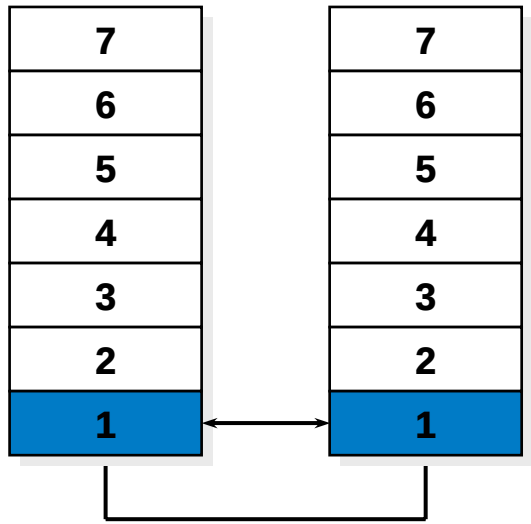
TI III: Operating & Communication Systems

Host-to-Network I

Physical Layer

Media, Signals

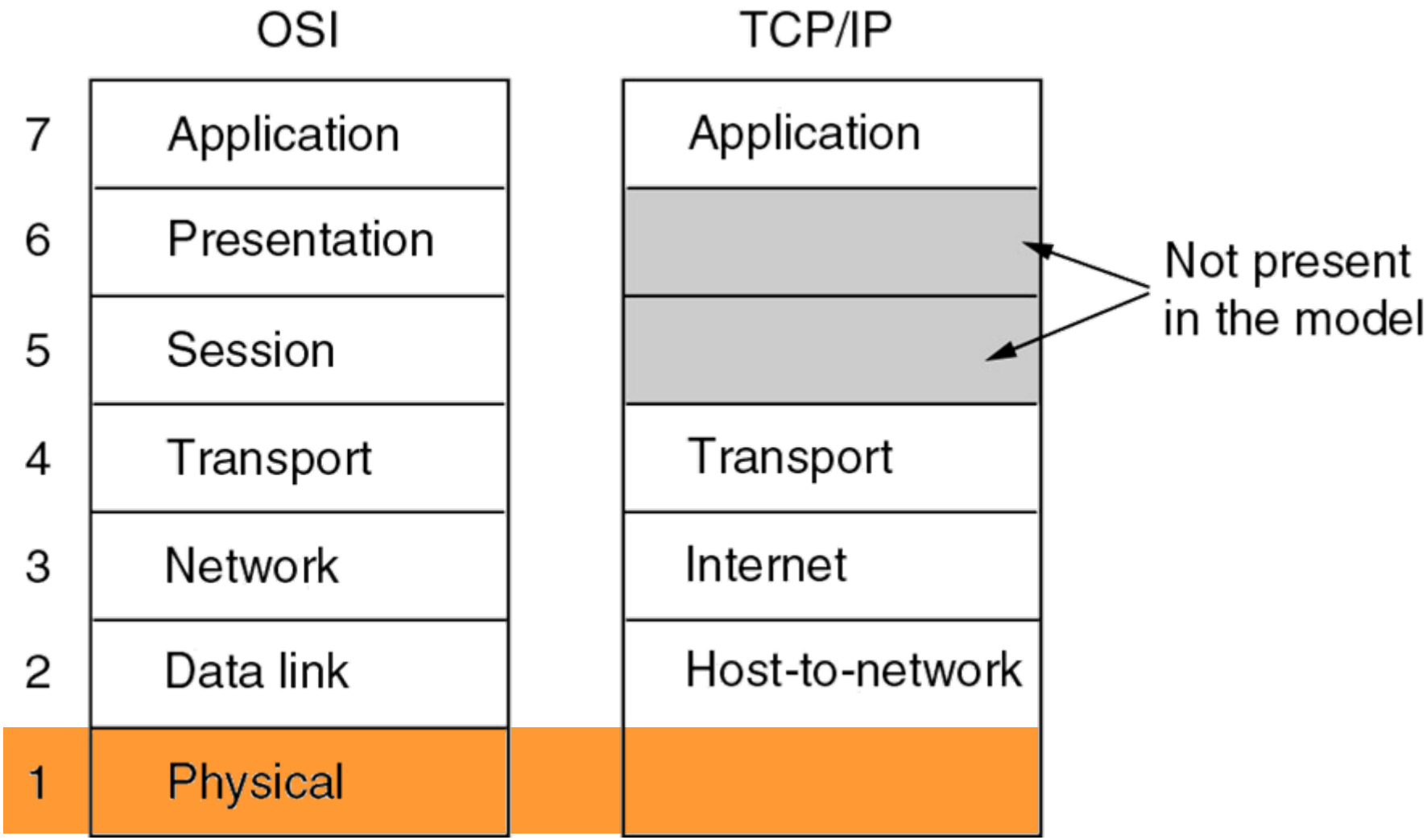
Modems



Content (2)

8. Networked Computer & the Internet
   - Sockets
   - Internet
   - Layers, Protocols
9. **Host-to-Network I**
   - **Physical Layer**
   - **Media, Signals**
   - **Modems**
10. Host-to-Network II
11. Host-to-Network III
12. Internet Protocol
13. Transport Protocols
14. Application Support

Physical Layer

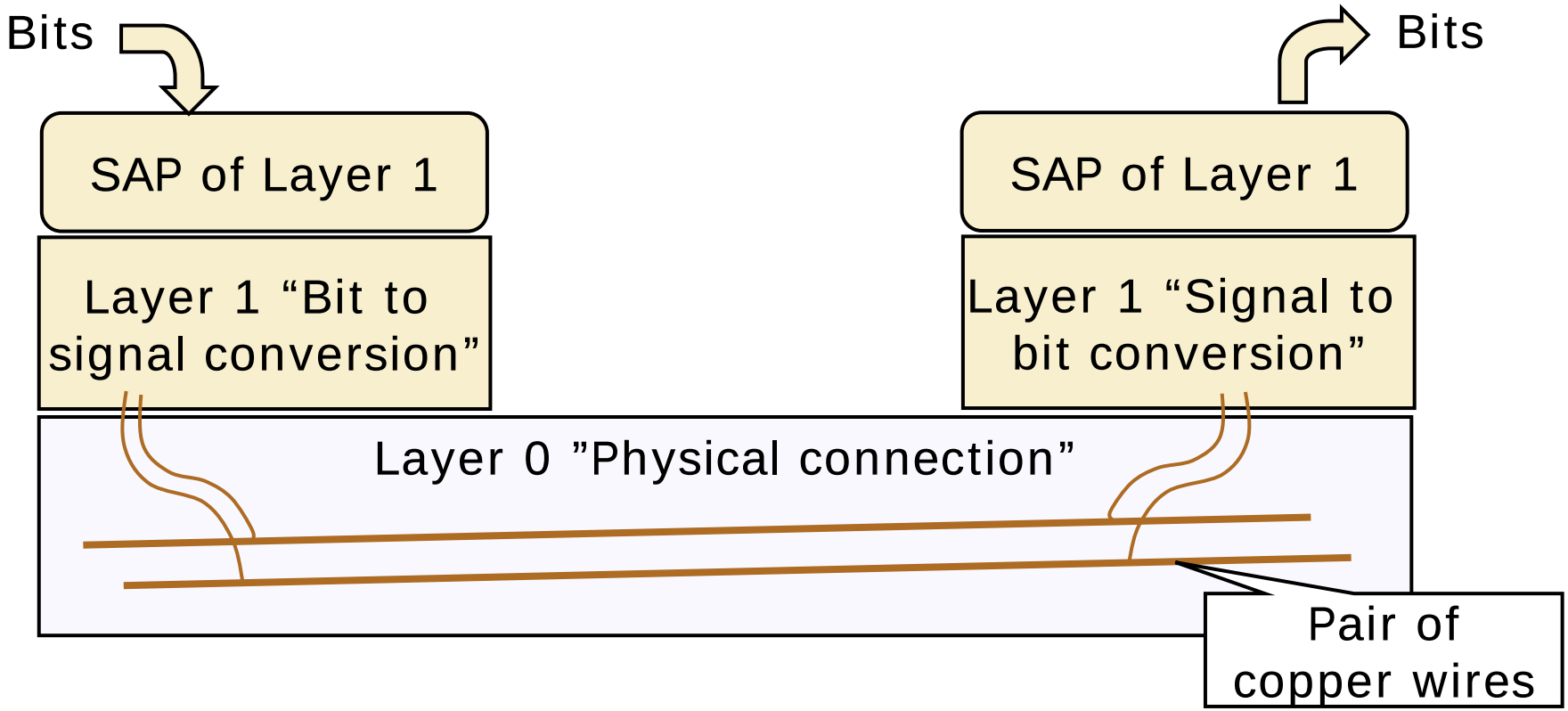


Tasks of the Physical Layer

- Responsible for turning a logical sequence of bits into a physical signal that can propagate through space
  - Many different forms of physical signals are possible
  - Signals are limited by their propagation in a physical medium (limited bandwidth, attenuation, dispersion) and by noise
- Bits can be combined into multi-valued symbols for transmission
  - Gives rise to the difference in data rate and baud rate
- Baseband transmission is fraught with problems, partially overcome by modulating a signal onto a carrier (broadband transmission)
- Includes connectors, media types, voltages…
- No error correction!

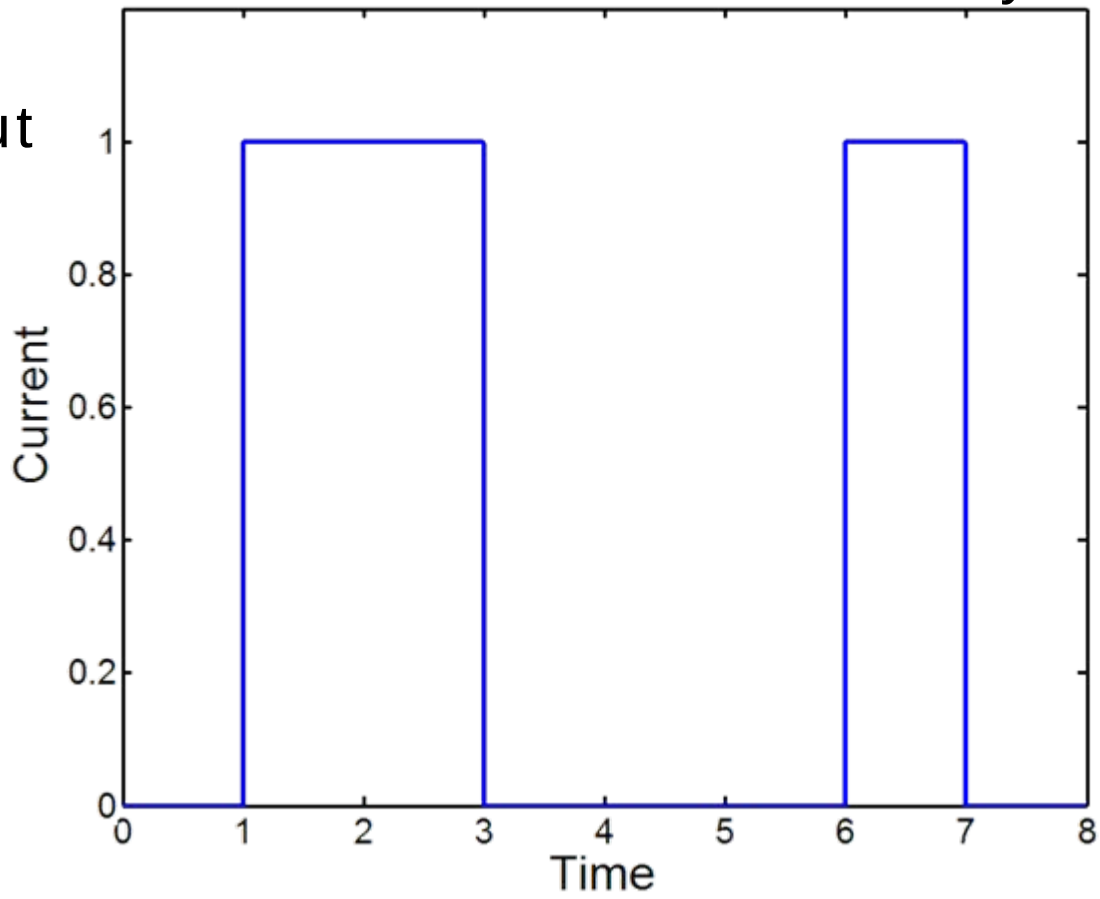
Basic service of physical layer: transport bits

- Physical layer should enable the transport of bits between two locations A and B
- Abstraction: Bit sequence – in order delivery
  - but maybe wrong bits



Example: Transmit bit pattern for character “b”

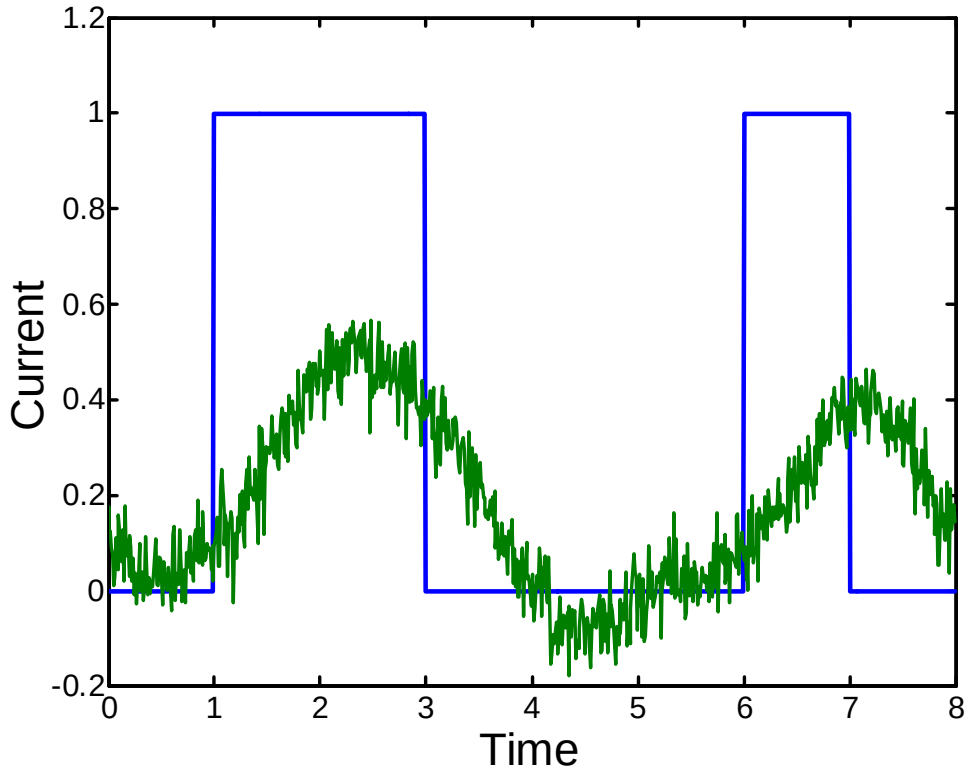
- Character “b” needs a representation as a sequence of bits
- One option: Use the ASCII code of “b”, 98, as a binary number 01100010
- Resulting current put on the wire:



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

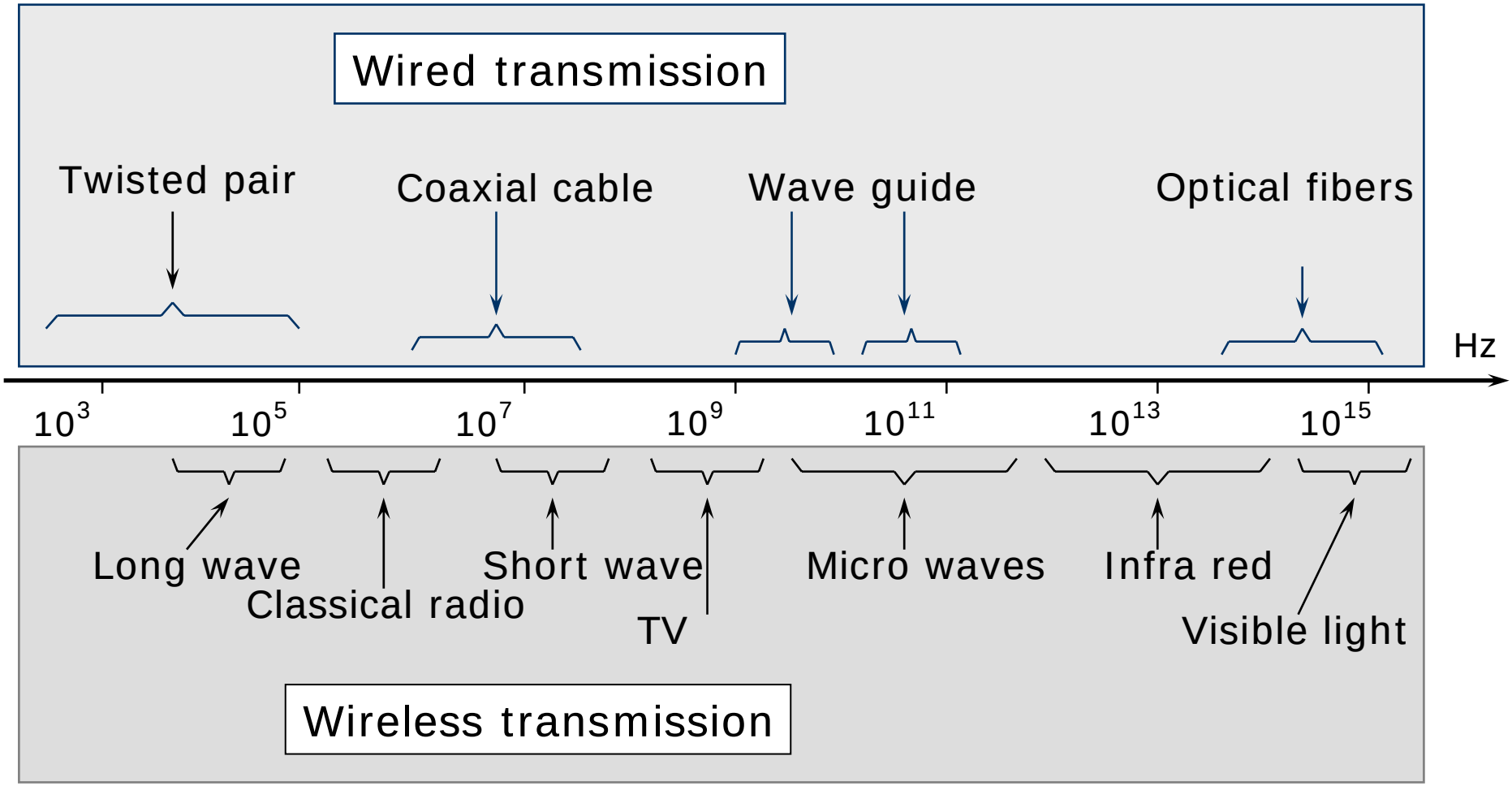
What arrives at the receiver?

- Typical pattern at the receiver:



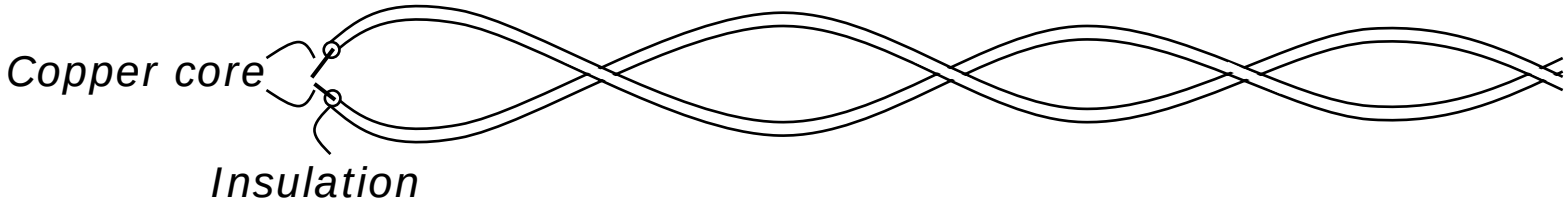
- What is going on here and how should we convert the signal back to a “b”?

Electromagnetic spectrum

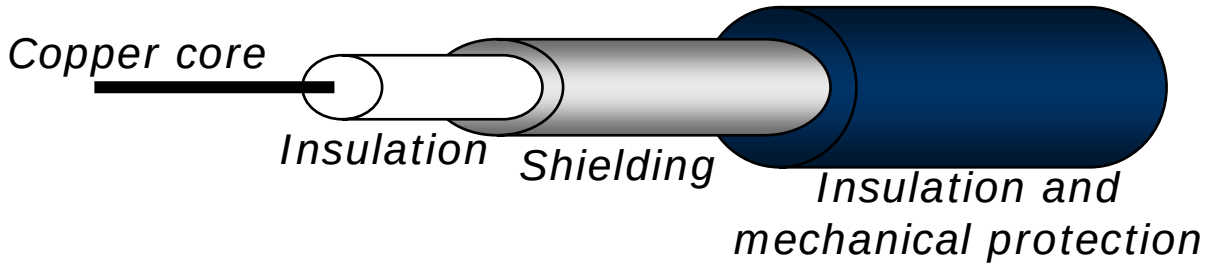


Wires

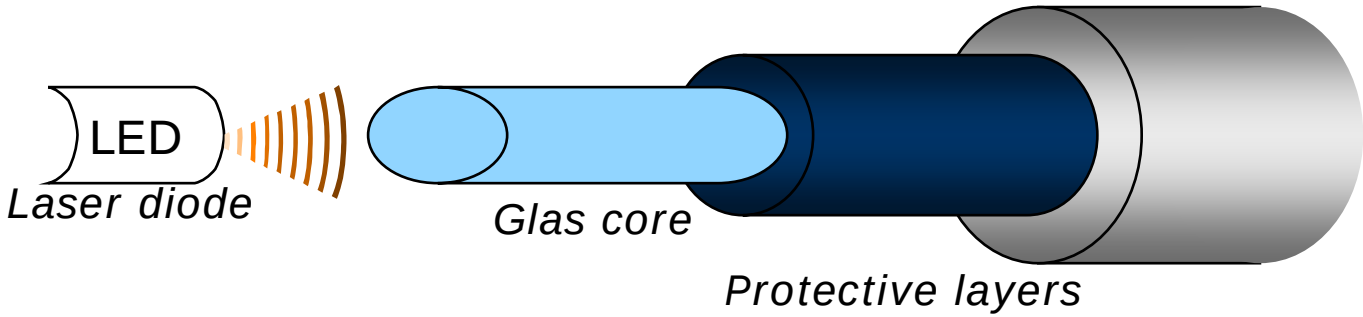
Twisted pair



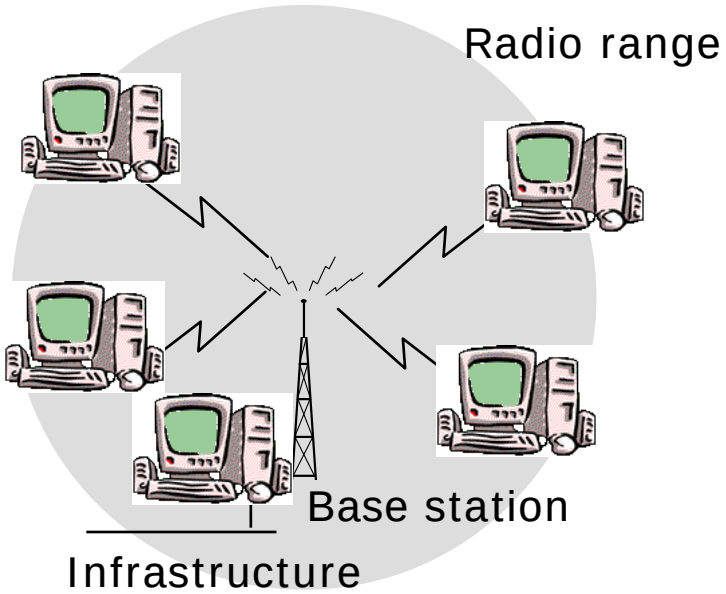
Coaxial

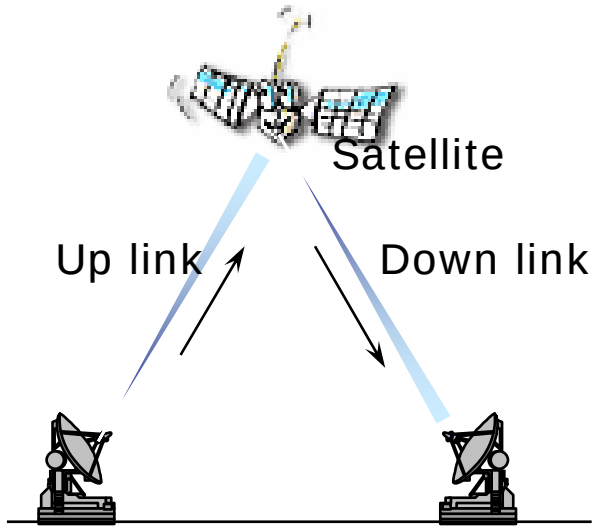


Optical fiber



Wireless transmission



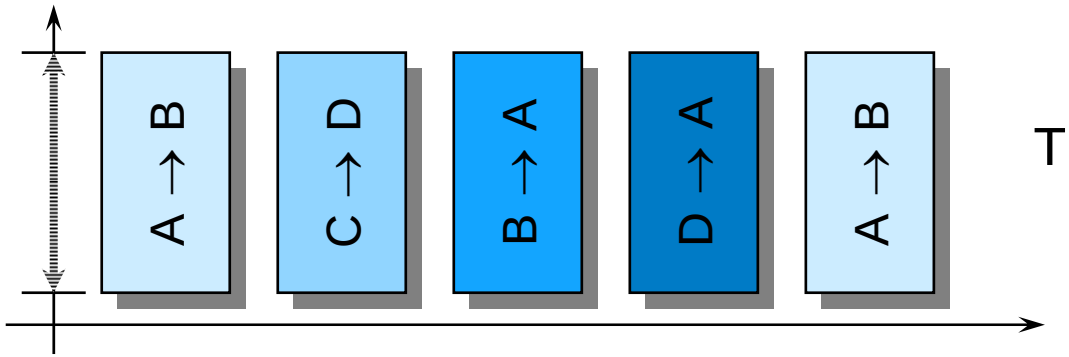


Multiplexing I

- Space division multiplexing (SDM)

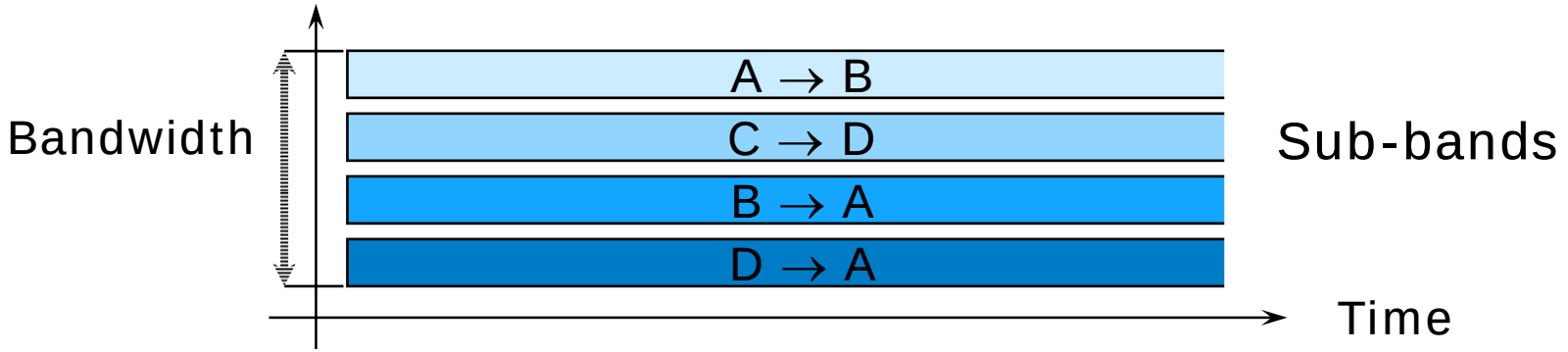
Bundling of wires

- Time division multiplexing (TDM)



Multiplexing II

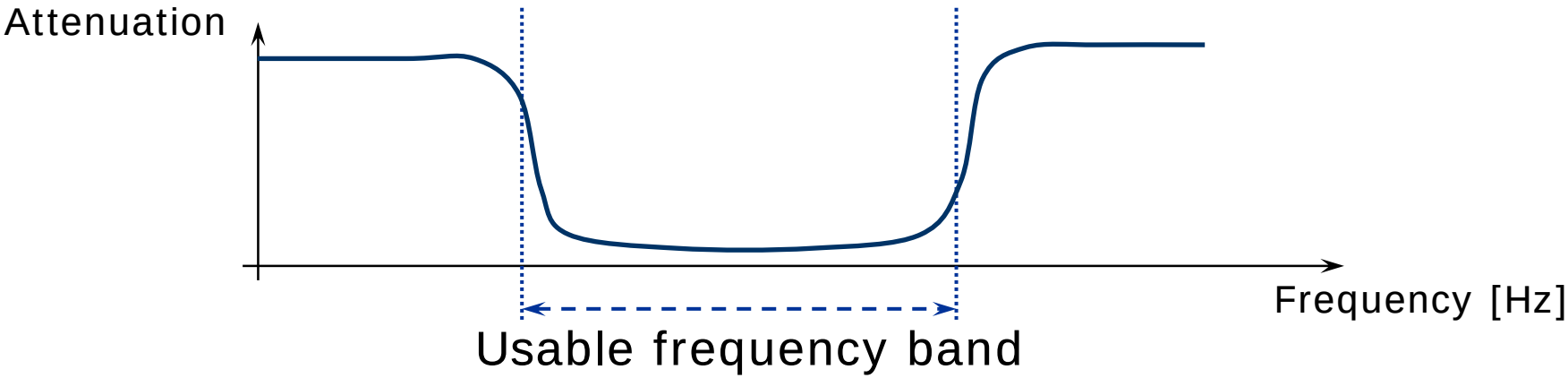
- Frequency division multiplexing (FDM)



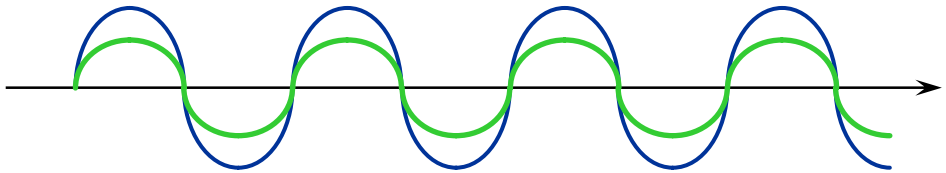
- Multiplexing in general allows for a more efficient usage of a medium

Typical effects of transmission

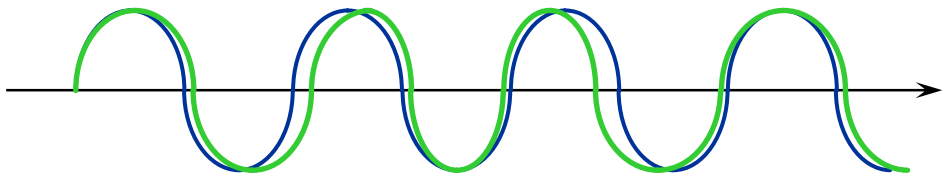
- Limited bandwidth



- Signal attenuation



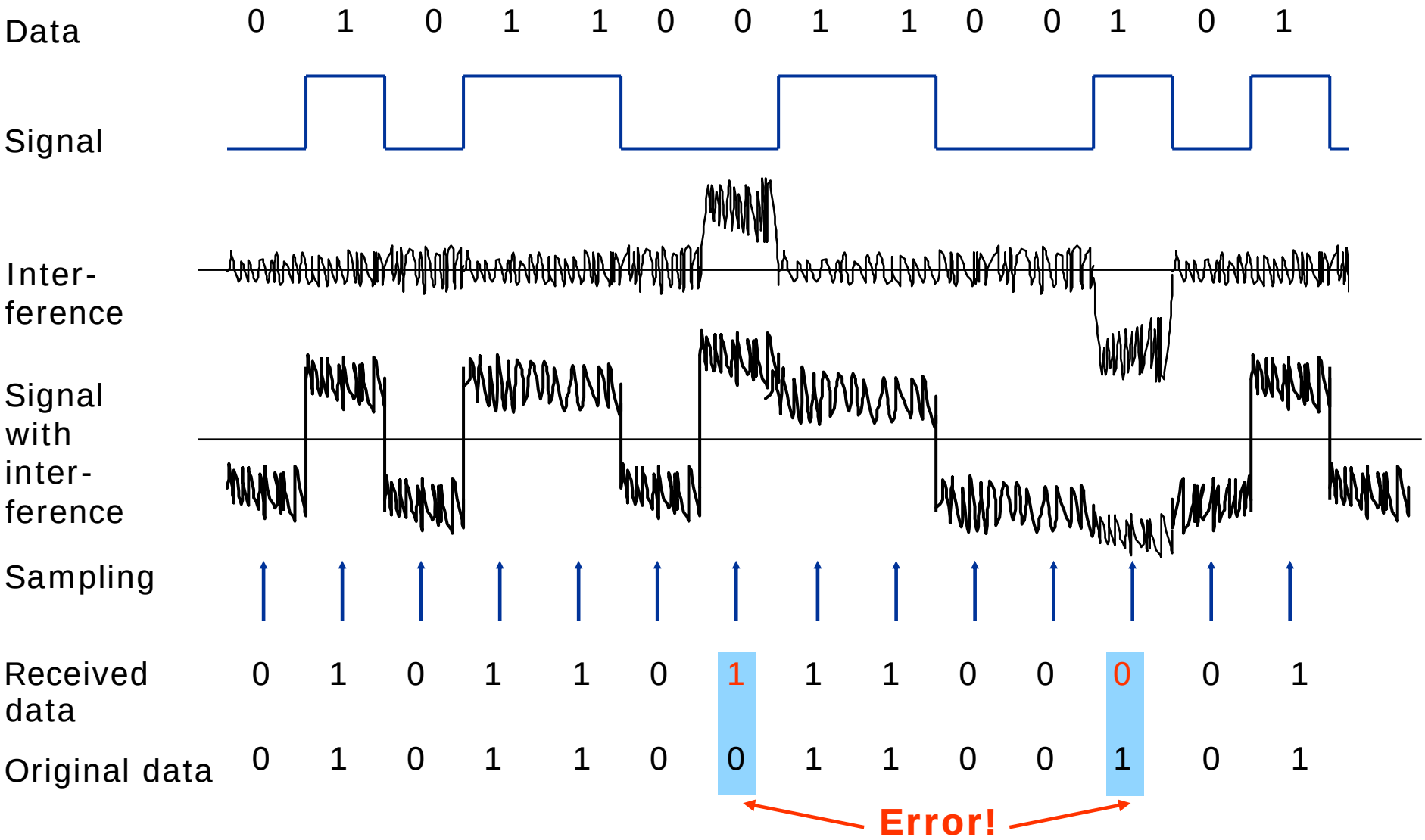
- Jitter, dispersion, ...



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)
- …

Example: Results of interference



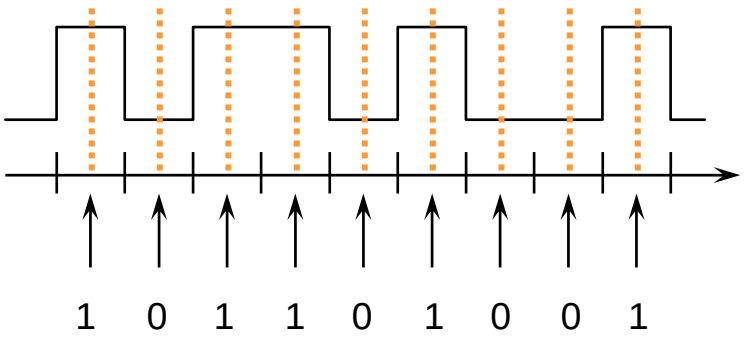
When to sample the received signal?

- 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 the ***bit*** level
  - The link layer will have to deal with frame synchronization
  - There is also “character” synchronization – omitted here

Overly simplistic bit synchronization

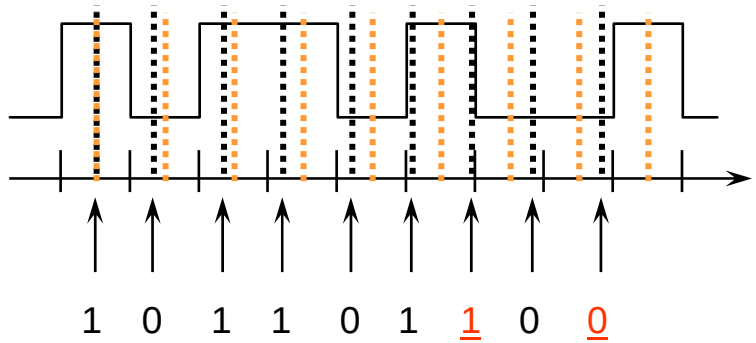
- One simple option:
  - Assume that sender and receiver at some point in time are synchronized
  - That both have an internal clock that tics at every symbol step
- Usually, this does not work
  - ***Clock drift*** is major problem – two different clocks never stay in perfect synchrony
- Errors if synchronization is lost:

Sender:



Channel

Receiver with a slightly faster clock:



Options to tell the receiver when to sample

Freie Universität Berlin

- Relying on clock synchronization does not work

- Provide an explicit clock signal
  - Needs parallel transmission over some additional channel
  - Must be in synch with the actual data, otherwise pointless

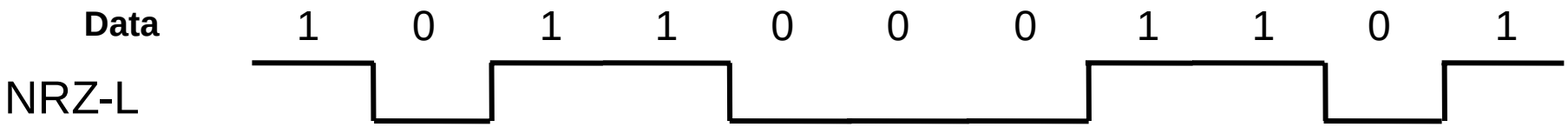
  → Useful only for short-range communication

- Synchronize the receiver at crucial points (e.g., start of a character or of a block)
  - Otherwise, let the receiver clock run freely
  - Relies on short-term stability of clock generators (do not diverge too quickly)

- Extract clock information from the received signal itself
  - Treated next in more detail

Extract clock information from signal itself

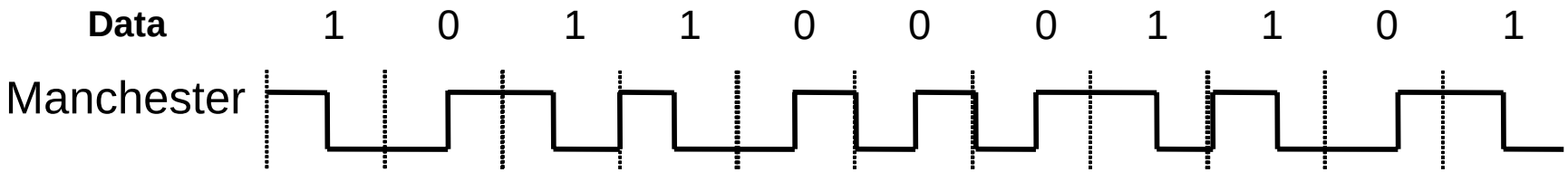
- Put enough information into the data signal itself so that the receiver can know immediately when a bit starts/stops
- Would the simple 0 → low, 1 → high mapping of bit→ symbol work?
- It should – after all, receiver can use 0-1-0 transitions to detect the length of a bit



- But it fails depending on bit sequences: think of long runs of 1s or 0s – receiver can loose synchronization
- Not nice not to be able to transmit arbitrary data

Extract clock information from signal itself – Manchester

- Idea: At each bit, provide indication to receiver that this is where a bit {starts/stops/has its middle}
  - Example: Manchester encoding
  - For a 0 bit, have the symbol change in the bit middle from low to high
  - For a 1 bit, have the symbol change in the bit middle from high to low



- Ensures sufficient number of signal transitions
  - Independent of what data is transmitted!

Baseband versus broadband transmission

Freie Universität Berlin

- The transmission schemes described so far: ***baseband transmission***
  - Baseband transmission directly puts the digital symbol sequences onto the wire
  - At different levels of current, voltage, …
- Baseband transmission suffers from
  - Limited bandwidth reshapes the signal at receiver
  - Attenuation and distortion depend on frequency and baseband transmissions have many different frequencies because of their wide Fourier spectrum
    - Rectangular signal causes “infinite” spectrum!
- Possible alternative: ***broadband transmission***
  - needed for wireless transmission (antennas) and frequency division multiplex

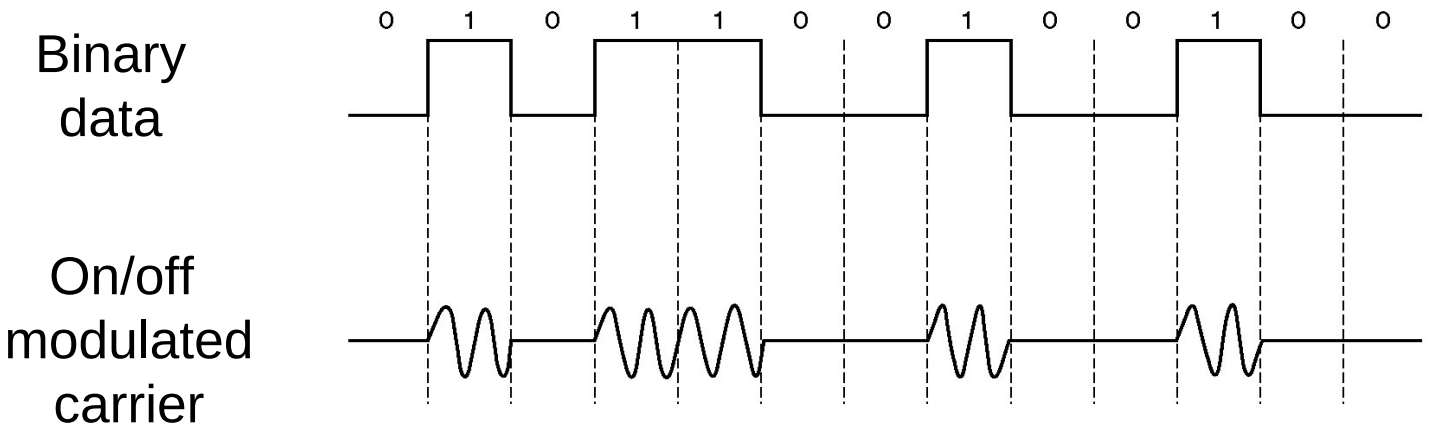
Broadband transmission

- Idea: get rid of the wide spectrum needed for baseband transmission
- Use a ***sine wave*** as a carrier for the symbols to be transmitted
  - Typically, the sine wave has high frequency
  - But only a *single* frequency!
- Pure sine waves has no information, so its shape has to be influenced according to the symbols to be transmitted
  - The carrier has to be ***modulated*** by the symbols (widening the spectrum)
- Three parameters that can be influenced
  - Amplitude a
  - Frequency f
  - Phase ϕ

$$a \sin(2\pi f t + \phi)$$

Amplitude modulation

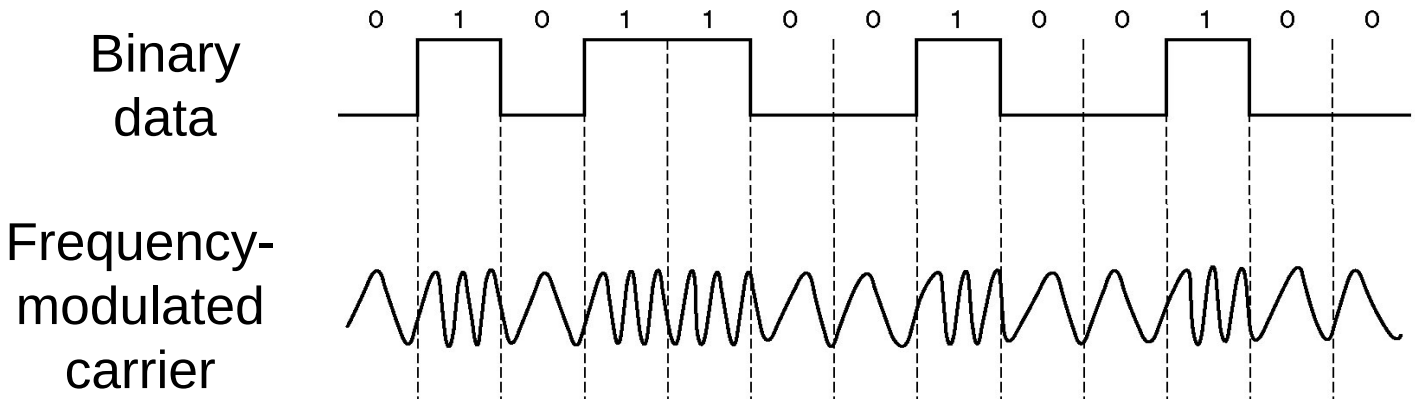
- The amplitude modulated sine wave $f_A(t)$ is given as
  - $f_A(t) = s(t) \sin(2\pi ft + \phi)$
  - The amplitude is given by the signal $s(t)$ to be transmitted
- Special cases:
  - s(t) is an ***analog*** signal – ***amplitude modulation***
  - s(t) is a ***digital*** signal – also called ***amplitude keying***
  - s(t) only takes 0 and *a* as values – ***on/off keying***



Frequency modulation

- The frequency modulated sine wave $f_F(t)$ is given by
  - $f_F(t) = a \sin(2\pi s(t)t + \phi)$
  - Modulation/keying terminology like for AM

- Example



Note: s(t) has an additive constant in this example to avoid having frequency zero

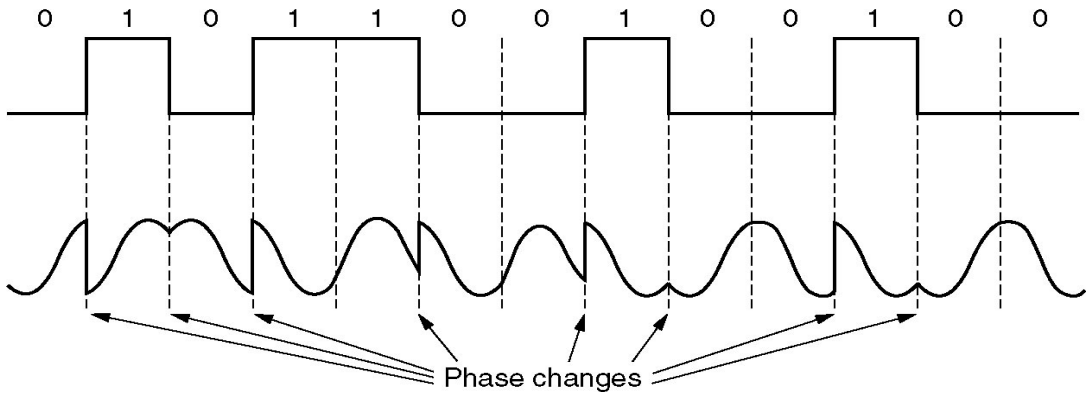
Phase modulation

- Similarly, a phase modulated carrier is given by
  - $f_P(t) = a \sin(2\pi ft + s(t))$
  - Modulation/keying terminology again similar

- Example:

Binary data

Phase-modulated carrier



- s(t) is chosen such that there are phase changes when the binary data changes
  - Typical example for ***differential coding***

Use more than just 0 and 1 in the channel

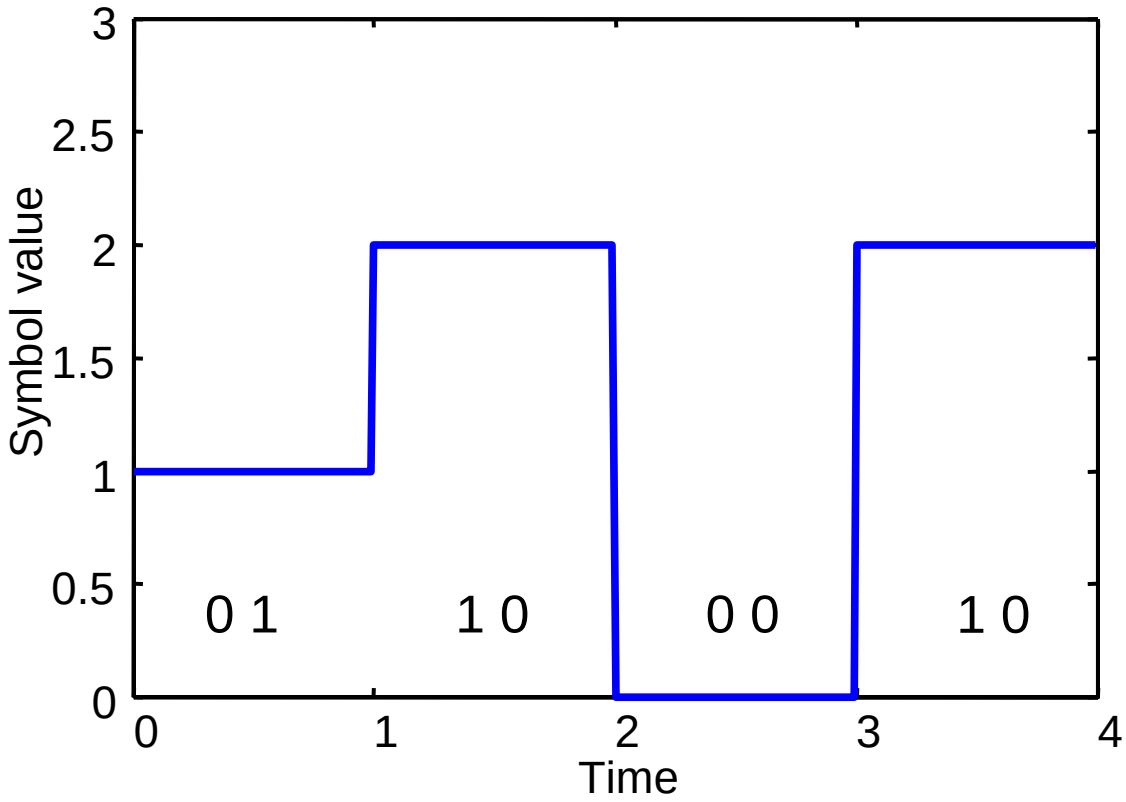
- Who says we can only use 0 and 1 as possible levels for the transmitted signal?
- Suppose the transmitter can generate signals (current, voltage, …) at *four different levels*, instead of just two
  - Then: to select one of four levels, *two bits* are required

- Distinction
  - “Bits” are 0 or 1, used in “higher” layers
  - “Symbols” can have multiple values, are transmitted over the channel
  - *Symbol rate*: Rate with which symbols are transmitted
    - Measured in baud
  - *Data rate*: Rate with which physical layer processes incoming data bits
    - Measured in bit/s

Example: Use four-level symbols to encode two bits

- Example:
  - Map 00 ➔ 0, 01 ➔ 1, 10 ➔ 2, 11 ➔ 3
  - Symbol rate is then only half the data rate as each symbol encodes two bits

- Today's systems:
  - many bits per symbol
  - often Phase Shift Keying/Amplitude Shift Keying combined



Data rate with multi-valued symbols – Nyquist

- Using symbols with multiple values, the data rate can be increased

- ***Nyquist formula*** summarizes:

$$\text{maximum data rate} \leq 2H \log_2 V \text{ bit/s}$$

where V is the number of discrete symbol values and H the available bandwidth in Hz

Unlimited data rate with many symbol levels?

- Nyquist's theorem appears to indicate that unlimited data rate can be achieved when only enough symbol levels are used
- Is this plausible?

- More and more symbol levels have to be spaced closer and closer together
- What then about noise?
  - Even small random noise would then result in one symbol being misinterpreted for another

- So not unlimited?

Shannon limit on achievable data rate

- Achievable data rate is fundamentally limited by noise
  - More precisely: by the relationship of signal strength compared to noise
  - The relatively fewer noise there is at the receiver, the easier it is for the receiver to distinguish between different symbol levels
- Relationship characterized by ***Shannon, 1948***

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

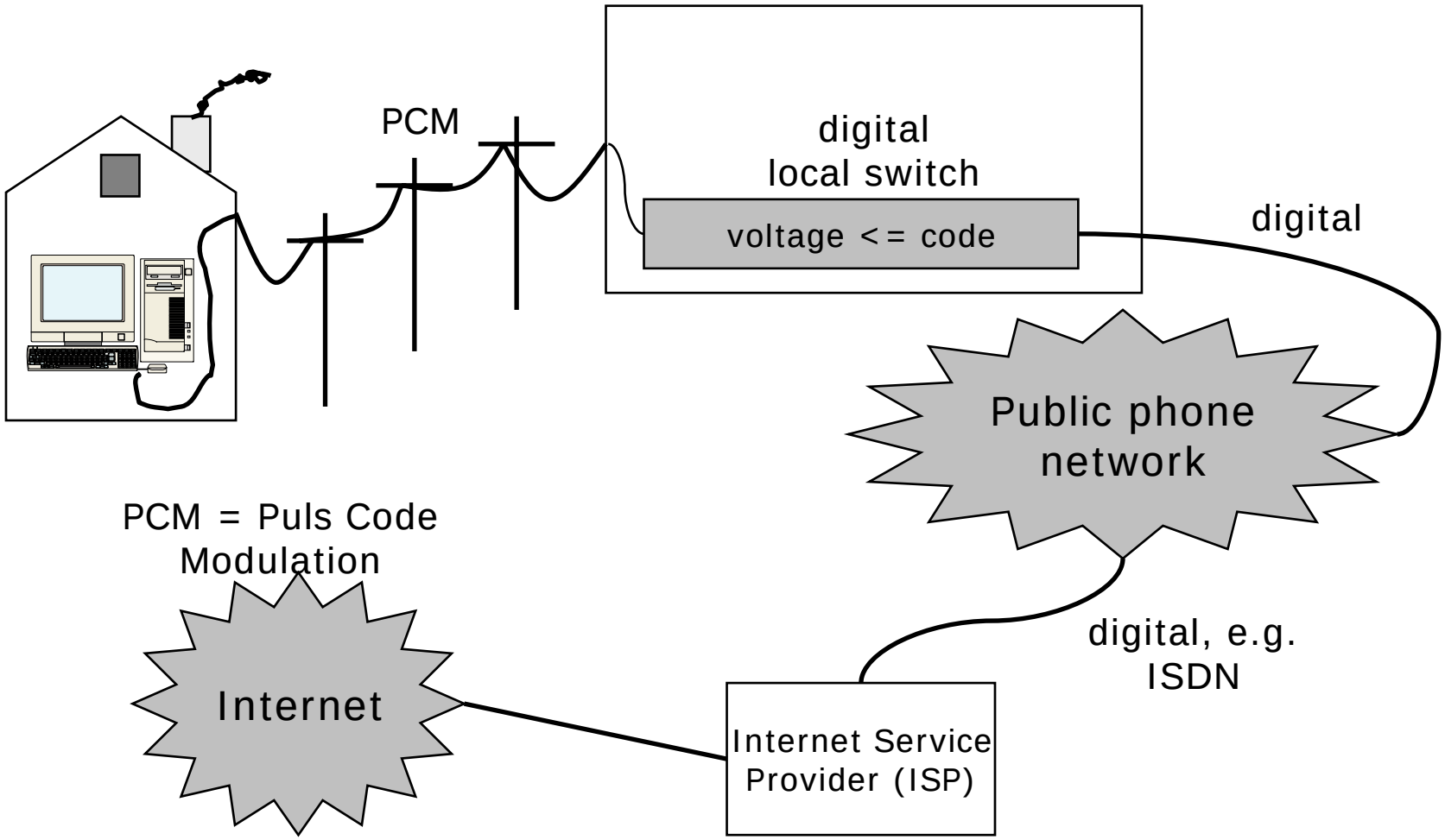
where S is signal strength, N is noise level and H the available bandwidth in Hz

- This theorem formed the basis for ***information theory***

Classical modems for analog subscriber lines: V.90/V.92

- Idea
  - Modulation/Demodulation of signals depending on data
  - Adaptation to medium characteristics
- Key parameters
  - Data rates up to 56 kbit/s downstream (provider to user)
  - Upstream 33.6 kbit/s (based on V.34)
    - V.92: 48 kbit/s upstream plus data compression (V.44)
- Technology
  - Digital infrastructure required, only last mile analog
  - Digital transmission from provider to last switch
  - PCM signals from switch to modem
  - Data rates depend much on line quality
    - Line probing required plus adaptation

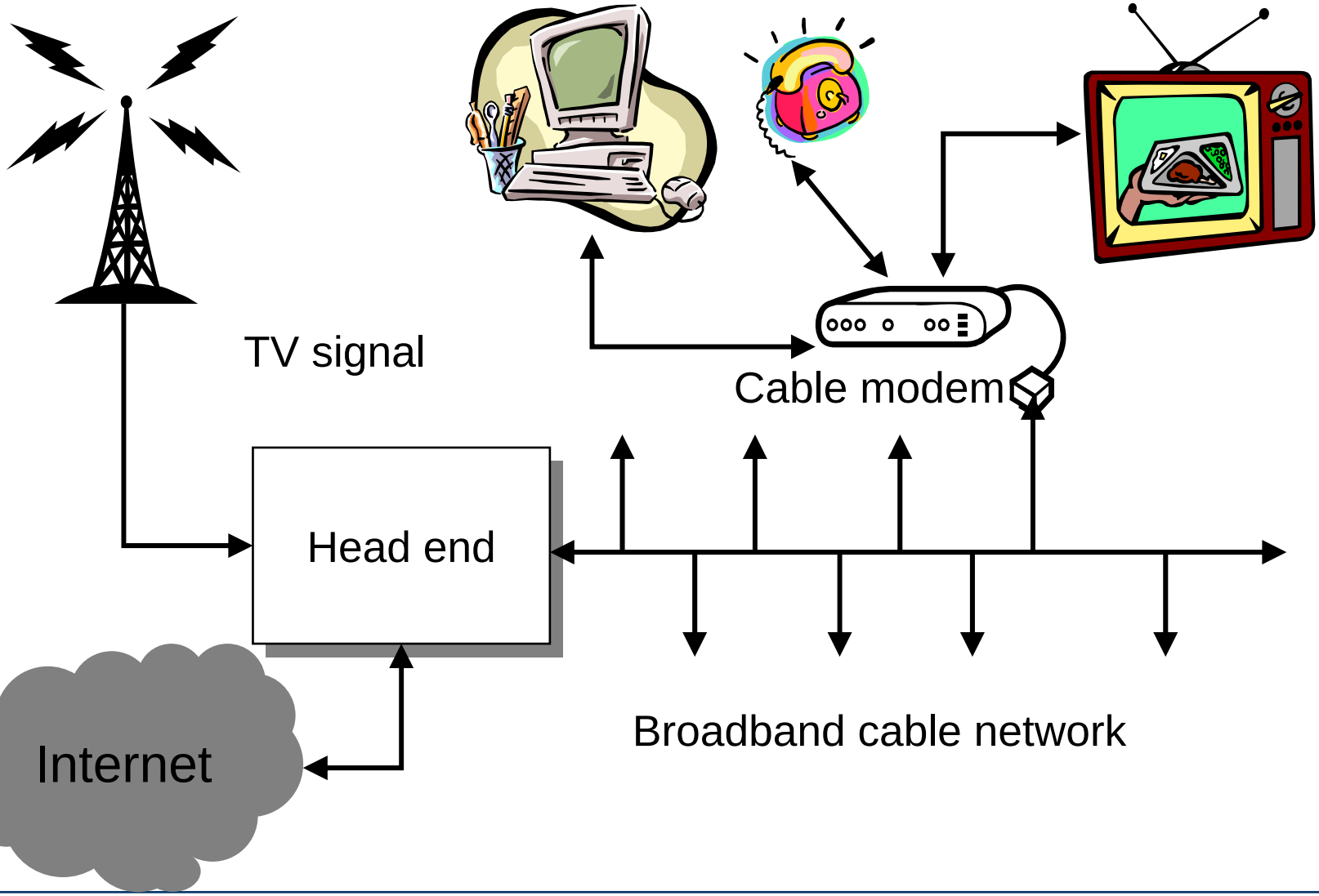
V.90 basic architecture



Newer modem technologies

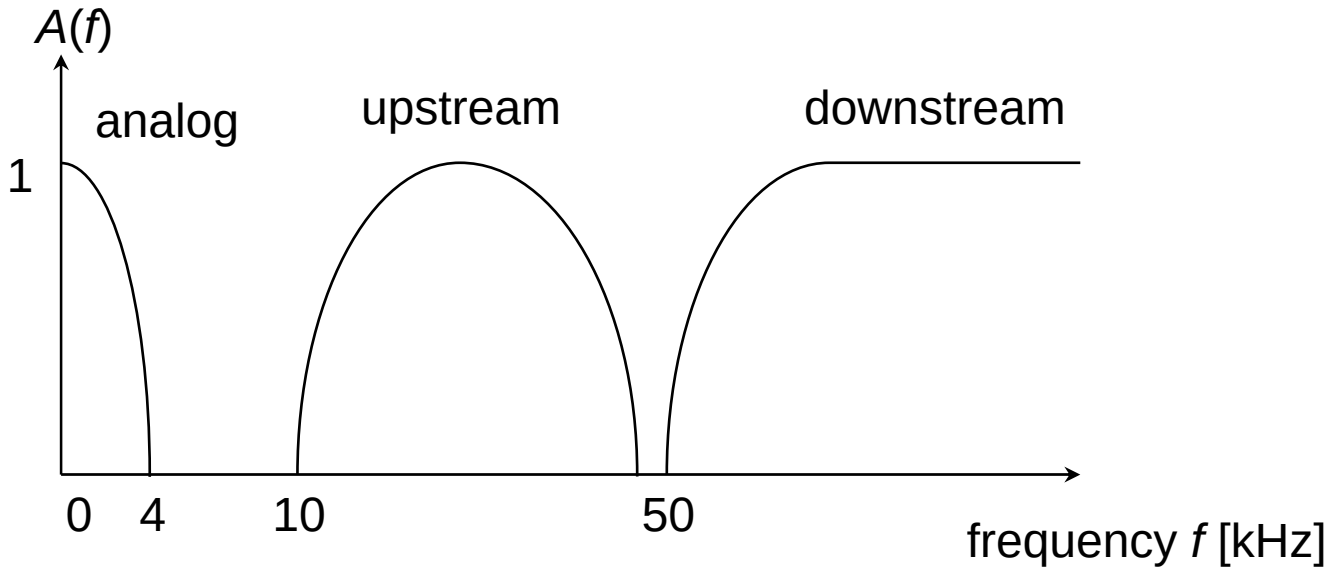
- Cable modem
  - Data transmission via broadband cable (TV)
  - Infrastructure must be bi-directional
  - Data rates in the Gbit/s range but always shared medium
- Powerline communications
  - Data transmission via power lines
  - Couple high-frequency signals into standard power lines
  - Data rates up to several Mbit/s but shared medium
- xDSL modems
  - Higher data rates using conventional phone lines
  - Typical data rates up to 16 Mbit/s downstream, up to 1 Mbit/s upstream (ADSL)
  - Not everywhere, rates depend on location, interference ...
  - Special technologies for faster response („FastPath“)
  - Up to 50 Mbit/s at certain places (VDSL)
  - Symmetrical for companies/servers

Broadband cable: digital TV plus Internet

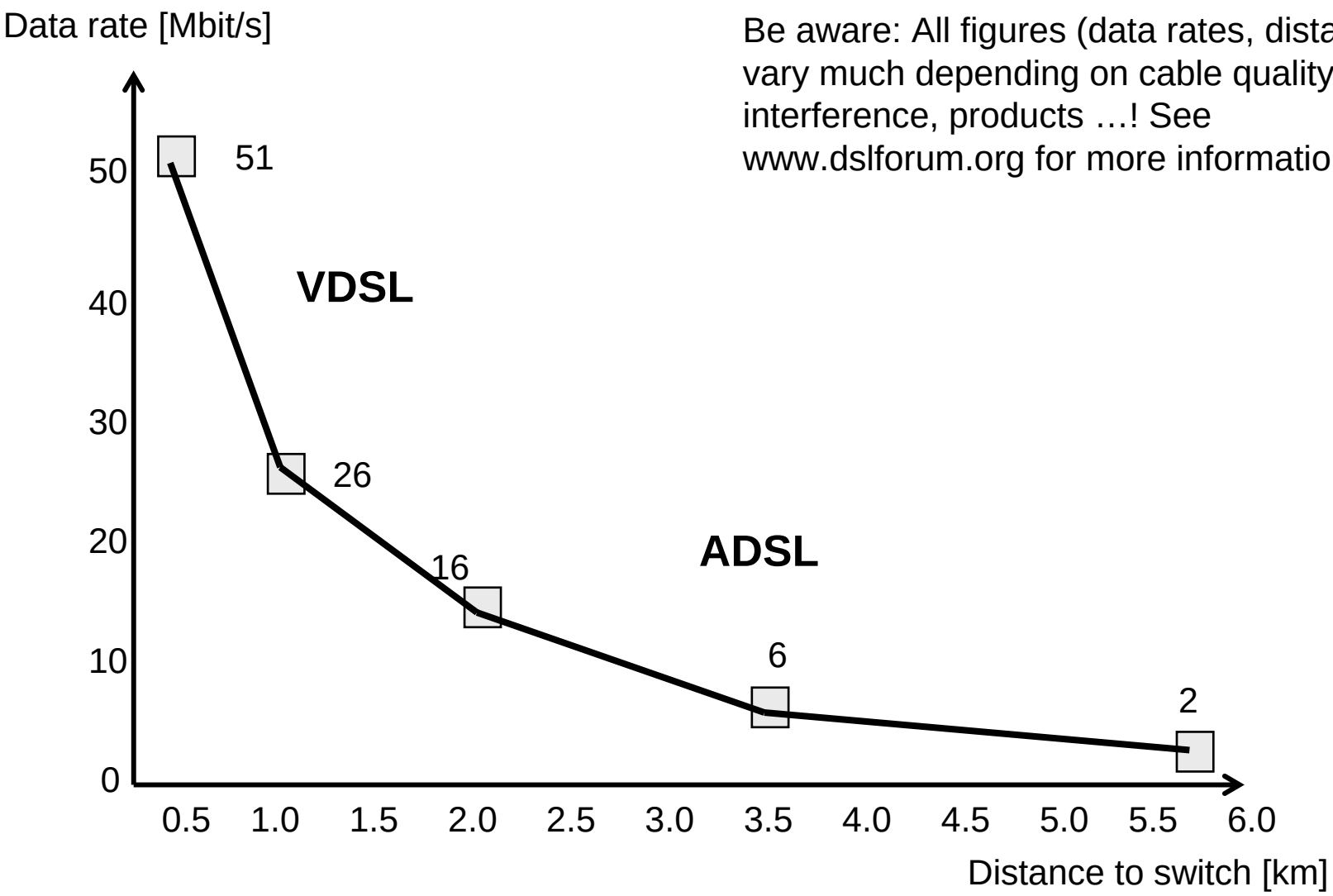


xDSL

- xDSL: different DSL (Digital Subscriber Line) technologies
- Goal: use already existing phone lines (simple twisted pair, unshielded) for higher data rates
  - ISDN: replacement of the analog phone system by a fully digital system (offering n times 64 kbit/s)
- Co-existence of classical analog (or digital ISDN) phone system plus high data rates



Typical downstream data rates



Be aware: All figures (data rates, distances) vary much depending on cable quality, interference, products …! See www.dslforum.org for more information.