Exercise Sensor Networks

Lecture 2: Communication (MAC and error handling)

Exercise 2.1:

In a Hamming code word, rather than a data bit, a check bit toggles. Can the mistake be detected and corrected and if yes, how?

Exercise 2.2:

The following Hamming code word is given: 01111001111. Create an error with as few changes as possible that can not be detected.

Exercise 2.3:

A number of d bit errors should be corrected. Explain why a distance of 2d is not sufficient for the code?

Exercise 2.4:

In the last lecture we have seen an estimation of how many redundant bits are necessary to detect and correct 1 bit errors. Now do the same estimation for 2 bit errors. It is not necessary to find a particular code, only a lower bound for the number of check bits is of interest.

How many bits are necessary to protect a 7 bits ASCII code against at most 2 toggled bits?

Exercise Sensor Networks

Lecture 2: Communication (MAC and error handling)

Exercise 2.5: Cost for error correction vs. retransmission

a)

For a given transmission channel there occurs 1 error in 4000 bits on average. These single bit errors are statistically independent. A single packet consists of 128 bytes. It is either transmitted fully and correctly or not at all. The receiver can detect whether a packet was transmitted free of errors without any additional costs. If an error occurred, the receiver asks the sender only once for retransmission. The request for retransmission is considered to be an ordinary packet of 128 bytes. If an error occurs in such a request it is treated as if no request was ever sent.

How high is the overall data rate in this scenario (in percent of the data rate that could theoretically be achieved if no error occurred)?

b)

To make things easier we assume that a bit error occurs only once per packet. Rather than asking the sender for retransmission, we choose to employ forward error correction, using the Hamming code from the lecture (the code itself is not of importance here).

How high is the actual data rate (in percent) compared to the one that would theoretically be possible without forward error correction and if no errors occurred.

Exercise 2.6: (opt) Explain why protecting n bits of data against single bit errors with forward error correction requires only O(log(n)) check bits while protecting a fixed number of c bits against a number of n toggled bits requires an exponential number of check bits. Hint: Think of the table consisting of valid and invalid code words and of the method in which bits are checked in the Hamming code.

Exercise Sensor Networks

Lecture 2: Communication (MAC and error handling)

Exercise 2.7: Fountain Codes

A sender wants to transmit the following 32 bits

10110100 01011011 01010101 10110110

in four chunk-packets, each of which contains 8 bits. Both, sender and receiver use the same random number generator which produces the following bit-stream:

1110 0101 1001 0110

For data transmission, the **Random Linear Fountain Code** from the lecture is used.

Proceeding:

Sender side
- Divide the message into chunks. Combine the chunks bit-wise according to the bit-merging vector which is taken from the output of the random number generator.
- Stop sending further chunks as soon as a sufficient number of XORed chunks with linear independent merging vectors have been sent.

Receiver side
- Collect the incoming chunks until a sufficient number is received. Sufficient means, that their merging vectors are linearly independent. The merging vectors are taken from the output of the random number generator as was done on the sender side.
- After having gathered enough data, calculate the modulo 2 inverse matrix of the matrix formed by the merging vectors.
- XOR the received chunks according to the inverse matrix.