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 : parité croisée

Principe de la parité croisée (1)

- Emission d'une trame de $n^2 + 2 \times n$ bits plutôt que n bits
- Explication sous forme matricielle

$$
\begin{array}{c|ccccccl}
p_0 & a_{0,n-1} & \dots & a_{0,j} & \dots & a_{0,0} & & \text{(mot 0)} \\
p_1 & a_{1,n-1} & \dots & a_{1,j} & \dots & a_{1,0} & & \text{(mot 1)} \\
\vdots & & \vdots & & \vdots & & & \\
p_{n-1} & a_{n-1,n-1} & \dots & a_{n,j} & \dots & a_{n-1,0} & & \text{(mot n-1)} \\
\hline
 & pc_{n-1} & \dots & pc_j & \dots & pc_0 & &
\end{array}
$$

Dans cette matrice

- le bit $a_{i,j}$ est le bit j du mot i
- les mots $[a_{i,n-1} \dots a_{i,0}]$ sont les mots d'information à transmettre
- le bits p_i est le mot de parité du mot i, c'est à dire $p_i = \bigoplus_{k=0}^{n-1} a_{i,k}$
- le bit pc_j est le bit de parité des bits $a_{i,j}$ c'est à dire : $pc_j = \bigoplus_{k=0}^{n-1} a_{k,j}$

Principe de la parité croisée (2)

L'émetteur

- Calcule les $2 \times n$ bits de parité et de parité croisée
- Opérations totalement combinatoires (réalisation électronique 🙂)
- Transmet les n mots d'information et les $2 \times n$ bits de parité et parité croisée dans une même **trame**

Le récepteur

- Reçoit la trame transmise et **connaît sa structure**
- Calcul de $Cp_i = p_i \oplus \bigoplus_{k=0}^{n-1} a_{i,k}$ (test de parité du mot i)
- Calcul de $Cc_j = pc_j \oplus \bigoplus_{k=0}^{n-1} a_{k,j}$ (test de parité des bits de rang j)

pas d'erreur de transmission $\Rightarrow \nexists (Cp_i, Cc_j)\ s.t.\ Cp_i.Cc_j = 1$

Exemple illustratif (1)

- Soit à transmettre quatre mots [1101] [0010] [1111] [0101]
- Emetteur : calcul des parités et parités croisées

1	1	1	0	1
1	0	0	1	0
0	1	1	1	1
0	0	1	0	1
	0	1	0	1

- Emetteur : transmission de la trame [110100101111010111000101]
- Récepteur : calcul des contrôles de parité Cp_i et Cc_j

test de parité Cp_i ↓					
0	1	1	1	0	1
0	1	0	0	1	0
0	0	1	1	1	1
0	0	0	1	0	1
		0	1	0	1
(test de parité croisée Cc_j) →		0	0	0	0

Pas d'erreur $\Rightarrow$ tous les Cp_i et $Cc_j = 0$

Exemple illustratif (2)

- Supposons une erreur sur un bit d'information
- Récepteur : calcul des contrôles de parité Cp_i et Cc_j

test de parité Cp_i	$\downarrow$					
	0	1	1	1	0	1
	1	1	0	0	0	0
	0	0	1	1	1	1
	0	0	0	1	0	1
			0	1	0	1
(test de parité croisée Cc_j)	$\rightarrow$		0	0	1	0

Une erreur unique $\Rightarrow$ $Cp_i = 1$ et $Cc_j = 1$ désignent le bit $a_{i,j}$

Tous les calculs sont des opérations combinatoires (XOR)

Propriétés de la parité croisée (1)

- Erreur unique sur un bit de parité ou parité croisée

$$\begin{array}{c|c|cccc}
0 & 1 & 1 & 1 & 0 & 1 \\
\color{red}{1} & \color{red}{0} & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 1 \\
\cline{2-6}
 & & 0 & 1 & 0 & 1 \\
\cline{2-6}
 & & 0 & 0 & 0 & 0
\end{array}
\qquad
\begin{array}{c|c|cccc}
0 & 1 & 1 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 1 \\
\cline{2-6}
 & & \color{red}{1} & 1 & 0 & 1 \\
\cline{2-6}
 & & \color{red}{1} & 0 & 0 & 0
\end{array}$$

- sur bit de parité (à gauche) : $\forall j \in [0, n-1], Cc_j = 0$
- sur bit de parité croisée (à droite) : $\forall i \in [0, n-1], Cp_i = 0$
- $\Rightarrow$ détection et correction possible, mais sans intérêt (l'information est correcte)

Propriétés de la parité croisée (2)

- Deux erreurs

0	1	1	1	0	1
1	1	1	0	1	0
0	0	1	1	1	1
1	0	0	1	1	1
		0	1	0	1
		1	0	1	0

0	1	1	1	0	1
1	1	1	0	1	0
0	0	1	1	1	1
1	0	1	1	0	1
		0	1	0	1
		0	0	0	0

$\Rightarrow$ Toujours au moins deux possibilités pour les erreurs (cas détecté mais non corrigé)

Propriétés de la parité croisée (3)

- Nombre impair d'erreurs sur une même colonne (ou une ligne)

$$
\begin{array}{c|c|cccc}
0 & 1 & 1 & 1 & 0 & 1 \\
1 & 1 & 1 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 & 0 & 1 \\
\cline{2-6}
 & & 0 & 1 & 0 & 1 \\
\cline{2-6}
\multicolumn{2}{c}{} & 1 & 0 & 0 & 0
\end{array}
\qquad
\begin{array}{c|c|cccc}
0 & 1 & 1 & 1 & 0 & 1 \\
1 & 1 & 1 & 0 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 1 & 0 & 1 \\
\cline{2-6}
 & & 0 & 1 & 0 & 1 \\
\cline{2-6}
\multicolumn{2}{c}{} & 1 & 0 & 0 & 0
\end{array}
$$

$\Rightarrow$ Toujours au moins deux possibilités pour les erreurs (cas détecté mais non corrigé)

- Raisonnement similaire sur les lignes (détecté mais non corrigé)

Propriétés de la parité croisée (4)

- Cas le plus simple d'erreurs multiples non détecté

$$\begin{array}{c|c|cccc} 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & \color{red}{1} & \color{red}{1} & 1 & 0 \\ 0 & 0 & \color{red}{0} & \color{red}{0} & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ \cline{2-6} & & 0 & 1 & 0 & 1 \\ \cline{2-6} & & 0 & 0 & 0 & 0 \end{array}$$

- Quatre erreurs dans des positions précises (aux quatre coins d'un rectangle)
- Probabilité extrèmement faible (10^{-28}) : risque acceptable !

Conclusions

- Simple à mettre en œuvre en informatique comme en électronique
- Corrige correctement une erreur quelle que soit sa localisation
- Détecte 2 ou 3 erreurs, probabilité d'occurrence très faible
- Ne détecte pas quatre erreurs "positionnées sur les coins d'un rectangle" probabilité d'occurrence quasi nulle
- Bon principe ? oui mais ...
  $2 \times n$ bits de contrôle pour 2^n bits d'information (peu efficace)
- Bien mieux : codes linéaires (Hamming vidéo suivante)