Lecture 5: Source coding

Exercise 5.1: CRC polynomials

Divide the message 1011010011 by the generator polynomial 10011 as done in the lecture. Write down the whole message as if it was transmitted to a receiver.

Exercise 5.2: CRC polynomials

(a) Find an easy to identify case in which a given polynomial will fail for a given error.

(b) How long does a generator polynomial have to be at least in order to detect every possible bit error if the message has n bits?
Exercise Data Networks

Lecture 5: Source coding

Exercise 5.3: 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 generation which produces the following bit-stream:

1110 0101 1001 0110

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

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

Receiver
- Collect the incoming chunks until a sufficient number is received.
  Sufficient means, that their merging vectors are linearly
  Side 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.
Lecture 5: Source coding

Exercise 5.4: Fountain codes

The following considerations should prepare you for the last assignment. A good understanding of this issue will enable you to solve it with fewer lines of code and less effort.

In the previous example you have simulated a sender that produces packets and a receiver that receives them and reconstructs the data. While the sender can produce an arbitrary number of packets without problems, the receiver may fail if the random bit vectors prove to be co-linear.

(a) Describe how the receiver finds out that the bit vectors it got so far are not suitable for the reconstruction of the packets.

(b) How can the receiver solve this problem?