

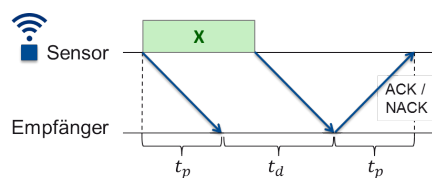
Übungsblatt 4

20. April 2017

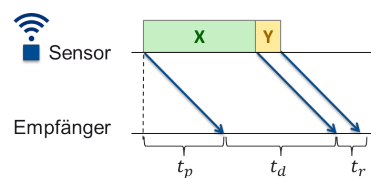
Besprechung: 27. April 2017

Problem 4.1: Anwendungsfall: Internet of Things

Es werden batteriebetriebene Sensoren betrachtet, die zweimal pro Stunde ein Datenpaket der Länge 100 Byte an einen Empfänger senden. Der Übertragungskanal hat eine Übertragungskapazität $C = 400$ kbps und eine Bitfehlerrate (BER) von $p_b = 10^{-4}$. Betrachten Sie im folgenden beide Möglichkeiten zur Übertragung der Sensordaten, ARQ und FEC.



(a) Automatic Repeat Request (ARQ)



(b) Forward Error Correction (FEC)

1. *Auslastung*. Wie ist die maximal mögliche Auslastung der Übertragungsstrecke für (1) kurze Übertragungsstrecken $t_p = 0.2$ ms und (2) lange Übertragungsstrecken $t_p = 99$ ms?
2. *Internet of Things*. Welche Übertragungsart eignet sich für batteriebetriebene Sensoren? Berechnen Sie, wann FEC zu bevorzugen ist! Was müssen Sie in der Praxis beachten?

Problem 4.2: Zweidimensionale Paritätskontrolle

Mit der aus der Vorlesung bekannten eindimensionalen Paritätskontrolle können nur Einfachfehler (und ungerade Fehleranzahl) erkannt werden. Das Verfahren wird nun auf zwei Dimensionen erweitert und ist auch bekannt als Kreuzparität oder Blockparität. Die Nutzdatenwörter zu 8 Bit Länge werden in 8 Blöcke gruppiert und in einem Quadrat (Array) zusammengefasst. Zu jeder Zeile und zu jeder Spalte wird ein Paritätsbit berechnet. Die Konstruktion der Paritätsbits wird in folgender Tabelle dargestellt.

$x_{1,1}$	$x_{1,2}$	\dots	$x_{1,8}$	$y_{1,9}$
$x_{2,1}$	$x_{2,2}$	\dots	$x_{2,8}$	$y_{2,9}$
\vdots	\vdots	\ddots	\vdots	\vdots
$x_{8,1}$	$x_{8,2}$	\dots	$x_{8,8}$	$y_{8,9}$
$y_{9,1}$	$y_{9,2}$	\dots	$y_{8,8}$	$y_{9,9}$

1. *Bitfehler*. Welche der Fehler in den Fehlerszenarien werden erkannt? Können Fehler korrigiert werden? Diese Bits werden in den fünf Szenarien fehlerhaft übertragen.
(1) $x_{1,1}$ (2) $x_{1,1}, x_{1,2}$ (3) $x_{1,1}, x_{2,1}$ (4) $x_{1,1}, x_{2,2}, x_{3,3}$ (5) $x_{4,3}, x_{4,5}, x_{6,3}, x_{6,5}$
2. *Fehlererkennung*. Wie groß ist die Hamming-Distanz? Wieviele Fehler werden erkannt und korrigiert? Wie ist die Codeeffizienz der Blockparität? Wieviele Codeworte gibt es?
3. *Ohne Längsparität*. Wie ändert sich die Hamming-Distanz, wenn die letzte Zeile (sog. Längsparität) weggelassen wird?

Problem 4.3: Repetition Code und Code-Effizienz

Der Repetition Code wiederholt jedes übertragene Bit n mal. Zum Beispiel für $n = 3$ wird die Bitfolge 0010 codiert in 000 000 111 000.

1. *Fehlerkorrektur*. Berechnen Sie, wieviele Fehler korrigiert werden können. Wie ist die minimale Hamming-Distanz? Was passiert bei einer Bitfehlerwahrscheinlichkeit von $p_b > 50\%$?
2. *Restfehlerwahrscheinlichkeit*. Berechnen Sie die Restfehlerwahrscheinlichkeit für gegebenen n und p_b . Stellen Sie das Ergebnis grafisch dar.
3. *Matlab Code*. Der folgende Matlab Code implementiert einen Encoder für den Repetition Code. Implementieren Sie einen einfachen Decoder für die empfangenen Daten `rcvd`.

```
k=3; % Anzahl der Nutzbits
data = randi([0,1],1,k); % k zufaellige Bits werden erzeugt
r=5; % Jedes Nutzbit wird r-mal wiederholt
codeword=repmat(data,r,1); % Repetition Code
p_b=0.1; % Bit Error Rate (BER)
errorVector=rand(r,k)<p_b; % Bitfehler bei Uebertragung
rcvd = mod(codeword+errorVector,2); % empfangene Daten
```

Problem 4.4: Verständnisfragen

1. *Modulo-Arithmetik*. Ist die Addition im Galois-Feld $\mathbb{F}_2 = \{0, 1\}$ distributiv? Gilt $(a + b) \cdot c \bmod 2 = (ac \bmod 2) + (bc \bmod 2) \bmod 2$?
2. *Hamming-Schranke*. Bei t Fehlern gibt es $\sum_{i=0}^t \binom{n}{i}$ mögliche Fehlerkombinationen für Codeworte der Länge n . Wieviele Fehlersyndrome sind durch m Prüfbits darstellbar? Was bedeutet dies für die Anzahl der Nutzbits?
3. *Code Effizienz*. Wodurch ist die Code Effizienz $\frac{k}{m+k}$ für m Prüfbits und k Nutzbits in der Praxis beschränkt? Welche Parameter wählen Sie für Ethernet (BER $p_b = 10^{-12}$)?