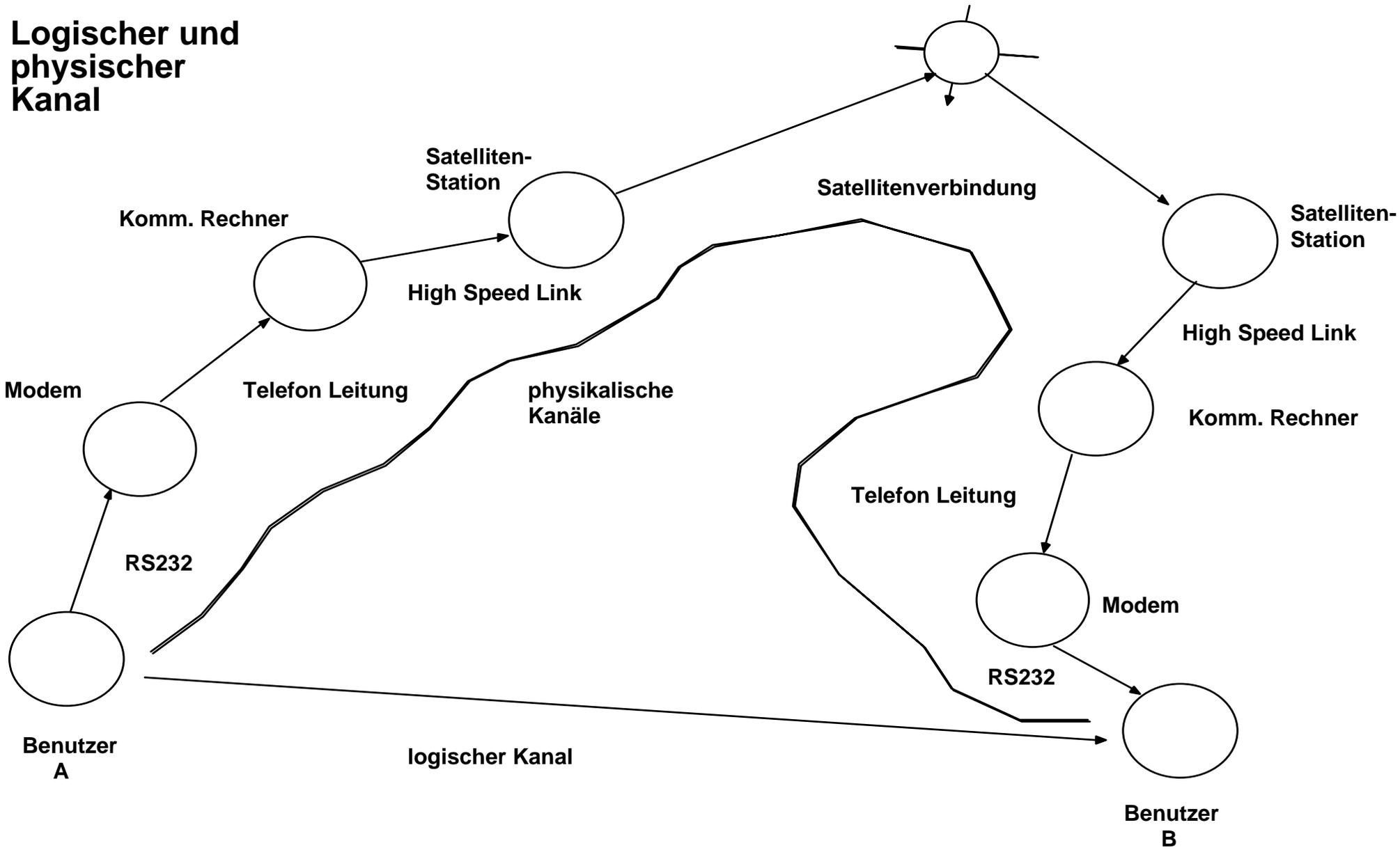


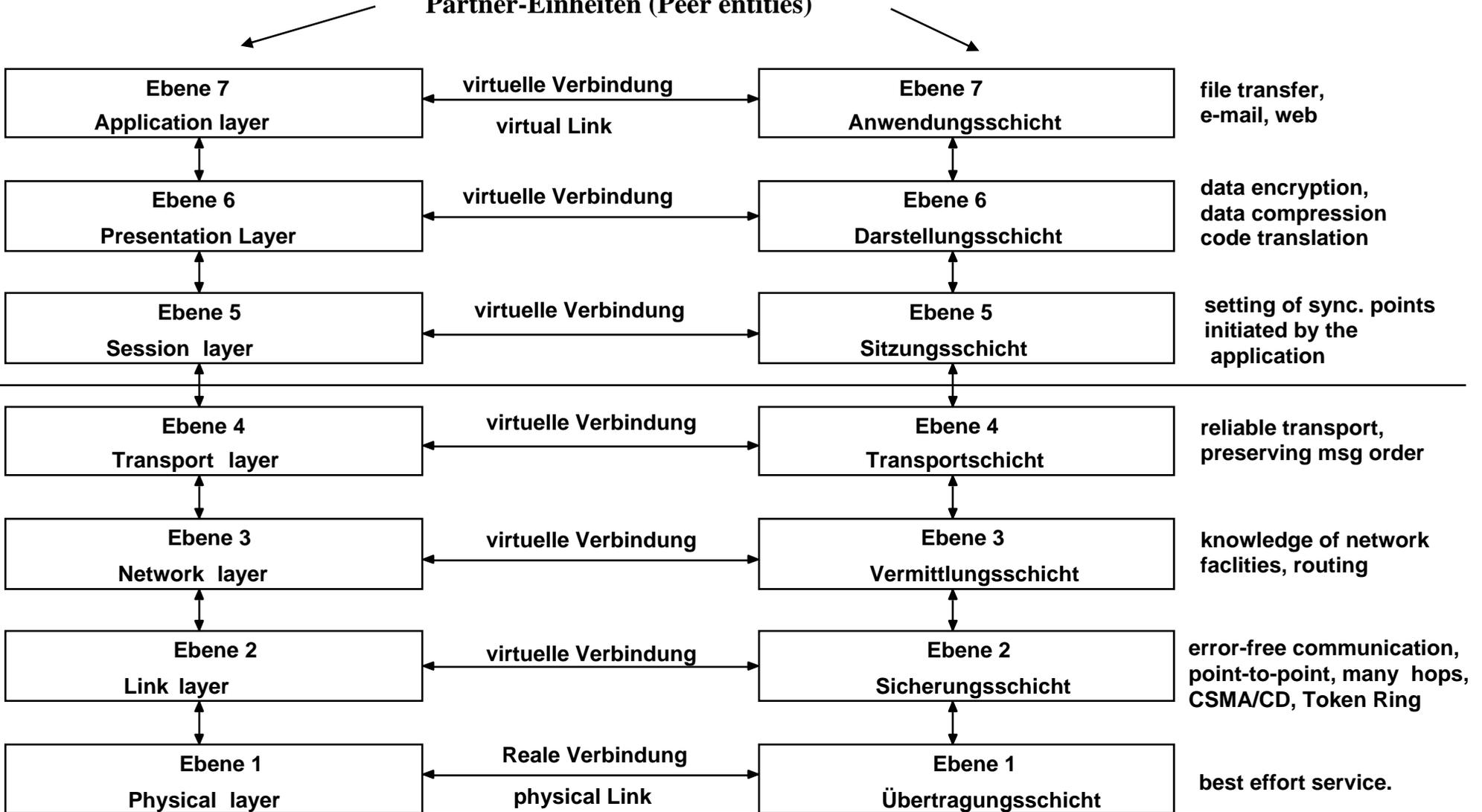
# **Die physische Übertragungsschicht**

# Logischer und physischer Kanal



**ISO / OSI**  
**ISO: International Standard Organization**  
**OSI: Open System Interconnection**

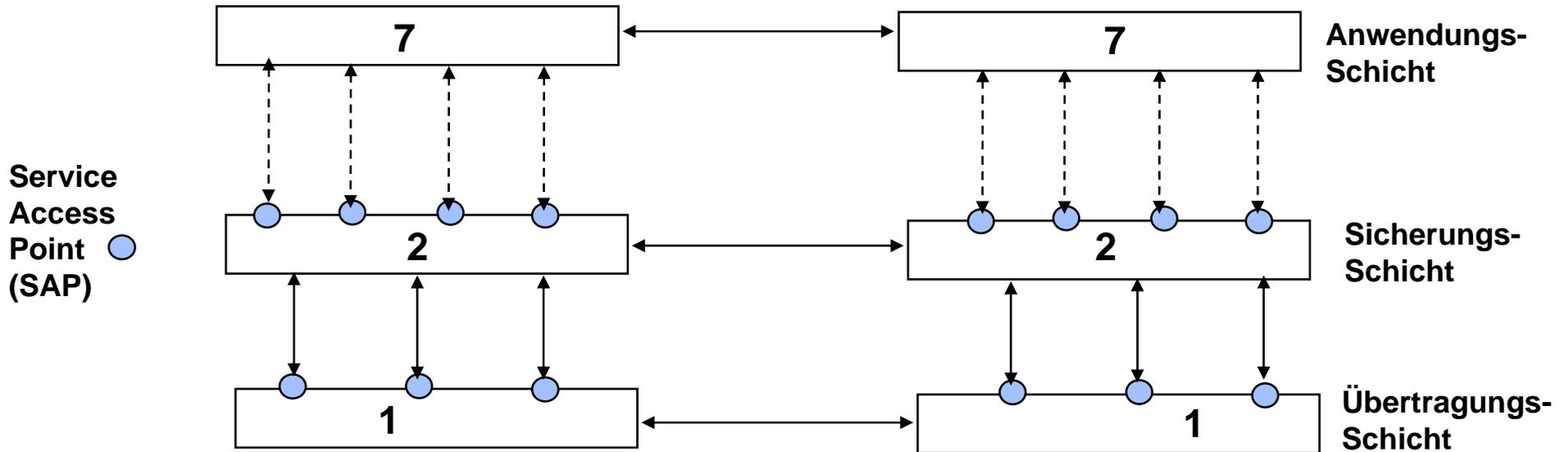
Partner-Einheiten (Peer entities)



Anwendungsorientiert

Netzwerkorientiert

## Schichtenmodell für RT-Kommunikationssysteme im Feldbereich



**Voraussetzung: Homogenes, geschlossenes System**

- ➡ Nicht alle Ebenen notwendig (Beispiele: Wegsuche (Routing) oder Verbindungsaufbau über mehrere Teilnetze hinweg)
- ➡ Leere Schichten im OSI- Modell
- ➡ Höhere Schichten greifen direkt auf SAPs der niederen Schichten zurück

- ➡ Erhöhung der Effizienz
- ➡ Direkte Abbildung von Schicht 7 auf Schicht 2
- ➡ Funktionalität anderer Schichten muß neu festgelegt und verteilt werden (z.B. Codierung Schicht 6 muß in einer anderen Schicht realisiert werden).

# Die Übertragungsschicht (Physical Layer)

- **Asynchrone serielle Übertragung** (Synchr. / Zeichen)
- **Synchrone serielle Übertragung** (Bitsynchronisation)

- **Bitcodierung:**
  - NRZ
  - Manchester Code
  - MFM

- **Modulation und Datenübertragung:**
  - **Basisband** Beisp.: Morsetel. / Ethernet
  - **Bandpaß (Breitband)** Beisp.: Radio, TV, Kabel-TV, Modem
- Modulationsarten: AM, FM

- **Übertragungsmedien:**
  - **Glasfaser** (Multi-Mode, Single-Mode)
  - **Kupferkabel** (Twisted Pair, Koaxial)
  - **Radio** (Frequenztafel)
  - **Satellitenverbindung** (Geostationär, umlaufend)

# Eigenschaften von Kommunikationsnetzen

## Begrenzende Faktoren:

- **Transferrate, (Übertragungskapazität, Bandbreite)**
- **Verzögerung (Latency)**

## Transferrate:

**Morsetelegraph: < 100Bit/sek**

**Telegraphie: < 150 Bit/sek**

**Telefon: ca. 50Kbit /sek**

**Serielle RS232: ~ 100Kbit/sek**

**Feldbusse: wenige Kbit/sek ... ~ 1Mbit/sek**

**Ethernet: 10-1000Mbit/sek**

**Hochleistungsglasfaser: mehrere Gbit/sek**

**Verzögerung: Satellitenverbindung (2 x 35700 km): ~ 240 ms,  
Transatlantikkabel (ca. 6.000 km): ~ 20 ms**

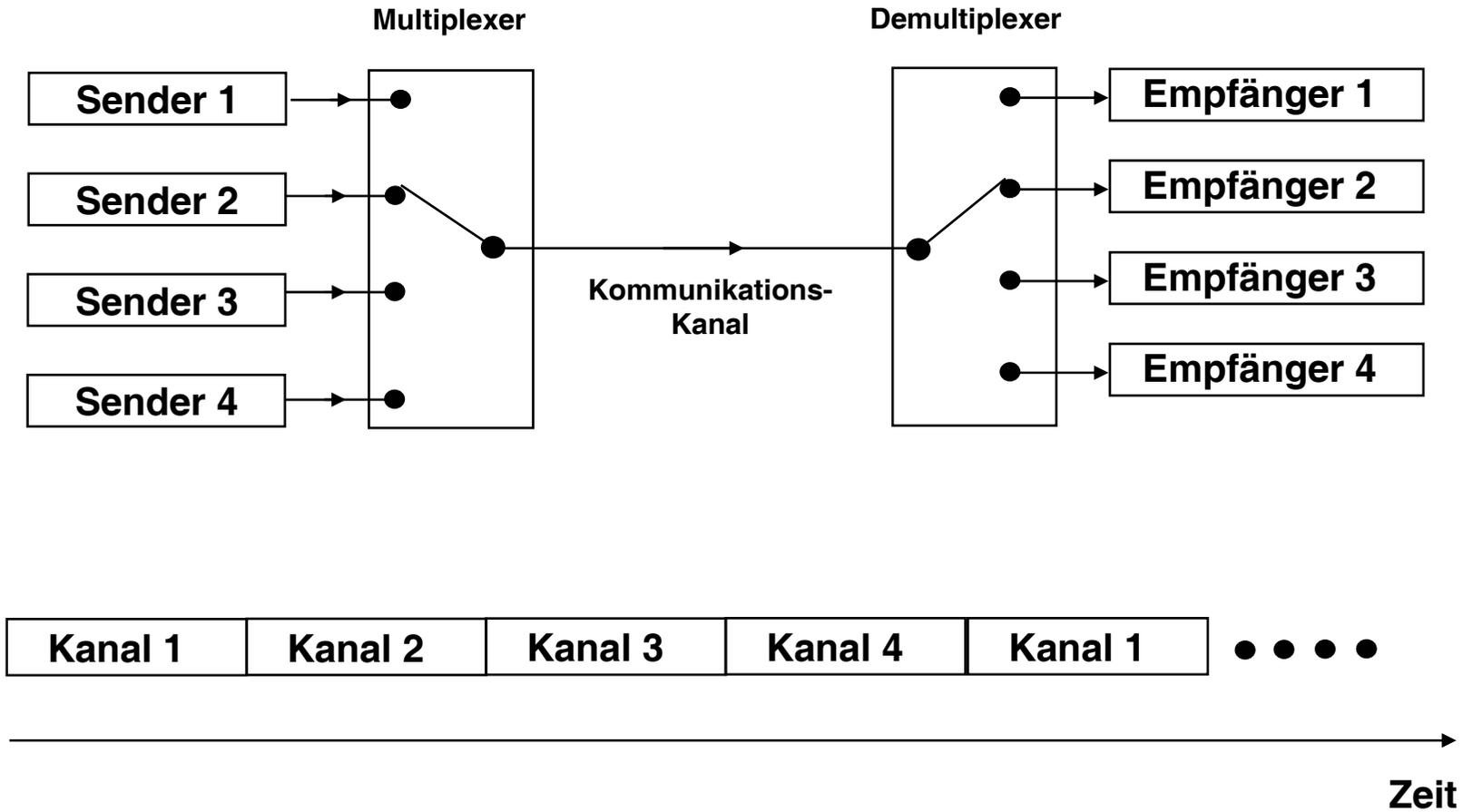
**Topologie: Anforderungen werden aus der Anwendung hergeleitet.  
Anzahl der Teilnehmer, Punkt-zu-Punkt, Broadcast.**

**Optimierung: Prinzip der Lokalität ,**

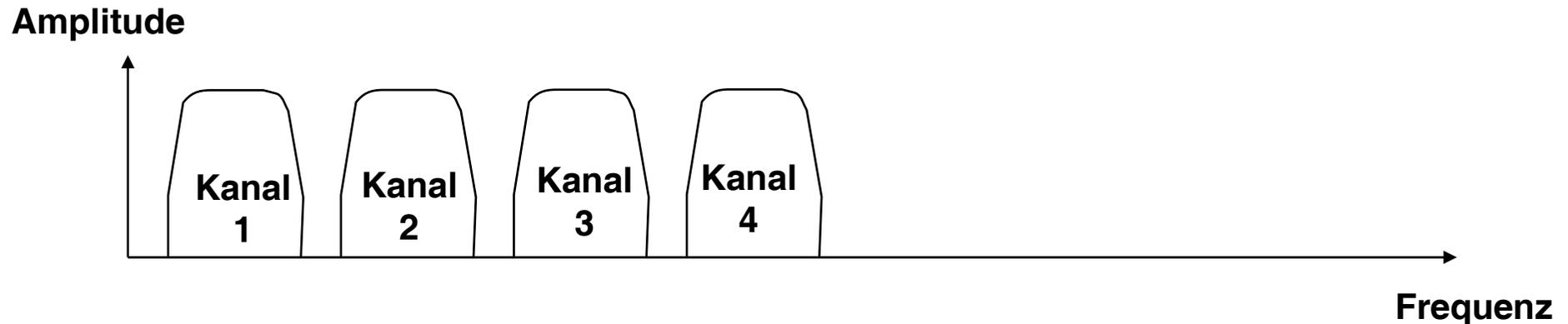
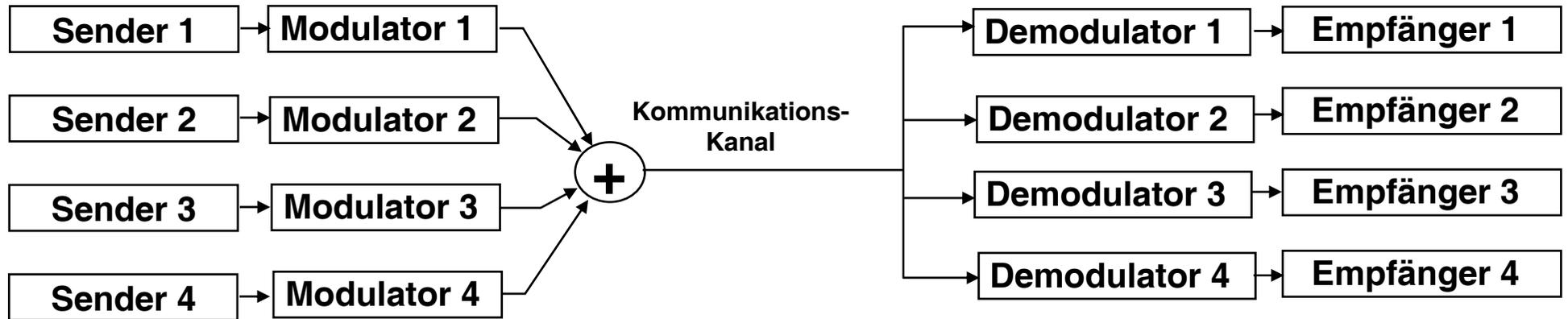
**Problem: Routing,**

**Struktur: Vollständig vernetzt, Bus, Ring, Baum, Stern, Gitter**

# Zeitschlitz-Multiplexing (Time-Division multiplexing)



## Frequenzband Multiplexing (frequency-division multiplexing)



# Wieviel Information kann man über ein Medium übertragen?

**Begrenzende Faktoren:**

→ **Bandbreite des Kanals**

→ **Störungen**

→ **Die Bandbreite begrenzt die Schaltfrequenz mit der man von einem zum anderen Signalpegel umschalten kann.**

→ **Die Störungen begrenzen die Anzahl der unterscheidbaren Signalpegel.**

Dämpfung der  
relativen  
Amplitude  
(in dB)



# Übertragungskapazität: (Shannon)

$$C = B \cdot \log_2 (P_s + P_n) / P_n = B \cdot \log_2 (1 + P_s / P_n)$$

**C** : Kanalkapazität (gemessen in Bit/sek (bps))

**P<sub>s</sub>** : Signalleistung (gemessen in  $\mu$ W, mW, W)

**P<sub>n</sub>** : Störleistung (gemessen in  $\mu$ W, mW, W)

**B**: Bandbreite

**P<sub>s</sub> / P<sub>n</sub>** : Störabstand (dB) =  $20 \cdot \log_{10} (P_s/P_n)$

**Beisp.**

**Telefon: Bandbreite 3000Hz, Störabstand 60 dB**  
(entspricht einem Verhältnis von 1/1000)

$$C = 3000 \cdot \log_2 (1+1000) = 3000 \cdot 9,97 = 29900 \text{ Bit/sek (bps)}$$

Ein Pegel ist definiert als Ausdruck für das Verhältnis elektrischer oder akustischer Größen gleicher Einheit zueinander. Er hat das Formelzeichen  $L$  (für level). Pegel werden in Bel (nach **Alexander Graham Bell**), dessen zehntem Teil Dezibel (Einheitenzeichen dB) oder in Neper (Einheitenzeichen Np) angegeben.

Der Pegel quadratischer Größen ist als **dekadischer Logarithmus** des Verhältnisses zweier Größen  $P_1$  und  $P_0$  anzugeben:

$$L = \lg \frac{P_1}{P_0} \quad (\text{in B}).$$

1 Bel entspricht daher dem Leistungsverhältnis 10:1.

Wegen der handlicheren Zahlenwerte ist die Angabe in Dezibel üblich. So ergibt sich:

$$L = 10 \cdot \lg \frac{P_1}{P_0} \quad (\text{in dB}).$$

Bei Leistungen gilt es zu berücksichtigen, dass hier ja bereits eine Multiplikation vorliegt (z.B. Watt = Strom \* Spannung = Spannung<sup>2</sup> / Widerstand).

$$P \propto U^2$$

$$L = 10 \cdot \lg \frac{P_1}{P_0} = 10 \cdot \lg \frac{U_1^2}{U_0^2} \quad (\text{in dB})$$

und somit

$$L = 20 \cdot \lg \frac{U_1}{U_0} \quad (\text{in dB}).$$

*Bel (B)* bzw. *Dezibel (dB)* sind keine physikalischen Einheiten sondern Hilfsgrößen. Sie stehen für bestimmte Rechenvorschriften.

$U_1/U_0$	dB	Beschreibung
1000	60	Verstärkung
100	40	Verstärkung
10	20	Verstärkung
3,16	10	Verstärkung
2	6	Verstärkung
1,414	3	Verstärkung
1	0	(1:1) Übertragung
0,7071	-3	Dämpfung
0,5	-6	Dämpfung
0,316	-10	Dämpfung
0,1	-20	Dämpfung
0,01	-40	Dämpfung
0,001	-60	Dämpfung

$$L = 20 \cdot \lg \frac{U_1}{U_0}$$

# Bps und BAUD

**Bps (Bit/sek) definiert die Bitrate**

**BAUD definiert die Anzahl der Pegeländerungen**

**Bit/sec wird durch die Kanalkapazität begrenzt !**

**BAUD wird durch die Bandbreite begrenzt !**

**Ansätze zur Erhöhung der bps-Rate bei gegebener BAUD-Rate des Kanals:**

- **viele unterscheidbare Signalpegel**
- **Codierung mit möglichst wenigen Übergängen**

## **Möglichkeiten der Codierung (Basisband)**

**Pegel  
Pulsbreite  
Übergänge**

### **Bitcodierung:**

<b>NRZ</b>	<b>(Non-Return-To-Zero)</b>
<b>Manchester</b>	
<b>MFM</b>	<b>(Modified-Frequency-Modulation)</b>

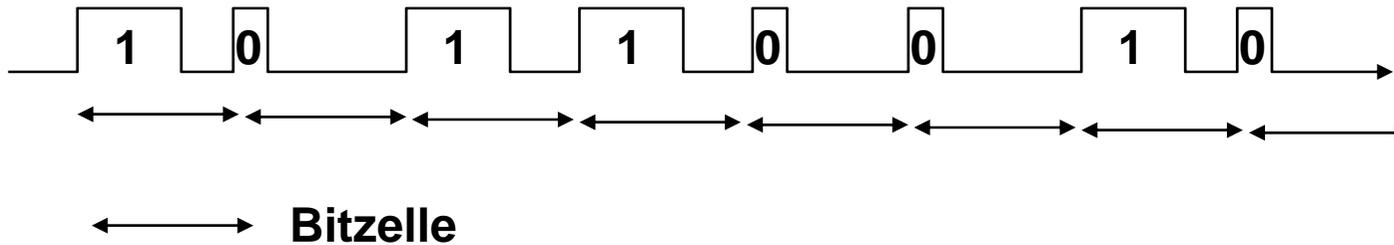
### **Probleme:**

**Synchronisation  
Anzahl der Flußwechsel  
Konstante/variable Paketlänge**

# Ein (schlechtes) Beispiel

RZ: (Always) Return to Zero (PWM)

Beisp.: 1011 0010

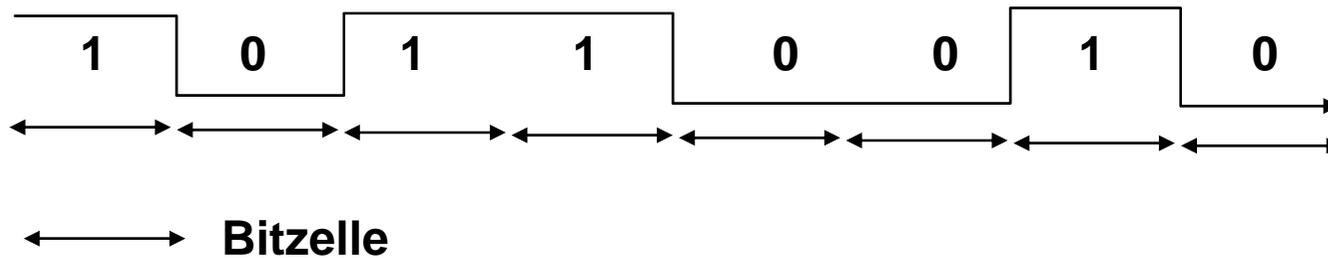


# NRZ Codes

---

NRZ: Non Return to Zero

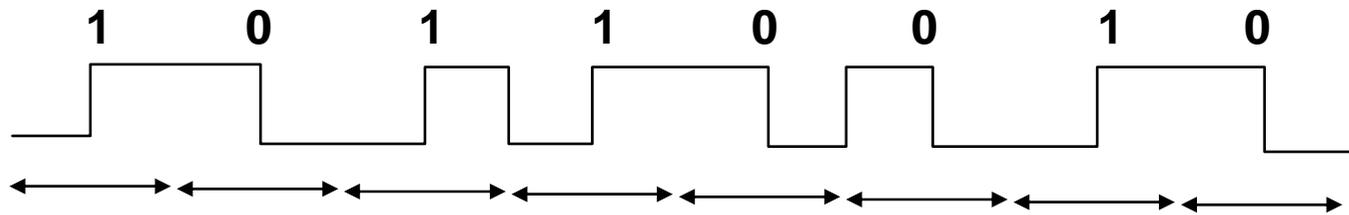
Beisp.: 1011 0010



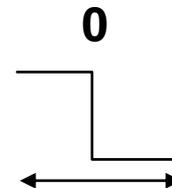
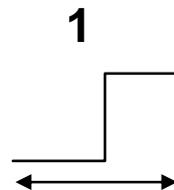
# Manchester Codierung

---

Beisp.: 1011 0010



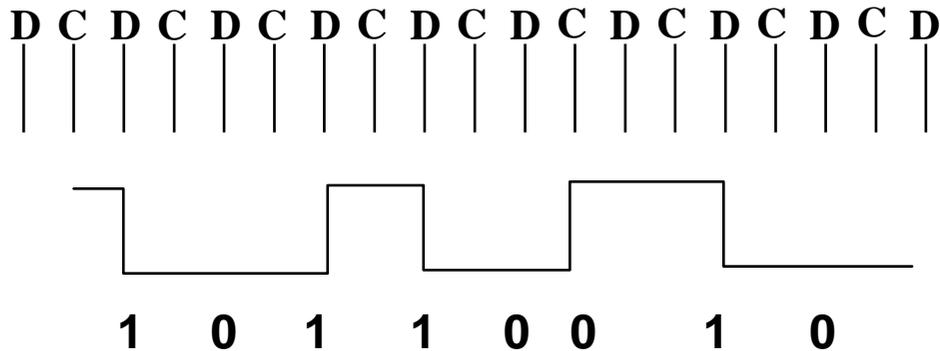
Bitzelle



# MFM (Modified Frequency Modulation) Codierung

---

Beisp.: 1011 0010



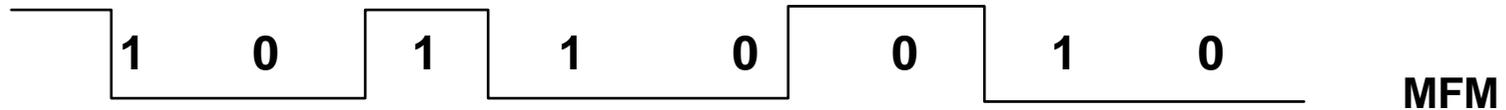
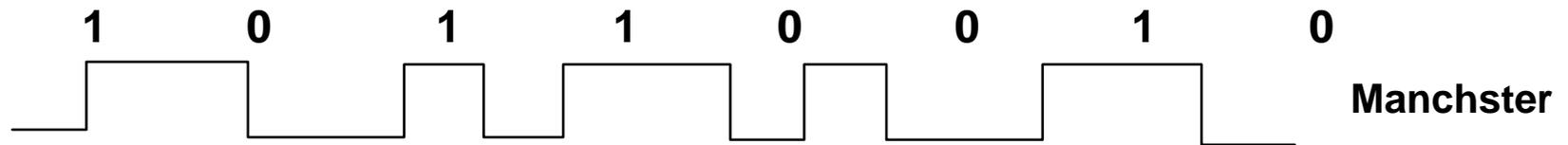
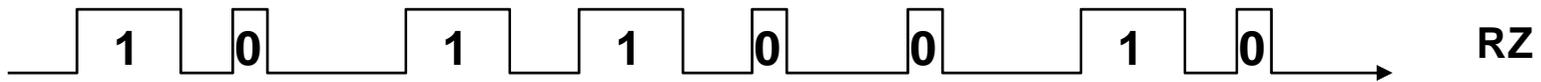
1: immer Pegeländerung an einem data point (D)

0: keine Pegeländerung an einem data point (D)

mehrere "0" hintereinander: Änderung am clockpoint (C)

# Vergleich der Codierungen

---



# Vergleich der Codierungen

---

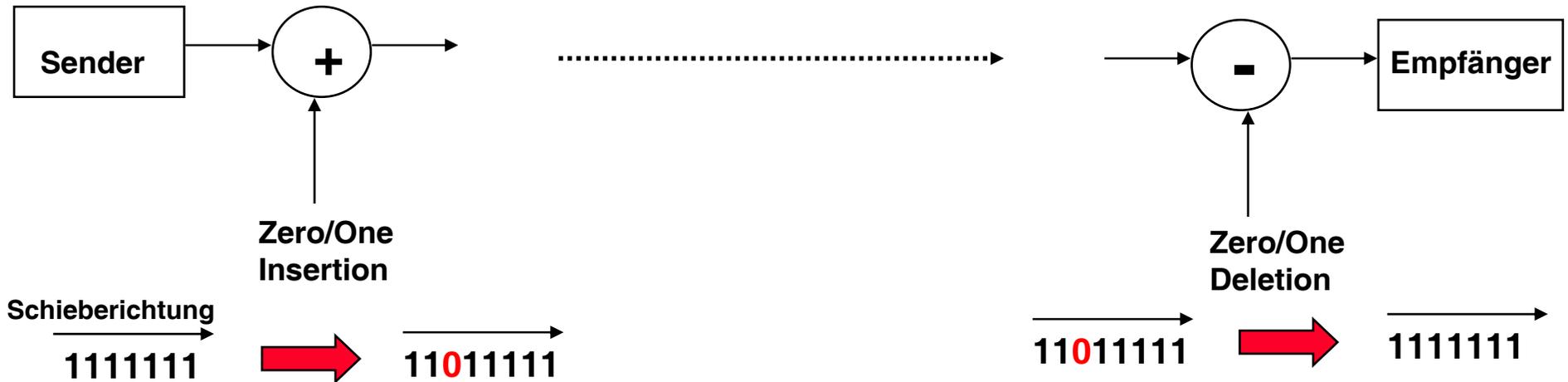
Typ	Synchronisierung	Übergänge/Bit im Mittel/Max		Feste Länge
RZ	ja	2	2	ja
NRZ	nein	>0,5	1	ja
NRZ*	ja	>0,5	1	nein
Manchester	ja	1,5	2	ja
MFM	ja	>0,5	1	ja

**NRZ\*: NRZ mit Bit Stuffing**

# Bit-Stuffing

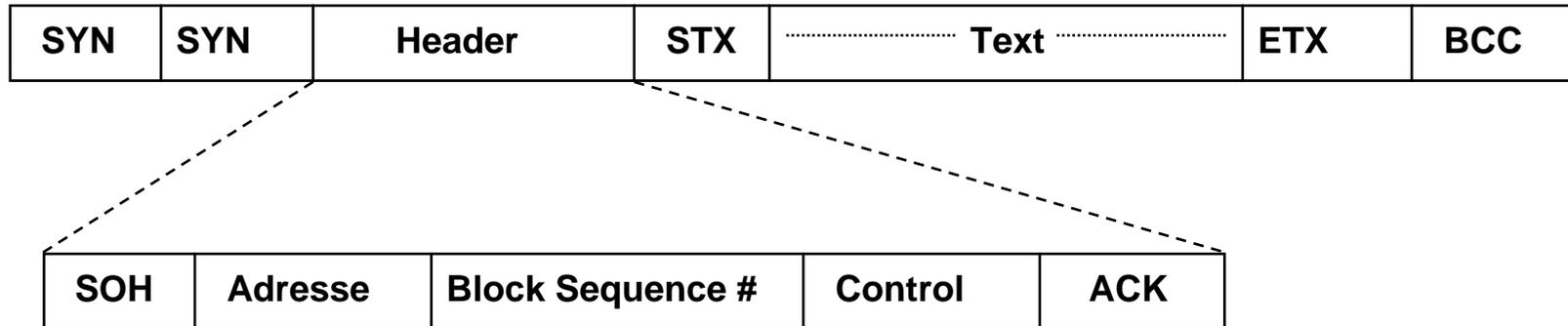
Nach einer bestimmten Anzahl  $n$  erkannter konsekutiver Pegel auf der Senderseite wird der komplementäre Pegel eingefügt, so daß nach maximal  $n$  aufeinanderfolgende Einsen oder Nullen sicher ein Übergang auftritt, der zur Synchronisation benutzt werden kann.

Auf der Empfängerseite wird die Nachricht durch das Entfernen der eingefügten Nullen wieder in den ursprünglichen Zustand versetzt.



# BiSync-Protokoll (IBM)

Byte-orientiert  
Character-orientiert



## ASCII-Code

**SYN :**  $16_{16}$   
**STX :**  $2_{16}$       **Start of Text**  
**ETX:**  $3$         **End of Text**  
**BCC:**            **Error Correcting Code**

**SOH:**  $1_{16}$       **Start of Header**  
**ACK:**  $6_{16}$       **Acknowledge**  
**NAK:**  $15_{16}$     **Negative Ack.**  
**ENQ:**  $5_{16}$       **Enquire Request.**

## HDLC (High Level Data Link Control)

### Problem:

Bei Character-orientierten Protokollen kann man Steuer- und Trennzeichen identifizieren (STX, ETX, etc.)

Bei Bit-orientierten Protokollen ist jede Kombination der Bits als Daten zulässig.

➡ Wie werden Steuerzeichen erkannt ?

### HDLC-Frame

Flag	Adresse	Control	Info	FCS	Flag
0111 1110	8	8	optional	FCS	0111 1110

I-Frame - Information Frame: Datentransport  
S-Frame - Supervisor Frame: Flußkontrolle, z.B. ACK, Retransmission  
U-Frame - Unnumbered Frame: weitere Steuerfunktionen z.B. connect, disconnect

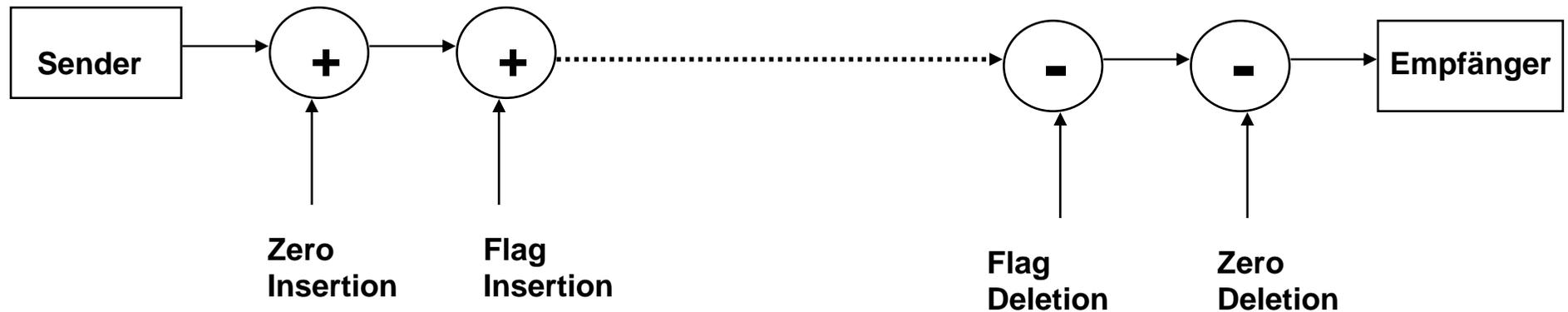
Wie wird das Flag 01111110 von Daten unterschieden ?

# Bit-Stuffing

Ziel: die Sequenz 01111110 tritt nur als Flag auf.

Verfahren: Nach 5 erkannten konsekutiven Einsen auf der Senderseite wird eine 0 eingefügt, so daß maximal 5 aufeinanderfolgende Einsen auftreten können. Das Flag wird in einer späteren Stufe an die Nachricht angehängt.

Auf der Empfängerseite wird die Nachricht durch das Entfernen des Flags und der eingefügten Nullen wieder in den ursprünglichen Zustand versetzt.



Schieberichtung

1111111

➔ 11011111

➔ 1101111101111110

➔ 11011111

➔ 1111111

# Flußkontrolle

**Flußkontrolle regelt des Informationsfluß zwischen den Kommunikationspartnern. Ziel ist die Synchronisation zwischen Sender und Empfänger, damit der Empfänger mit dem Sender Schritt halten kann.**

**Der Empfänger bestimmt die Geschwindigkeit der Kommunikation.**

**Unterscheidung zwischen:**

- **Impliziter Flußkontrolle**
- **Expliziter Flußkontrolle**

## Implizite Flußkontrolle

**Sender und Empfänger vereinbaren a priori, d.h. vor der Laufzeit, eine maximale Kommunikationsrate, die vom Sender nicht überschritten wird. Der Empfänger kann dann garantieren, daß er bei dieser Rate alle Nachrichten abnehmen kann.**

—————→ **Synchrone Kommunikation**

### **Eigenschaften:**

- **Zur Laufzeit werden keine Quittungen benötigt, d.h. der Kommunikationskanal ist unidirektional.**
- **Fehlererkennung erfolgt auf der Empfängerseite (Ausbleibende (aber erwartete) Nachrichten, fehlerhafte Übertragung).**
- **Implizite Flußkontrolle erleichtert die Realisierung von Multicast-Kommunikation (keine Quittungen von allen Kommunikationspartnern erforderlich).**

# Explizite Flußkontrolle

—————→ **Asynchrone Kommunikation**

**Der Sender schickt eine Nachricht an den Empfänger und wartet, bis er ein explizites Quittungssignal bekommen hat, das den Empfang der Nachricht bestätigt.**

**Der Sender steht damit unter der Kontrolle des Empfängers (liegt in dessen “Sphere of Control”). Der Empfänger bestimmt die Geschwindigkeit der Kommunikation, bzw. wann der Sender eine Nachricht senden darf (auch “back pressure flow control” genannt).**

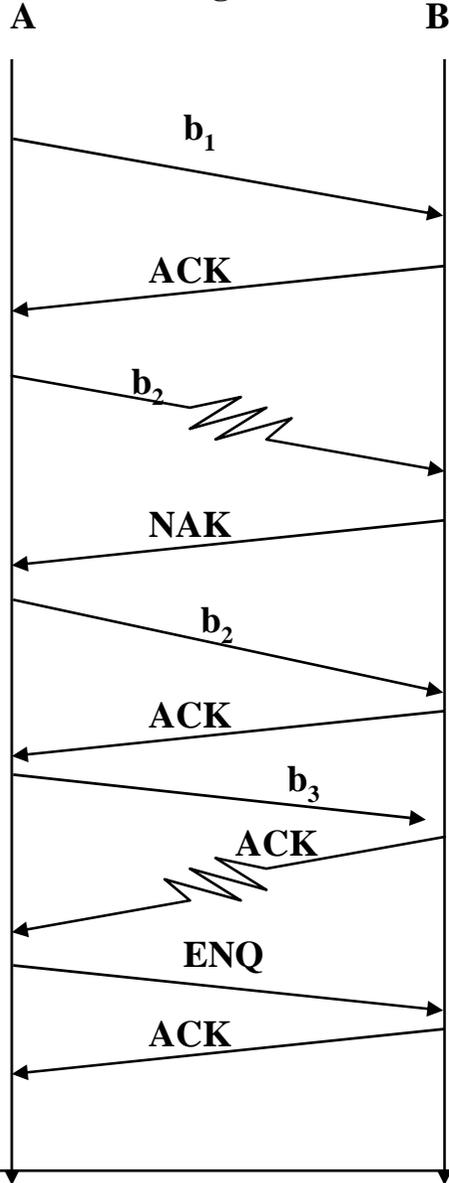
**Fehlererkennung liegt beim Sender auf der Basis des expliziten Quittungssignals.**

**Gründe für ein ausbleibendes Quittungssignal:**

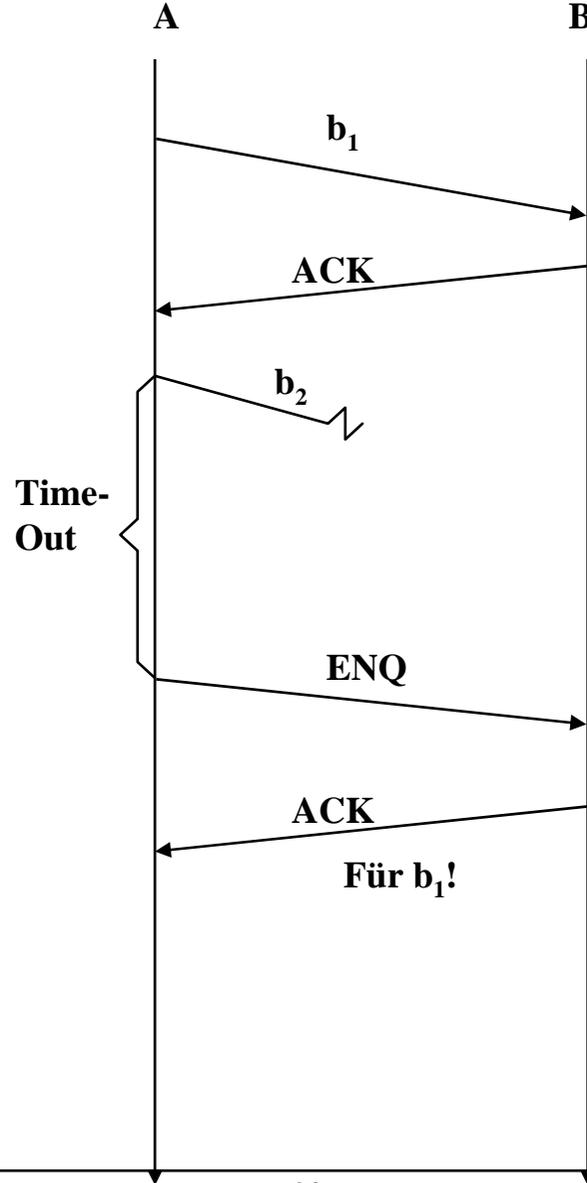
- **der Empfänger hat die Nachricht nicht bekommen,**
- **der Empfänger hat das Quittungssignal noch nicht geschickt,**
- **das Quittungssignal ging verloren,**
- **der Empfänger ist ausgefallen.**

# Beispiel für ein PAR (Positive Acknowledge and Retransmission) Protokoll - IBM BiSync

Nachrichtenaustausch mit Störungen



Problem:  
Verlorene Nachrichten



Lösung:  
Alternierende ACKs

