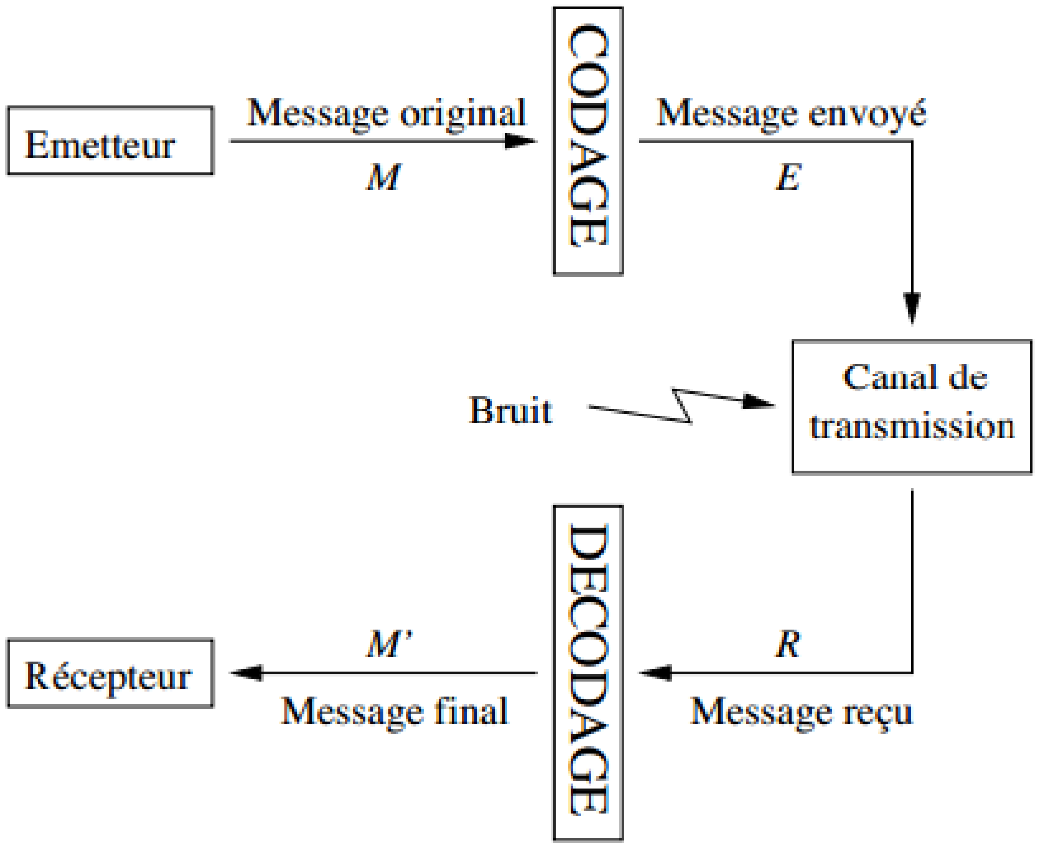
Détection d'erreurs

Schéma général de communication



Comment détecter et/ou corriger des erreurs ?

Un exemple simple :

Supposons qu'on veuille me transmettre un numéro de téléphone. Mon correspondant à deux possibilités :

1. Il me transmet 0169336000.
   S'il y a des erreurs de transmission, par exemple si je reçois 0167336010, je ne peux pas les détecter.
2. Il me transmet `zero un soixante-neuf trente-trois soixante zero zero`
   S'il y a des erreurs de transmission, par exemple si je reçois `zero an sogxante-seuf trepte-troisksoixanqe zoro zerb`, je suis capable de corriger les erreurs et de retrouver le numéro.

Exemples de redondance

- Comportement habituel au téléphone : répétition des numéros, épellation d'un nom ("I comme Irma, U comme Ursule, T comme Thérèse"...)
- Alphabet radio international : chaque lettre de l'alphabet est remplacée par un mot (Alpha Bravo Charlie Delta Echo . . .)
- L'orthographe traditionnelle comporte des redondances $\rightarrow$ moins sensible aux erreurs qu'une écriture phonétique (ou type SMS).

Le CRC

(Code Cyclique de Redondance)

- Principe général : représentation sous forme polynomiale de la suite de bits à transmettre

Exemple : la suite binaire **1100101** est représentée par le polynôme : $\mathbf{1}x^6 + \mathbf{1}x^5 + \mathbf{0}x^4 + \mathbf{0}x^3 + \mathbf{1}x^2 + \mathbf{0}x + \mathbf{1}$

$$= x^6 + x^5 + x^2 + 1$$

- Application du code lors de l'émission d'un mot :

  - Conversion d'un « polynôme générateur » en mot binaire

Exemple : avec le polynôme générateur **x^4+x^2+x**,

on obtient **10110**

- On ajoute *n* zéros au mot binaire à transmettre

(*n* est le degré du polynôme générateur)

Exemple : le mot à transmettre est **11100111**.

On obtient alors : **111001110000**

On ajoute à ce mot (111001110000) le mot correspondant au polynôme générateur (10110) jusqu’à ce que le mot obtenu soit inférieur au polynôme générateur.

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

Ce mot obtenu (**1110**) correspond au CRC à ajouter à la suite du mot avant de l'émettre.

Le mot à transmettre est donc : **11100111** **1110**.

• Vérification après réception du mot :
On ajoute le mot correspondant au polynôme générateur (10110) au mot reçu (11100111 1110)

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

Le résultat est nul, il n'y a donc pas d'erreur.

Codes correcteurs

- Détecte et corrige l’erreur

Le Code de Hamming

- k: bits de contrôle
- Longueur du message: m=(2^k-1)-k
- Longueur totale du message: n=2^k-1

k=3	m=4	n=7
k=4	m=11	n=15
k=5	m=26	n=31

7	6	5	4	3	2	1
1	0	1	**C2**	0	**C1**	**C0**

- C0 est calculé par rapport aux bits d'indice 7, 5, 3
- C1 est calculé par rapport aux bits d'indice 7, 6, 3
- C2 est calculé par rapport aux bits d'indice 7, 6, 5

Ainsi on a:

- C2=0
- C1=1
- C0=0

Le code reçu est donc: 1 0 1 0 0 1 0

7	6	5	4	3	2	1
1	0	1	0	0	1	0

$$
\begin{array}{lr}
7 & 0\,1\,1\,1 \\
5 & +\,0\,1\,0\,1 \\
2 & \underline{+\,0\,0\,1\,0} \\
 & 0\,0\,0\,0
\end{array}
$$

La somme de contrôle est égale à 0 => Message valide.

7	6	5	4	3	2	1
1	0	**0**	0	0	1	0

$$
\begin{array}{lr}
7 & 0\,1\,1\,1 \\
2 & +\,\underline{0\,0\,1\,0} \\
 & 0\,1\,0\,1
\end{array}
$$

La somme de contrôle n'est plus égale à 0=> Il y a une erreur.

0101=>5 L'erreur se trouve au bit n°5.