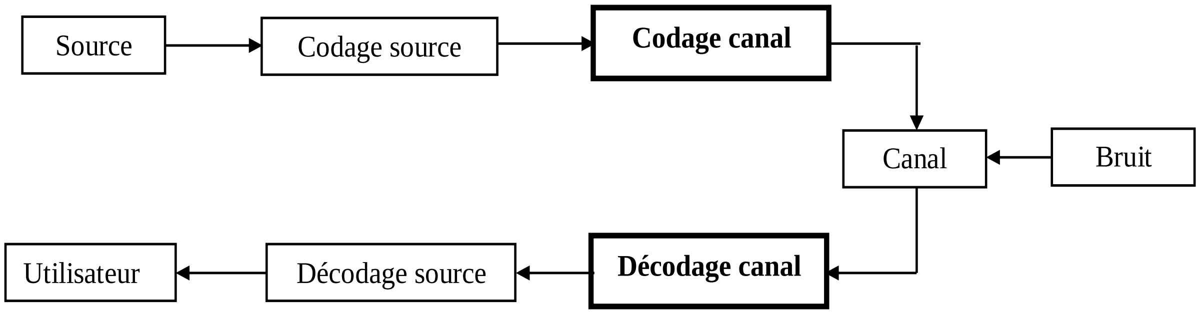
Chapitre

3

Codage Canal

I.1 Introduction

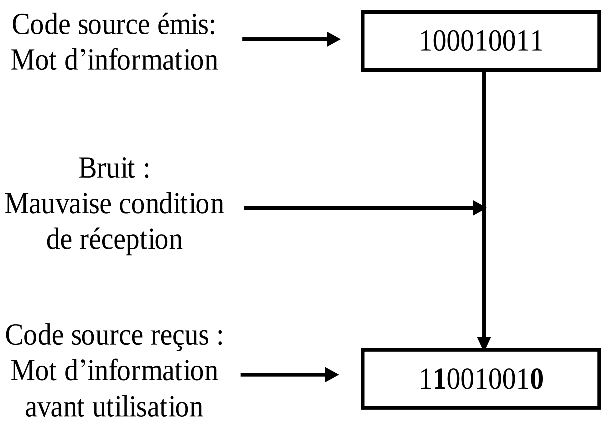
Cette partie s'intéresse à l'étude et la conception de code permettant de réaliser des transmissions fiables en présence de bruit dans un canal de communication.



Le codage canal de communication introduit systématiquement une redondance et donc une augmentation du flux de données en ajoutant des bits aux mots d'information. Ces bits ajoutés représentent les bits de contrôle qui permettent de détecter et de corriger les bits erronés afin d'améliorer la fiabilité de la transmission.
Le décodage du canal utilise toute la redondance du code pour connaitre les bits du message réellement transmis.

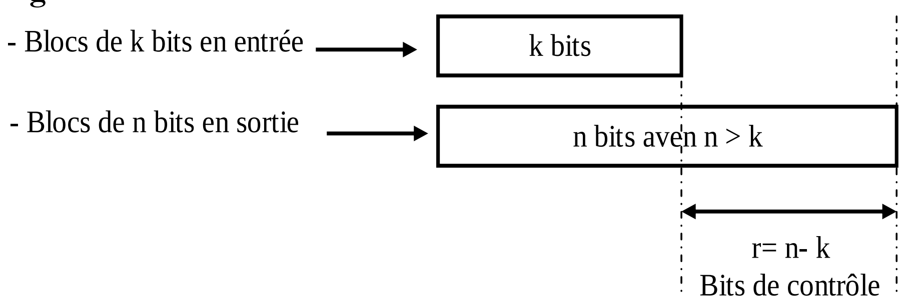
I.2 Principe du codage canal

Le principe du codage canal dans un système de communication numérique est de réduire la probabilité d'erreur par bit dans un canal bruité.



Le mot d'information reçus avant utilisation est différent du mot d'information émis par la source. Pour lutter contre ces erreurs causées par un canal bruité, nous allons développés une étude des codes correcteur d'erreurs. Ces codes correcteurs d'erreurs sont basés sur l'ajout des bits de redondance aux bits d'information et exploite cette redondance pour détecter et éventuellement corriger les erreurs.

Codage :



Décodage :
Le décodage permet de retrouver les blocs de k bits à partir des blocs de n bits.

I.2.1 Définitions liées au codage canal

Rendement
Le rendement d'un code représente le taux de codage qui est le rapport du nombre des bits des symboles utiles sur le nombre des bits des symboles utilisés.

$$\eta = \frac{k}{n} \quad \text{avec } k : \text{longueur des mots d'information}$$

$$n = \text{longueur des mots de code}$$

$$r = n - k : \text{bits de contrôle}$$

Poids de Hamming
Le poids de Hamming présente le nombre de bits non nuls que le mot de code possède.
Exemple : P(**1**0**1**00)=2
P(**1**0**11**000**1**)= 4

Distance de Hamming
La distance entre deux mots de code de même longueur au sens de Hamming présente le nombre de bits où les mots de code sont différents.
Exemple : D (1010**0**, 1010**1**)=1
D (101**1**00**0**1, 101**0**00**1**1)= 2

Distance minimale d'un code
La distance minimale d'un code présente la plus petite distance qu'on peut trouver entre tous les mots de code pris deux à deux.
Exemple : Soit une source discrète sans mémoire X avec trois symboles X_1, X_2 et X_3. On propose un codage canal $C(X_i)$ comme l'indique le tableau suivant :

X_i	Fréquences	$C(X_i)$
X_1	0.2	00
X_2	0.3	01
X_3	0.5	11

Déterminer la distance minimale de ce codage.
D (00, 01)=1
D (00, 11)=2
D (01, 11)= 1
La distance minimale de ce codage est d_{min}= 1

I.2.2 Notion de bit de parité

La plus part des codes utilisent la notion de bit de parité comme bits de contrôle afin d'assurer une transmission numérique efficace. Prenons un codage de source quelconque et ajoutons à chaque mot d'information un bit supplémentaire dit bit de parité formé par la somme ou exclusif de ces bits.

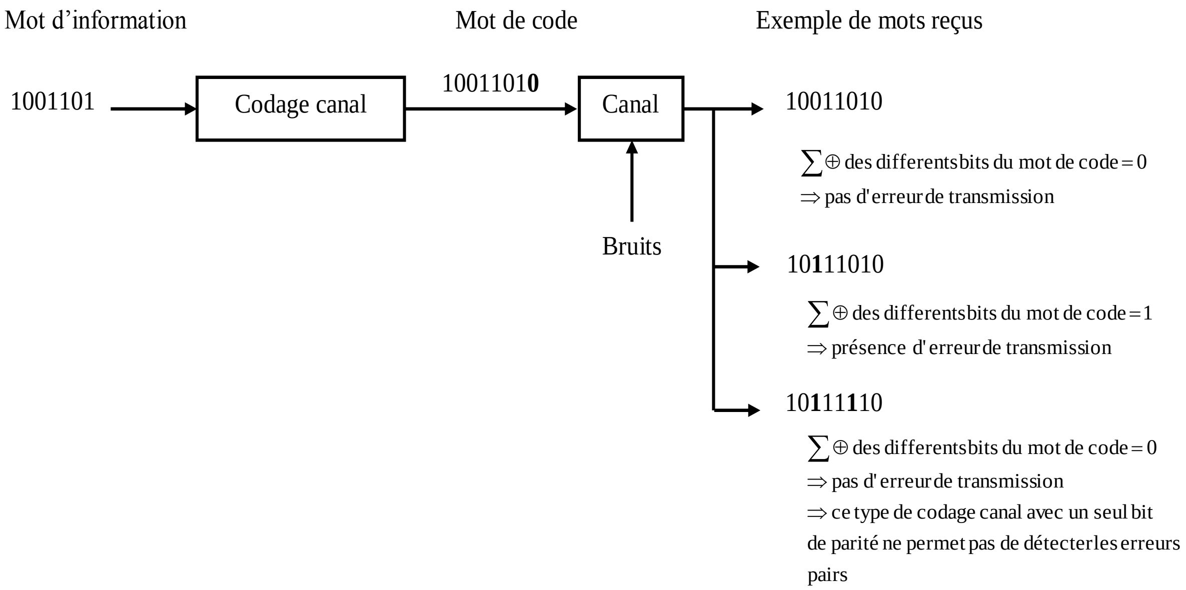
Mot d'information		Mot de code	
1010011	→	1010011**0**	Bit de parité
10100111	→	10100111**1**	

Chaque mot de code obtenu contiendra un nombre pair de « 1 ». Dans ce contexte, pour détecter une erreur d'un mot reçu il suffit de vérifier la parité à la réception.

$$- si \sum \oplus \text{ des differents bits du mot de code} = 0 \Rightarrow \text{pas d'erreur de transmission}$$

$$- si \sum \oplus \text{ des differents bit du mot de code} = 1 \Rightarrow \text{présence d'erreur de transmission}$$

Exemple :

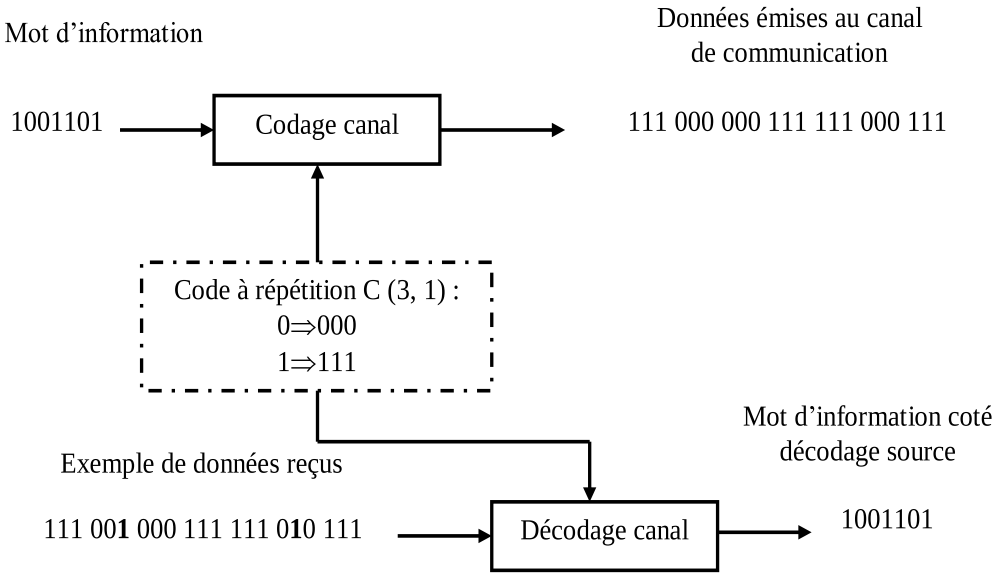


I.3 Différents types de codage canal

I.3.1 Code à répétition

Les codes binaires à répétition sont des codes R_K pour lesquels chaque bit du mot d'information est répété n=2K+1 fois avec K>1. Le décodage canal est assuré bit par bit à la majorité et on reconstitue le mot d'information émis. Le décodeur exploite donc cette redondance pour détecter et éventuellement corriger les erreurs.

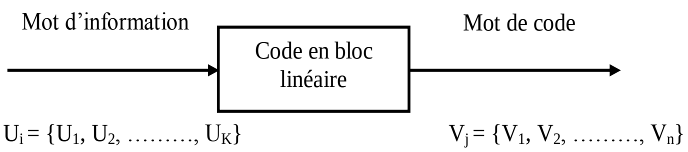
Exemple de code à répétition C (3, 1) :



Le codage canal par les codes à répétition est une solution couteuse car pour la même quantité d'information transmise il y'a plus de bits à transmettre par unité de temps. Les codes à répétition présente donc un rendement très faible. Pour le code C (3, 1) le rendement est de 1/3.

I.3.2 Code en bloc linéaire

Le code en bloc linéaire est considéré comme une généralisation de la notion de bit de parité. Dans ces codes on calcul les bits de parité sur des sous-ensembles des bits d'information.



Avec n > K
k : longueur des mots d'information
n : longueur des mots de code
r=n-k : longueur des bits de parité

Avec k bits on a 2^k combinaisons possible $\Rightarrow 2^k$ mots d'information. On notera le code en bloc linéaire C (n, k), l'association de 2^k mots de code de n bits aux 2^k mots d'information de 2^k bits. Dans la suite on s'intéresse à étudier les codes en bloc linéaire systématique. Un code en bloc linéaire est dit systématique si les k premiers bits du mot de code sont constitués par les k bits du mot d'information. Ces k bits sont dits systématiques et les (n-k) bits restants sont appelés bits de parité.

I.3.2.1 Matrice génératrice des codes linéaires systématiques

Dans un code linéaire par bloc systématique, nous pouvons définir un vecteur donnée U et un vecteur sortie V. U= $\{U_1, U_2, \ldots\ldots\ldots, U_k\}$ et V=$\{V_1, V_2, \ldots\ldots\ldots, V_n\}$

La matrice qui permet de générer les mots de code à partir des mots d'information appelée matrice génératrice est définie comme suit :

$$G = \left[I_K P^T \right]$$

Avec I_k matrice identitée d'ordre k

$$P = \begin{bmatrix} P_{11} & P_{12} & . & . & . & P_{1k} \\ P_{21} & P_{22} & . & . & . & P_{2k} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ P_{m1} & P_{m2} & . & . & . & P_{mk} \end{bmatrix} \text{matrice de parité} \Rightarrow P^T = \begin{bmatrix} P_{11} & P_{21} & . & . & . & P_{m1} \\ P_{12} & P_{22} & . & . & . & P_{m2} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ P_{1k} & P_{2k} & . & . & . & P_{mk} \end{bmatrix}$$

$$\Rightarrow G = \begin{bmatrix} 1 & 0 & . & . & 0 & P_{11} & P_{21} & . & . & P_{m1} \\ 0 & 1 & . & . & . & P_{12} & P_{22} & . & . & P_{m2} \\ . & . & . & . & . & . & . & . & . & . \, . \\ . & . & . & . & . & . & . & . & . & . \\ 0 & 0 & . & . & 1 & P_{1k} & P_{2k} & P_{2k} & . & P_{mk} \end{bmatrix}$$

Codage :

$$V = U * G$$

$$\Rightarrow V = \left[U_1, U_2 \ldots\ldots\ldots\ldots, U_k \right] * \begin{bmatrix} 1 & 0 & . & . & 0 & P_{11} & P_{21} & . & . & P_{m1} \\ 0 & 1 & . & . & . & P_{12} & P_{22} & . & . & P_{m2} \\ . & . & . & . & . & . & . & . & . & . \, . \\ . & . & . & . & . & . & . & . & . & . \\ 0 & 0 & . & . & 1 & P_{1k} & P_{2k} & P_{2k} & . & P_{mk} \end{bmatrix}$$

I.3.2.2 Matrice de test de parité des codes systématiques

Etant donné un mot reçu m, la question qui se pose comment peut-on connaitre s'il appartient ou non au code générer par la matrice génératrice.

La matrice qui permet de détecter les erreurs appelée matrice de test de parité H de dimension [r x n] est définie comme suit :

$H = [PI_r]$

Avec I_r matrice identitée d'ordre r

On a $V = U * G$

$\Rightarrow V * H^T = U * G * H^T$

$\Rightarrow V * H^T = U * [I_k P^T] * \begin{bmatrix} P^T \\ I_r \end{bmatrix}$

$V * H^T = U * [P^T \oplus P^T]$

$V * H^T = [0\,0\,0 ....0]$ matrice nulle

Donc tout mot de code multiplier par le transposé de la matrice de test de parité est un vecteur nul. Dans ce contexte pour contrôler l'appartenance du code d'un mot reçu il suffit de calculer le syndrome S définie par : S = m * H^T

$- si\ S = \vec{0} \Rightarrow$ Le mot recçu est un mot de code

$- si\ S \neq \vec{0} \Rightarrow$ Le mot recçu présente des erreurs de transmission

Exercice d'application

Pour un code systématique linéaire par bloc C (5,3) de matrice génératrice G = $[d_1d_2d_3C_1C_2]$ et avec les bits des vecteurs de test de parité c_1 et c_2 sont formés à partir des équations suivantes :

$C_1 = d_1 \oplus d_2$

$C_2 = d_1 \oplus d_3$

1- Déterminer la matrice génératrice de ce code.
2- Construire tous les mots de code possibles.
3- Déterminer la matrice de test de parité H.
4- En supposant la réception du mot m_1 = [11100], décoder ce mot reçu en localisant l'erreur et les bits transmis.
5- En supposant la réception du mot m_2 = [11111], décoder ce mot reçu en localisant l'erreur et les bits transmis.

Correction

1- Matrice génératrice du code C (5, 3)

$G = [I_k P^T]$

Longueur des mots d'information k=3

Longueur des mots de code n=5

Nombre de bits de redondance r=2

$k = 3 \Rightarrow I_k = I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$\Rightarrow G = \begin{bmatrix} 1 & 0 & 0 & P_{11} & P_{21} \\ 0 & 1 & 0 & P_{12} & P_{22} \\ 0 & 0 & 1 & P_{13} & P_{23} \end{bmatrix}$

On a $C_1 = d_1 \oplus d_2$

$\Rightarrow P_{11}=1\oplus0=1$, $P_{12}=0\oplus1=1$ et $P_{13}=0\oplus0=0$

On a $C_2 = d_1 \oplus d_3$

$\Rightarrow P_{21}=1\oplus0=1$, $P_{22}=0\oplus0=10$ et $P_{13}=0\oplus1=1$

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

2- Mots de code :

k= 3 $\Rightarrow 2^3=8$ mots d'information U={U_1, U_2, U_3, U_4, U_5, U_6, U_7, U_8}

Message	U1	U_2	U_3	U_4	U_5	U_6	U_7	U_8
Mot d'information	000	001	010	011	100	101	110	111

On a $V=U*G$

$$\Rightarrow V=[U_1,U_2..........,U_8]*\begin{bmatrix}1&0&0&1&1\\0&1&0&1&0\\0&0&1&0&1\end{bmatrix}$$

$$\Rightarrow V_1=U_1*G$$

$$\Rightarrow V_1=[000]*\begin{bmatrix}1&0&0&1&1\\0&1&0&1&0\\0&0&1&0&1\end{bmatrix}=[00000]\Rightarrow V_1=[00000]$$

$$\Rightarrow V_2=U_2*G$$

$$\Rightarrow V_2=[001]*\begin{bmatrix}1&0&0&1&1\\0&1&0&1&0\\0&0&1&0&1\end{bmatrix}=[00101]\Rightarrow V_2=[00101]$$

$$\Rightarrow V_3=U_3*G$$

$$\Rightarrow V_3=[010]*\begin{bmatrix}1&0&0&1&1\\0&1&0&1&0\\0&0&1&0&1\end{bmatrix}=[01010]\Rightarrow V_3=[01010]$$

$$\Rightarrow V_4=U_4*G$$

$$\Rightarrow V_4=[011]*\begin{bmatrix}1&0&0&1&1\\0&1&0&1&0\\0&0&1&0&1\end{bmatrix}=[01111]\Rightarrow V_4=[01111]$$

$$\Rightarrow V_5=U_5*G$$

$$\Rightarrow V_5=[100]*\begin{bmatrix}1&0&0&1&1\\0&1&0&1&0\\0&0&1&0&1\end{bmatrix}=[10011]\Rightarrow V_5=[10011]$$

$$\Rightarrow V_6=U_6*G$$

$$\Rightarrow V_6=[101]*\begin{bmatrix}1&0&0&1&1\\0&1&0&1&0\\0&0&1&0&1\end{bmatrix}=[10110]\Rightarrow V_6=[10110]$$

$$\Rightarrow V_7=U_7*G$$

$$\Rightarrow V_7=[110]*\begin{bmatrix}1&0&0&1&1\\0&1&0&1&0\\0&0&1&0&1\end{bmatrix}=[11001]\Rightarrow V_7=[11001]$$

$$\Rightarrow V_8 = U_8 * G$$

$$\Rightarrow V_8 = [111] * \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} = [11100] \Rightarrow V_8 = [11100]$$

3- Matrice de test de parité H:

$$H = [PI_r]$$

Avec :

$$P = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad \text{et } I_r = I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\Rightarrow H = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

4- Décodage du mot reçu m_1=[11100] :

Calcule de syndrome S :

S= m * H^T

$$On \text{ a } H = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix} \Rightarrow \mathrm{H}^\mathrm{T} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathrm{S}_1 = \mathrm{m}_1 * \mathrm{H}^\mathrm{T} \Rightarrow S_1 = [11100] * \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = [0 \quad 0]$$

$S_1 = \vec{0}$ $\Rightarrow$ Pas d'erreur de transmission

5- Décodage du mot reçu m_2=[11111] :

Calcule de syndrome S :

$$\mathrm{S}_2 = \mathrm{m}_2 * \mathrm{H}^\mathrm{T} \Rightarrow S_2 = [11111] * \begin{bmatrix} \boxed{1 \quad 1} \\ 1 \quad 0 \\ 0 \quad 1 \\ 1 \quad 0 \\ 0 \quad 1 \end{bmatrix} = \boxed{[1 \quad 1]}$$

$S_1 \neq \vec{0}$ $\Rightarrow$ Présence d'erreur de transmission dans le premier bit donc le mot transmis est [01111]. Ce mot de code est réellement le vecteur V_4.