Physical Network Layer

Properties of communication networks

Constraining factors:

- **Transfer rate, (capacity, bandwidth)**
- **Propagation latency**

Transfer rates:
- **Morse-telegraph: < 100Bit/sec**
- **Telegraphy: < 150 Bit/sec**
- **Phone: ~ 50Kbit /sec**
- **Serial RS232: ~ 100Kbit/sec**
- **Field bus: few Kbit/sec ... ~ 1Mbit/sec**
- **Ethernet: 10-1000Mbit/sec**
- **High speed networks: >> Gbit/sec**

Latency: **Satellite connection (2 x 35700 km): ~ 240 ms,**
cabel (trans-atlantic) (~ 6.000 km): ~ 20 ms

Topology: point-to-point, star, bus, tree, grid, multi-level....

The Physical Layer Issues

- **Asynchronous serial transmission   (character oriented)**
- **Synchronous serial transmission   (bitsynchronization)**

- **Bit coding:**
  - **NRZ (Non-Return-to-Zero**
  - **Manchester Code**
  - **MFM (Modified-Frequency-Modulation)**

- **Modulation and data transmission:**
  - **Base band   Example: Morsetel. / Ethernet**
  - **Broad band   Example: Radio, TV, Cabel-TV, Modem**
    Modulation: AM, FM

- **Transmissionmedia:**
  - **Fiber   (Multi-Mode, Single-Mode)**
  - **Copper   (Twisted Pair, Coaxial)**
  - **Radio   (Frequency band)**
  - **Satellite   (Geostationary, orbiting)**

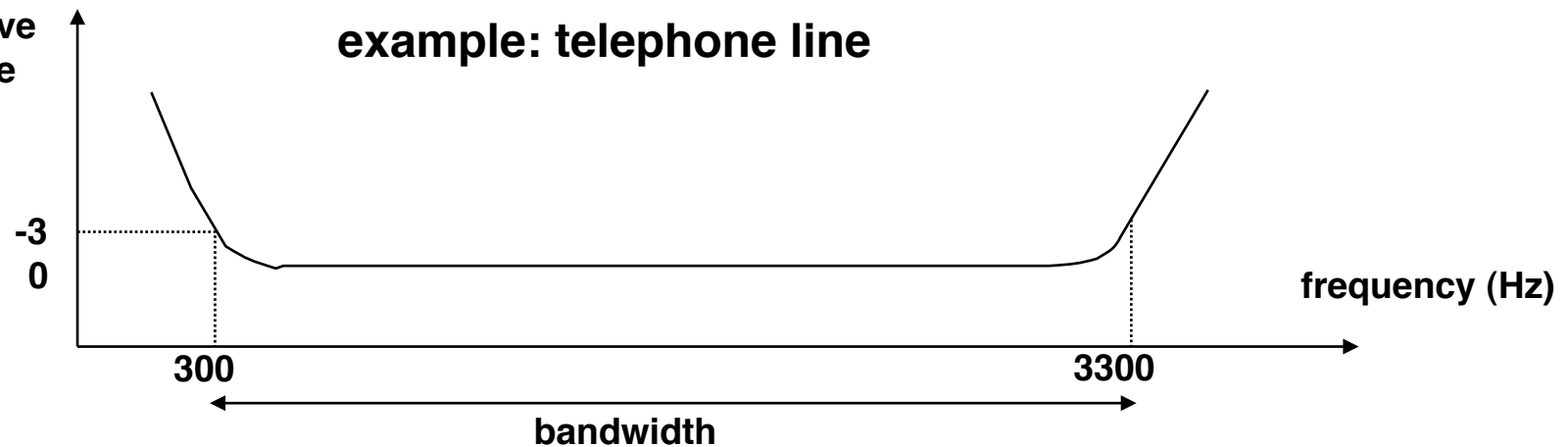# How much information can be transfered over a line?

limiting factors:

⟶ **bandwidth of the channel**

⟶ **noise**

⟶ The bandwidth limits the number of transitions, i.e. the frequency of switching from one signal level to the other

⟶ Noise (informally) limits the ability to distinguish between multiple signal levels

example: telephone line

Attenuation of the relative amplitude (in dB)

-3
0

300   3300   frequency (Hz)

bandwidth

Capacity of a channel (Shannon):

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

C : capacity of a channel   (measured in Bit/sec (bps))
P_s : signal strength (measured in μW, mW, W)
P_n : noise (measured in μW, mW, W)
B: bandwidth
$P_s\,/P_n$: signal-to-noise ratio

Example:
Telephone: bandwidth 3000Hz, signal-to-noise ratio 1000/1

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

Bps and BAUD

Bps (Bit/sec) defines a Bit rate (capacity of the channel)

BAUD defines the number of level transitions

Bit/sec is contraint by the channel capacity !

BAUD is constraint by the channel bandwidth !

Basic methods to increase the bps-Rate at a given BAUD rate of the channel:

- distinguish multiple levels
- Coding with the smallest number level transitions

Coding options (base band)

>   level
>   pulse width
>   transitions
>   phase

Bit coding:

>   NRZ   (Non-Return-To-Zero)
>   Manchester
>   MFM   (Modified-Frequency-Modulation)
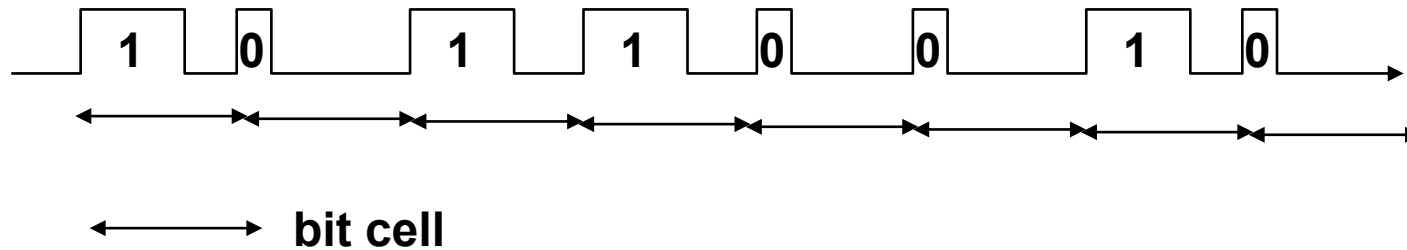
Problems:

>   synchronization
>   number of transitions
>   constant/variable frame length

A simple coding scheme
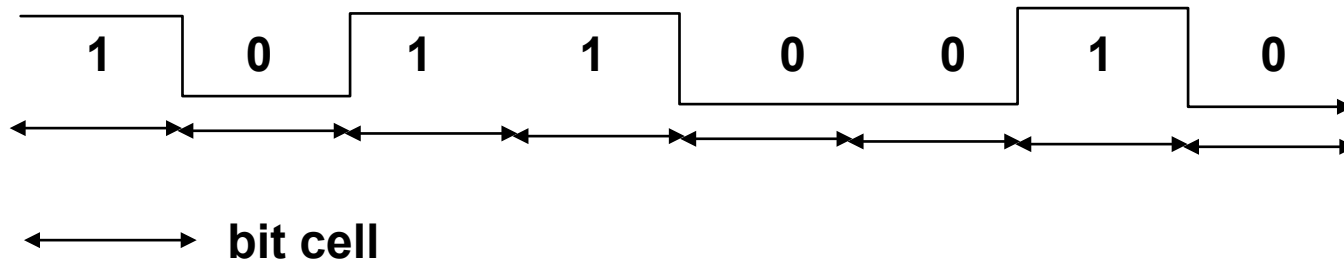
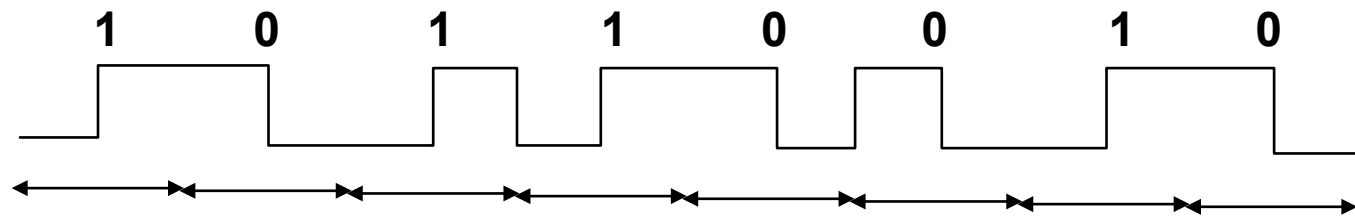RZ: (Always) Return to Zero (PWM)

Example: 1011 0010

bit cell

9

NRZ Codes

NRZ: Non Return to Zero

Example: 1011 0010

| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |

bit cell

Manchester Coding

Example: 1011 0010

1   0   1   1   0   0   1   0

bit cell

1   0

MFM (Modified Frequency Modulation)

Example: 1011 0010

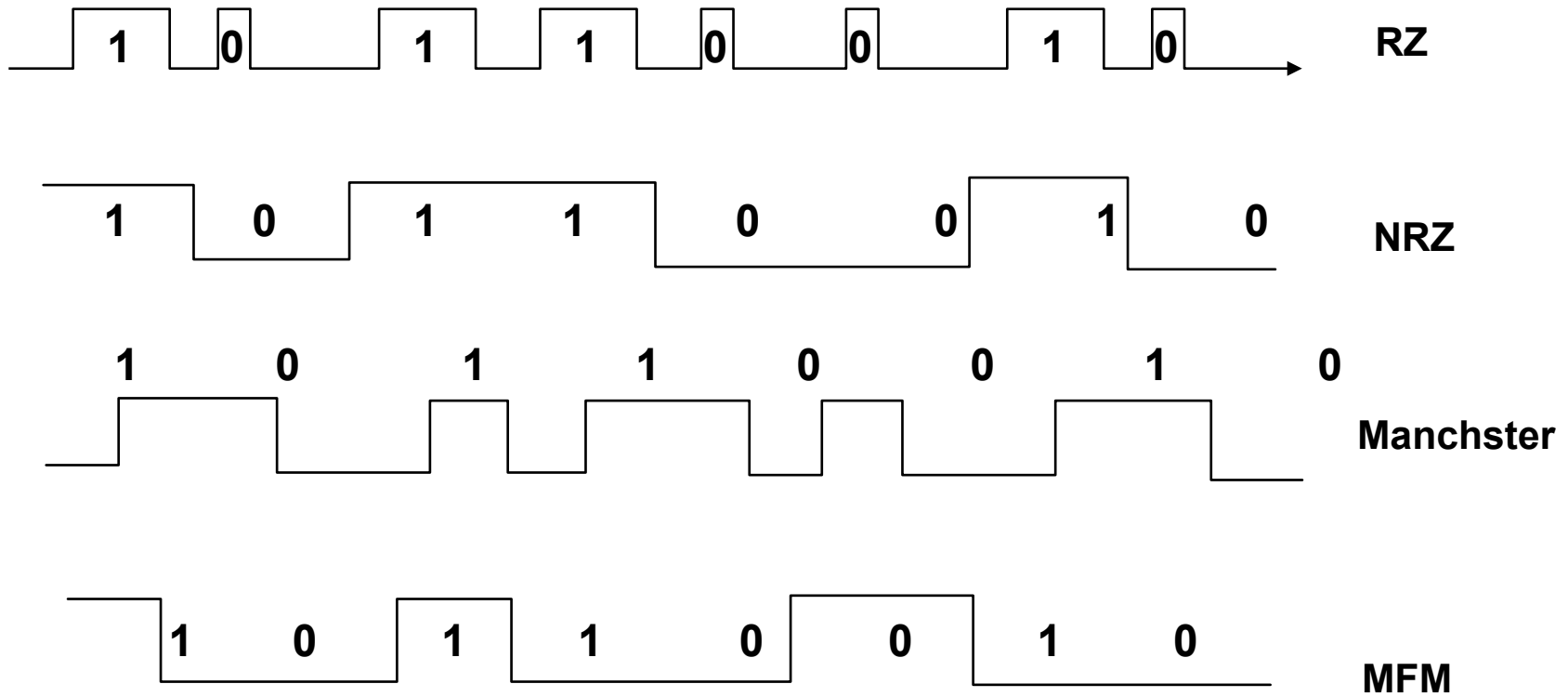D C D C D C D C D C D C D C D C D C D C D C D C D

1 0 1 1 0 0 1 0

1: always transition at a data point (D)

0: no transition at a data point (D)

multiple consecutive "0" : transistion at a clock point (C)

Comparison of Codes

RZ

NRZ

Manchster

MFM

Comparison of Codes

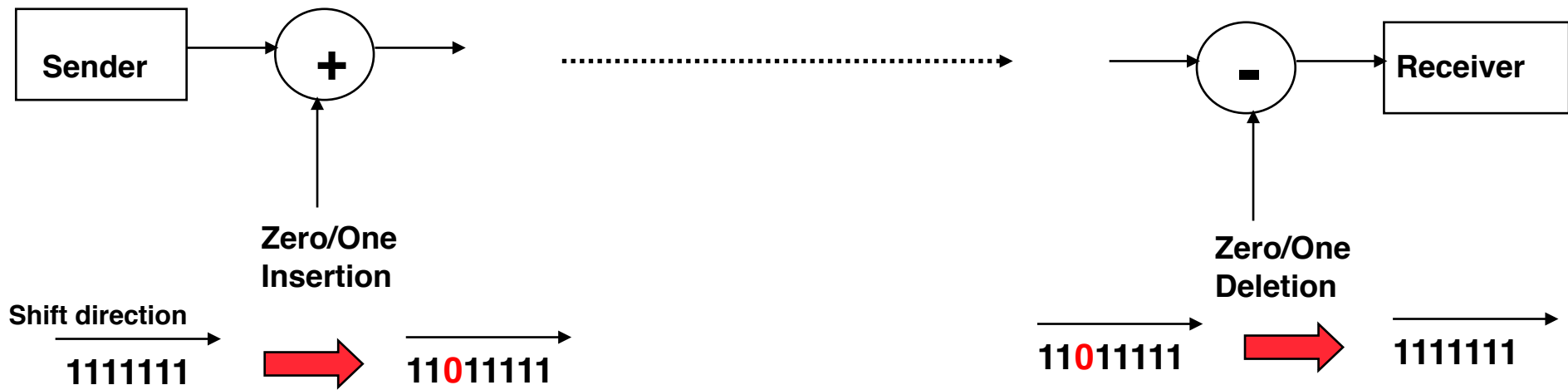| Type	Synchronization	transitions/Bit average/max		fixed length
RZ	Y	2	2	Y
NRZ	N	>0,5	1	Y
NRZ*	Y	>0,5	1	N
Manchester	Y	1,5	2	Y
MFM	Y	>0,5	1	Y

NRZ*: NRZ mit Bit Stuffing

Bit-Stuffing

When a long sequence of identical values "0" or "1" occurs, bit stuffing inserts a complemetary signal level after a fixed specified number of equal signal levels.

Sender transparently inserts stuff bits. The receiver re-establishes the original message by removing the respective stuff bits.

Sender $\rightarrow$ **+** $\cdots\cdots\cdots\rightarrow$ **-** $\rightarrow$ Receiver

Zero/One Insertion

Zero/One Deletion

Shift direction

1111111 $\Rightarrow$ 11011111

11011111 $\Rightarrow$ 1111111
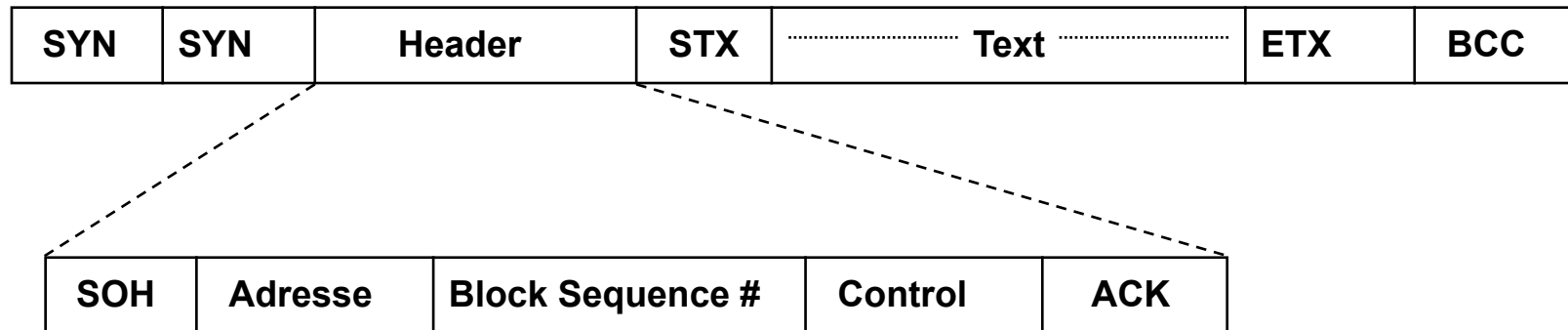
Bit-Stuffing and framing

Motivating examples:

      **- character-oriented protocols
e.g. IBM BiSync**

      **- binary protocols
e.g. IBM HDLC**

BiSync-Protokoll (IBM)

Byte-orientiert
Character-orientiert

SYN	SYN	Header	STX	Text	ETX	BCC

SOH	Adresse	Block Sequence #	Control	ACK

ASCII-Code

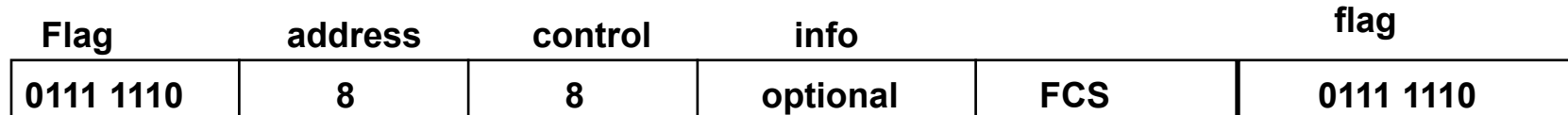|		
SYN :	16_{16}	
STX :	2_{16}	Start of Text
ETX:	3	End of Text
BCC:		Error Correcting Code
SOH:	1_{16}	Start of Header
ACK:	6_{16}	Acknowledge
NAK:	15_{16}	Negative Ack.
ENQ:	5_{16}	Enquire Request.

Bit stuffing to identify message boundaries

Example: HDLC (High Level Data Link Control)

Problem:
In character-oriented protocols, control and seperation characters (STX, ETX, etc.) can be identified easily..
In bit-oriented protocols any combination of bits as data is possible!

➡ How to identify control information?

HDLC-Frame

Flag	address	control	info		flag
0111 1110	8	8	optional	FCS	0111 1110

I-Frame - Information Frame: data transport
S-Frame - Supervisor Frame: flow control, e.g. ACK,
                                                      re-transmission
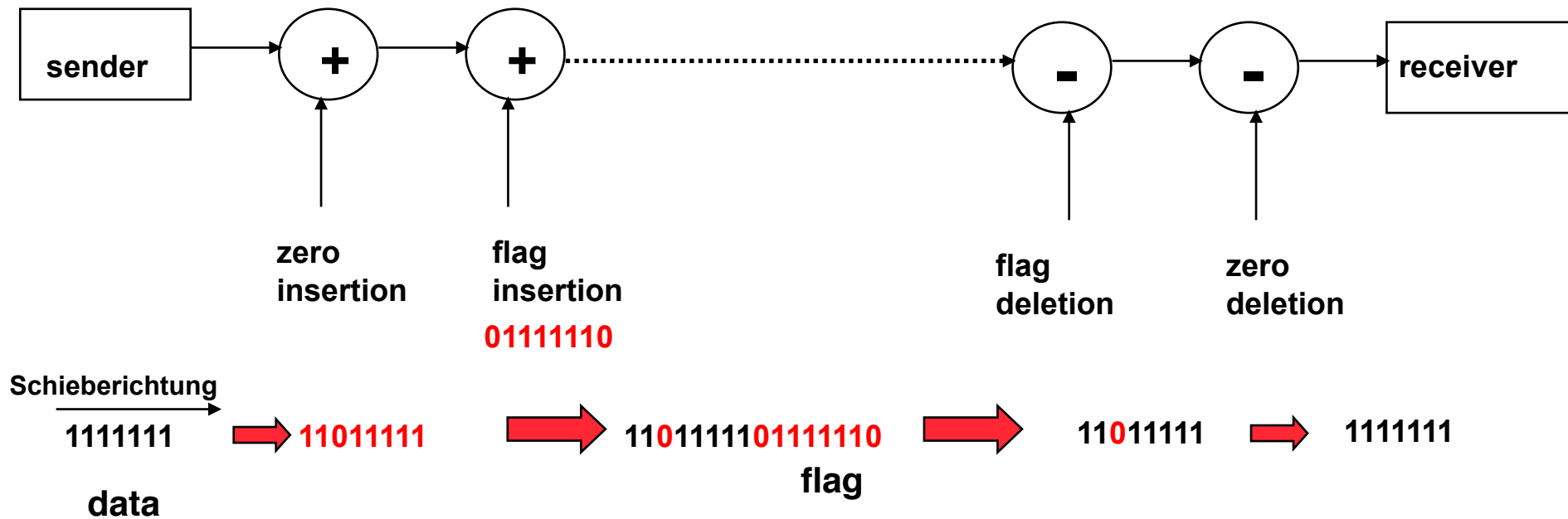U-Frame - Un-numbered Frame: additional control info
                                                      e.g. connect, disconnect

How to distinguish the control info 01111110 from data ?

Bit stuffing to identify message boundaries

goal: recognizing the flag "01111110".

method: The sender normally inserts a stuff bit "0" after 5 consecutive "1".
Therefore there is a max. number of five consecutive "1". The flag is
inserted AFTER the bist stuffing stage in the sender and detected and removed before the
receiver stuffing stage.

zero insertion

flag insertion
01111110

flag deletion

zero deletion

Schieberichtung
1111111 ⇒ 11011111 ⇒ 110111101111110 ⇒ 11011111 ⇒ 1111111
data   flag

Bit stuffing

- ensures synchronization

- needed for framing in binary protocols

- exploited for special signals