02 Information theory

02.04 Special encodings

- Error models
- Parity bits
- Hamming codes

Error model

- Independent errors - error probability per bit *p*
- Error probability per word

$$p_{word} = 1-(1-p)^n \cong np \qquad (1)$$

- Multiple-error probability

$$p_{word}(1) = \binom{n}{1} p(1-p)^{n-1} \cong np = o(p)$$

$$p_{word}(2) = \binom{n}{2} p^2(1-p)^{n-2} \cong \frac{n(n-1)}{2} p^2 = o(p^2)$$

...

$$p_{word}(n) = \binom{n}{n} p^n = p^n$$

Detection/correction target

- Multiple errors are much less likely than single errors
- The minimum target for error detection/correction are single errors per word
- No encodings can detect/correct errors of any multiplicity

Hamming distance

- Hamming distance between two words w1 and w2

$$d_H(w1,w2)$$

- Hamming distance of a code: minimum Hamming distance between pairs of code words

$$d_H(\text{code}) = \min_{w1,w2 \in \text{code}} \{d_H(w1,w2)\}$$

- Irredundant codes (using the minimum number of bits) have Hamming distance $d_H(\text{code})=1$

Detection/correction requirements

- If $d_H(w1,w2)=1$, a single error may transform $w1$ in $w2$
- If $w1$ and $w2$ belong to the same code, the error cannot be detected nor corrected, thus impairing reliability
- The minimum Hamming distance required to detect up to e errors per word is $d_H(\text{code})=e+1$
- The minimum Hamming distance required to correct up to e errors per word is $d_H(\text{code})=2e+1$

Code classification

- Error detecting codes (e-EDC)
  - e errors transform any *code* word in a *non-code* word
- Error correcting codes (e-ECC)
  - e errors transform any *code* word in a *non-code* word that is closer to the original word than to any other code word
- Any e-ECC is also $2e$-EDC
- Any code with $d_H > 1$ is a redundant code
- A n-bit redundant code is called *separable* if each codeword is composed of:
  - r information bits (belonging to an irredundant encoding)
  - m control bits (added to increase the Hamming distance between the codewords)

$n=r+m$

Replication codes

- Simple separable codes with the desired Hamming distance d_H can be obtained by replicating d_H times an irredundant code
- Error detection technique
  - bit-wise comparison
- Error correction technique
  - bit-wise majority voting

$$n = d_H r = r + (d_H\text{-}1)r$$
$$m = (d_H\text{-}1)r$$

Parity codes

- *Parity bit*: control bit computed in such a way that the codeword (or a subset of its bits) contains an even number of 1's
- *Parity code*: separable code obtained by adding parity control bits to an irredundant code
- 1-EDC parity code: code with m=1 using a single parity bit computed over all the bits of the codeword
- Correction technique:
  - Parity test of the number of 1's in the word

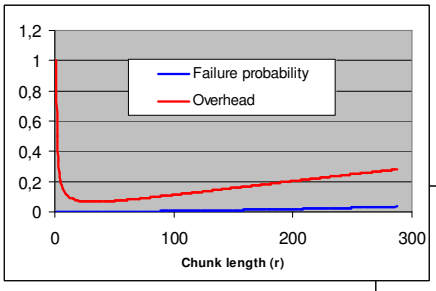
Performance of parity code

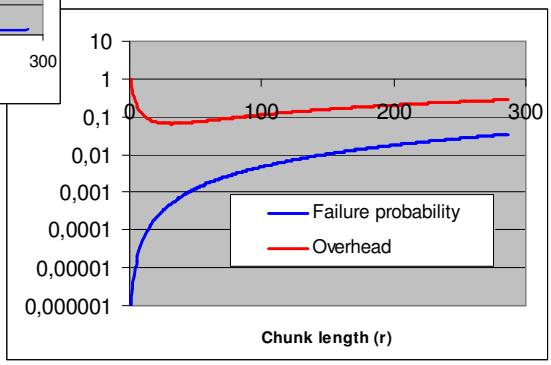
- Given:
  - *p* error probability per bit
  - *L* original length of a bit stream to be encoded
  - *r* size of a words (chunk of the bit stream)
- We can compute:
  - *p_c* = 1-*p* complement of *p*
  - *n* = *r*+1 size of a codeword
  - *P_w_correct* = (1-*p*)^n
  - *P_w_error* = 1-*P_w_correct*
  - *P_w_1error* = *n***p**(1-*p*)^(*n*-1)
  - *P_w_Merror* = *P_w_error* – *P_w_1error*

Performance of parity code

- We can also compute:
  - *N_words*0 = *L/r*
  - *N_reTx* = *P_w_1error* / (1-*P_w_1error*)
  - *N_words* = *N_words*0 * (1+*N_reTx*)
  - *L_total* = *N_words* * *n*
- The value of *r* (chunk length) can be decided in order to optimize the performance of the code, expressed in terms of:
  - Overhead = L_total / L - 1
  - Failure prob = *P_w_Merror*

Performance of parity code





d_H of Parity codes

- A single parity bit guarantees d_H=2
- Parity codes with d_H>2 can be obtained by using more than 1 parity bits
- The configuration of all parity bits is called *syndrome*
- Parity bits must be *independent from each-other*, or otherwise they will be ineffective in increasing the Hamming distance of the code
- A parity code is an ECC if all tolerated errors give rise to different syndromes. In this case, the syndrome uniquely encodes the error, enabling correction.

Hamming codes

- ECC parity codes using the minimum number of control bits required to grant single error correction capabilities (i.e., d_H=3)
- Hamming rule:

$$2^m \geq r + m + 1 = n + 1$$

control (parity) bits must have enough configurations to encode all possible error positions and the error-free case.

r	m	n
1	2	3
2	3	5
3	3	6
4	3	7
5	4	9

Hamming codes (encoding)

- Parity bits are placed in *power-of-2 positions*
- The i-th parity bit is computed as the EXOR of all information bits whose position contains a 1 in the i-th bit of its binary encoding

	n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
	r			1		2	3	4		5	6	7	8	9	10	11		12	13
position		**1**	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
		0	**1**	1	0	0	1	1	0	0	1	1	0	0	1	1	1	0	1
		0	0	0	**1**	1	1	1	0	0	0	0	1	1	1	1	0	0	0
		0	0	0	0	0	0	0	**1**	1	1	1	1	1	1	1	0	0	0
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	1	1

Hamming codes (decoding)

- In case of a single error, the syndrome is the binary representation of the position of the error in the word
  - 01**1**101 instead of 010101 => syndrome 110
    (error in position 3)
  - 0**0**0101 insetad of 010101 => syndrome 010
    (error in position 2)
- Multiple errors cannot be corrected
  - 01**1**1**1**1 instead of 010101 => syndrome 011
    (error in position 6 – wrong information)