

Tutoriumsblatt 3

Diskussion: 18. + 19. Mai 2021

Aufgabe 3.1: Abtasttheorem

Mit der Digitalisierung des Telefonnetzes (durch ISDN) wurde festgelegt, dass zur digitalen Telefonie ein Kanal mit 64 kBit/s benötigt wird.

- Auf welchen Annahmen basiert die Bitrate von 64 kBit/s?
- Hängt diese Bitrate auch mit dem verwendeten Medium (Twisted Pair) zusammen?

Aufgabe 3.2: Code Division Multiplexing (CDM)

In modernen Mobilfunknetzen wird teilweise Code Division Multiplex (CDM) als Mehrfachzugriffsverfahren eingesetzt. Bei CDM wird jedes Datenbit $x \in \{0, 1\}$ zur Übertragung mittels eines Spreizcodes $a = a_0 a_1 \cdots a_{m-1}$ mit $a_i \in \{-1, +1\}$ in eine Signalfolge übersetzt, wobei gilt:

x	gesendete Bitfolge
0	$a = a_0 a_1 \cdots a_{m-1}$
1	$-a = -a_0 - a_1 \cdots - a_{m-1}$

Es gibt also pro Spreizcode genau zwei Codeworte.

Damit mehrere Sender gleichzeitig senden können, ohne die Übertragung der anderen Sender zu stören, ist es notwendig, dass die verwendeten Codes orthogonal sind. Zwei Codes a mit $a_i \in \{-1, +1\}$ und b mit $b_i \in \{-1, +1\}$ sind orthogonal, wenn ihr Skalarprodukt 0 ergibt, d.h. wenn gilt:

$$\sum_{i=0}^{m-1} a_i * b_i = 0$$

Betrachten Sie in dieser Aufgabe Spreizcodes der Länge $m = 4$.

- Wie viele verschiedene Spreizcodes sind bei dieser Länge möglich?
Sind alle Spreizcodes orthogonal zueinander?
- Geben Sie alle zu $a = 1, 1, 1, 1$ orthogonalen Spreizcodes an.

Aufgabe 3.3: Bit-Stuffing

Führen Sie anhand des folgenden Beispiels Bit-Stuffing durch:

- Die zu übertragende Bitfolge sei 1011 0001 0011 0110 0110 0101 1101 01.
- Das Flag zur Kennzeichnung von Start und Ende der Übertragung sei 1001101.
- Bit Stuffing soll erfolgen, indem eine 0 eingefügt wird.

Aufgabe 3.4: CRC

- Gegeben ist die Bitsequenz 1011 1001, die mittels CRC gesichert übertragen werden soll. Berechnen Sie die Prüfsumme unter Verwendung des Generatorpolynoms $G(x) = x^4 + x + 1$.
- Warum hat sich CRC zur Fehlerprüfung in der Datenkommunikation durchgesetzt? Welche Vorteile bietet sie verglichen mit z.B. der Verwendung eines Paritätsbits?
- Kann man mittels einer CRC auch Fehler korrigieren?
- Warum wird die berechnete Prüfsumme nicht wie alle anderen Kontrollinformationen mit im Header übertragen, sondern an die Nutzdaten angehängt?

Aufgabe 3.5: Fehlerkorrigierende Codes

Sie wollen Nachrichten übertragen, welche nur aus den Zeichen „A“, „B“, „C“ und „D“ bestehen können. Gegeben sei die folgende Codierungsvorschrift, die regelt, welche Bitfolgen zur Repräsentation der jeweiligen Zeichen übertragen werden:

- „A“: 000000
- „B“: 111000
- „C“: 000111
- „D“: 111111

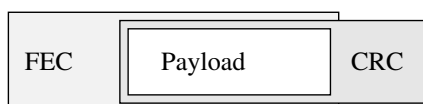
- Sie empfangen nun die im Folgenden dargestellten Bitfolgen. Welche Zeichen werden jeweils dekodiert? Lässt sich dies in jedem Fall eindeutig entscheiden?
 - 100000
 - 001111
 - 101111
 - 101010
- Wie viele Bitfehler lassen sich mit der gegebenen Codierung erkennen?
- Wie viele Bitfehler lassen sich mit der gegebenen Codierung korrigieren?
- Sie wollen nun fehlererkennende (CRC) und fehlerkorrigierende (FEC) Codes gemeinsam verwenden, um Fehler auf jeden Fall zu erkennen und soweit möglich auch korrigieren zu können. Dazu können Sie zwischen drei Szenarien wählen:



Szenario 1



Szenario 2



Szenario 3

Im ersten Szenario wird zunächst die CRC-Prüfsumme zu den Nutzdaten (Payload) berechnet, anschließend werden Prüfbits zur Fehlerkorrektur über Payload und CRC-Prüfsumme berechnet.

Im zweiten Szenario geht man genau umgekehrt vor: erst werden Korrekturbits berechnet, danach erfolgt über Payload und Korrekturbits eine Berechnung der CRC-Prüfsumme.

Im dritten Szenario werden sowohl CRC- als auch FEC-Prüfbits nur über den Payload berechnet.

Welches dieser Szenarien halten Sie für das Sinnvollste?