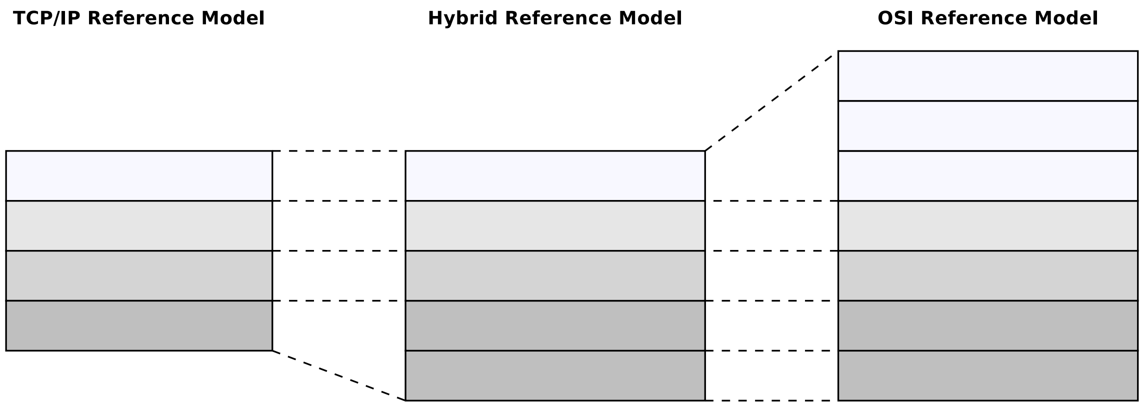
Exercise Sheet 2

Exercise 1 (Layers of Reference Models)

1. Fill in the names of the layers of the reference models in the figure.



2. Assign to technical terms *Frames*, *Packets*, *Segments* and *Signals* to the layers of the reference models in the figure.
3. Why are the Presentation Layer and the Session Layer not intensively used?
4. Why is the hybrid reference model closer to reality, compared with the TCP/IP reference model?

Exercise 2 (Quantization and Sampling)

1. Why do quantization and sampling create errors? Can we avoid these errors?
2. Taking the classical telephony example: How often should the system sample the signal?
3. What is the maximum data rate without noise? Is this realistic?

Source: Prof. Dr. Jochen Schiller, FU Berlin (2015)

Exercise 3 (Bit Rate and Symbol Rate)

The unit of bit rate is bit/s, whereas the symbol rate is given in baud.

1. How are these two units related?
2. Under which circumstances are symbol rate and bit rate equal?
3. Is it possible that the bit rate is smaller than the symbol rate?
4. Why can a symbol not carry an arbitrary amount of bits?

Source: Prof. Dr. J. Seitz, M. Aumüller, TU Ilmenau (2018)

Exercise 4 (Data Rate)

In classical telephone network the maximum data rate without digital telephone switch is limited to max. 33.6 kbit/s. A *33.6k Modem* uses the *trellis coded modulation (TCM)* to transmit the bits over the telephone line.

1. In order to reach this data rate a symbol rate of 3429 baud has been achieved. How many bits must be encoded in a single symbol?
2. Explain why the system uses a modulation scheme to transmit the data instead of line coding.
3. Calculate SNR a telephone line has to provide in order to achieve this data rate.
4. Calculate the signal strength if there exists noise of 0.1 kW on the line.

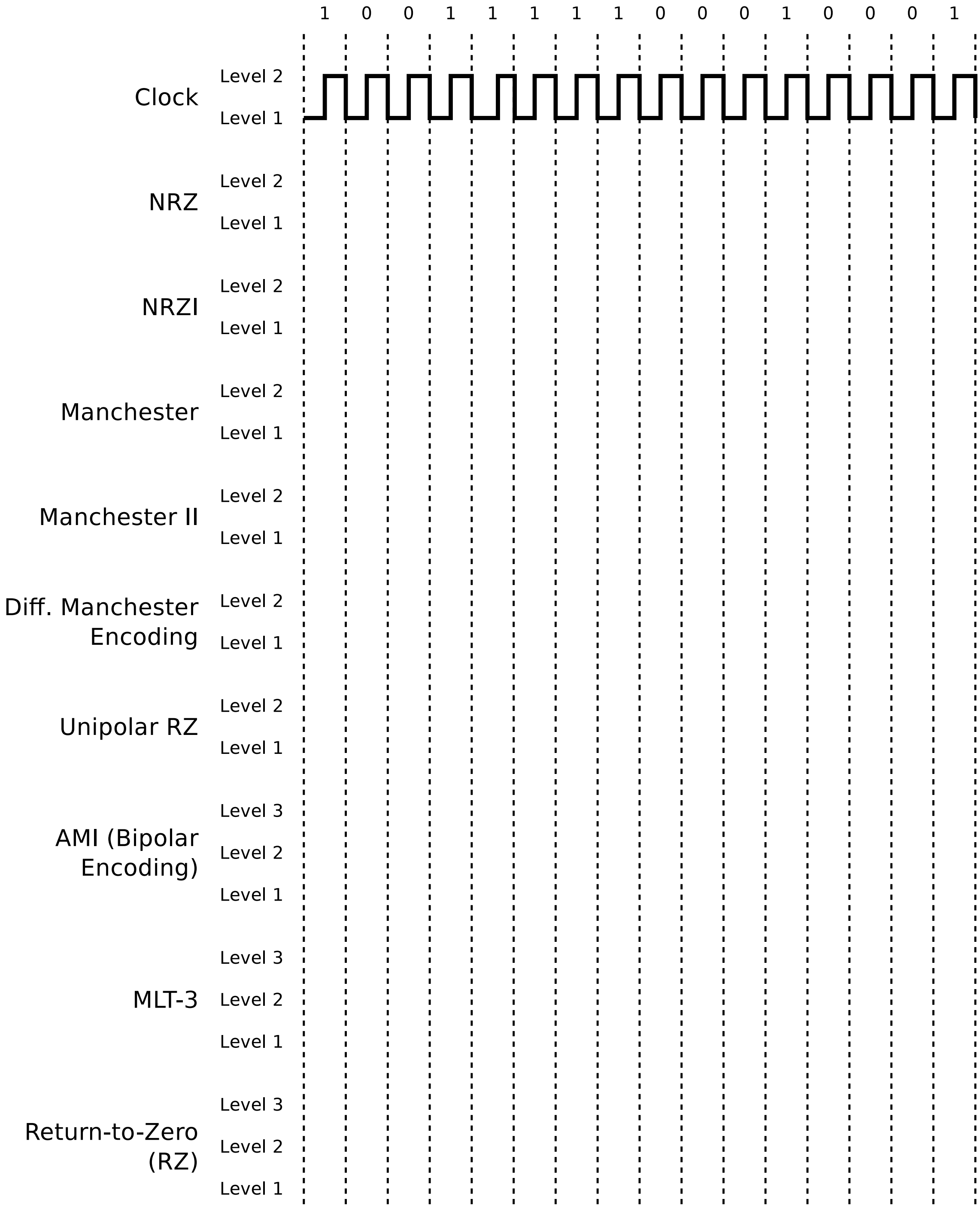
Exercise 5 (Line Codes)

1. Explain why computer networks require line codes.
2. Many different line codes exist. Explain why it is impossible to use one single line code for every network technology.
3. Explain the way Non-Return-To-Zero (NRZ) works.
4. Name the two problems that can occur when NRZ is used to encode data.
5. Explain both problems from subtask 4 in detail.
6. Explain how the problems from subtask 4 can be avoided.

7. Name at least 5 line codes that use 2 signals levels.
8. Name at least 3 line codes that use 3 signal levels.
9. Which line codes ensure a signal level change for each logical 1 bit?
10. Which line codes ensure a signal level change for each transmitted bit?
11. Why do not all line codes ensure a signal level change for each transmitted bit?
12. Which line codes ensure that the signal levels are equally distributed?
13. Why is it important for the receiver of signals, which are encoded according to the Differential Manchester Encoding, to know the initial signal level?
14. What is a scrambler?
15. All line codes have drawbacks. What can be done to avoid the problems, that can result from these drawbacks?
16. Which line code maps groups of **4** payload bits onto groups of **5** code bits?
17. Which line code maps groups of **5** payload bits onto groups of **6** code bits?
18. Why do some line codes, that map groups of payload bits onto groups of code bits, implement variants with neutral inequality, positive inequality and negative inequality?
19. How is the efficiency of a line code calculated?

Exercise 6 (Encoding Data with Line Codes)

1. Give the encodings for the given bit pattern. *Attention: Please assume that the initial signal level of NRZI and Differential Manchester Encoding is signal level 1 (low signal).*



2. Encode the bit sequences with 4B5B and NRZI and draw the signal curve.

   - `0010 1111 0001 1010`
   - `1101 0000 1001 1110`

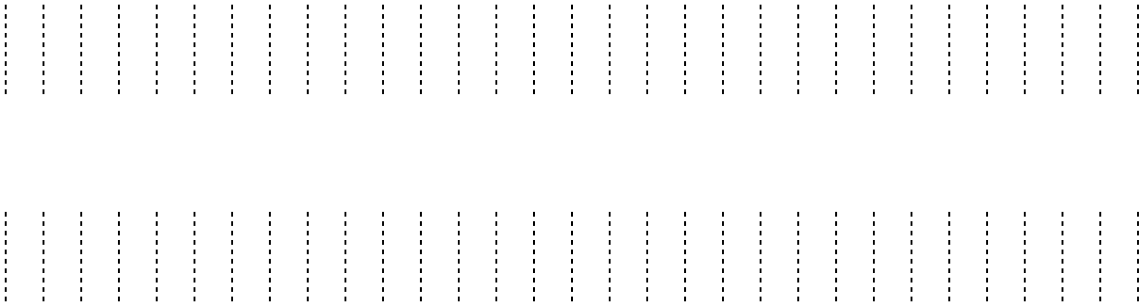
   Attention: Please assume that the initial signal level of NRZI is signal level 1 (low signal).

Label	4B	5B	Function
0	0000	11110	0 hexadecimal
1	0001	01001	1 hexadecimal
2	0010	10100	2 hexadecimal
3	0011	10101	3 hexadecimal
4	0100	01010	4 hexadecimal
5	0101	01011	5 hexadecimal
6	0110	01110	6 hexadecimal
7	0111	01111	7 hexadecimal
8	1000	10010	8 hexadecimal
9	1001	10011	9 hexadecimal
A	1010	10110	A hexadecimal
B	1011	10111	B hexadecimal
C	1100	11010	C hexadecimal
D	1101	11011	D hexadecimal
E	1110	11100	E hexadecimal
F	1111	11101	F hexadecimal

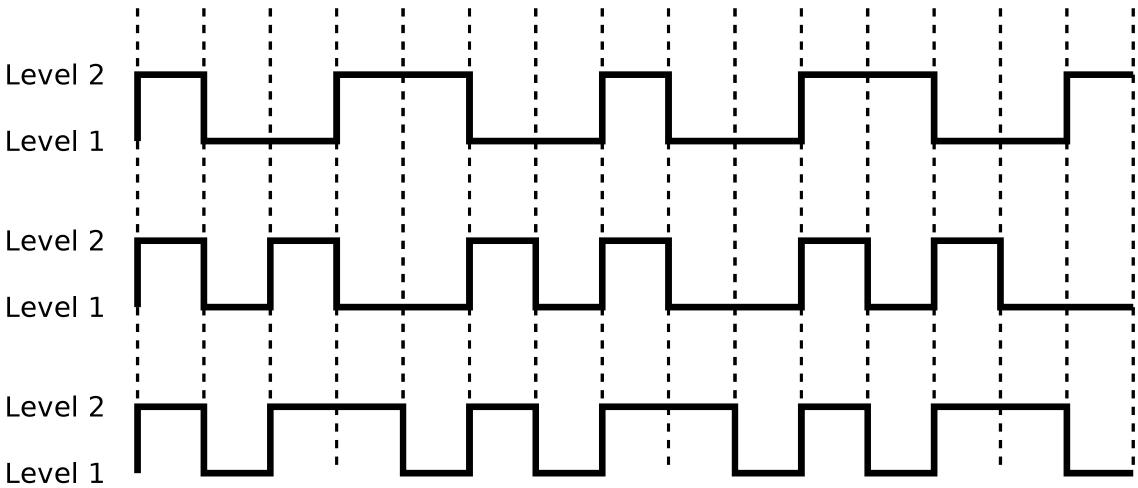
3. Encode the bit sequences with 5B6B and NRZ and draw the signal curve.

   - `00001 01011 11000 01110 10011`
   - `11010 11110 01001 00010 01110`

5B	6B neutral	6B positive	6B negative	5B	6B neutral	6B positive	6B negative
00000		001100	110011	10000		000101	111010
00001	101100			10001	100101		
00010		100010	101110	10010		001001	110110
00011	001101			10011	010110		
00100		001010	110101	10100	111000		
00101	010101			10101		011000	100111
00110	001110			10110	011001		
00111	001011			10111		100001	011110
01000	000111			11000	110001		
01001	100011			11001	101010		
01010	100110			11010		010100	101011
01011		000110	111001	11011	110100		
01100		101000	010111	11100	011100		
01101	011010			11101	010011		
01110		100100	011011	11110		010010	101101
01111	101001			11111	110010		



4. These signal curves are encoded with NRZI and 4B5B. Decode the data.



Source: Jörg Roth. Prüfungstrainer Rechnernetze. Vieweg (2010)

Exercise 7 (Do some research)

1. In the late 1980s modems typically achieved a data rate of 9.6 kbit/s (2400 baud). Which modulation scheme was used and how many bits could be employed per symbol?

2. Find out which (historical) data storage used Differential Manchester Encoding.

3. An Internet access over ISDN (Integrated Services Digital Network) offers a data rate of 64 kbit/s (single B channel). Why did it still provide a much more significant advantage over, for instance, 56k modem connections?

4. The (in)famous *hacker* John Thomas Draper is widely known as **Captain Crunch**. Explain the origin of this nickname and how it related to the principles of the physical layer.

Exercise 8 (Lab Exercise: Traffic Monitoring)

For this exercise you have to boot the image labeled **PCNA-Lab** on the lab PCs again.

1. After booting and logging in, start a terminal emulator again and activate your network interface by calling:
   `sudo ip link set eno0 up`
2. Now you should start the *Wireshark* application and start to capture the traffic on that device (double click on *eno2*).
3. Go back to the terminal emulator and call
   `sudo NetworkManager`
4. Analyze the captured traffic:
   - Which protocols are involved during network interface initialization?
   - Map the identified protocols to the individual layers of the OSI reference model.
   - Identify the encapsulation (the header and potentially trailer data) on each layer.
   - Open a *web browser* and open *http://example.teaching.dahahm.de*. Filter the traffic in Wireshark to display only the communication towards the web server.
   - Run an *iperf3* measurement between your computer and your neighbor's one and identify the corresponding traffic in Wireshark.