Computer Networks

Exercise Session 03

Prof. Dr. Oliver Hahm

Frankfurt University of Applied Sciences
Faculty 2: Computer Science and Engineering
oliver.hahm@fb2.fra-uas.de
https://teaching.dahahm.de

November 11, 2022

General Schedule

All exercises will follow this general schedule

- Identify potential understanding problems
  $\rightarrow$ Ask your questions
  $\rightarrow$ Recap of the lecture
- Address the understanding problems
  $\rightarrow$ Answer your questions
  $\rightarrow$ Repeat certain topics
- Walk through the exercises/solutions $\rightarrow$ Some hints and guidance
  $\rightarrow$ Work time or presentation of results

Fundamentals of Data Signals

You have seen . . .

- how an analog signal can be transformed into a digital signal (and vice versa) using quantization and sampling
- how often a channel needs to be sampled to reconstruct the original analog signal
- how a square wave signal can be constructed by a fundamental frequency and its harmonics
- the difference between bandwidth, data rate, and symbol rate
- what data date can be achieved on a noiseless and a noisy channel with finite bandwidth

Data Encoding

You have seen . . .

- what a baseband transmission is
- which requirements exist for a good encoding (robustness, efficiency, and clock recovery)
- several line codes and how they relate to these requirements
- what the problems of baseline wander and clock recovery are and how to tackle them
- how an encoding of group of bits in combination of another encoding can be used to address all requirements → e.g., 4B/5B

Modulation

You have seen . . .

- how data can be modulated onto a carrier frequency in broadband
- what amplitude, frequency, and phase modulation are
- which advantages and drawbacks these methods have

Transmission Media

You have seen . . .

- which categories of transmission media exist
- common types of guided transmission media (coaxial, twisted pair, and fiber optic)
- what the common challenges of wireless networks are
- how the last mile can be bridged in a cost-efficient manner

Technologies

You have seen ...

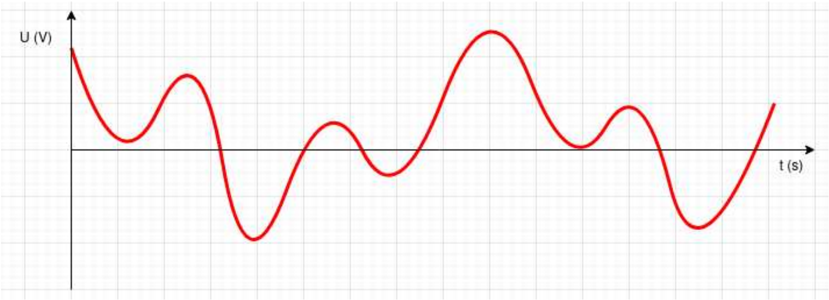
- how Ethernet has evolved to become the most popular wired LAN technology
- what Token Ring was and why it became obsolete
- which types of WLAN exist and how they differ
- what Bluetooth, piconets, scatternets, and BLE are

Any other questions left?

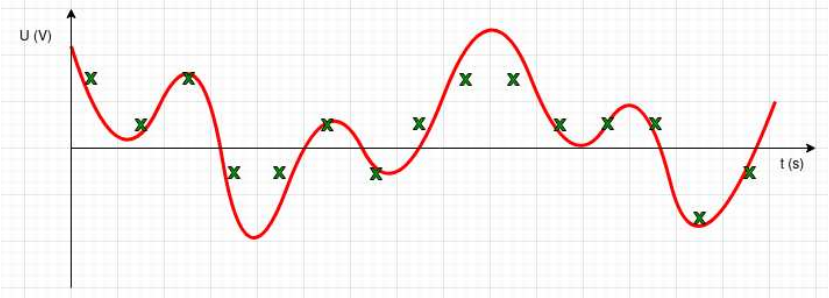
Exercise 1: Layers of Reference Models

- Protocol example for the session layer
  - Point-to-Point Tunneling Protocol (PPTP) was used for Virtual Private Networks (VPNs)
  - Encapsulate layer 2 frames into a TCP control channel
  - Layer 3 protocols like IP can be transported over PPTP
  - Password Authentication Protocol (PAP) can be used for password-based authentication
- Protocol example for the presentation layer
  - External Data Representation (XDR) is a data serialization format
  - It allows for de- and encoding between different representations of data types
  - Supported data types comprise: `boolean`, `int`, `float`, `enumerations` ...
  - An example can be found here: `https://github.com/brendanhay/xdr/blob/master/example.xdr`

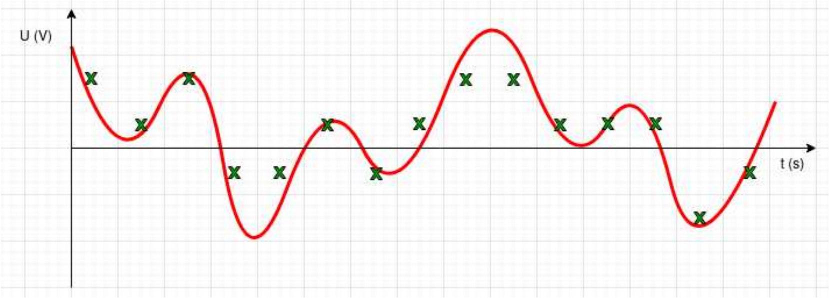
Exercise 2: Quantization and Sampling



Exercise 2: Quantization and Sampling



Exercise 2: Quantization and Sampling



- Until the 1980s the whole telephone system was voice only

⇒ the lowest frequency was 300 Hz, the highest frequency was 3.4 kHz

Exercise 3 and 4: Bit, Symbol and Data Rate

Remember the differences between bit rate and symbol rate:

Bit rate

- The number of bits that can be transmitted per unit of time
- Typically expressed as bit/s, or bps in conjunction with a SI prefix
- The physical layer defines the gross bitrate
- It is also used in digital multimedia to represent the number of bits used to encode audio or videos

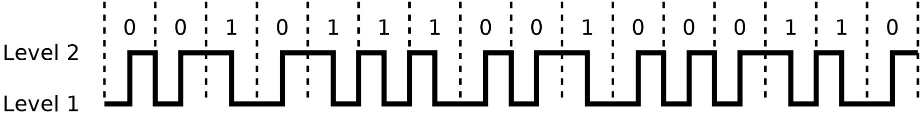
Baud rate

- Number of symbol or waveform changes per unit of time
- Typically expressed as baud (Bd)
- The symbol duration time or unit interval is T_s
  $T_s = \frac{1}{f_s}$
  where f_s is the symbol rate

The bit rate depends on the bandwidth of the communication channel and the number of bits per symbol

Exercise 5 and 6: Manchester II Encoding

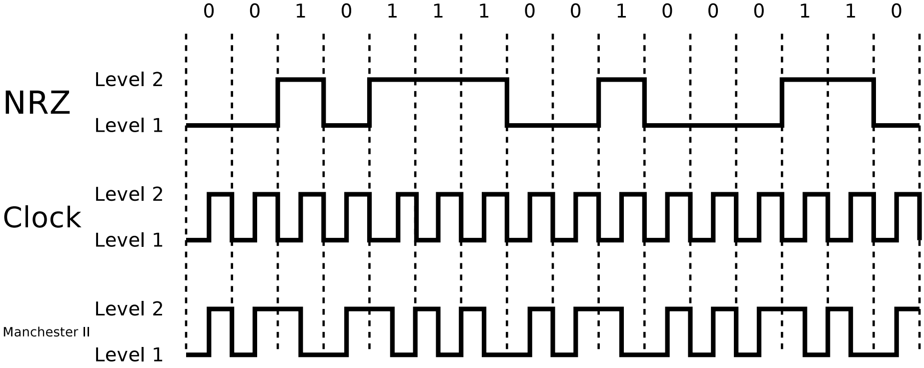
- This line code (also called Biphase-L is the opposite of the Manchester encoding
  - Manchester encoding:
    - Transition from high to low signal corresponds to a logical 0 bit
    - Transition from low to high signal corresponds to a logical 1 bit
  - Manchester II encoding:
    - Transition from low to high signal corresponds to a logical 0 bit
    - Transition from high to low signal corresponds to a logical 1 bit
- Just as for the Manchester encoding, clock recovery is possible for the receiver and baseline wander cannot occur because the usage of the signal levels is distributed equally



Manchester II Code

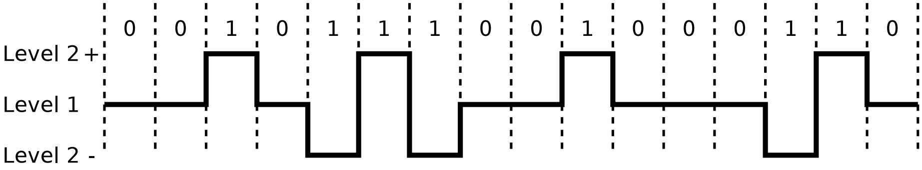
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

- The Manchester II encoding is calculated via exclusive or (XOR) of the NRZ encoded data and the clock



Alternate Mark Inversion (AMI code) = Bipolar Encoding

- Uses 3 signal levels (+, 0 und –)
  - Logical 0 bits are encoded as middle signal level (0)
  - Logical 1 bits are alternating encoded as high (+) or low signal level (–)
- Benefit: Baseline wander cannot occur
- Drawback: Clock recovery is impossible for series of logical 0 bits
- Error detection is partly possible because the signal sequences ++, ––, +0+ and –0– are illegal



AMI Line Code in Practice and Scramblers

The ISDN S_0 bus uses a modified version of the AMI line code

- With this variant, logical 1 bits are encoded as middle signal level and logical 0 bits are alternating encoded as high signal level or low signal level

- To allow for clock recovery a scrambler is often used, after AMI line code encoding

⇒ A scrambler is a device, which modifies a bit stream according to a simple algorithm in a way, that it is simple to reverse back to the original bit stream

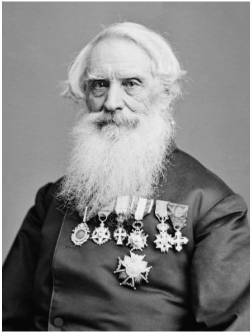
- In this case, scramblers are used, to interrupt long series of logic 0 bits

Encoding Data

Image source: Wikipedia (CC0)

- Efficient data encoding is important not only since the rise of computer networks
- An example for an efficient encoding is the Morse Code, invented by Samuel Morse from 1838

A	· —	M	— —	Y	— · — —
B	— · · ·	N	— ·	Z	— — · ·
C	— · — ·	O	— — —	1	· — — — —
D	— · ·	P	· — — ·	2	· · — — —
E	·	Q	— — · —	3	· · · — —
F	· · — ·	R	· — ·	4	· · · · —
G	— — ·	S	· · ·	5	· · · · ·
H	· · · ·	T	—	6	— · · · ·
I	· ·	U	· · —	7	— — · · ·
J	· — — —	V	· · · —	8	— — — · ·
K	— · —	W	· — —	9	— — — — ·
L	· — · ·	X	— · · —	0	— — — — —



Samuel Morse (1791 – 1872)