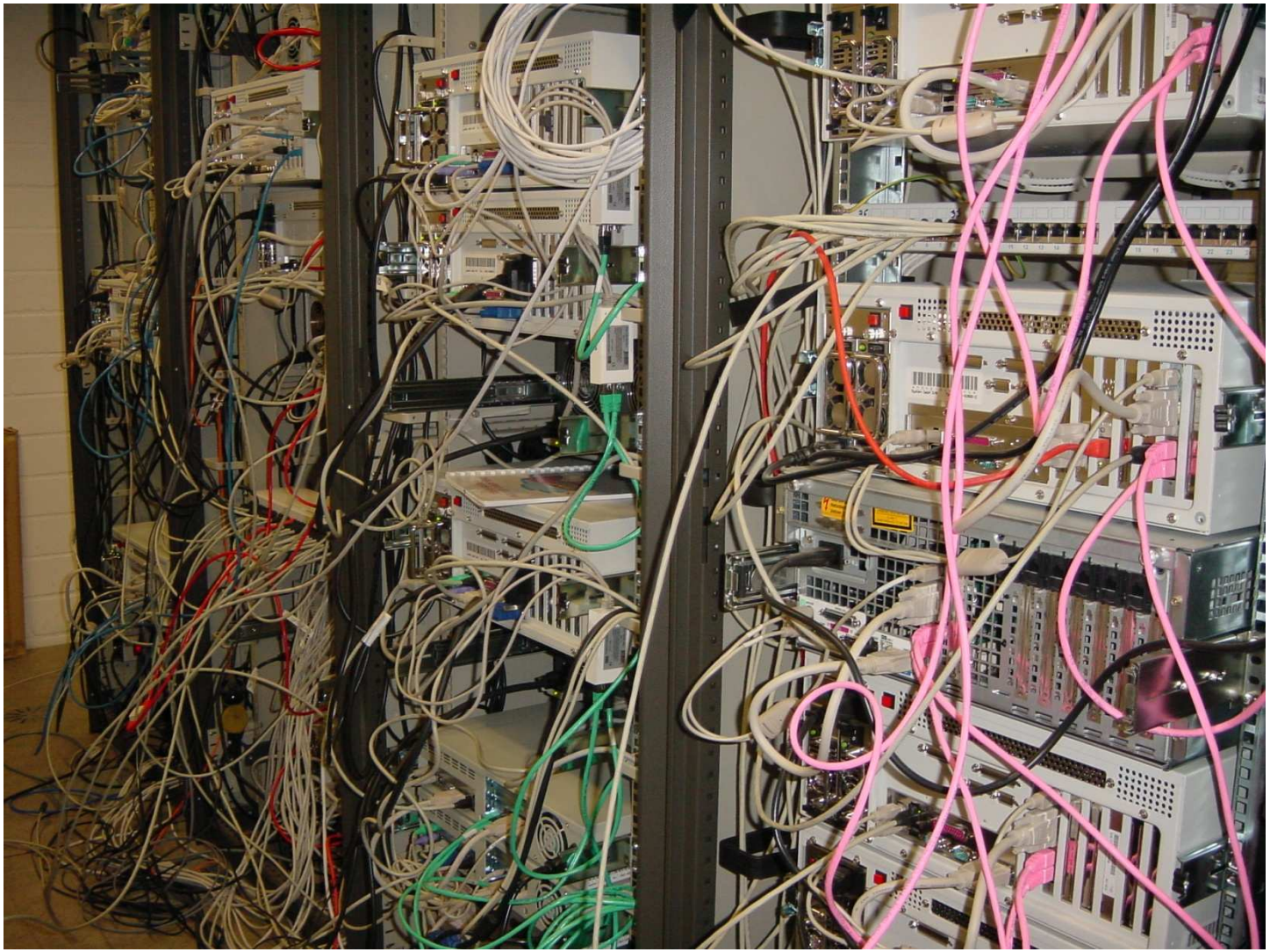
# Beispiele

# Beispiele



Patchfeld

# Beispiele



# Ethernet-Weiterentwicklung: Gigabit Ethernet

- „Gigabit Ethernet“ (auch **1000Base-X** genannt):
  Gigabit Ethernet Alliance (GEA) - Zusammenschluss mehrerer Hardware-Hersteller
  → Informationen u.a. bei http://www.gigabit-ethernet.org
- **Grundgedanke:**
  - logischer Schritt von 10 Mbit/s Ethernet über Fast-Ethernet (100 Mbit/s) zu Gigabit-Ethernet (1.000 Mbit/s)
  - ursprüngliches Ziel: Beibehaltung des CSMA/CD-Verfahrens
  - Ideale Ergänzung zu (Fast-) Ethernet zur Fortführung im Backbone-Bereich
  - Einsatz auf Glasfaser und Kupferadern
- **Probleme:**
  - Min. Rahmenlänge 64 byte von (Fast-) Ethernet zu klein, um Kollisionen zu erkennen
    - Kompatibilität zu (Fast-) Ethernet möglich?
  - Strenge Normen zur elektromagnetischen Verträglichkeit müssen erfüllt werden
  - Elektro-physikalische Eigenschaften der Kupferadern: Übersprechen, Dämpfung, ...

# Gigabit-Ethernet im Detail

- Duplex-Betrieb und Halbduplex-Betrieb möglich
  - Praktisch alle neueren Komponenten sind vollduplexfähig
- Problem: Beim Halbduplex-Betrieb Kollisionserkennung notwendig $\rightarrow$ min. Rahmenlänge von (Fast-) Ethernet zu klein!
- Lösung:
  - Bei Gigabit-Ethernet wurde daher die minimale Rahmengröße von 64 byte auf 512 byte erhöht
  - Auffüllen kleinerer Rahmen mit Füllzeichen (Extension Symbols: "Carrier Extension"), die an FCS angehängt werden, aber keine Bedeutung haben
    - Weiterhin minimale Rahmengröße von 64 byte
  - Packet Bursting: Zusammenfassen mehrerer kleiner Pakete möglich
- Burst Limit: Beschränkung der max. Sendezeit auf 65.536 bit
- Gleiches Rahmenformat wie bei (Fast-)Ethernet auf MAC-Ebene
  - Auf der physikalischen Ebene jedoch leichte Unterschiede, die aber für MAC transparent sind
- Umschalten zwischen 10, 100, 1.000 Mbit/s mittels Auto-Negotiation möglich

# Gigabit-Ethernet: 1000Base-T

- Wegen der weiten Verbreitung ungeschirmter verdrillter Doppeladern soll Gigabit-Ethernet auch für diese Kabeltypen (UTP-5) möglich sein. Aber:
  - Länge von bis zu 100 m soll beibehalten werden
  - Wegen 8B/10B-Codierung: 1.250 Mbaud zur Übertragung notwendig!
  - Probleme mit Übersprechen, Dämpfung, Abstrahlung, ...
- Lösung: Parallele Nutzung aller vier Doppeladern:
  - Sternförmige Verkabelung, keine Kollisionen
  - Steuerung des Senders durch Sternpunkt
  - Modulationsverfahren: PAM5 (Pulsamplitudenmodulation mit fünf Zuständen)
    - 125 Mbaud pro Adernpaar
    - 2bit/Symbol $\rightarrow$ 250 Mbps pro Adernpaar
    - Über die 4 Adernpaare kann pro Signalschritt 1 byte übertragen werden; insg. 1 Gbps
  - Codierungsverfahren: Trellis-Codierung und Scrambling (Verwürfeln)
  - Duplexbetrieb durch Echokompensation
  - Standardisierung: IEEE 802.3ab

# Gigabit-Ethernet: Glasfaser und geschirmte Doppelader

- Standards für Glasfaser:
  - 1000Base-SX (short wavelength, Standard, Multimode-Glasfaser),
  - 1000Base-LX (long wavelength, „luxury“, Multi- und Monomode-Glasfaser)
- 1000Base-CX (twinax, geschirmte Doppelader)
- Codierung: 8B/10B anhand einer Codetabelle
- Standardisierung durch IEEE 802.3z

| Standard | Kabeltyp | Durchmesser | Wellenlänge | Segmentlänge |
|---|---|---|---|---|
| 1000Base-SX | Multimode | 62,5 µm | 830 nm | 2-275 m |
| | Multimode | 50 µm | 830 nm | 2-550 m |
| 1000Base-LX | Multimode | 62,5 µm | 1270 nm | 2-550 m |
| | Multimode | 50 µm | 1270 nm | 2-550 m |
| | Monomode | 10 µm | 1270 nm | 2-5.000 m |
| 1000Base-CX | Twinax | | | 25 m |

Twinax-Kabel: 

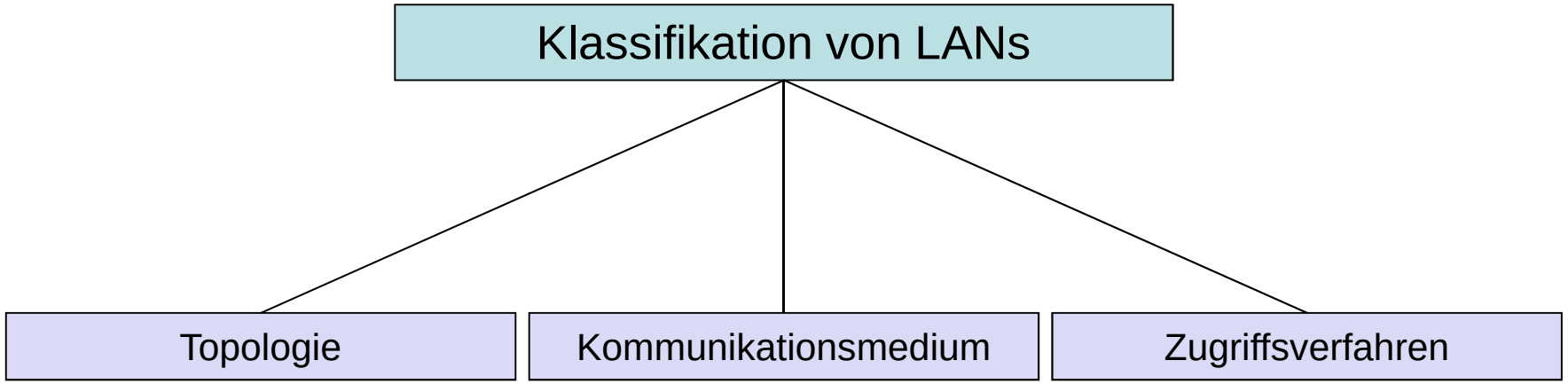
# Weiterentwicklungen: 10 Gigabit Ethernet

- Glasfasermedium: IEEE 802.3ae,
- Kupfermedium: IEEE 802.3ak und IEEE 802.3an.
- http://www.10gea.org/
- Datenrate 10.000 Mbit/s
- Auch ungeschirmte Doppeladern möglich (802.3ab)

**Neuste Entwicklung:**
**IEEE 802.3ba**
**40 Gbit/s-Ethernet,**
**100 Gbit/s-Ethernet**
**über Glasfaser**
**sowie <10m Kupferkabel**

| Bezeichnung | Kabel | Segmentlänge |
|---|---|---|
| 10GBase-T | 4 ungeschirmte Adernpaare (CAT6a) | 100m |
| 10GBase-CX4 | 2 geschirmte Adern-paare (Doppel-Twinax) | 15m |
| 10GBase-SR | Multimode-Glasfaser | 26 - 300 m |
| 10GBase-LR<br>10GBase-ER | Monomode-Glasfaser | 10.000 m<br>40.000 m |
| 10GBase-LX4<br>10GBase-LW4 | Multi-/<br>Monomode-Glasfaser | 240-300m<br>10.000 m |

# 3.6 Klassifikation Lokaler Netze




- Stern
- Bus
- Baum
- Ring
- ...

- Verdrilltes Adernpaar
- Koaxialkabel
- Glasfaser
- ‚Luft‘

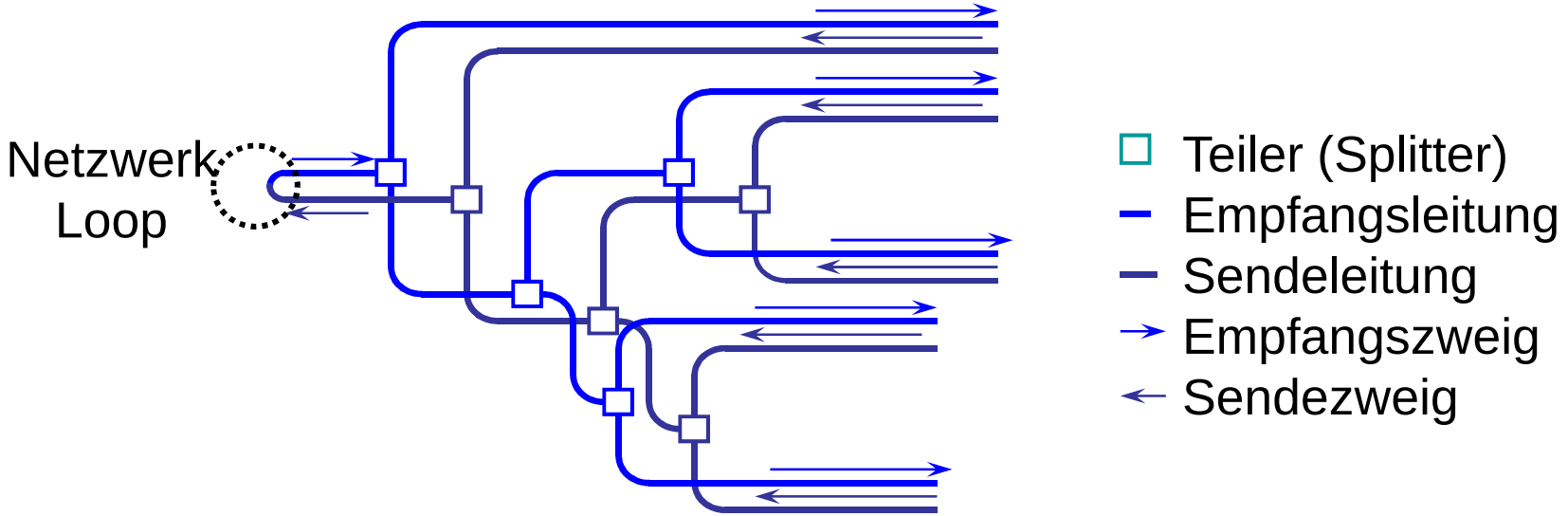
- kontrolliert
- konkurrierend

# LAN-Topologie Stern – Grundprinzipien

- Grundprinzipien:
  - Exklusive Punkt-zu-Punkt-Verbindung zwischen Station und Zentrale
  - Kommunikation zwischen Stationen ausschließlich über Zentrale
  - Stationen sind z.B. Telefon, PC, Notebook, Gateways als Internetzwerkeinheit
  - Vermittlungstechniken:
    - Raummultiplex (analog/digital): ZVK mit Durchschaltevermittlung
      Bsp: Telefon-Nebenstellenanlage, Private Branche Exchange PBX
    - Paketvermittlung: ZVK mit Speichervermittlung
      (meist Datagramm-Vermittlung)
  - Anwendungsbeispiele:
    - Digitale Nebenstellenanlagen
    - Strukturierte Verkabelung bei Ethernet

# LAN-Topologie Baum

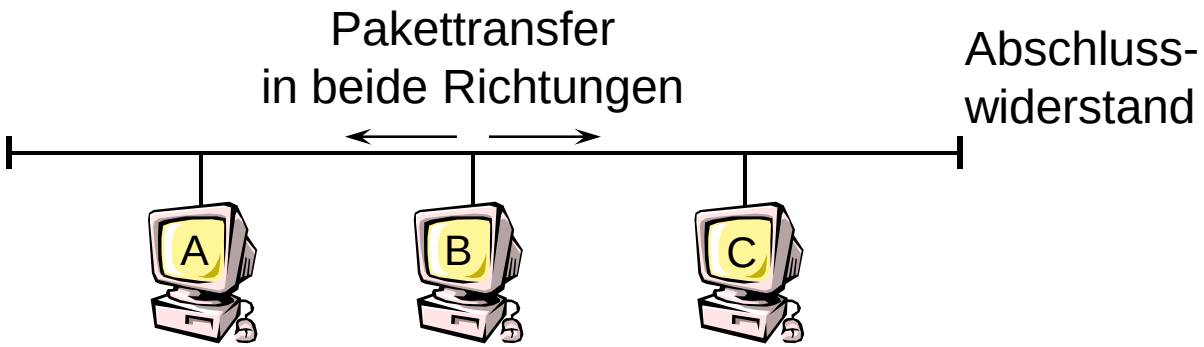
- Verbindungsstruktur:
  - Mehrpunkt (multi-point)
- Vielfachzugriff (multiple access):
  - Angeschlossene Teilnehmer haben Zugriff zum gleichen Übertragungskanal
- Verteilnetz (broadcast medium):
  - Alle Teilnehmer empfangen sämtliche Nachrichten.



# LAN-Topologie Bus

- Verbindungsstruktur: Mehrpunkt (multi-point)
- Vielfachzugriff (multiple access)
- Verteilnetz (broadcast medium)
  - Passive Ankopplung der Stationen. Keine Verstärkung/Signalformung/ Signalwandlung an den Ankopplungspunkten der Stationen
  - Empfangen von Daten durch Kopieren

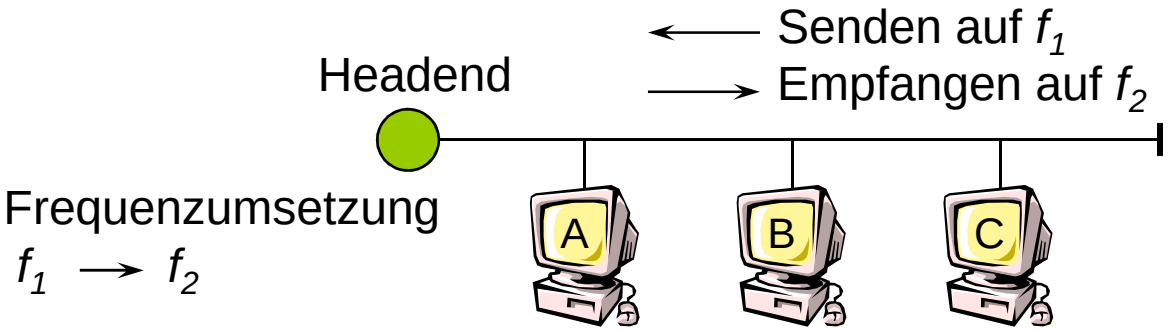
- Übertragungstechniken:
  1. Basisband-Bussystem




Stationsankopplung mit Lese- und Schreibeinrichtung

# LAN-Topologie Bus für Breitbandtechnologie

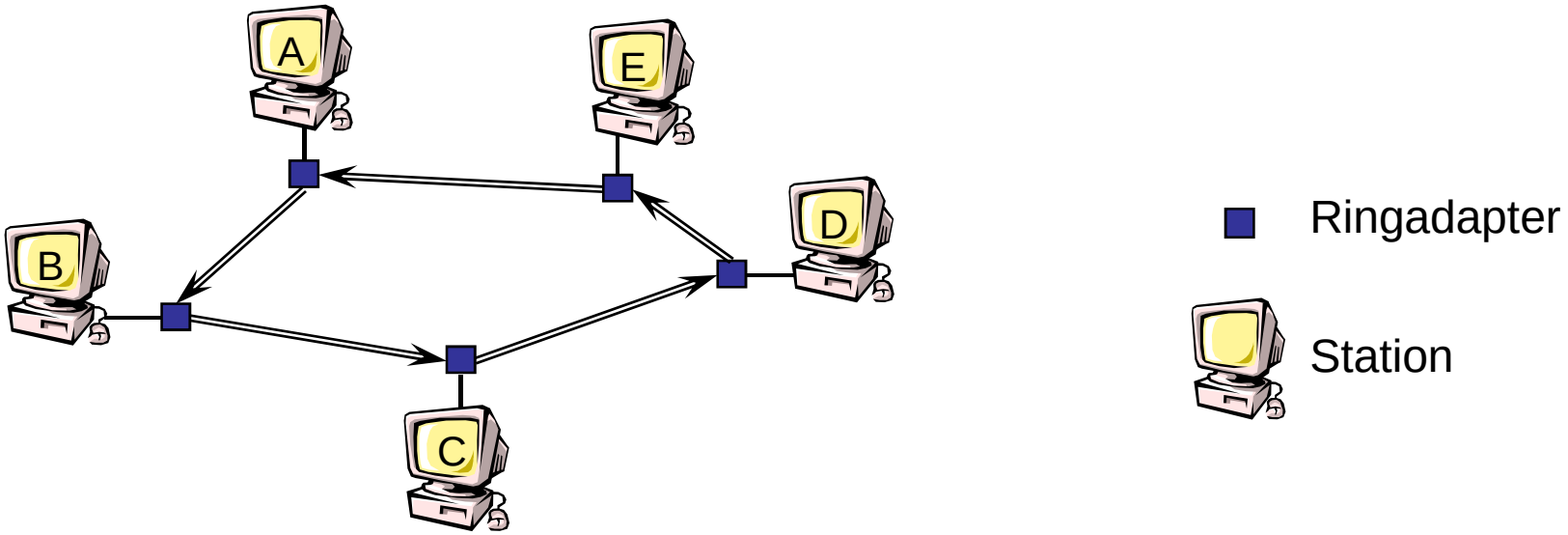
- Breitband-Bussystem (z.B. Kabelfernsehen)




| | Sendefrequenz | Empfangsfrequenz |
|---|---|---|
| Subsplit | 5-30 MHz | 54-400 MHz |
| Midsplit | 5-116 MHz | 168-400 MHz |
| Highsplit | 5-174 MHz | 232-400 MHz |

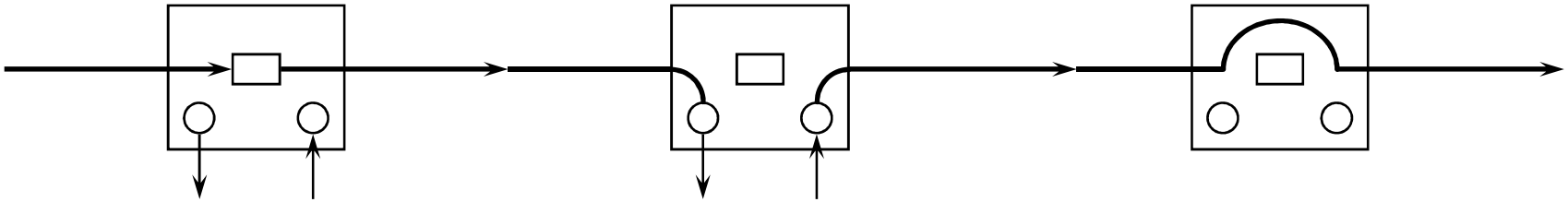
# LAN-Topologie Ring

- Verbindungsstruktur:
  - geschlossene Folge von unidirektionalen Punkt-zu-Punkt-Verbindungen.
- Zugriff:
  - zum Ring über Ringschnittstelle/Ringadapter
- Aktiver Ring:
  - Ringadapter sind aktive Signalregeneratoren mit einem Zwischenpuffer und damit einer Verzögerung um mindestens ein Bit (1-Bit-Delay).



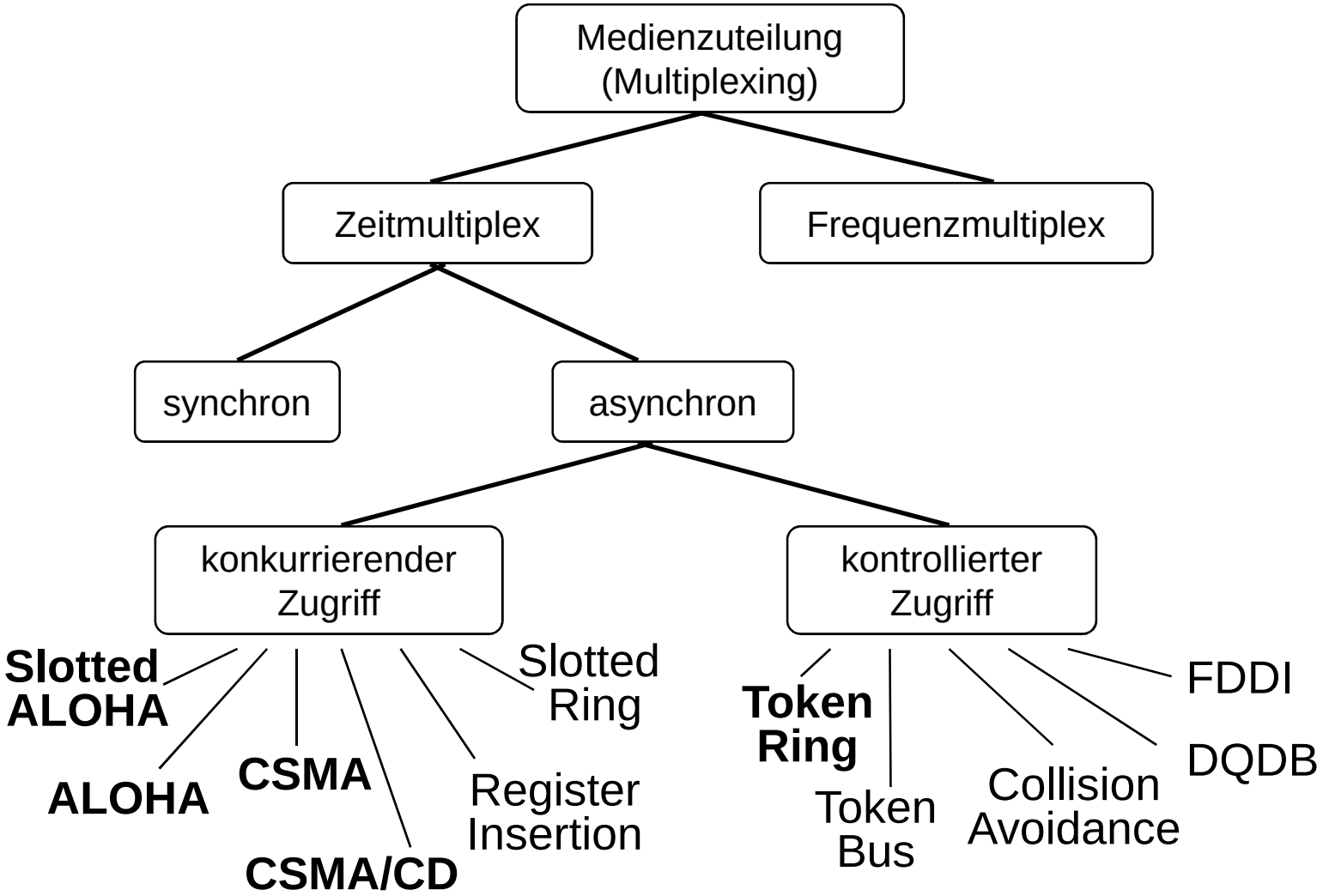
# LAN-Topologie Ring – Ringadapter

## Zustände des Ringadapters:

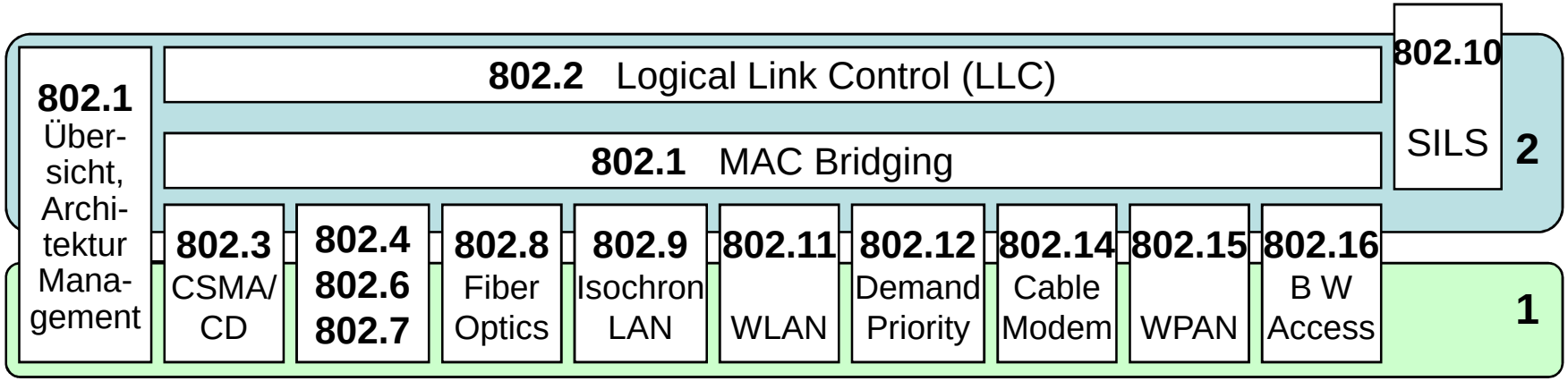


- Abhör-Zustand
  - Abhören des vorbeilaufenden Bitstroms
  - Kopieren des Bitstroms „Kopieren im Fluge“
  - Modifikation des Bitstroms möglich
- Sende-Zustand
  - Aussenden der Sendebits
  - Einbehalten und Überprüfen des ankommenden Sendeblocks
- Überbrückungszustand
  - passiver Anschluss

# LAN/MAN: Zugriffsverfahren in der Übersicht



# LAN/MAN: Standardisierung nach IEEE 802

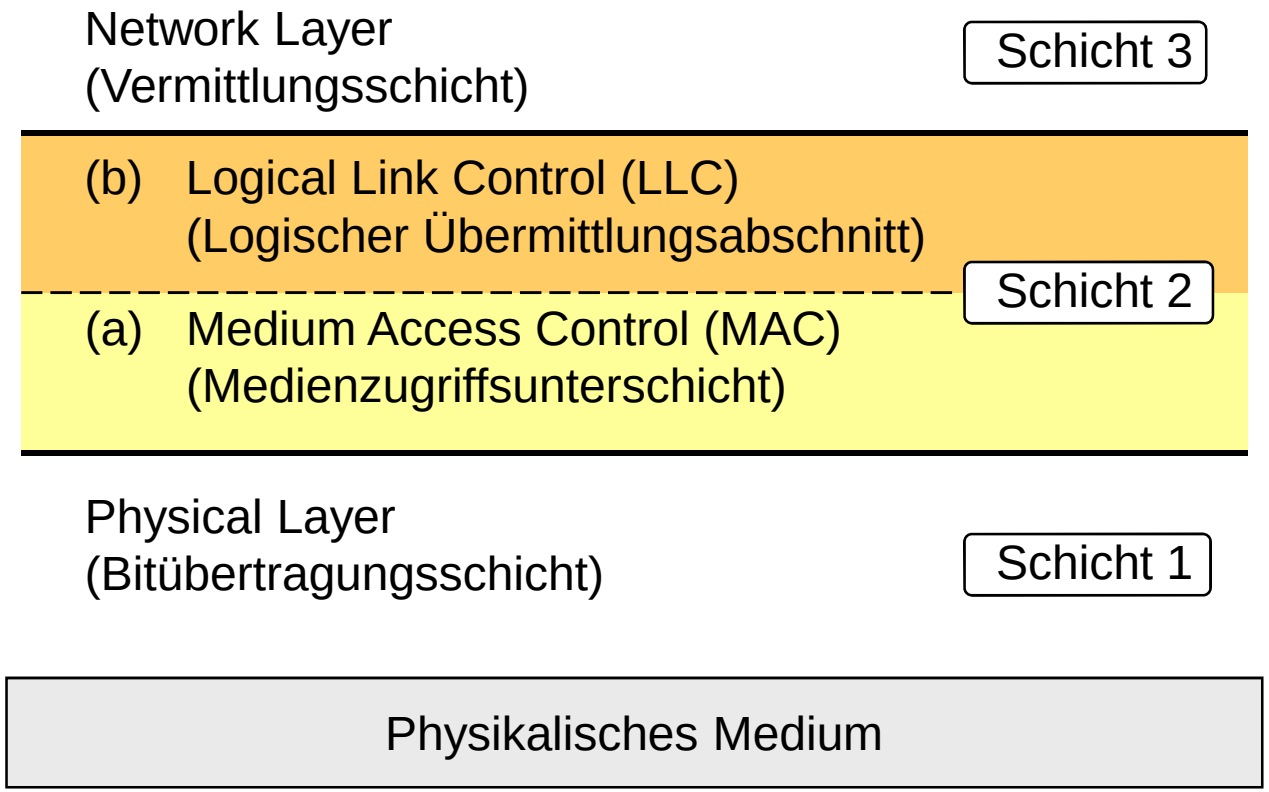



*Themen:*

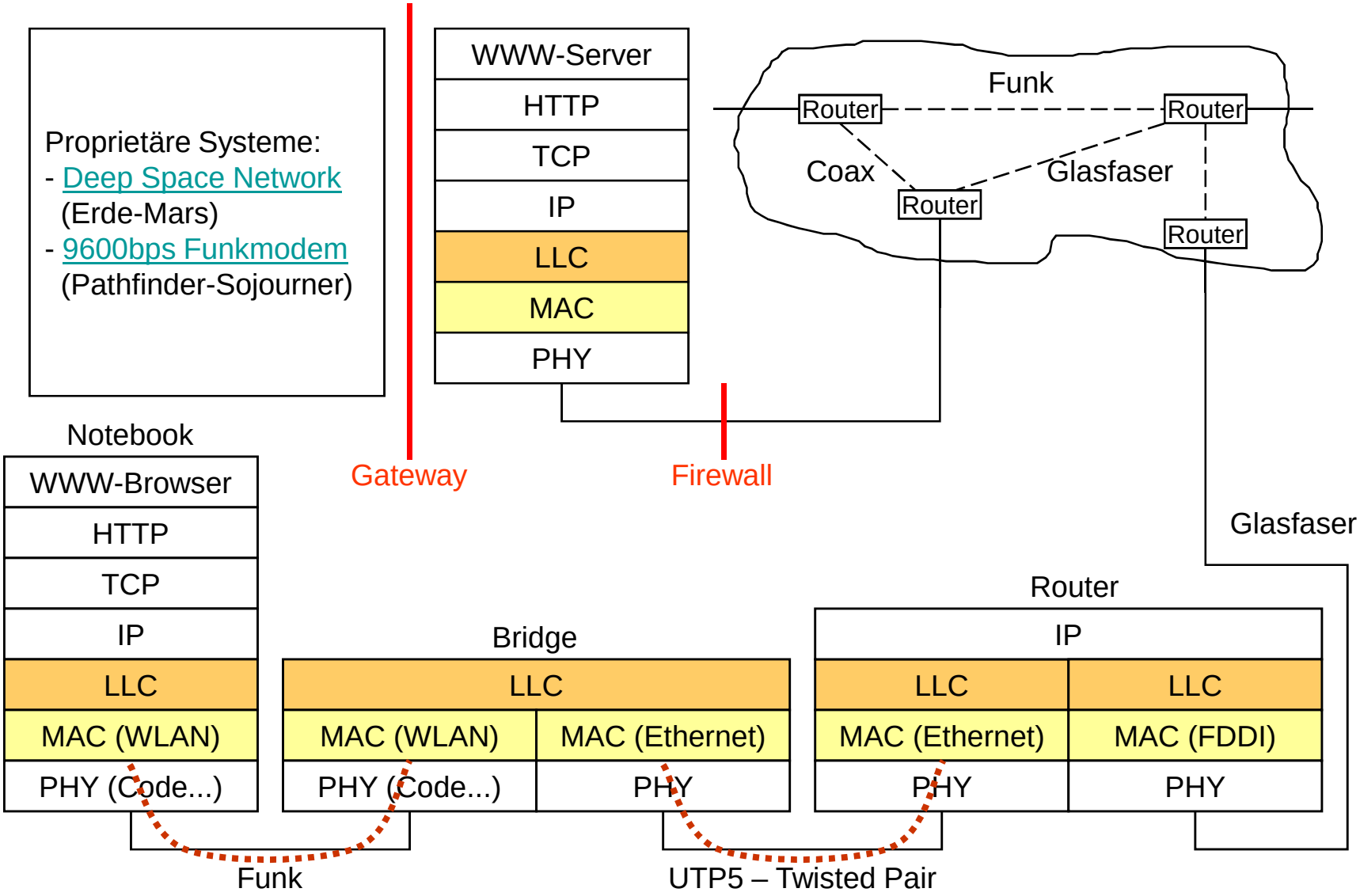
| | |
|---|---|
| **802.1:** | Zusammenhang der Standards und MAC Bridging |
| **802.2:** | Logical-Link-Control-Dienste/Protokolle (LLC) *inaktiv* |
| **802.3:** | CSMA/CD-Protokoll auf Bustopologie |
| **802.4:** | Token-Bus-Protokoll auf Bustopologie *inaktiv* |
| **802.5:** | Token-Ring-Protokoll auf Ringtopologie |
| **802.6:** | Metropolitan Area Network *inaktiv → 802.14* |
| **802.7:** | Broadband TAG (Technical Advisory Group) *inaktiv* |
| **802.8:** | Fiber Optic TAG |
| **802.9:** | Isochronous LAN |
| **802.10:** | Sicherheitsstruktur für 802-Protokolle *inaktiv* |
| **802.11:** | Wireless LANs |
| **802.12:** | Demand Priority Working Group *inaktiv* |
| **802.14:** | Cable Modem: Datenübertragung über TV-Kabelmodems |
| **802.15:** | Wireless Personal Area Networks: Netzwerke über kurze Distanzen |
| **802.16:** | Broadband Wireless Access: Drahtloser Zugriff auf Breitband-Systeme |
| … | |
| **802.21:** | Enable handover and interoperability between heterogeneous networks |
| **802.22:** | Wireless Regional Area Networks ("WRANs") http://ieee802.org/22/ |

# LAN: Standardisierung nach ISO/OSI

- Erweiterung des OSI-Schichtenmodells:
- Unterteilung der Schicht 2 in zwei Unterschichten.

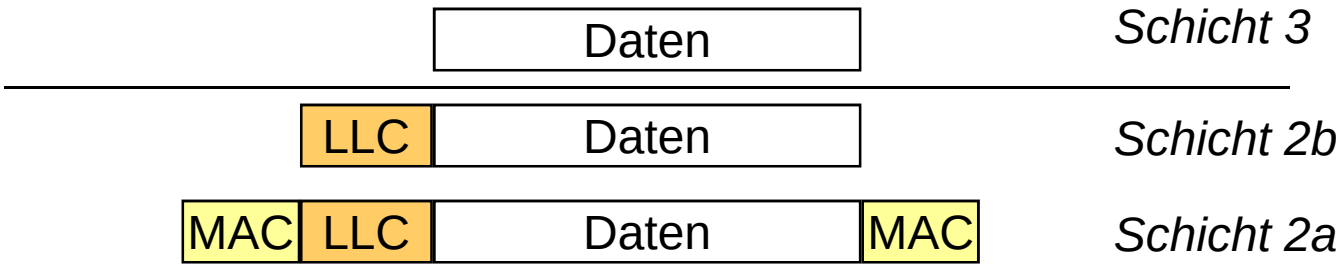


# MAC und LLC im Beispiel



# 3.7 LAN: Logical Link Control (LLC) IEEE802.2

- **Hauptaufgabe:**
  - Verdecken unterschiedlicher MAC-Verfahren
  - Drei Diensttypen:
    - **LLC-Typ 1: unzuverlässiger Datagrammdienst (typisch für LANs)**
    - LLC-Typ 2: verbindungsorientierter Dienst
    - LLC-Typ 3: bestätigter Datagrammdienst
- **Format:**
  - vereinfachte Version von HDLC (nur Asynchronous Balanced Mode Extended)
  - Adressen implizieren das verwendete Protokoll der Schicht 3



# HDLC-Protokoll

- High-Level Data Link Control (HDLC)
  - Bit-orientiertes, code-transparentes Sicherungsschichtprotokoll
  - Codetransparenz durch Bit-Stuffing
  - Halb- und vollduplexfähig
  - Punkt-zu-Punkt- und Mehrpunkt-Konfiguration
  - Symmetrische und unsymmetrische Konfiguration
  - Piggybacking
  - Flusskontrolle durch „Sliding Window“-Technik
  - Varianten des HDLC-Protokolls:
    - SDLC von IBM (Synchronous Data Link Control)
    - LAPB (Link Access Procedure, Balanced)
    - LAPD (ISDN, Link Access Procedure, D-Kanal)
    - LLC (IEEE 802.2, Logical Link Control)
    - PPP (Point-to-Point Protocol)

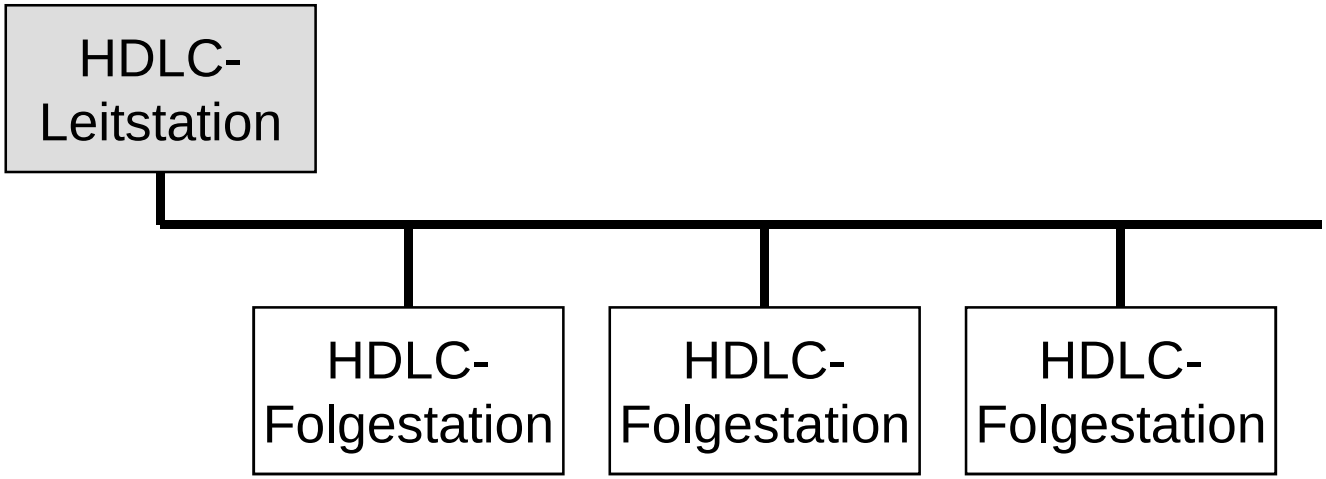
# HDLC: Konfigurationen

- Zu unterscheiden:
  - *Leitsteuerung*, die Befehle aussendet;
  - *Folgesteuerung*, die Meldungen als Reaktion auf Befehle aussendet.
- Drei Fälle des Datenflusses:
  - **unsymmetrische (zentrale) Steuerung:** *Empfangsaufruf*
    Die Leitsteuerung mit Datenquelle (Leitstation) fordert die Folgesteuerung (Folgestation) durch Befehl zum Datenempfang auf.
  - **unsymmetrische (zentrale) Steuerung:** *Sendeaufruf*
    Die Leitsteuerung mit Datensenke fordert die Folgesteuerung mit Datenquelle durch Befehl zum Senden von Daten auf (z.B. für Mehrpunkt-Verbindung).
  - **symmetrische (gleichberechtigte) Steuerung**
    Zwei *Hybridstationen*, die als Überlagerungen von je einer Leit- und Folgesteuerung aufzufassen sind, können Meldungen und Befehle ausgeben.

# HDLC: Verbindungstopologien

- Punkt-zu-Punkt-Verbindung:


- Asymmetrische Mehrpunktverbindung:



# HDLC: Betriebsarten

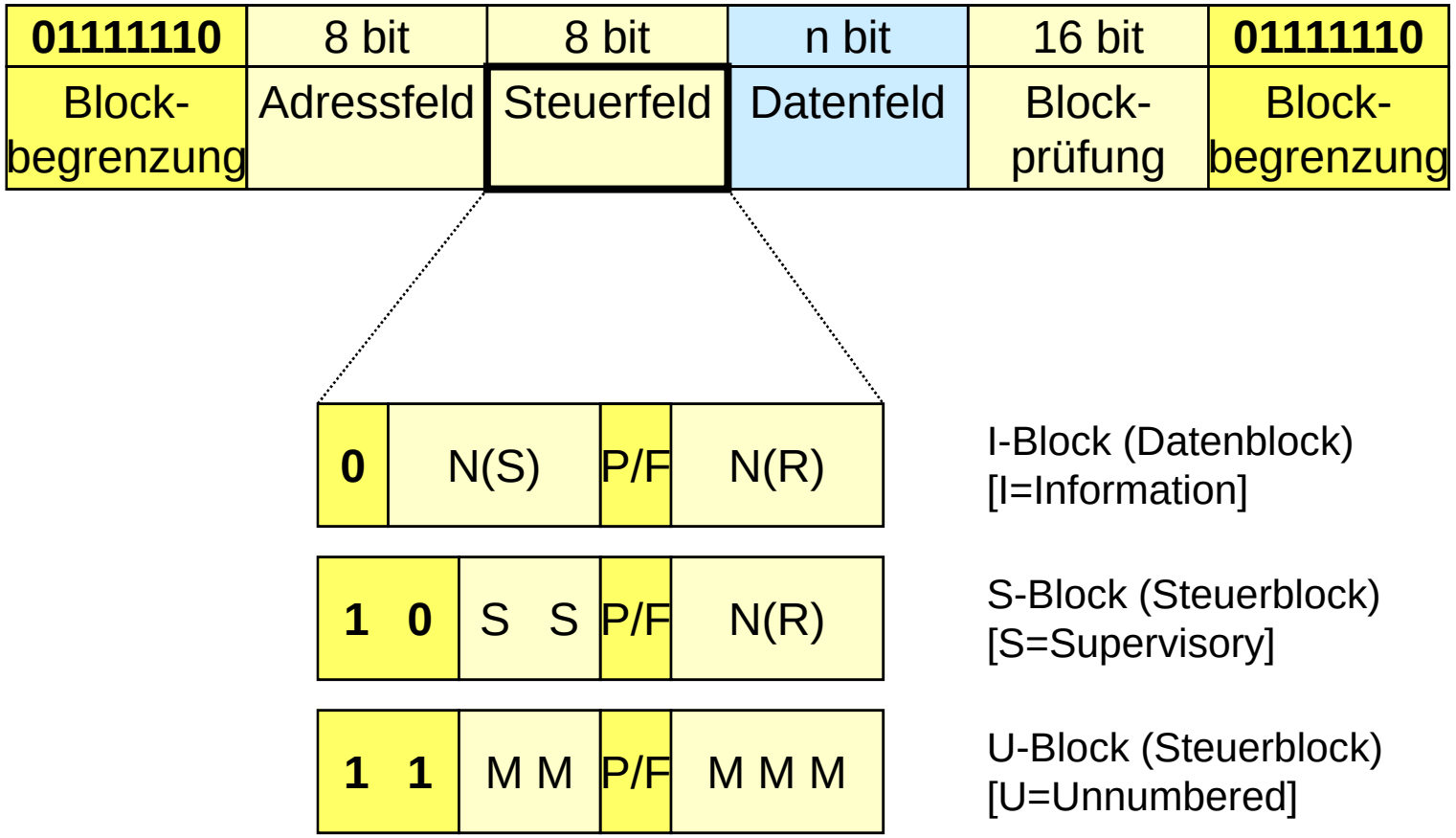
- **Aufforderungsbetrieb** (NRM — *Normal Response Mode*).
  - Folgestation darf nur nach ausdrücklicher Erlaubnis durch Leitstation Meldungen senden.
- **Spontanbetrieb** (ARM — *Asynchronous Response Mode*)
  - Folgestation kann jederzeit Meldungen an Leitstation senden.
- **Gleichberechtiger Spontanbetrieb** (ABM — *Asynchronous Balanced Mode*)
  - Beide Hybridstationen dürfen jederzeit Meldungen und Befehle übermitteln.

# HDLC: Datenübermittlungsblock

| **01111110** | 8 bit | 8 bit | n bit | 16 bit | **01111110** |
|---|---|---|---|---|---|
| Block-begrenzung | Adressfeld | Steuerfeld | Datenfeld | Block-prüfung | Block-begrenzung |

- Datenübermittlungsblock liefert Rahmen (HDLC-Frame) für Übermittlung von Befehlen, Meldungen und Daten.
- Blockbegrenzung (Flag): Feste Codierung zur Synchronisation
- Adressfeld (Address Field):
  - Befehl: Zieladresse;
  - Meldung: Herkunftsadresse
- Steuerfeld (Control Field): Festlegung von Befehlen und Meldungen
- Datenfeld (Information Field): Beliebige Bit-Folge ($n \geq 0$; n nicht notwendigerweise Vielfaches von 8), benötigt Bit Stuffing
- Blockprüfungsfeld (Frame Check Sequence, FCS): CRC-Verfahren

# HDLC: Datenübermittlungsblock und Steuerfeld




I-Block (Datenblock)
[I=Information]

S-Block (Steuerblock)
[S=Supervisory]

U-Block (Steuerblock)
[U=Unnumbered]

# HDLC: Aufbau des Steuerfeldes

| Steuerfeldformat für | Bit-Nummer 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| I-Block (Datenblock) [I=Information] | **0** | N(S) | | | P/F | N(R) | | |
| S-Block (Steuerblock) [S=Supervisory] | **1** | **0** | S | S | P/F | N(R) | | |
| U-Block (Steuerblock) [U=Unnumbered] | **1** | **1** | M | M | P/F | M | M | M |

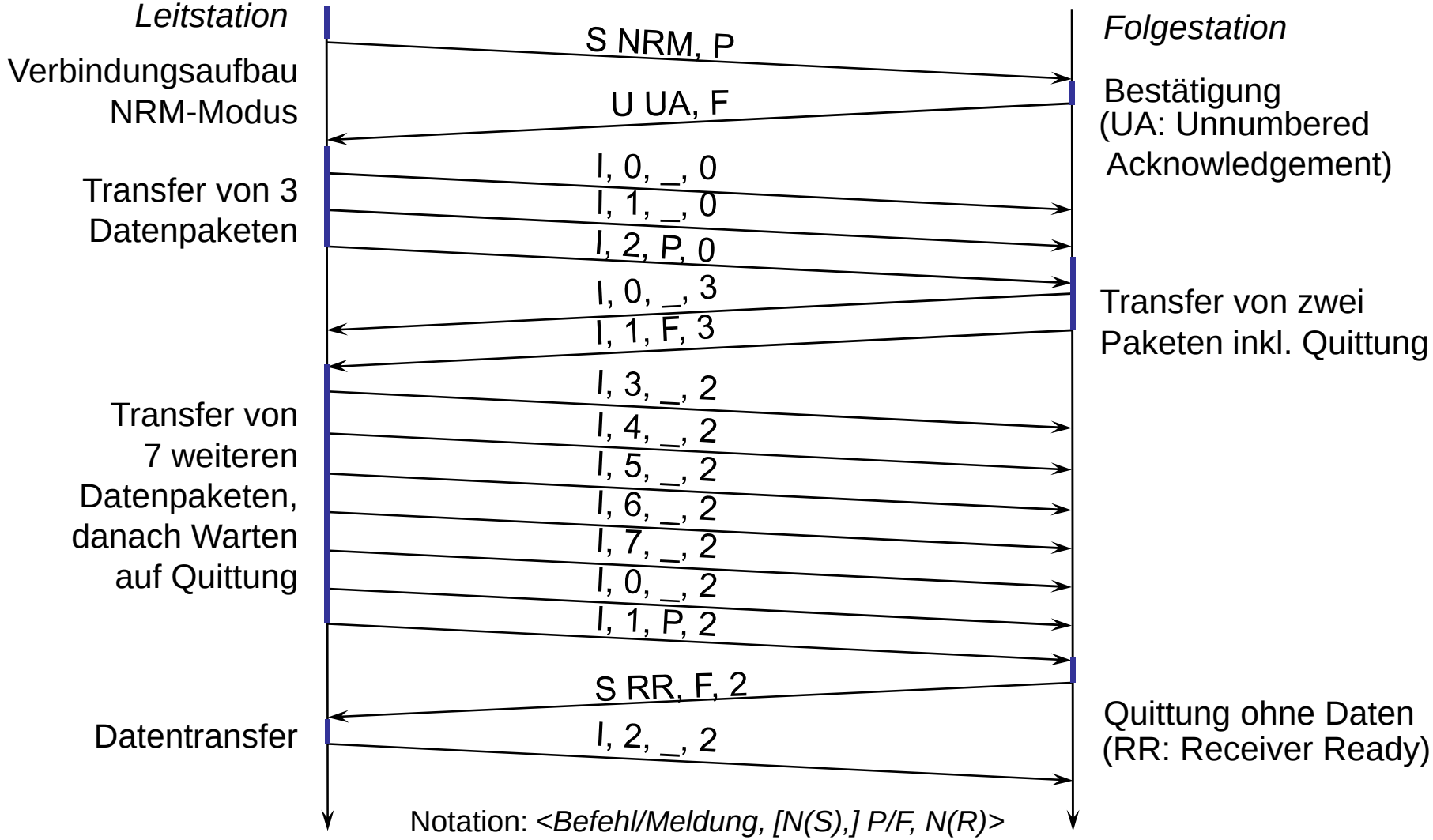
- Sendesequenznummer N(S); Empfangssequenznummer N(R) je 3 bit lang
- I-Block: Übertragung von Nutzdaten
- S-Block: Steuerblock, Übertragungssteuerung (Befehle, Meldungen) mittels S-Bits wie z.B. Sendeaufruf, Bestätigung empfangener DÜ-Blöcke
- U-Block: Steuerblock ohne Folgenummer
  Zusätzliche Übertragungssteuerungsfunktionen, jedoch ohne Empfangsfolgenummer; Codierung durch M-Bits (max. 32 Befehle; z.Zt. 13 Befehle und 8 Meldungen festgelegt.)

# HDLC: Markierungsbit — Poll/Final (P/F)

- P/F-Bit hat unterschiedliche Bedeutung in Befehlen und Meldungen sowie in den einzelnen Betriebsarten:
  - *P/F=1 in Befehlen:*
    Anforderung einer Meldung, bzw. einer Folge von Meldungen (Poll)
  - *P/F=1 in Meldungen:*
    Bestätigung des Empfangs eines Befehls mit PF=1,
    d.h. Meldungen als Antwort auf Befehle mit PF=1 (Final).
  - *Gebrauch im Normal Response Mode:*
    Folgestation darf nach Senden einer Meldung mit P/F=1 als Antwort auf Befehl mit P/F=1 keine weiteren DÜ-Blöcke ohne Erlaubnis durch die Leitstation senden.
  - *Gebrauch im Asynchronous Response / Balanced Mode:*
    Auf einen Befehl mit P/F=1 muss vorrangig durch eine oder mehrere Meldungen mit P/F=1 geantwortet werden, jedoch sind weiterhin Meldungen mit P/F=0 möglich.

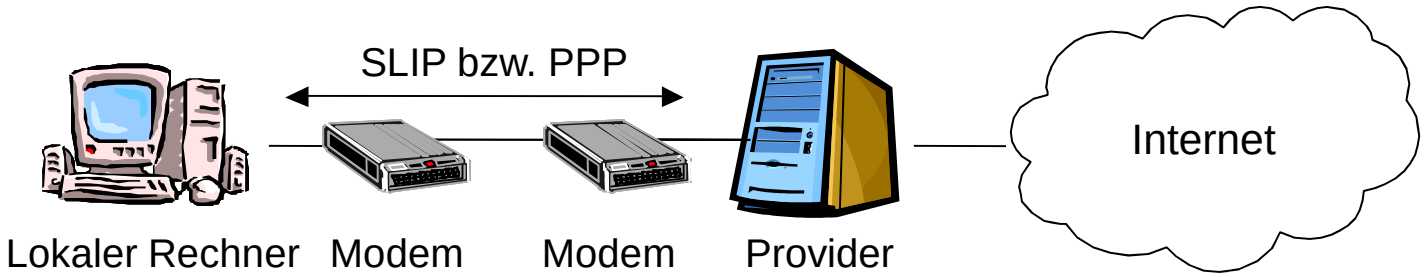
# HDLC: Beispielablauf

- Aufforderungsbetrieb (halbduplex), Modulo 8, Fenstergröße 7



# PPP (Point-to-Point Protocol)

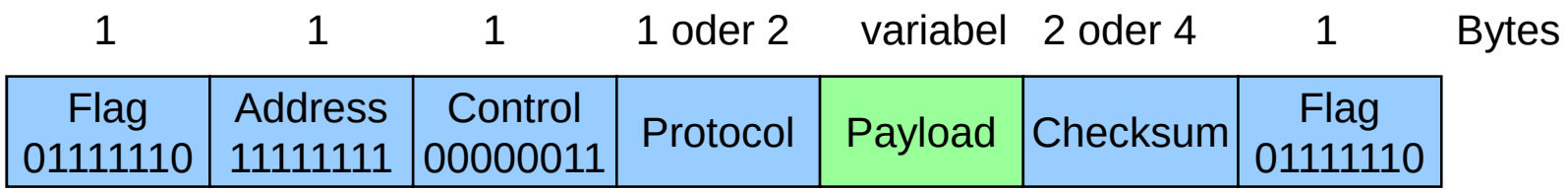
- Der größte Teil des Internets beruht auf Punkt-zu-Punkt-Verbindungen:
  - Verbindungen im WAN zwischen Routern / Heimanbindung über Modem und Telefonleitung
- SLIP (serial line IP, RFC 1055): keine Fehlererkennung, nur IP, keine dynamische Adresszuweisung, keine Authentifizierung




- PPP (RFC 1661/1662):
  - Schicht-2-Rahmenformat mit Fehlererkennung, Rahmenbegrenzung
  - Steuerprotokoll (LCP, Link Control Protocol) zum Verbindungsaufbau,
  - Verbindungstest, Verbindungsverhandlung, Verbindungsabbau
  - Verhandlung von Schicht-3-Optionen unabhängig vom Schicht-3-Protokoll
  - (separates NCP, Network Control Protocol, für alle unterstützten Schicht-3-Protokolle)

# PPP-Paketformat

- Paketformat an HDLC angelehnt




- zeichenorientiert (anstatt bitorientiert), d.h. die Länge das Nutzdatenfeldes endet immer an einer Byte-Grenze
- Codetransparenz durch Character Stuffing
- typischerweise werden nur *unnumbered*-frames übertragen, bei hohen Fehlerraten (Mobilkommunikation) kann jedoch auch der zuverlässigere Modus mit Sequenznummern und Bestätigungen gewählt werden
- als Protokolle im Nutzlast-Feld sind u.a. IP, AppleTalk, IPX definiert
- falls nicht anderweitig verhandelt, ist die maximale Länge der Nutzlast auf 1500 Byte begrenzt
- durch zusätzliche Verhandlung kann der Paketkopf verkleinert werden

# PPP-Verbindung

- Typisches Szenario beim Zugriff eines PCs auf das Internet via Modem
  - Anruf beim Service-Provider via Modem und Aufbau einer physikalischen Verbindung
  - Anrufer sendet mehrere LCP-Pakete im PPP-Rahmen zur Auswahl der gewünschten PPP-Parameter
  - Austausch von NCP-Paketen, um Vermittlungsschicht zu konfigurieren
    - z.B. kann hier dynamisch mittels DHCP (s.u.) eine IP-Adresse zugewiesen werden falls IP als Protokoll gewählt wurde
  - Der Anrufer kann nun genauso wie ein fest verbundener Rechner Internet-Dienste nutzen
  - Zur Beendigung der Verbindung wird via NCP die IP-Adresse wieder freigegeben und die Vermittlungsschichtverbindung abgebaut
  - Über LCP wird die Schicht 2-Verbindung beendet, schließlich trennt das Modem die physikalische Verbindung