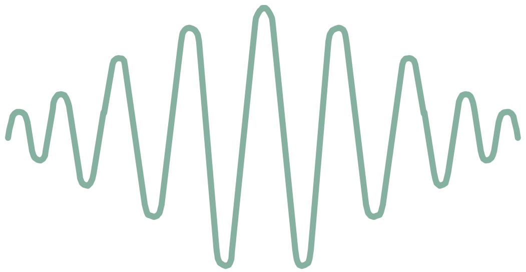
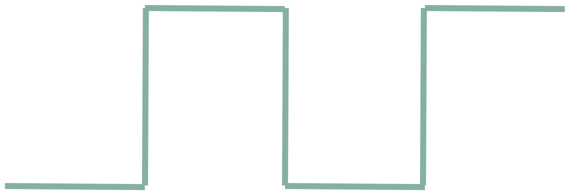
Transmission of data requires physical representation of data

A Signal is the physical representation of data

An analog signal is a sequence of continuous values

A digital signal is a sequence of discrete values



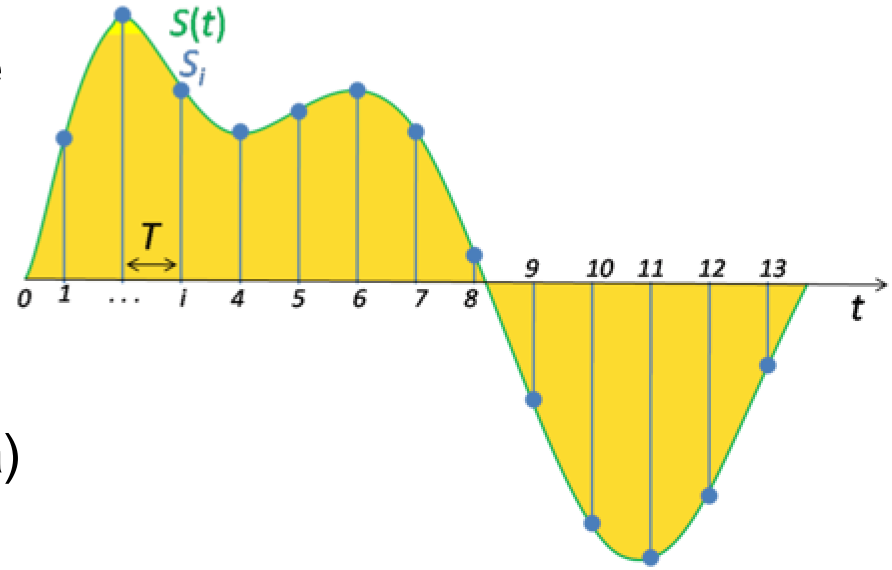


Computers deal with digital signals

The physical layer transmits data based on signals through space

The need to convert - **Quantization**

- Computers can only deal with digital data => discrete signal
- Physical mediums are by nature analog => continuous signal
- Must convert from digital signal to analog signal (and vice versa)



Source: https://en.wikipedia.org/wiki/Sampling_(signal_processing)

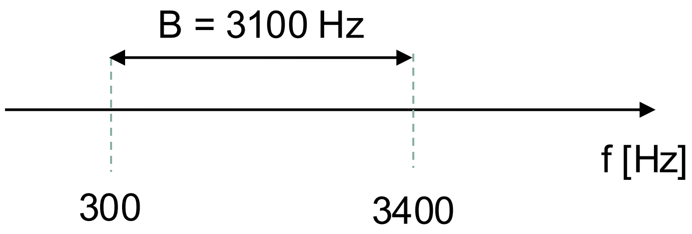
The need to measure - **Sampling**

- Computers can only deal with discrete time
- Physical mediums' state vary continuously
- Must rely on periodical measurements of the physical medium (> 2 * bandwidth of the signal)

Definition: Bandwidth & Throughput

Bandwidth

- Interval of the spectrum, difference between upper and lower frequencies, measured in Hertz
- Example: classical telephony over copper wires supports 300 – 3400Hz, thus bandwidth B = 3400Hz - 300Hz = 3100 Hz



Throughput

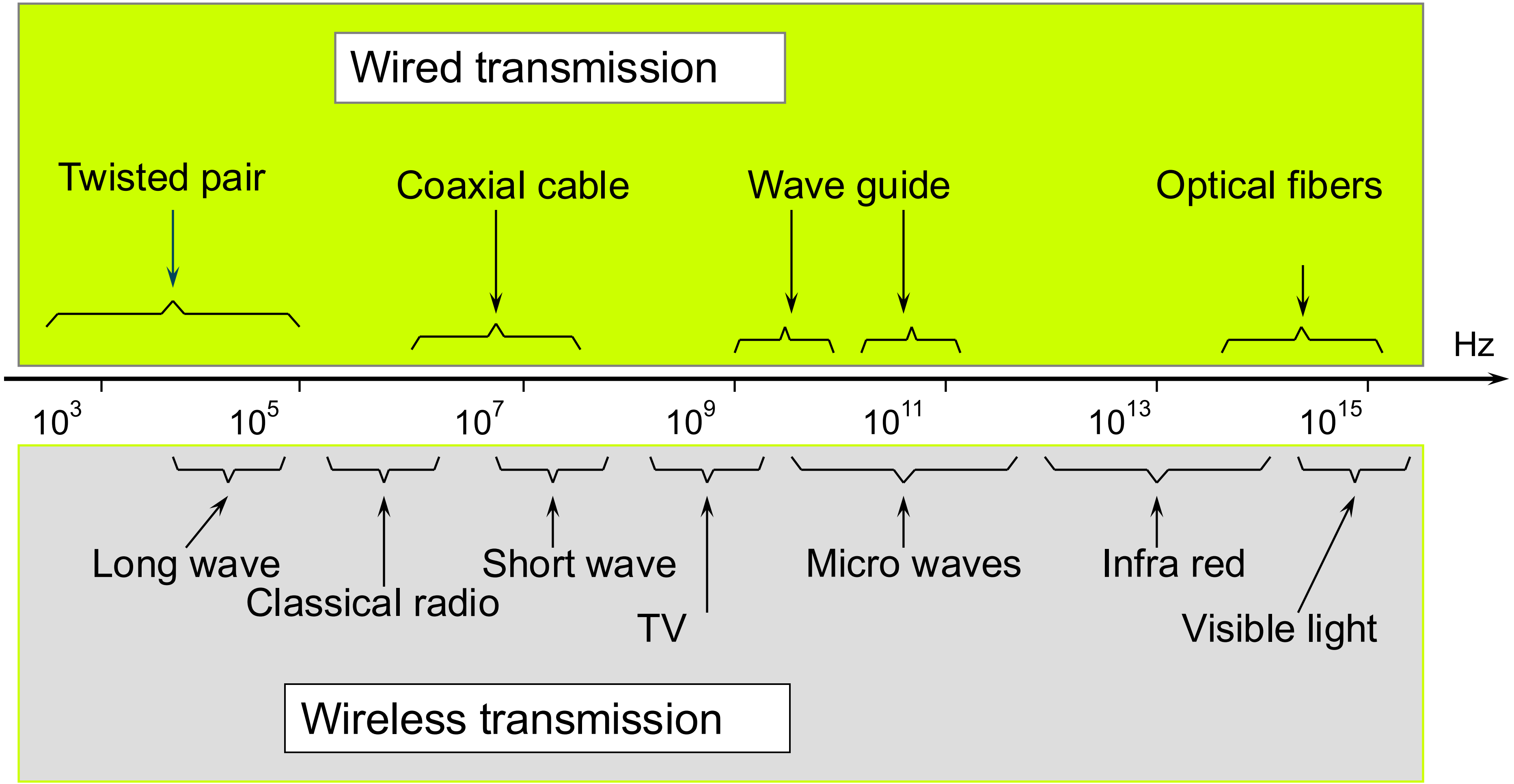
- Amount of bits per second transmitted over a system, measured in bit/s

Shannon brings this together

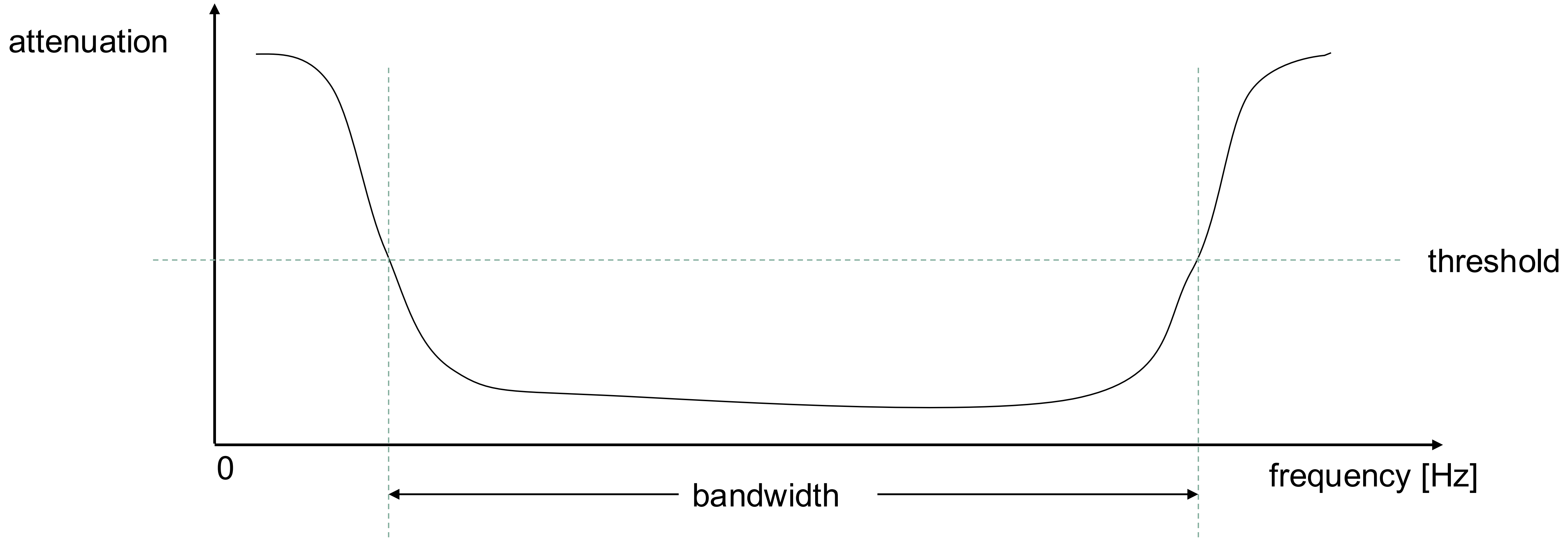
- Achievable data rate is limited by noise in real systems
- More precisely: by relationship of signal strength compared to noise, i.e., Signal-to-Noise Ratio (SNR, S/N)
- S: signal strength; N: noise level; B: bandwidth in Hz;
- S/N commonly expressed in dB: S/N [dB] = $10 \times \log_{10}(S/N)$

$$\textbf{maximum throughput [bit/s]} = \boldsymbol{B \log_2 (1 + S/N)}$$

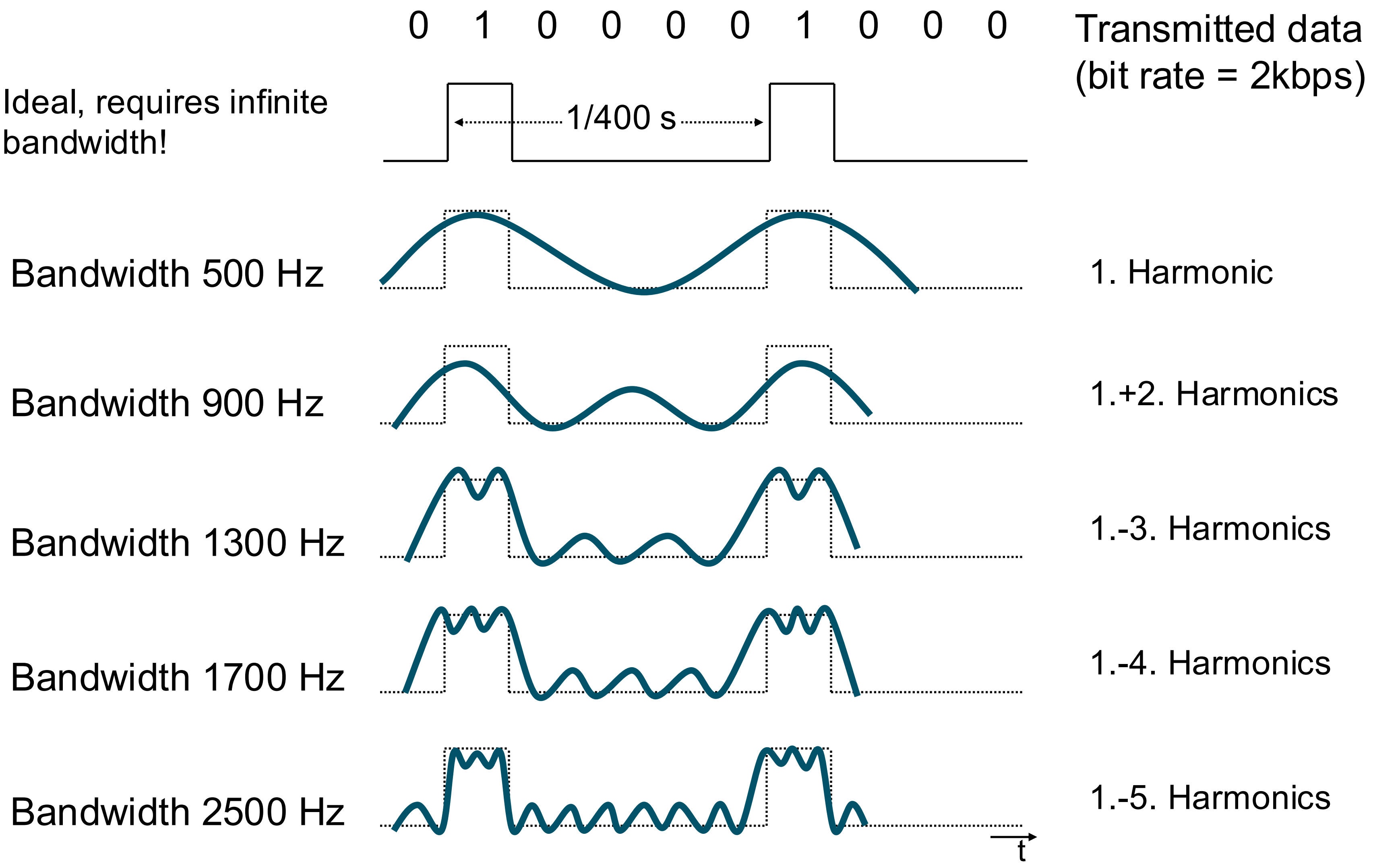
Electromagnetic Spectrum



Real technical systems are always bandwidth-limited



Bit rate *vs.* bandwidth: Medium limiting harmonics



Tasks of Physical Layer

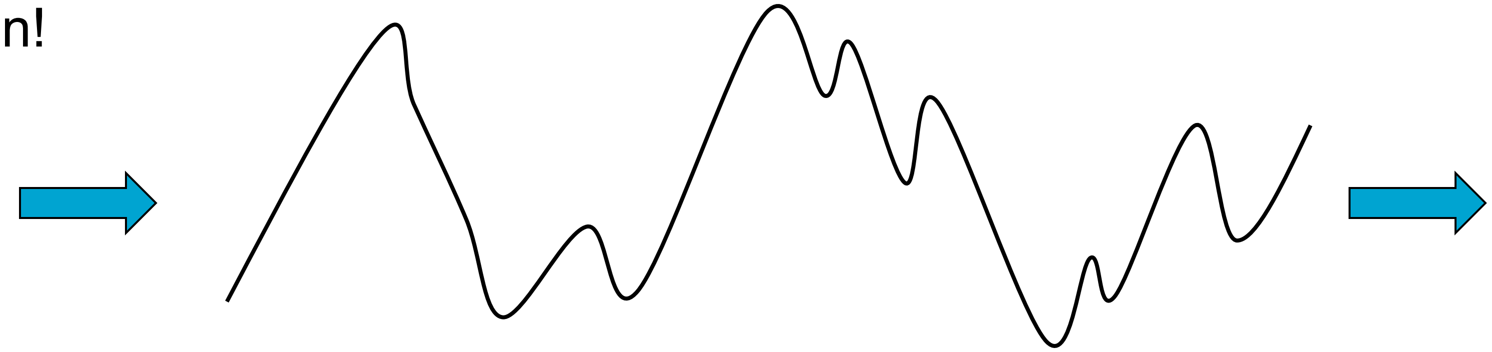
Responsible for turning a logical sequence of bits into a physical signal that can propagate through a medium

- Many forms of physical signals
- Signals are limited by their propagation in a physical medium
- Limited bandwidth, attenuation, dispersion, and by noise

Includes connectors, media types, voltages, …

No error correction!

01001010110101



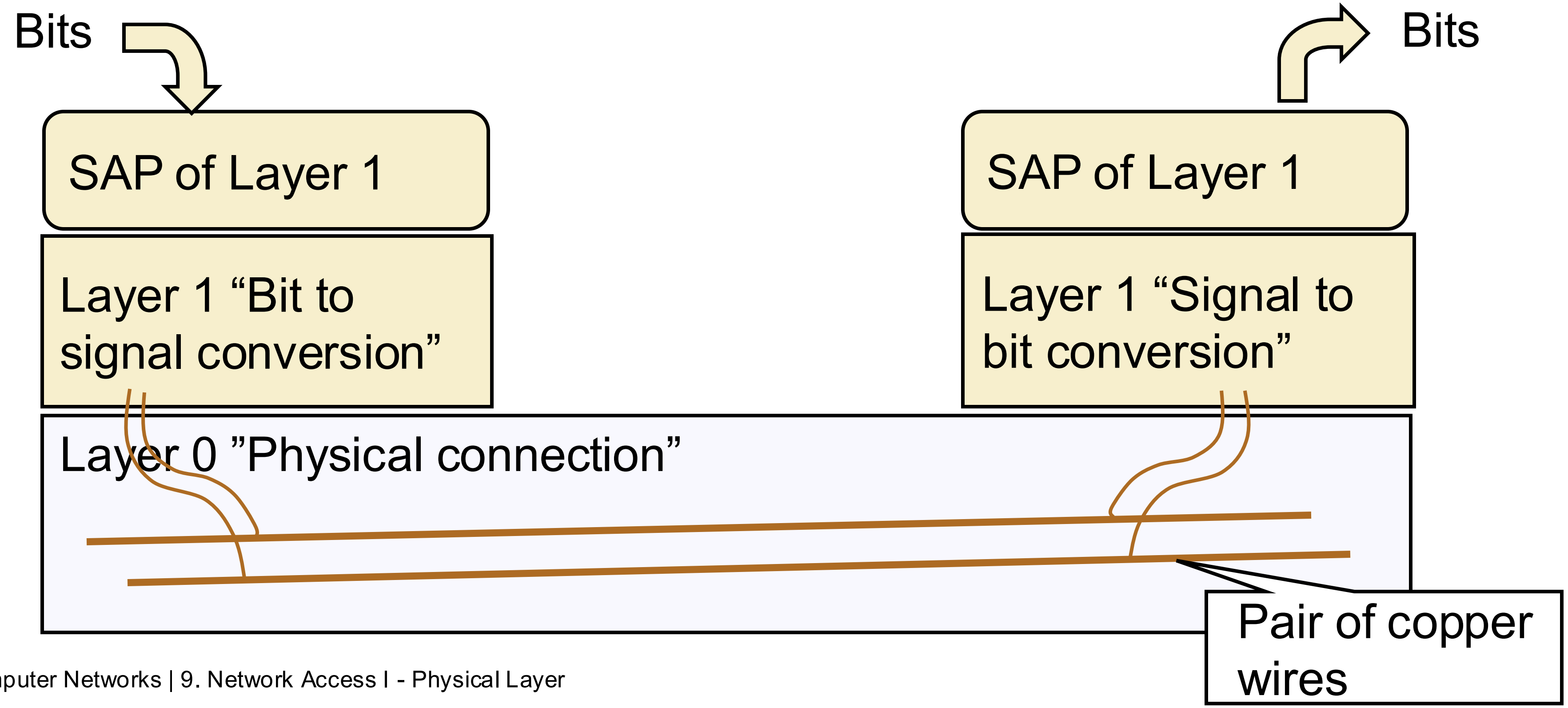
01001011001101

Basic Service of Physical Layer: Transport Bits

Physical layer should enable transport of bits between two locations A and B

Abstraction: Bit sequence (in order delivery)

- But no guarantee on correct transmission of bits

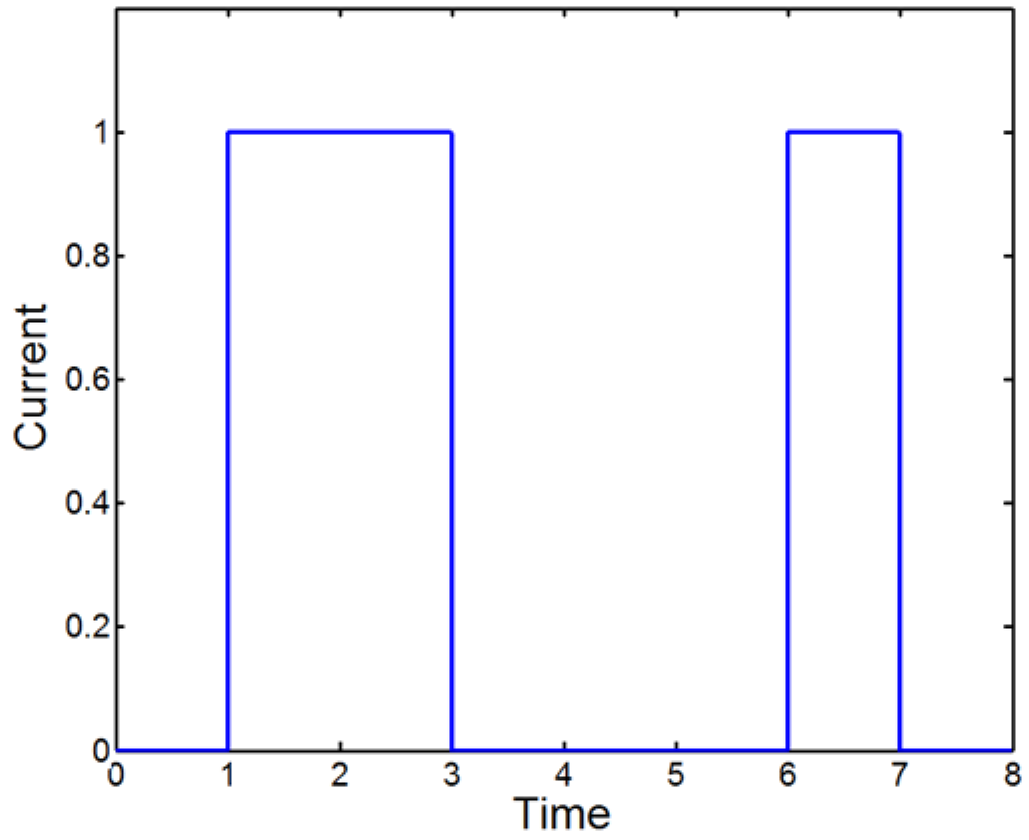


Example: Transmit Bit Pattern for Character “b”

Represent character “b” as a sequence of bits

Use ASCII code → “b” = 98, as binary number 01100010

Resulting current on the wire:



Note: Abstract ***data*** is represented by physical ***signals*** – changes of a physical quantity in time or space!

Basic Signal Processing: Periodic Signals

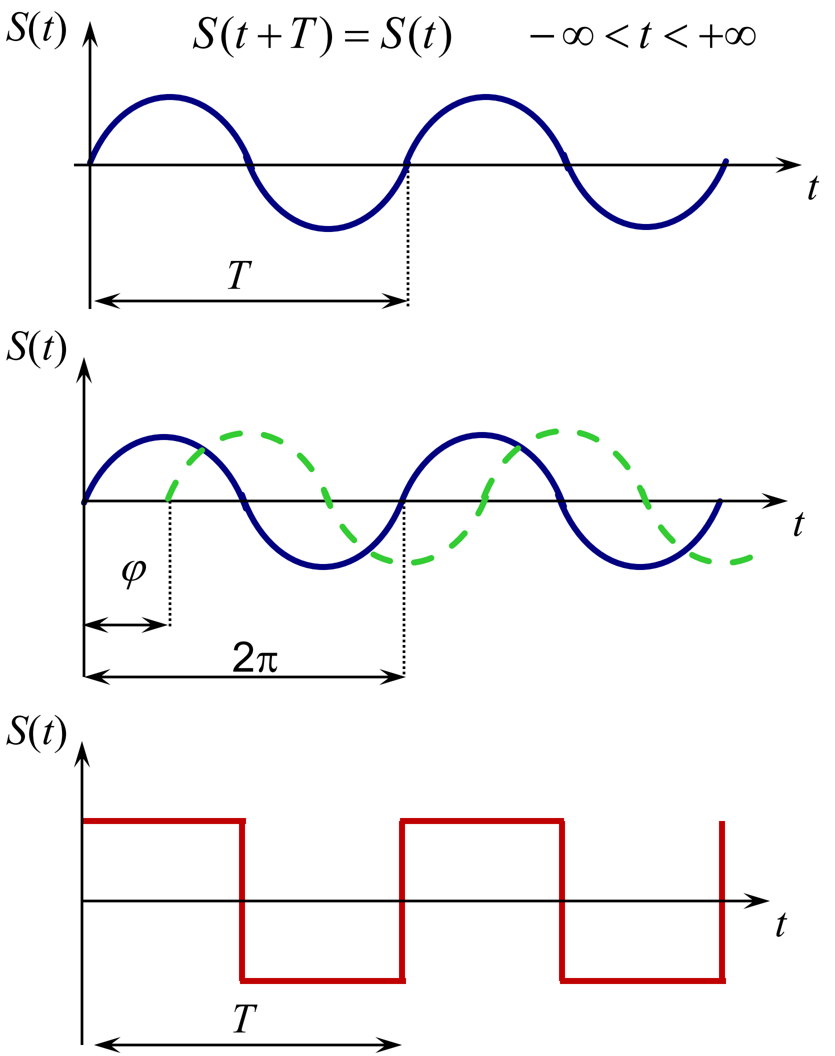
Periodic signals: the simplest signals

Parameters of periodic signals:

- Period T
- Frequency $f = 1/T$
- Amplitude $S(t)$
- Phase φ

Examples:

- Sinus (period = 2π)
- Phase shift φ
- Square wave

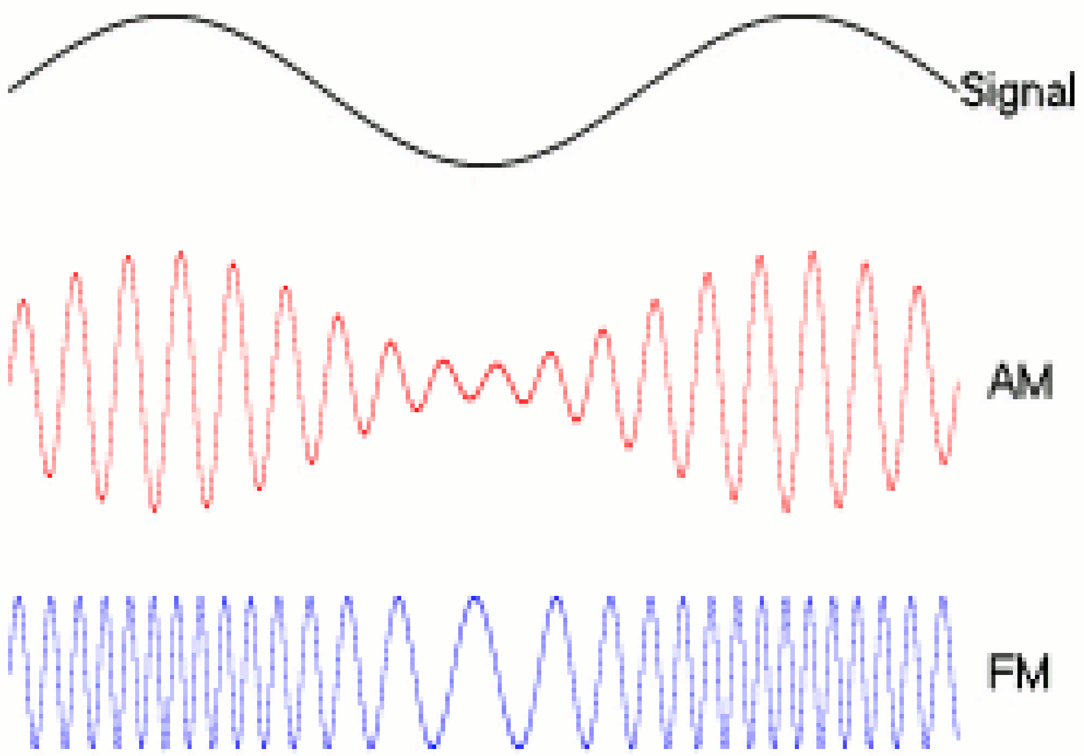


Basic Signal Processing: Modulation

Periodic signals: the simplest signals

Parameters of periodic signals:

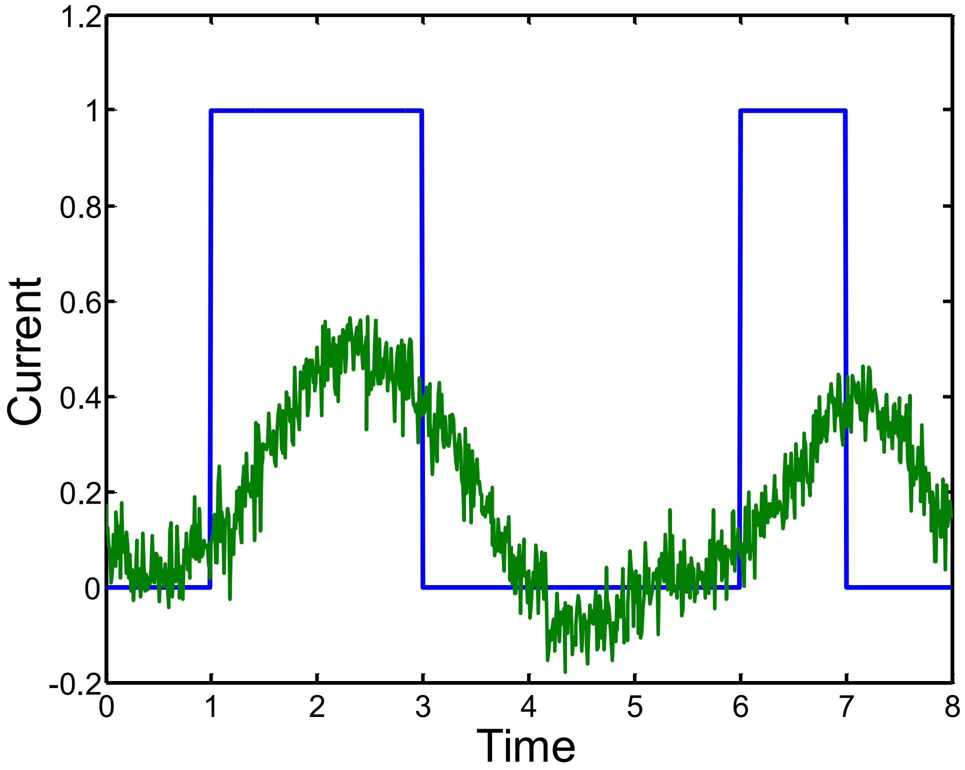
- Period T
- **Frequency $f = 1/T$**
- **Amplitude $S(t)$**
- **Phase φ**



Bildquelle: https://de.wikipedia.org/wiki/Modulation_(Technik)

What Arrives at the Receiver?

Use ASCII code → "b" = 98, as binary number 01100010
Typical pattern at the receiver:



➢What is going on here and how should we convert the signal back to a "b"?

Interference

Noise

- Background noise
- Thermal

Echoes

- E.g. at connections

Crosstalk

- E.g. interference across wires

ELF

- Extreme low frequency, e.g. 50/60 Hz AC

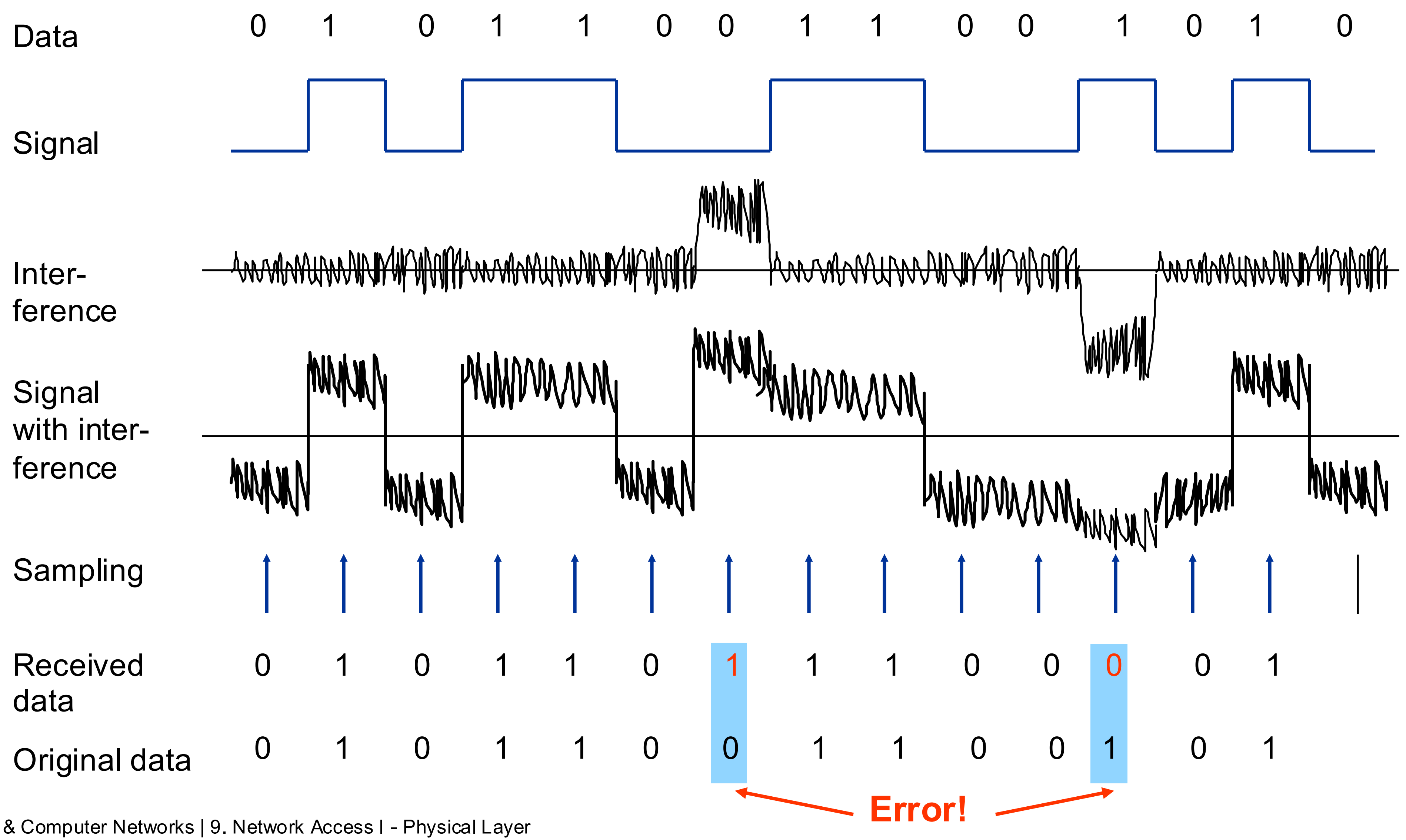
Spikes

- Short, high amplitude

…

Plus: attenuation, dispersion, refraction, phase shift …

Example: Results of Interference



When to Sample Received Signal?

How does the receiver know *when* to check the received signal for its value?

- One typical convention: In the middle of each symbol
- But when does a symbol start?
  - The length of a symbol is usually known by convention via the symbol rate

The receiver has to be *synchronized* with the sender at bit level

- Link layer will have to deal with *frame synchronization*
- There is also *character synchronization* – omitted here

Overly Simplistic Bit Synchronization

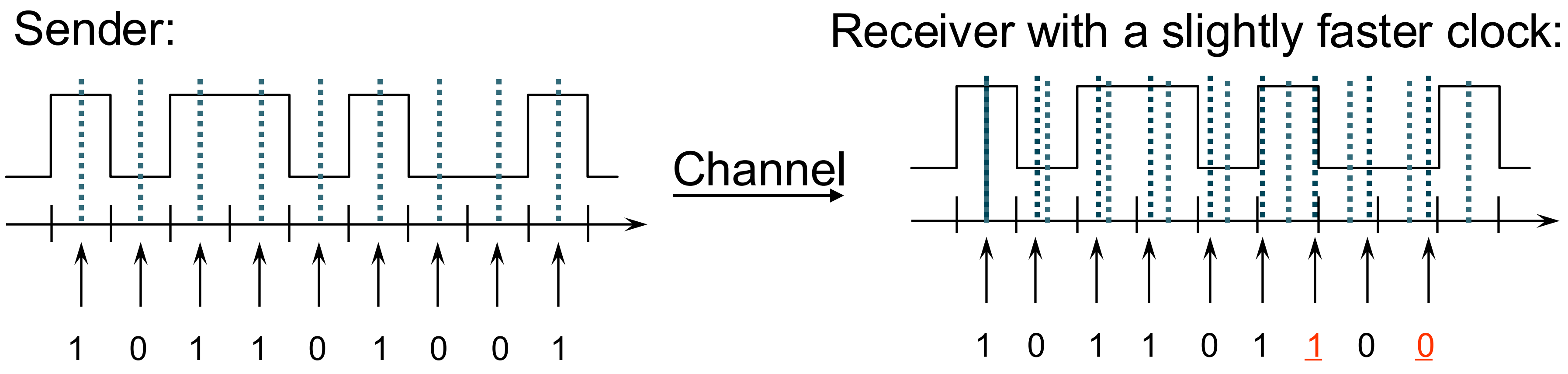
One simple option, assume

- ... that sender and receiver are synchronized at some point in time
- ... that both have an internal clock that tics at every symbol step

➢Usually, this does not work due to *clock drift*

- Two different clocks never stay in perfect synchrony

➢Errors, if synchronization is lost:

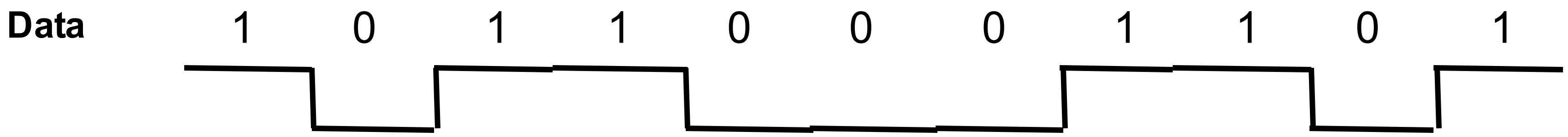


Extract Clock Information from Signal

Put enough information into data signal itself so that receiver can know immediately when a bit starts/stops

➢Would the simple 0/low, 1/high mapping of bit/symbol work?

Receiver can use 0-1-0 transitions to detect length of a bit



Fails depending on bit sequences, e.g. long runs of 1s/0s

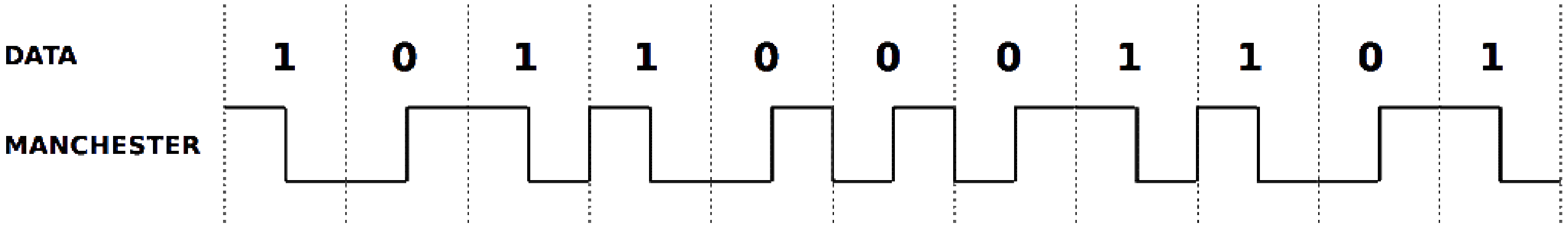
➢ Receiver can loose synchronization

➢Not nice not to be able to transmit arbitrary data

Manchester Encoding

Idea: At each bit, provide indication to receiver that this is where a bit starts/stops/has its middle

- For a 0 bit, have symbol change in middle of bit from low to high
- For a 1 bit, have the symbol change in middle of bit from high to low



The signal is self-clocking since one transition per period is guaranteed

Disadvantage: bit rate is as half as high as baud rate (i.e. changes in signal values)

Achievable Data Rate with Noise

Achievable data rate is limited by noise

- More precisely: by relationship of signal strength compared to noise, i.e., Signal-to-Noise Ratio (SNR, S/N)

Shannon's formula:

$$\text{maximum data rate [bit/s]} = H \log_2 (1 + S/N)$$

S: signal strength

N: noise level

H: bandwidth in Hz

S/N commonly expressed in dB:

- S/N [dB] = $10 \times \log_{10}(S/N)$

This theorem formed the basis for information theory