Exercise Data Networks (due till November 24, 2008)

Lecture 4: Source coding

Exercise 4.1:

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

Solution:

When checking the parity of a certain check bit, the wrong parity will be detected and the counter is incremented by the number of the particular check bit. Obviously other check bits will not fail because all other bits are correct. Note that the counter already contains the number of the erroneous check bit. In other words: The normal algorithm of verifying bits considers data bits as well as check bits. No special care has to be taken.
Exercise 4.2:

The following Hamming code word is given: 0111001111. Create an error with as few changes as possible that can not be detected. How do you know that less changes of bits can not pass undetected?

Solution:

Check bits are turquoise

toggled bits are red

1234 5 6 7 8 9 0 1
0 0 0 1 1 1 0 0 1

For a minimum of toggled bits a data bit should be chosen which influences as few check bits as possible. Some data bits have only 2 check bits which care for them, e.g., bit number 9 (=8+1). So toggle 9, 8 and 1 and check whether 1 and 8 still result in an even parity (together with the data bits for which they are responsible).

Can less than three changes ever pass undetected? The Hamming code can correct 1 (e=1) bit error in a code word. Thus, the minimal distance between two valid code words must be at least \( d=2e+1 \) which is 3.
Exercise Data Networks

Lecture 4: Source coding

Exercise 4.2: (continued)

Now generate a bit error changing as many bits as possible.

Solution:

\[
\begin{array}{cccc}
12345678901 & 12345678901 & 12345678901 \\
01111001111 & 01111001111 & \text{result: } 10000110000
\end{array}
\]

Exercise 4.3:

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

Solution:

The reason why the distance has to be \(2e+1\) and not \(2e\) only is that for every incorrect code word there has to be a short and a long way to the nearest correct code word. In the case of a distance of 5 it is always possible to change 1 or 2 bits (namely those which are incorrect) in order to get to the next correct code word. Changing 3 or more bits makes no sense because it would already be sufficient to change 2 or 1 bit to obtain a valid symbol. However, if the distance between valid symbols of a code was e.g., 4, there could be two possibilities to change 2 bits in order to obtain a valid code word. As a consequence, the choice of the nearest valid code word would not be unique.
Lecture 4: Source coding

Exercise 4.4:

In the last lecture we have seen an estimate of how many redundant bits are necessary to detect and correct 1 bit error. 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?

Solution:

The code should contain $2^m$ data bits and a yet unknown number of $r$ check bits.
Remark: The code will contain $2^{m+r}$ symbols which are divided into $2^m$ valid symbols and $2^{m+r}-2^m$ invalid ones.

Either one- or two bit errors can occur:

1 bit errors: For each of the valid code words an invalid one can be created by simply swapping one of the $n$ bits.

2 bit errors: Again one of the $n$ bits can be toggled. For the second error bit $(n-1)$ possibilities remain. At a first glance $n(n-1)$ possibilities evolve. But since the order of the toggled bits does not matter the number of possibilities shrinks by 50%. This means $n(n-1)/2$ erroneous code words have to be considered for every valid code word.

No error: Of course we must not forget to consider the valid code words.
Exercise Data Networks

Lecture 4: Source coding

Exercise 4.4: (continued)

Solution:

\[
\left( n + \frac{n(n-1)}{2} + 1 \right) 2^m \leq 2^{r+m}
\]

valid code word

2 bit errors

1 bit error

\[
\left( m + r + \frac{(m+r)(m+r-1)}{2} + 1 \right) 2^m \leq 2^{r+m}
\]

\[
\frac{m+r+(m+r)^2}{2} + 1 \leq 2^r
\]

\[
\frac{7+7+(7+7)^2}{2} + 1 \leq 2^7
\]

106 \leq 128

In order to protect the 7 bit ASCII code against 2 simultaneous bit errors at last 7 check bits are necessary.
Lecture 4: Source coding

Exercise 4.5: Explain why protecting n bits of data against single bit errors with forward error correction requires only $\mathcal{O}(\log(n))$ check bits.

Solution:

Protecting against single bit errors can e.g., be done with the Hamming code in which n check bits take care of $\mathcal{O}(2^n)$ data bits. Adding only one more check bit is enough to double the number of data bits which can be protected.

* $([2^n]-n]$ to be precise)