Détection d'erreur de transmission

M. Combacau - combacau@laas.fr

Université Paul Sabatier
LAAS-CNRS

10 novembre 2024

Objectif

Transmission d'information en informatique
Détection et correction d'erreur : codes de Hamming

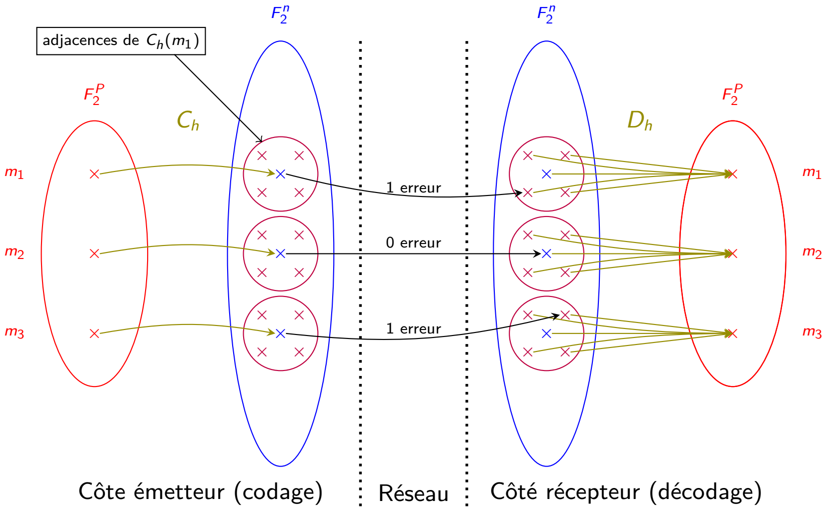
Distance de Hamming - Adjacence de deux codes

- Définie par
$$\begin{cases} (F_2^n)^2 \xrightarrow{dh} \mathbb{N} \\ (m_1, m_2) \in (F_2^n)^2 : dh(m_1, m_2) = \text{ nombre de bits à 1 dans } m_1 \oplus m_2 \end{cases}$$
- Nombre de bits de même rang ayant des valeurs différentes
- Exemple illustratif
$$\begin{cases} m_1 & = [01101] \\ m_2 & = [10110] \\ m_1 \oplus m_2 & = [11011] \end{cases} \Rightarrow d_h(m_1, m_2) = 4$$
- 2 mots m_1 et m_2 sont dits adjacents ssi $d_h(m_1, m_2) = 1$
- Un mot de n bits possède n mot adjacents
$m = [101]$ mots adjacents [001] [111] [100]

Principe d'un code de Hamming

- Soit la fonction codage $C_h : F_2^p \xrightarrow{c_h} F_2^n$
- Corriger une erreur,
  $\forall (m_1, m_2) \in Im(C_h)^2, m_1 \neq m_2 \Rightarrow d_h(m_1, m_2) \geq 3$
- Ainsi, le sous ensemble des codes adjacents à m_1 est disjoint du sous ensemble de codes adjacents à m_2
- La fonction décodage du récepteur est telle que tous les codes adjacents à un mot m ont pour image le code du mot m
- Ainsi, une erreur unique sur un code se traduira par un code adjacent à ce code, que la fonction décodage désigne par construction

Illustration du principe du code de Hamming



Nombre minimal de bits de contrôle

- Mots de p bits à transmettre : 2^p code sans erreurs et $2^p \times n$ codes adjacents (avec une erreur) nécessaires sur n bits
- Sur n bits, le mot codé à $n+1$ mots adjacents
- $2^p \times (n+1) \leq 2^n \Rightarrow$ le code correcteur existe
- $2^p \times (n+1) = 2^n \Rightarrow$ code **parfait**
- Les codes parfaits sont des codes de Hamming
- $2^p \times (n+1) = 2^n$
- Dans $\mathbb{N}$ cette égalité est possible pour $n+1 = 2^k$ dans ce cas, $n = 2^k - 1$ et $p = n - k$.

Liste des premiers codes parfaits

k	$n = 2^k - 1$	$p = n - k$	info	contrôle	**nom du code**
1	1	0	0	1	sans intérêt !
2	3	1	1	2	Code de répétition ou H(3,1)
3	7	3	4	3	code de Hamming : H(7,4)
4	15	11	11	4	code de Hamming : H(15,11)
5	31	26	26	5	code de Hamming : H(31,26)
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Remarque :

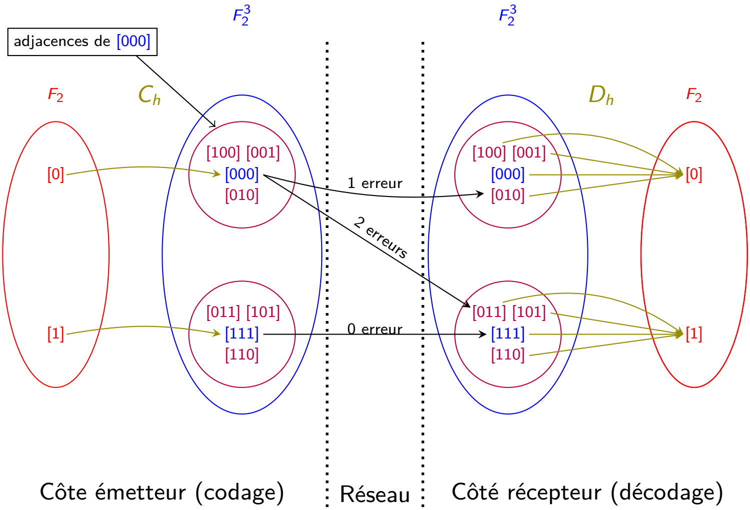
- Parité croisée 4×4 : 16 bit d'info + 8 bits de contrôle
- H(31,26) : 26 bits d'information, 5 bits de contrôle

Fonctionnement du code de Hamming H(3,1)

- Le seul pour lequel on peut donner une liste des codes utilisés !
- $\begin{cases} C_h(3,1)[0] = [000] \\ C_h(3,1)[1] = [111] \end{cases}$ d'où le nom "code de répétition"
- Décodage

  $$\begin{cases} D_h(3,1)[000] = D_h(3,1)[100] = D_h(3,1)[010] = D_h(3,1)[001] = [0] \\ D_h(3,1)[111] = D_h(3,1)[011] = D_h(3,1)[101] = D_h(3,1)[110] = [1] \end{cases}$$
- Coûteux : 2 bits de contrôle pour un bit d'information ! ! !

Fonctionnement du code de Hamming H(3,1)



Fonctionnement du code h(7,4) (1)

- 4 bit d'information et 3 bits de contrôle
- Génération des bits de contrôle

d_3	d_2	d_1	d_0	c_2	c_1	c_0
$\times$	$\times$	$\times$		$d_3 \oplus d_2 \oplus d_1$		
$\times$	$\times$		$\times$		$d_3 \oplus d_2 \oplus d_0$	
$\times$		$\times$	$\times$			$d_3 \oplus d_1 \oplus d_0$

- Changement d'un bit d_i changement d'au moins 2 bits c_j
  $\Rightarrow$ d_h entre deux codes est toujours au moins égale à 3
- Emetteur
  1. calcul des bits de contrôle $[c_2 c_1 c_0]$
  2. émission des septs bits $[d_i c_j]$ dans un ordre connu du récepteur

Fonctionnement du code h(7,4) (2)

- Récepteur : calcule le **syndrome**$=[p_2 p_1 p_0]$

$$\text{avec} \begin{cases} p_2 & = c_2 \oplus d_3 \oplus d_2 \oplus d_1 \qquad (c_2 = d_3 \oplus d_2 \oplus d_1) \\ p_1 & = c_1 \oplus d_3 \oplus d_2 \oplus d_0 \qquad (c_1 = d_3 \oplus d_2 \oplus d_0) \\ p_0 & = c_0 \oplus d_3 \oplus d_1 \oplus d_0 \qquad (c_0 = d_3 \oplus d_1 \oplus d_0) \end{cases}$$

- Absence d'erreur de transmission Syndrome [000]
- Une erreur sur un des 7 bits transmis

$$\begin{cases} d_3 \text{ faux} & \Rightarrow (p_2, p_1, p_0) = (1,1,1) \\ d_2 \text{ faux} & \Rightarrow (p_2, p_1, p_0) = (1,1,0) \\ d_1 \text{ faux} & \Rightarrow (p_2, p_1, p_0) = (1,0,1) \\ d_0 \text{ faux} & \Rightarrow (p_2, p_1, p_0) = (0,1,1) \\ c_2 \text{ faux} & \Rightarrow (p_2, p_1, p_0) = (1,0,0) \\ c_1 \text{ faux} & \Rightarrow (p_2, p_1, p_0) = (0,1,0) \\ c_0 \text{ faux} & \Rightarrow (p_2, p_1, p_0) = (0,0,1) \end{cases}$$

Syndrome différent pour chaque erreur

Mise en œuvre électronique (1)

- Rappel $\left\| \; x = a \oplus b \oplus c = a.b.c + \overline{a}.\overline{b}.c + \overline{a}.b.\overline{c} + a.\overline{b}.\overline{c} \right.$
- Emetteur : calcul des trois bits de contrôle

$$\left\{ \begin{array}{ll} c_2 & = d_3.d_2.d_1 + \overline{d_3}.\overline{d_2}.d_1 + \overline{d_3}.d_2.\overline{d_1} + d_3.\overline{d_2}.\overline{d_1} \\ c_1 & = d_3.d_2.d_0 + \overline{d_3}.\overline{d_2}.d_0 + \overline{d_3}.d_2.\overline{d_0} + d_3.\overline{d_2}.\overline{d_0} \\ c_0 & = d_3.d_1.d_0 + \overline{d_3}.\overline{d_1}.d_0 + \overline{d_3}.d_1.\overline{d_0} + d_3.\overline{d_1}.\overline{d_0} \end{array} \right.$$

- Rappel $\left\| \begin{array}{lll} x & = & a \oplus b \oplus c \oplus d \\ & = & \overline{a}.\overline{b}.\overline{c}.d + \overline{a}.\overline{b}.c.\overline{d} + \overline{a}.b.\overline{c}.\overline{d} + a.\overline{b}.\overline{c}.\overline{d} \\ & + & \overline{a}.b.c.d + a.\overline{b}.c.d + a.b.\overline{c}.d + a.b.c.\overline{d} \end{array} \right.$
- D'où le calcul du syndrome

$$\left\{ \begin{array}{ll} p_2 & = c_2 \oplus d_3 \oplus d_2 \oplus d_1 \\ & = \overline{c_2}.\overline{d_3}.\overline{d_2}.d_1 + \overline{c_2}.\overline{d_3}.d_2.\overline{d_1} + \overline{c_2}.d_3.\overline{d_2}.\overline{d_1} + c_2.\overline{d_3}.\overline{d_2}.\overline{d_1} \\ & +\overline{c_2}.d_3.d_2.d_1 + c_2.\overline{d_3}.d_2.d_1 + c_2.d_3.\overline{d_2}.d_1 + c_2.d_3.d_2.\overline{d_1} \\ p_1 & = c_1 \oplus d_3 \oplus d_2 \oplus d_0 \\ & = \overline{c_1}.\overline{d_3}.\overline{d_2}.d_0 + \overline{c_1}.\overline{d_3}.d_2.\overline{d_0} + \overline{c_1}.d_3.\overline{d_2}.\overline{d_0} + c_1\overline{d_3}.\overline{d_2}.\overline{d_0} \\ & +\overline{c_1}.d_3.d_2.d_0 + c_1\overline{d_3}.d_2.d_0 + c_1 d_3.\overline{d_2}.d_0 + c_1 d_3.d_2.\overline{d_0} \\ p_0 & = c_0 \oplus d_3 \oplus d_1 \oplus d_0 \\ & = \overline{c_0}.\overline{d_3}.\overline{d_1}.d_0 + \overline{c_0}.\overline{d_3}.d_1.\overline{d_0} + \overline{c_0}.d_3.\overline{d_1}.\overline{d_0} + c_0.\overline{d_3}.\overline{d_1}.\overline{d_0} \\ & +\overline{c_0}.d_3.d_1.d_0 + c_0.\overline{d_3}.d_1.d_0 + c_0.d_3.\overline{d_1}.d_0 + c_0.d_3.d_1.\overline{d_0} \end{array} \right.$$

Mise en œuvre électronique (2)

- Calcul des 4 mintermes cor_i pour corriger les bits d_i

$$\begin{cases} cor_3 & = p_2.p_1.p_0 \\ cor_2 & = p_2.p_1.\overline{p_0} \\ cor_1 & = p_2.\overline{p_1}.p_0 \\ cor_0 & = \overline{p_2}.p_1.p_0 \end{cases}$$

- Correction de $d_i \iff cor_i = 1$
- Et finalement le mot corrigé de son erreur

$$[d_3\ d_2\ d_1\ d_0] = [(cor_3 \oplus d_3)\ (cor_2 \oplus d_2)\ (cor_1 \oplus d_1)\ (cor_0 \oplus d_0)]$$

- Outil de simplification algébrique : forme somme de produits pour chacun des bits d'information corrigé
- Mise en œuvre électronique en $3 \times delta_t$ 🙂

Mise en œuvre informatique (1)

- $[p_2 p_1 p_0]$ coefficients du polynôme $S = p_2.2^2 + p_1.2 + p_0$
- Valeur de S en fonction du bit d'information erroné

bit erroné	valeur de S
d_3	7
d_2	6
d_1	5
d_0	3

- Emetteur : ordre des bits $[d_3 d_2 d_1 c_2 d_0 c_1 c_0]$
- Récepteur : $valeur(S) \neq 0 \Rightarrow$ complémenter le bit de rang $(valeur(S) - 1)$

L'ordre d'émission $[d_3 d_2 d_1 c_2 d_0 c_1 c_0]$ est toujours respecté dans H(7,4)

Cas de deux erreurs

- Le code erroné $dh = 1$ avec un autre mot du code
  Cet autre mot sera choisi par le décodage !
- Exemple, bits d_2 et d_1 erronés
  $p_1 = p_0 = 1$, $valeur(S) = 3$ correction du bit d_0 !
- Solution
  1. Emetteur : calculer la parite p du mot $[d_3 d_2 d_1 c_2 d_0 c_1 c_0]$
  2. Emetteur : émettre le mot $[d_3 d_2 d_1 c_2 d_0 c_1 c_0 p]$
  3. Récepteur :
     a. une erreur : parité globale des 8 bits $=1$
        $valeur(S)$ désigne le rang du bit à corriger
     b. parité globale des 8 bits $= 0$ et $valeur(S) \neq 0$
        2 erreurs : demander retransmission du mot

Réalisable en informatique et en électronique

—fin de cet enseignement Digital Data Processing—