III-1
Códigos detectores e correctores de erros

Comunicações

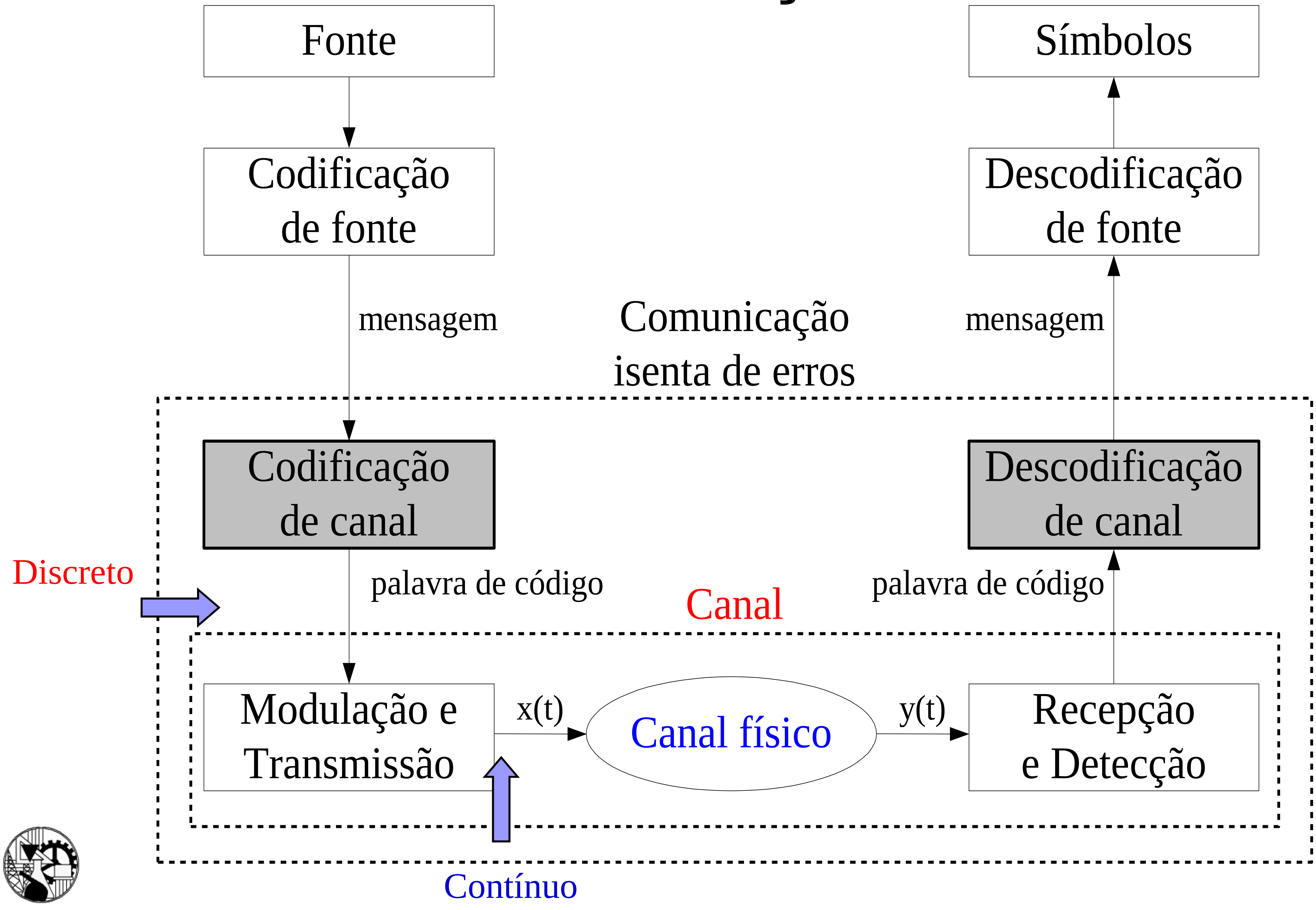
(29 Dezembro de 2008)

Sumário

1. Aspectos gerais sobre a comunicação digital
    1. Causa de erros

2. Códigos detectores e correctores de erros
    1. Códigos de bloco linear (n,k)
    2. Características dos códigos
    3. Capacidades de detecção e correcção de erros
    4. Códigos de repetição e bit de paridade
    5. Código de Hamming
    6. Descodificação baseada em síndroma
        1. Detecção e correcção
3. Aplicações e Bibliografia

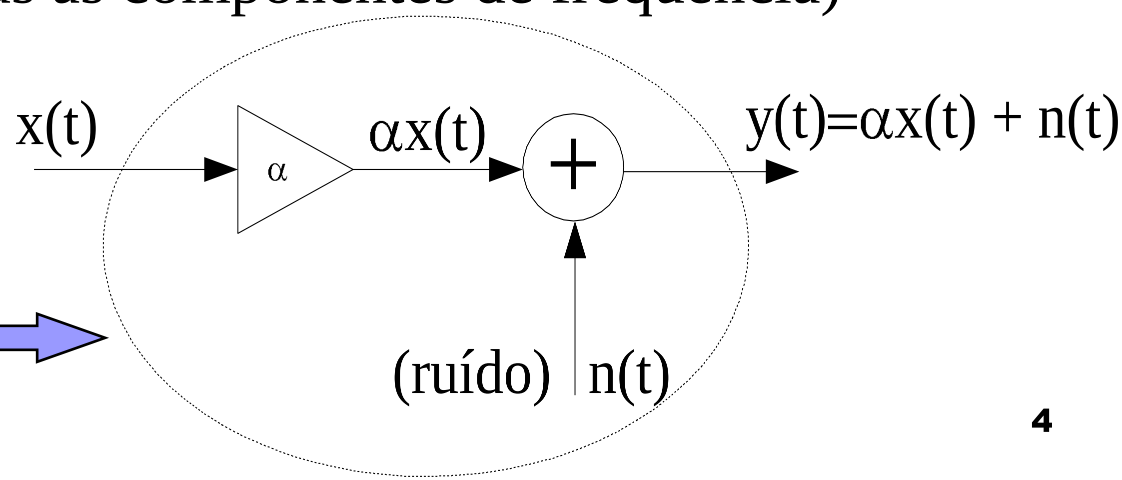
Cenário de utilização



Modelo de canal físico

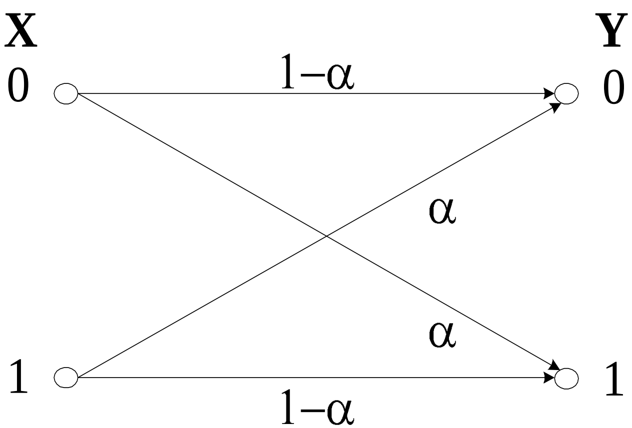
- A correcção e/ou detecção de erros é necessária vido aos erros introduzidos no canal de transmissão ou de armazenamento
- O modelo AWGN (Aditive White Gaussian Noise) é realista em muitos cenários
  - Ruído aditivo (soma-se ao sinal informação)
  - Gaussiano (a distribuição de amplitudes é gaussiana)
  - Branco (tem todas as componentes de frequência)

Canal físico



Modelo de canal discreto

- O canal é analisado através de modelo discreto usando variáveis aleatórias (v.a.)
- Modelo BSC - *binary symmetric channel*



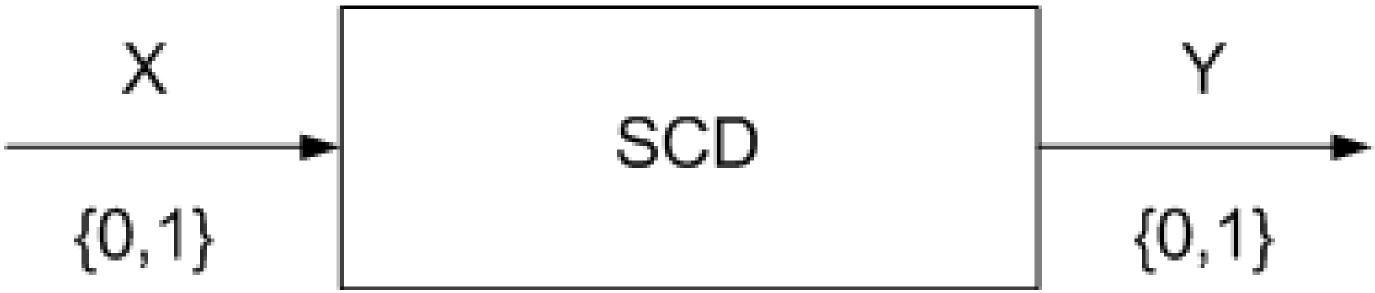
<u>Probabilidade de erro de bit</u>

$$
\begin{aligned}
P_e &= P(y_0, x_1) + P(y_1, x_0) \\
&= P(y_0|x_1)P(x_1) + P(y_1|x_0)P(x_0) \\
&= \alpha P(x_1) + \alpha P(x_0) \\
&= \alpha
\end{aligned}
$$

A probabilidade de erro define o **BER (*Bit Error Rate*)** do canal. É a taxa de erros por bit.

Modelo de canal discreto

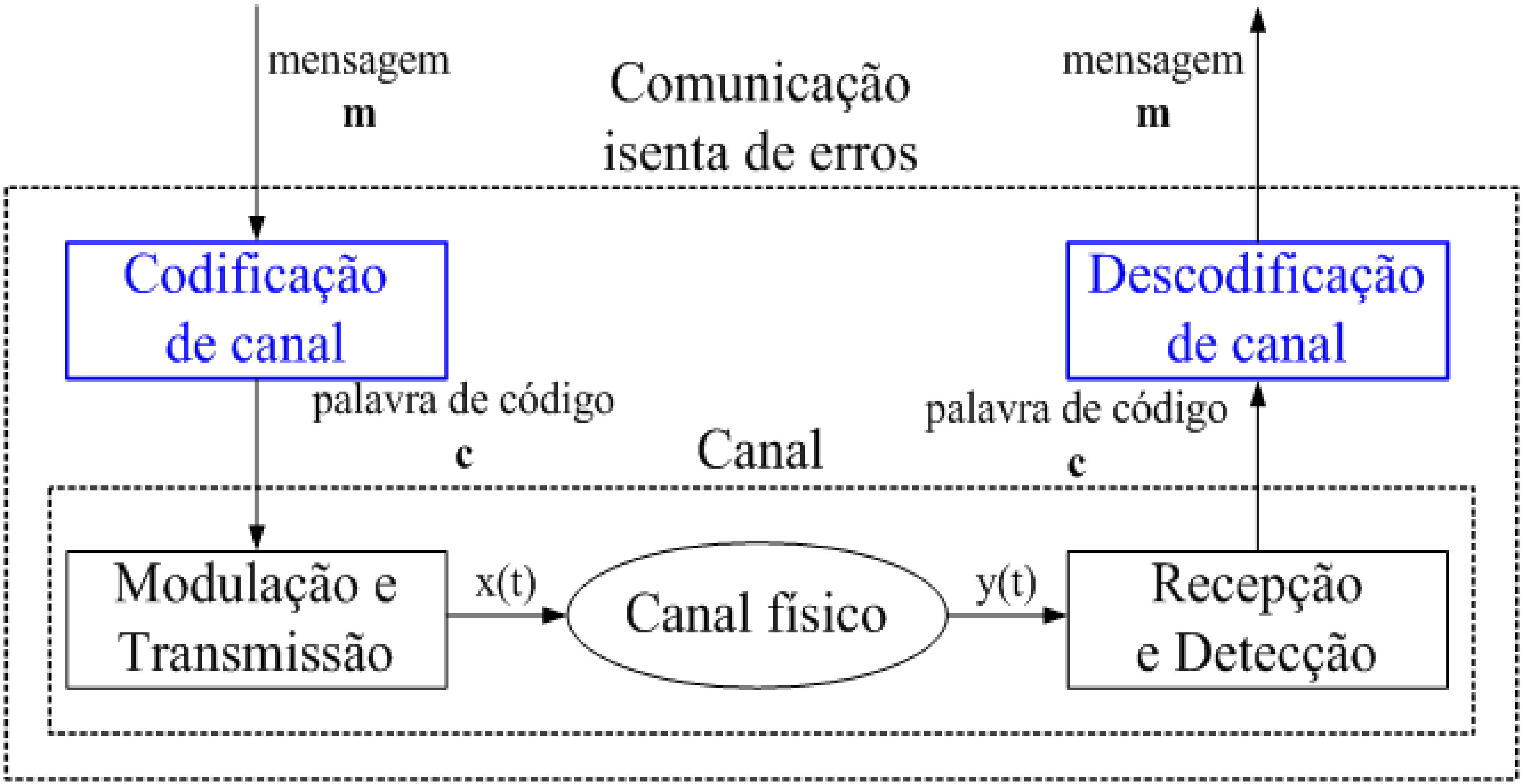
- O canal é analisado através de modelo discreto usando variáveis aleatórias (v.a.)
- Do ponto de vista da transmissão, um SCD pode ser visto através de modelo probabilístico
- A probabilidade de erro por troca de bit não é nula



Códigos de canal

- A detecção e correcção consegue-se ao introduzir <u>redundância</u> na mensagem original
- Essa redundância é função da mensagem
- Códigos a analisar: repetição; bit de paridade par; Hamming.
- Os códigos de canal são utilizados nos modos:
  - FEC - **F**orward **E**rror **C**orrection
  - ARQ - **A**utomatic **R**epeat Re**Q**uest

Cenário de Aplicação

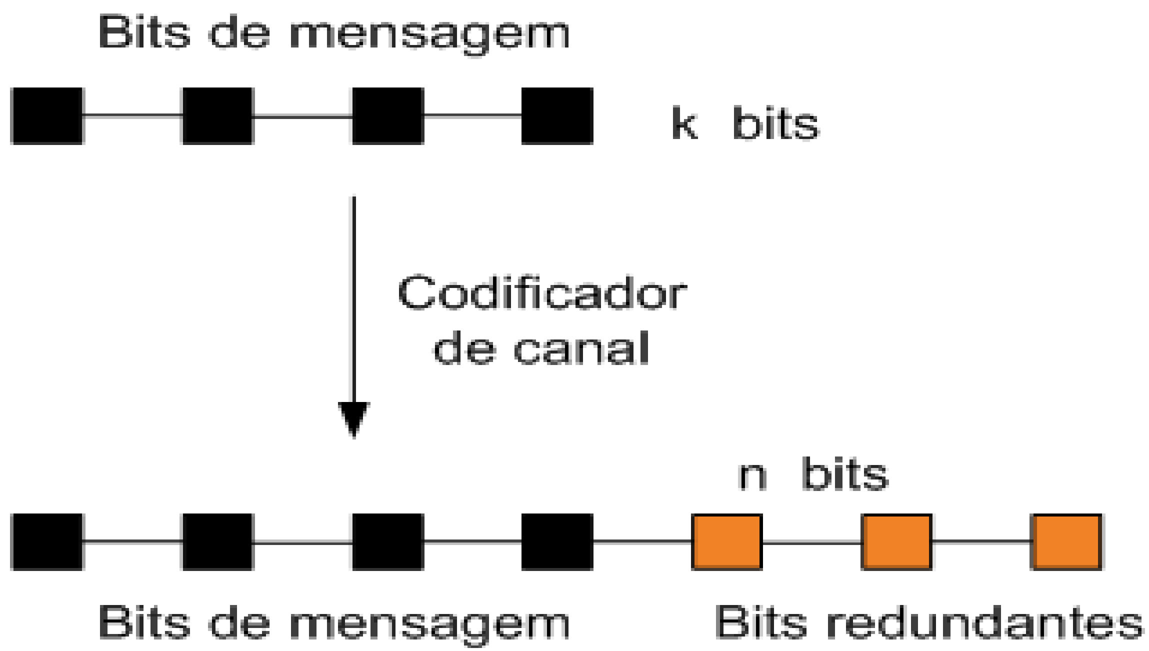


Modos de funcionamento

- FEC - **F**orward **E**rror **C**orrection
  - Modo correcção de erros
  - O receptor recebe as palavras de código, detecta eventuais erros e corrige-os
- ARQ - **A**utomatic **R**epeat Re**Q**uest
  - Modo de detecção de erros
  - O receptor recebe as palavras de código e detecta eventuais erros; em caso de erro, solicita a retransmissão

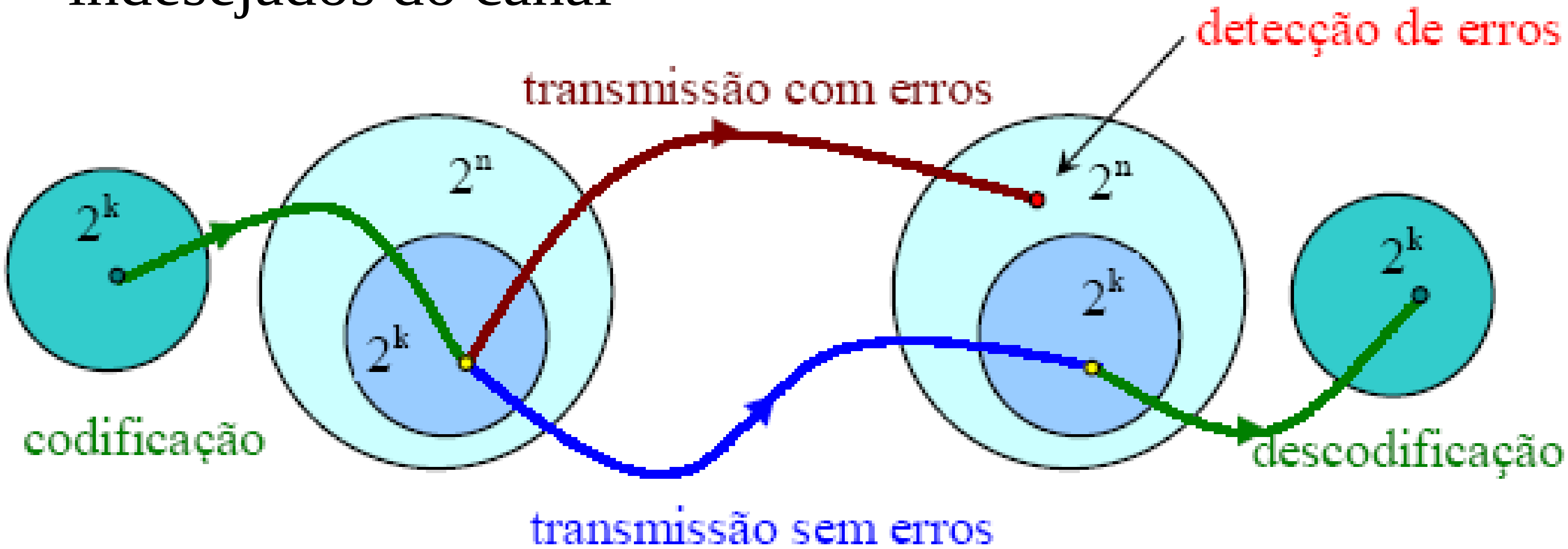
Codificador de canal (n,k)

- Codificador de bloco
- Cada bloco de k **bits de mensagem** origina uma **palavra de código** com n bits
- k = número de bits de mensagem
- n = número de bits de palavra de código



Objectivo da codificação de canal

- Aumentar a robustez do SCD relativamente aos efeitos indesejados do canal

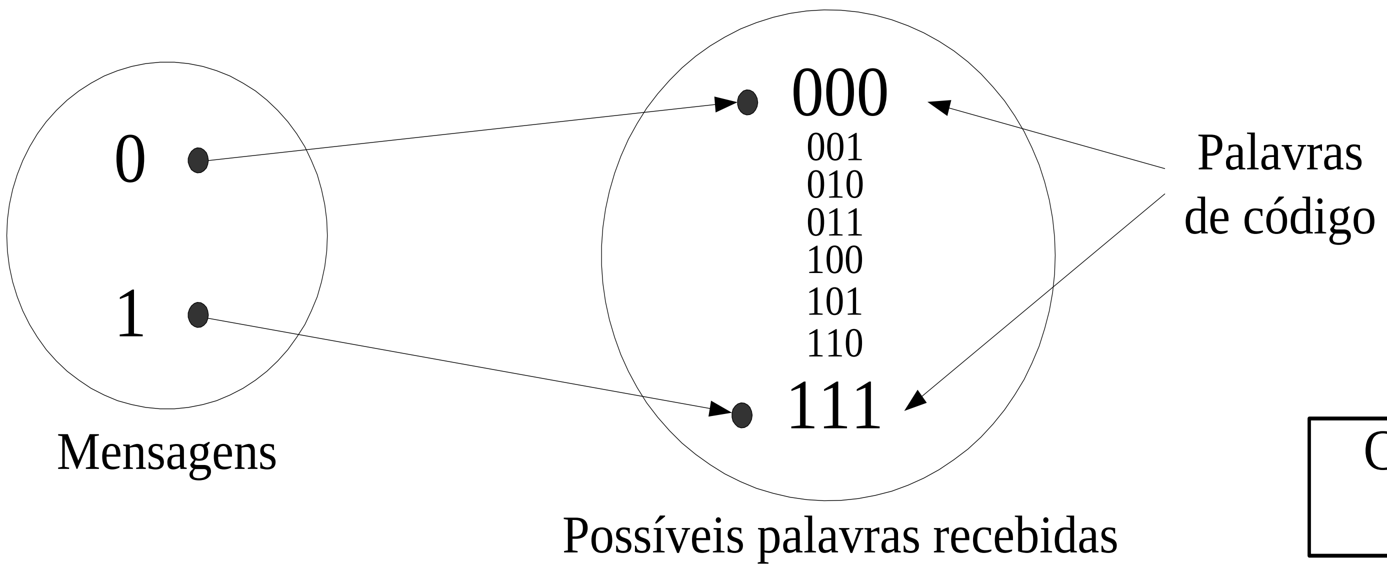


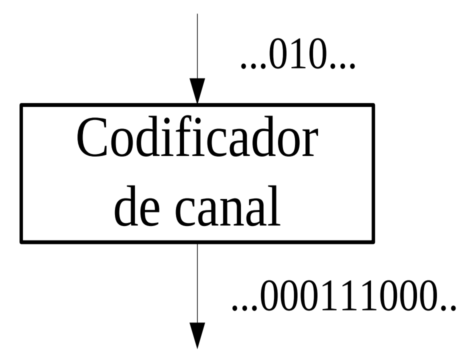
  - Cada bloco de k bits de mensagem dá origem a uma palavra de código com n bits
    - 2^k palavras de código no espaço de 2^n palavras

Código de repetição (3,1)

- Consiste na repetição da mensagem
- Exemplo: código (3,1), na forma (n,k) com k=1 bit de mensagem e n=3 bit na palavra de código

m	c
0	000
1	111

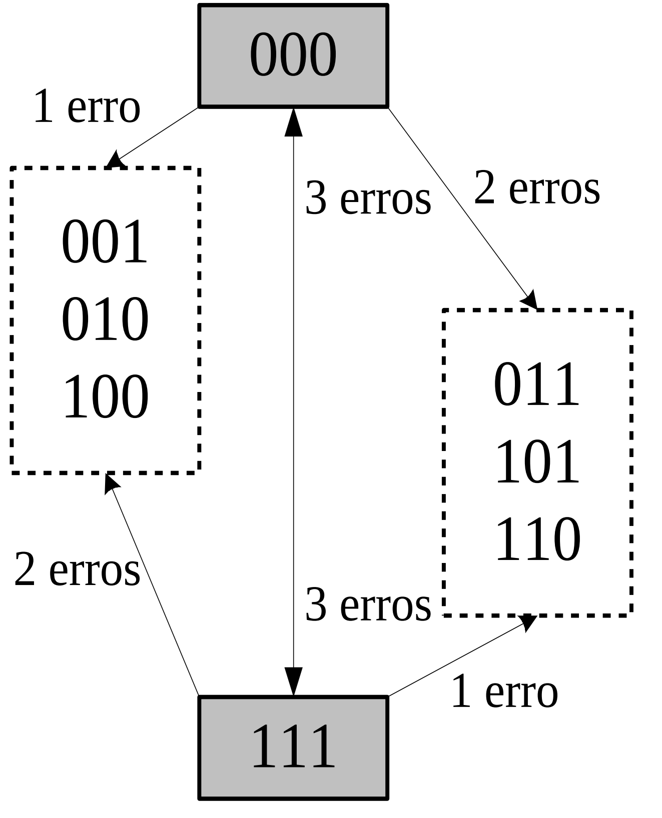




Usa $2^k = 2^1 = 2$ palavras de $2^n = 2^3 = 8$ possíveis

Código de repetição (3,1)

- Descodificação realizada por maioria
- A **distância** entre as palavras de código, garante que:
  - Detecta todos os erros de 1 e 2 bit
  - Corrige todos os erros de 1 bit



Considerando um BSC com $\alpha = 10^{-5}$,

tem-se que:

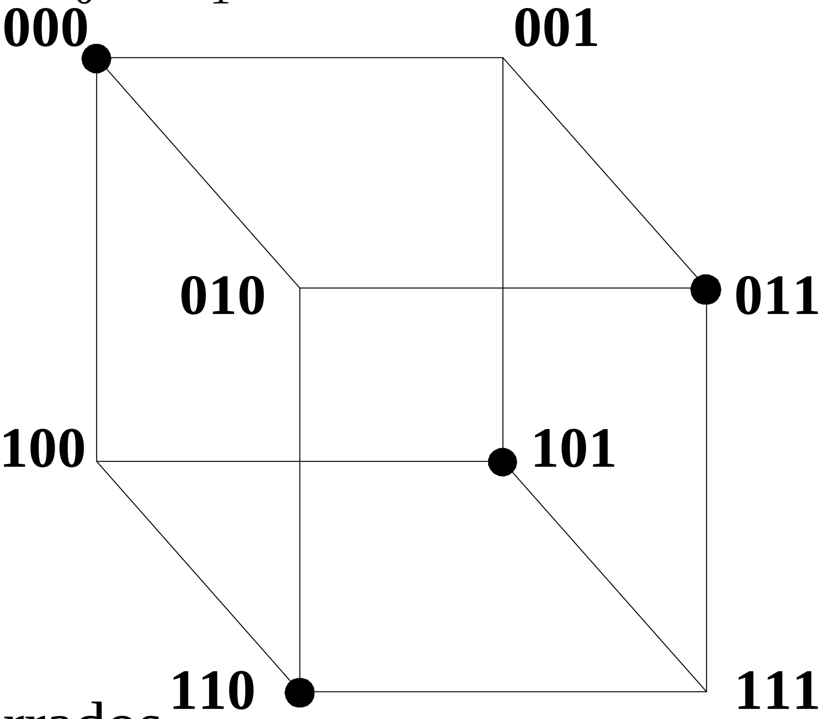
$$P(1,3) = C_1^3\ \alpha^1(1-\alpha)^2 = \frac{3!}{2!1!}\ \alpha(1-\alpha)^2$$
$$= 3\alpha - 6\alpha^2 + 3\alpha^3 \approx 3\times 10^{-5}$$

$$P(2,3) = C_2^3\ \alpha^2(1-\alpha)^1 = \frac{3!}{1!2!}\ \alpha^2(1-\alpha)$$
$$= 3\alpha^2 - 3\alpha^3 \approx 3\times 10^{-10}$$

Código bit de paridade (3,2) - paridade par

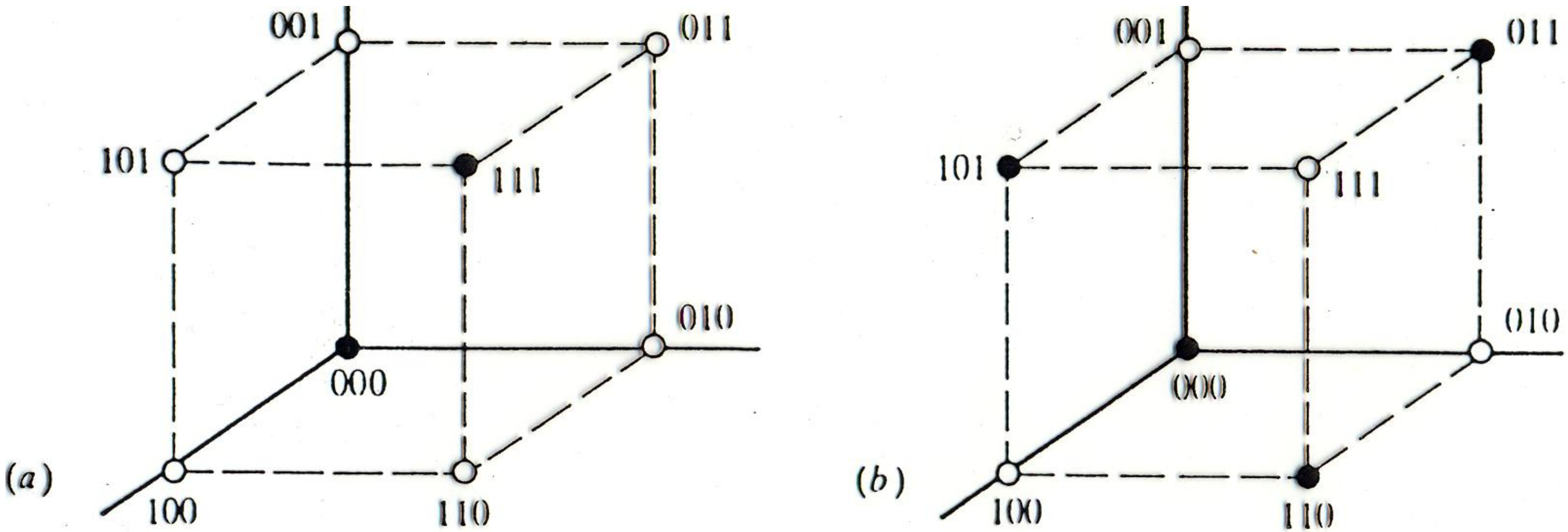
- Adicionar um bit no final da mensagem; este bit é a soma módulo 2 dos bits da mensagem
- A palavra de código é **c** $=[m_0\ m_1\ \ m_0 \oplus m_1]$

m	c
00	000
01	011
10	101
11	110



- <u>Detecta</u> a presença de 1 e 3 bits errados
- Não tem capacidade de <u>correcção</u>; não realiza FEC

Palavras de código: vectores



- Palavras de 3 bit
  - **(a)** código de repetição (3,1); 3 arestas entre as 2 palavras de código
  - **(b)** código de bit de paridade (3,2); 2 arestas entre 2 palavras de código mais próximas

Códigos de bloco (n,k): propriedades

1. *Code rate* (ritmo) $R = \frac{k}{n}$, medida de eficiência

2. Distância de Hamming (**dH**): número de dígitos em que diferem duas quaisquer palavras do código

3. **Distância mínima (dmin)**: é a menor distância de Hamming entre duas quaisquer palavras do código; depende da redundância: $\mathsf{dmin} \leq 1 + q, \quad q = n - k$

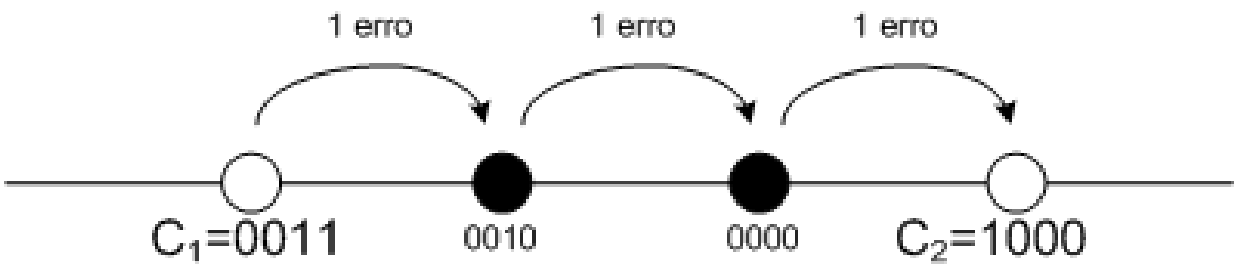
4. Detecta todos os padrões até "l" erros: $l \leq \mathsf{dmin} - 1$

5. Corrige todos os padrões até "t" erros: $t \leq \lfloor \frac{\mathsf{dmin}-1}{2} \rfloor$

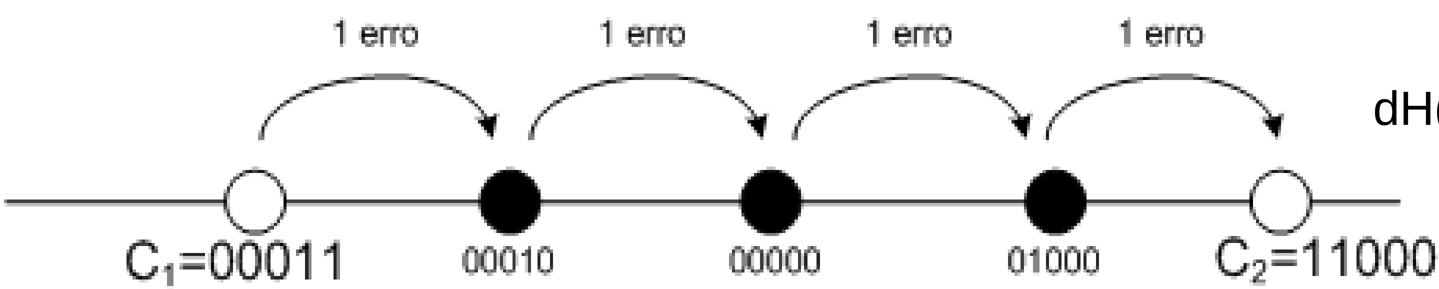
6. Detecta "l" erros e corrige "t" erros: $\mathsf{dmin} \geq l + t + 1, \quad \text{com } l > t$

Códigos de bloco (n,k): distância

- Distância de Hamming entre palavras



dH(c1,c2)=3



dH(c1,c2)=4

Códigos lineares: propriedades

- O desenho de códigos eficientes é um problema complexo: maximizar **dmin** com constrição ***R*** ou maximizar ***R*** com constrição **dmin**
- São problemas adicionais: <u>memória</u> ocupada e <u>complexidade</u> do codificador e do descodificador
- Através dos conceitos de estrutura algébrica e espaço vectorial definem-se os **códigos lineares** (elementos de sub-espaço vectorial)
- Os códigos lineares são um sub-conjunto de todos os códigos; requerem menos memória e existem codificadores e descodificadores simples

Códigos lineares de bloco (n,k)

- Bloco: todas as palavras têm a mesma dimensão
- Linear: o vector nulo pertence ao código; a soma modular de duas palavras do código é ainda uma palavra do código

---

n = número de bits da palavra de código — 2^n palavras possíveis
k = número de bits da mensagem — 2^k palavras de código
q = n – k, é o número de bits redundantes

Seja $\mathbf{m} = [m_0\ m_1\ \ldots\ m_{k-1}]$ a mensagem e **c** a palavra de código

Podem ser sistemáticos ou não sistemáticos; exemplos destas formas:

• sistemática: $\mathbf{c} = [m_0\ m_1\ \ldots\ m_{k-1}\ \ b_0\ b_1\ \ldots\ b_{q-1}]$

• não sistemática: $\mathbf{c} = [m_0\ b_1\ b_0\ m_1\ \ldots\ m_{k-1}\ \ldots\ b_{q-1}]$

Peso de Hamming

- Define-se peso de Hamming (**w**) como o número de dígitos não nulos numa palavra
- Sejam c_i e c_j duas palavras distintas de um código linear de bloco; tem-se por definição que
$$\mathrm{dmin} = \min_{i \neq j} dH(c_i, c_j)$$
- Dado que o código é linear, tem-se:

$$\mathrm{dmin} = \min \ \mathrm{w}(c_i \oplus c_j) = \min \ \mathrm{w}(c_k),$$

soma modular

sendo c_k palavra do código, diferente do vector nulo

Exemplos:

Código de repetição (3,1)

m	c	**w**(c)
0	000	0
1	111	3

$\mathrm{dmin} = 3$

$l = 2$

$t = 1$

Código de bit de paridade par (3,2)

m	c	**w**(c)
00	000	0
01	011	2
10	101	2
11	110	2

$\mathrm{dmin} = 2$

$l = 1$

$t = 0$

Matriz Geradora

- As palavras de código **c** são obtidas através do produto do vector mensagem **m** pela matriz geradora do código **G**

$$\mathbf{c} = \mathbf{m} \times \mathbf{G}$$

- **c** é vector de dimensões 1x n; **m** é vector 1x k
- **G** é matriz k x n; nos códigos sistemáticos temos
  - $\mathbf{G} = [\ \mathbf{I}_k \mid \mathbf{P}\ ]$ ou $\mathbf{G} = [\ \mathbf{P} \mid \mathbf{I}_k\ ]$ sendo **P** a **sub-matriz geradora de paridade**, ou seja, a matriz que estabelece as equações de paridade do código
  - Cada coluna de **P** constitui uma equação de paridade
  - **P** tem dimensões k x q

Matriz Geradora

- Exemplos de matrizes geradoras
- Código de repetição (3,1)

$$G = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

- Código de bit de paridade par (3,2)

$$G = \begin{bmatrix} I_2 & P \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad \text{ou} \quad G = \begin{bmatrix} P & I_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Códigos de Hamming

- Família de códigos lineares de bloco
- Têm dmin=3, logo corrigem todos os erros de 1 bit
- A motivação: $P(2,n) \ll P(1,n)$
- Definidos por um parâmetro inteiro m (≥ 2) tal que:

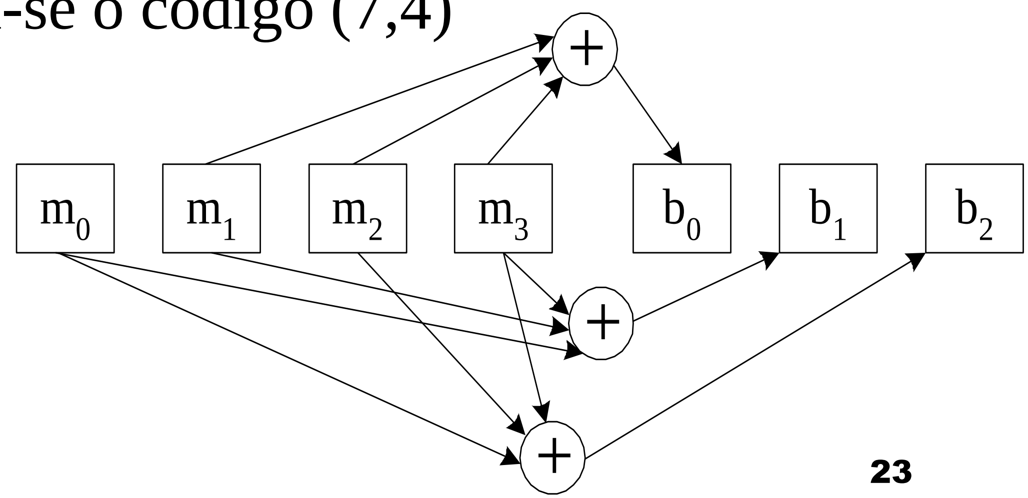
$$(n,k) = (2^m - 1, 2^m - 1 - m)$$

Por exemplo, com m=3 tem-se o código (7,4)

$\mathbf{c} = [m_0\ m_1\ m_2\ m_3\ \ b_0\ b_1\ b_2]$

Equações de paridade:

$$b_0 = m_1 \oplus m_2 \oplus m_3$$
$$b_1 = m_0 \oplus m_1 \oplus m_3$$
$$b_2 = m_0 \oplus m_2 \oplus m_3$$



Códigos de Hamming: forma matricial

$$\mathbf{c} = m\mathbf{G} = m\,[I_4 \mid P]$$

$$= \begin{bmatrix} m_0 & m_1 & m_2 & m_3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} m_0 & m_1 & m_2 & m_3 & b_0 & b_1 & b_2 \end{bmatrix}$$

- **G** é a matriz geradora do código
- Cada linha de **G** é uma palavra do código
- Todas as palavras do código são obtidas por combinação linear das linhas de **G**

- **G** é um conjunto de vectores linearmente independentes
- Gera 16 vectores de um total possível de 128; base de sub-espaço vectorial

Hamming (7,4): todas as palavras

Apresentam-se as 16 palavras de código e respectivos pesos de Hamming

Palavra de código							Peso	Palavra de código							Peso
0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	3
0	0	0	1	1	1	1	4	1	0	0	1	1	0	0	3
0	0	1	0	1	0	1	3	1	0	1	0	1	1	0	4
0	0	1	1	0	1	0	3	1	0	1	1	0	0	1	4
0	1	0	0	1	1	0	3	1	1	0	0	1	0	1	4
0	1	0	1	0	0	1	3	1	1	0	1	0	1	0	4
0	1	1	0	0	1	1	4	1	1	1	0	0	0	0	3
0	1	1	1	1	0	0	4	1	1	1	1	1	1	1	7

O menor peso de Hamming para palavras não nulas é 3, logo:

$$\text{dmin} = 3,\ \ l = 2 \text{ e } t = 1$$

Códigos de Hamming (características)

- Seja $\boldsymbol{k}$ o número de *bits* da mensagem a transmitir e $\boldsymbol{n}$ o número de *bits* efectivamente transmitidos
- Códigos de Hamming são códigos de bloco linear (n,k) onde:
  - $\boldsymbol{q} \geq 3$, sendo $q = n - k$ o número de *bits* redundantes
  - $\boldsymbol{n} = 2^q - 1$
  - para q = {1, 2, 3, ...}, temos então (7,4), (15,11), (31,26), ...
- A eficiência do código (*code rate*) é $\mathbf{r_c} = k/n = 1 - q/(2^q - 1)$
  - $\mathbf{r_c} \rightarrow 1$, se $q >> 1$
- $\mathbf{d_{min} = 3}$, independentemente de q
- A submatriz **P** geradora de paridade consiste em $\boldsymbol{k}$ linhas com $\boldsymbol{q}$ *bits*, com dois ou mais *bits* a **1** por linha

Cálculos em MATLAB

```
>> [H,G,n,k] = hammgen(3);
>> G
G =
  1   1   0     1   0   0
0
  0   1   1     0   1   0
0
  1   1   1     0   0   1
0
  1   0   1     0   0   0
1
>> n , k
n =
    7
k =
    4
```

m=3

- **G** está na forma sistemática

$$\mathbf{G} = [P \mid I_4]$$

- As equações de paridade são diferentes das apresentadas no exemplo anterior

Cálculos em MATLAB

```
>> m1 = [0 1 0 1]; m2 = [1 1 0 1];
>> c1 = mod ( m1*G, 2 )
c1 =
    1    1    0    0    1    0    1
>> c2 = mod ( m2*G, 2 )
c2 =
    0    0    0    1    1    0    1
```

- Entre as mensagens m1 e m2 muda apenas um bit; entre as palavras de código c1 e c2 mudam três bits
- c1 resulta da soma módulo 2 da segunda e quarta linhas de **G**
- c2 resulta da soma módulo 2 da primeira, segunda e quarta linhas de **G**

Descodificador de canal: características

- O descodificador:
  1. recebe a palavra **y** (possivelmente com erros)
  2. estima a palavra de código $\hat{x}$ que lhe deu origem
  3. estima a mensagem $\hat{m}$

$$\hat{x} = \arg\min_{x_i} dH(y, x_i), \forall x_i \in \mathcal{C}$$

$$\hat{x} \rightarrow \hat{m}$$

Um critério possível

- Funciona num dos modos:
  1. detecção
  2. correcção
  3. detecção e correcção

- Se a palavra recebida **y** não pertence ao código, houve erro(s)

Codificação / descodificação

- O codificador gera as palavras de código através da **matriz geradora G,** com

$$\mathbf{c = m \times G}$$

- No caso dos códigos sistemáticos temos $\mathbf{G = [\, I_k \mid P \,]}$ sendo **P** a **sub-matriz geradora de paridade**.
- A **matriz de controlo de paridade H** definida por $\mathbf{H = [\, P^T \mid I_{n-k} \,]}$ permite verificar se existem erros na palavra recebida **c,** através do cálculo do **síndroma** **(**conjunto de sintomas**)**

$$\mathbf{s = c \times H^T}$$

  - Caso **s** seja nulo, não se detectam erros
  - Caso contrário, existem erros detectados

Codificação / descodificação

- Codificação e descodificação matricial
- Na codificação temos

$$\mathbf{c = m \times G = [\ I_k \mid P\ ] = [m_o\ m_1\ \ldots\ m_{k\text{-}1}\quad b_o\ b_1\ \ldots\ b_{q\text{-}1}\ ],}$$

de forma a obter a concatenação *k bits mensagem* | *q bits de paridade*.

- Na descodificação é necessário obter os bits de mensagem, recalcular a paridade sobre estes e comparar com os bits de paridade enviados
- Para tal usa-se a **matriz de controlo de paridade H = [$\mathbf{P^T \mid I_{n\text{-}k}}$]** no cálculo do **síndroma**

$$\mathrm{s} = \mathrm{c}\mathrm{H}^{\mathrm{T}} = \mathrm{m}\,G\mathrm{H}^{\mathrm{T}} = m[I_k \quad P]\begin{bmatrix} P \\ I_q \end{bmatrix} = [s_0\, s_1 \cdots s_{q-1}]$$

Descodificação

- O **síndroma** é um vector de q bits (**H**$^{\mathbf{T}}$ tem dimensões n x q).
- Cada bit do síndroma corresponde à verificação da presença de erros no respectivo bit de paridade
- Na ausência de erros temos síndroma nulo porque GH$^{\mathrm{T}}$ são ortogonais

$$\mathrm{s} = \mathrm{c}\mathrm{H}^{\mathrm{T}} = \mathrm{m}\,G\mathrm{H}^{\mathrm{T}} = m[I_k \quad P]\begin{bmatrix} P \\ I_q \end{bmatrix} = [0\,0 \cdots 0]$$

- Erros são detectados sempre que o síndroma não é nulo
- O valor do síndroma só depende do padrão de erro **e**; não depende da palavra de código

$$\mathrm{s} = (\mathrm{c} + \mathrm{e})\mathrm{H}^{\mathrm{T}} = c\mathrm{H}^{\mathrm{T}} + e\mathrm{H}^{\mathrm{T}} = [0\,0 \cdots 0] + e\mathrm{H}^{\mathrm{T}} = e\mathrm{H}^{\mathrm{T}}.$$

Descodificação

- Cada padrão de 1 bit em erro, tem um síndroma único associado
- Sejam os padrões de erro
  - e1 = [1 0 .... 0], que corresponde ao primeiro bit errado
  - e2 = [0 1 .... 0], que corresponde ao segundo bit errado
- Para uma palavra de código **c** temos

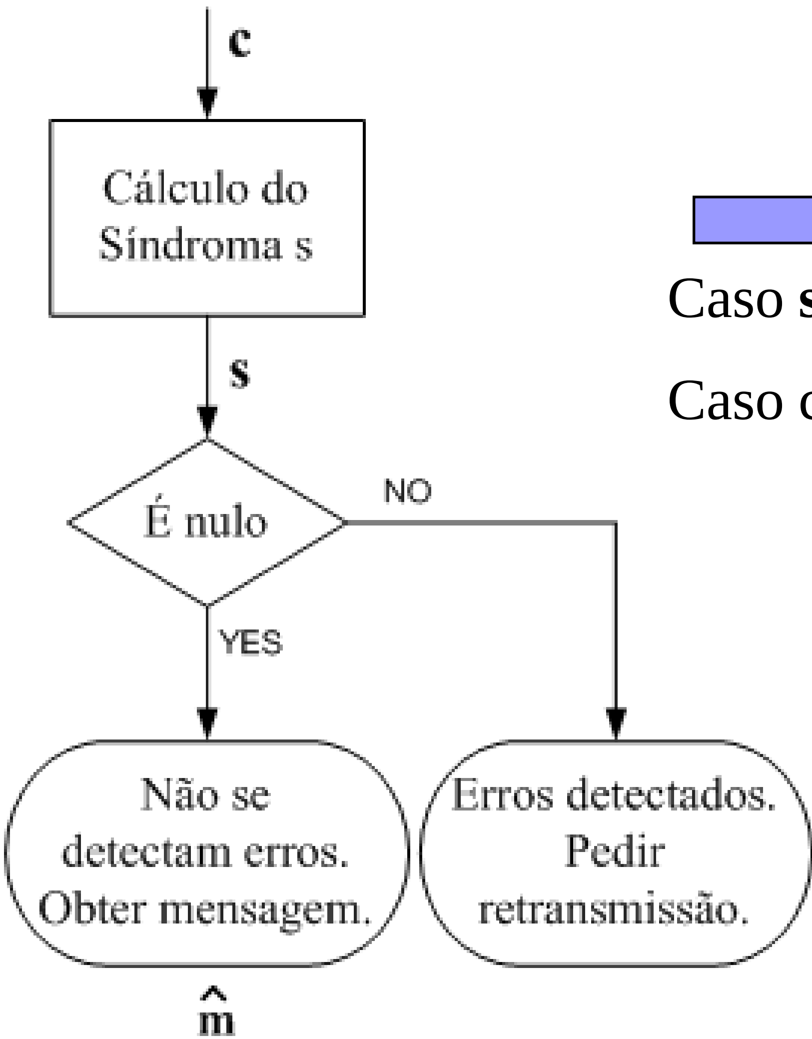
$$\mathrm{s}_1 = e_1 \mathrm{H}^{\mathrm{T}} = \text{primeira linha de } \mathrm{H}^{\mathrm{T}}.$$

$$\mathrm{s}_2 = e_2 \mathrm{H}^{\mathrm{T}} = \text{segunda linha de } \mathrm{H}^{\mathrm{T}}.$$

- As linhas de H^{T} são sempre <u>não nulas</u>

Descodificação: detecção

- Processo de descodificação em modo **detecção (ARQ)**



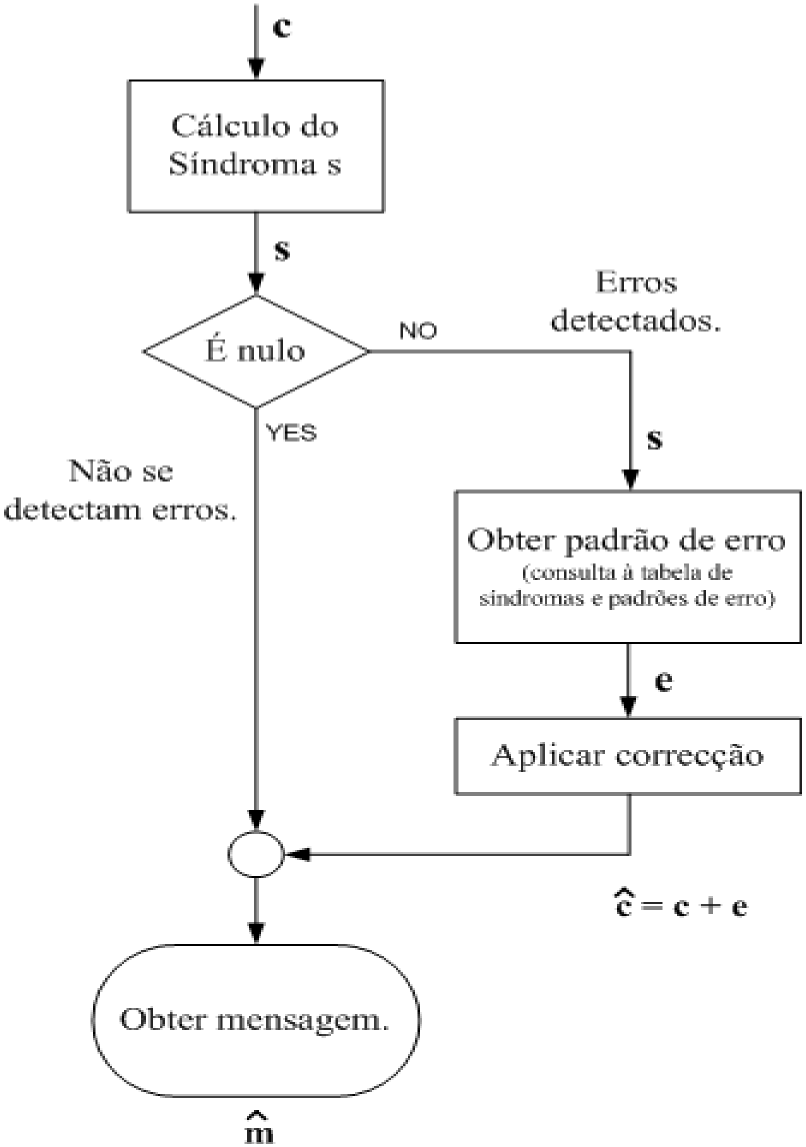
$\mathbf{s} = \mathbf{c} \times \mathbf{H}^T$

Caso **s** seja nulo, não se detectam erros

Caso contrário, existem erros detectados

É necessário que o SCD possibilite este pedido de retransmissão

Descodificação: correcção

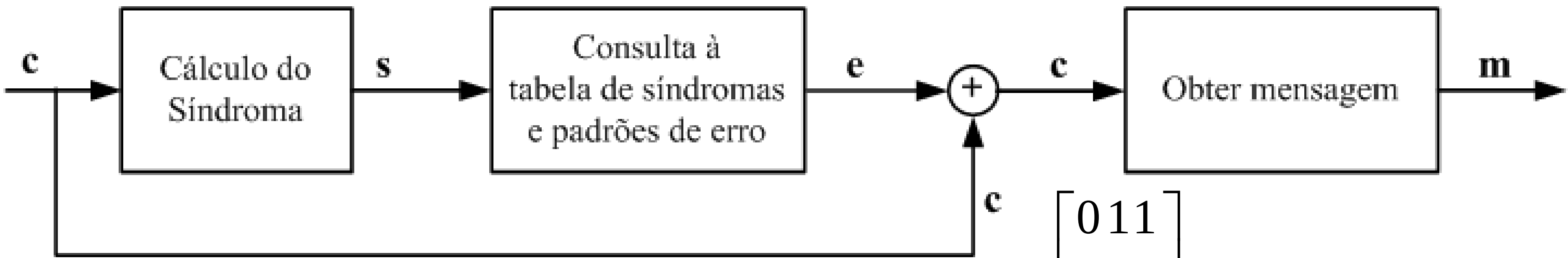


- Processo de descodificação em modo **correcção (FEC)**

Mecanismo de correcção

Descodificação: correcção

- Mecanismo de **correcção**: exemplo para o código Hamming (7,4)
- Matrizes geradora **G** e de teste de paridade $\mathbf{H}^{\mathbf{T}}$



$$G = \begin{bmatrix} 1\,0\,0\,0 & 0\,1\,1 \\ 0\,1\,0\,0 & 1\,1\,0 \\ 0\,0\,1\,0 & 1\,0\,1 \\ 0\,0\,0\,1 & 1\,1\,1 \end{bmatrix}$$

$$H^T = \begin{bmatrix} 0\,1\,1 \\ 1\,1\,0 \\ 1\,0\,1 \\ 1\,1\,1 \\ 1\,0\,0 \\ 0\,1\,0 \\ 0\,0\,1 \end{bmatrix}$$

Descodificação: correcção

- Tabela de síndromas para o código Hamming(7,4)
- O código tem $2^3=8$ síndromas: síndroma nulo - ausência de erro; os outros 7 correspondem aos padrões de um bit em erro por palavra

$$H^T = \begin{bmatrix} 011 \\ 110 \\ 101 \\ 111 \\ 100 \\ 010 \\ 001 \end{bmatrix}$$

Síndroma	Padrão de Erro	Observações
000	0000000	Ausência de erro
011	1000000	1.º bit em erro
110	0100000	2.º bit em erro
101	0010000	3.º bit em erro
111	0001000	4.º bit em erro
100	0000100	5.º bit em erro
010	0000010	6.º bit em erro
001	0000001	7.º bit em erro

Descodificação: correcção

s	e
000	0000000
011	1000000
110	0100000
101	0010000
111	0001000
100	0000100
010	0000010
001	0000001

- Sejam as palavras de código
  - $c_1 = [\mathbf{1\ 0\ 0\ 0}\quad 0\ 1\ 1]$
  - $c_2 = [\mathbf{0\ 0\ 1\ 1}\quad 0\ 1\ 0]$
- Sejam as palavras recebidas no descodificador
  - $y_1 = c_1 + \ [1\ 0\ 0\ 0\ 0\ 0\ 0] = [\underline{\mathbf{0}}\ 0\ 0\ 0\ 0\ 1\ 1]$
  - $y_2 = c_2 + \ [0\ 0\ 1\ 0\ 0\ 0\ 0] = [0\ 0\ \underline{\mathbf{0}}\ 1\ 0\ 1\ 0]$
  - $y_3 = c_1 + \ [1\ 1\ 0\ 0\ 0\ 0\ 0] = [\underline{\mathbf{0}}\ \underline{\mathbf{1}}\ 0\ 0\ 0\ 1\ 1]$
- Os síndromas obtidos são
  - $s_1 = y_1H^T = [0\ 1\ 1]$
  - $s_2 = y_2H^T = [1\ 0\ 1]$
  - $s_3 = y_3H^T = [1\ 0\ 1]$

Dois erros na palavra

Descodificação: correcção

s	e
000	0000000
011	**1000000**
110	0100000
101	**0010000**
111	0001000
100	0000100
010	0000010
001	0000001

- Os padrões de erro associados são
  - $e_1 = [1\ 0\ 0\ 0\ 0\ 0\ 0]$
  - $e_2 = [0\ 0\ 1\ 0\ 0\ 0\ 0]$
  - $e_3 = [0\ 0\ 1\ 0\ 0\ 0\ 0]$
- As palavras estimadas são
  - $c_1 = y_1 + e_1 = [\underline{\mathbf{0}}\ 0\ 0\ 0\ 0\ 1\ 1] + [1\ 0\ 0\ 0\ 0\ 0\ 0] = [1\ 0\ 0\ 0\ \ 0\ 1\ 1]$
  - $c_2 = y_2 + e_2 = [0\ 0\ \underline{\mathbf{0}}\ 1\ 0\ 1\ 0] + [0\ 0\ 1\ 0\ 0\ 0\ 0] = [0\ 0\ 1\ 1\ \ 0\ 1\ 0]$
  - $c_3 = y_3 + e_3 = [\underline{\mathbf{0}}\ \underline{\mathbf{1}}\ 0\ 0\ 0\ 1\ 1] + [0\ 0\ 1\ 0\ 0\ 0\ 0] = [0\ 1\ 1\ 0\ \ 0\ 1\ 1]$
- As mensagens obtidas após correcção
  - $m_1 = [1\ 0\ 0\ 0\ ]$
  - $m_2 = [0\ 0\ 1\ 1\ ]$
  - $m_3 = [0\ 1\ 1\ 0\ ]$

Os dois erros na palavra implicaram erro após correcção (t=1)

Comparação de códigos

Análise comparativa de códigos: ritmo e capacidades de detecção e correcção de erros.

Código	R = k/n	dmin	Detecta l	Corrige t
Repetição (2,1)	0.500	2	1	0
Repetição (3,1)	0.333	3	2	1
Repetição (4,1)	0.250	4	3	1
Repetição (5,1)	0.200	5	4	2
Paridade (3,2)	0.666	2	1	0
Paridade (8,7)	0.875	2	1	0
Hamming (7,4) m=3	0.571	3	2	1
Hamming (15,11) m=4	0.733	3	2	1
Hamming (31,26) m=5	0.838	3	2	1

Aplicações

- Comunicação série assíncrona
  - 1 bit de paridade por cada byte
- Memórias RAM
  - 1 bit de paridade por cada byte
  - mais do que 1 bit de paridade - ECC (*Error Correcting Code*) RAM
- Teletexto
  - Hamming (8,4) - extensão do Hamming (7,4)
- Discos rígidos
  - Alguns usam código de Hamming – existem bits de paridade por cada sector

Aplicações

- RAID (*Redundant Array of Independent Disks*)
  - RAID 1 – *mirroring*; código de repetição
  - RAID 2 - *Hamming system*; no caso do Hamming (7,4) usa 7 discos rígidos (4 dados + 3 paridade)
  - RAID 3 - *parallel transfer with parity drive*; usa código bit de paridade, no qual existem vários discos de dados e um de paridade

- *Bluetooth* (comunicação sem fios)
  - Usa código de repetição (3,1) - *packet header*
  - Usa Hamming modificado (15,10) - *application data*

Bibliografia

- Folhas de apoio
  - A. Ferreira, Códigos detectores e correctores de erros, disponível na página da unidade curricular http://www.deetc.isel.ipl.pt/sistemastele/cm/

- Livro
  - J. Moreira and P. Farrell, **Essentials of Error-Control Coding**, 2006, John Wiley and sons.