



Codes LDPC

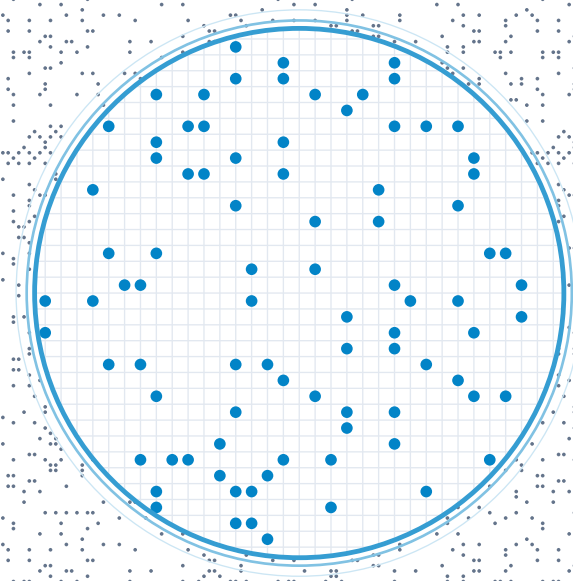
Anthony PERRONI

n°49871

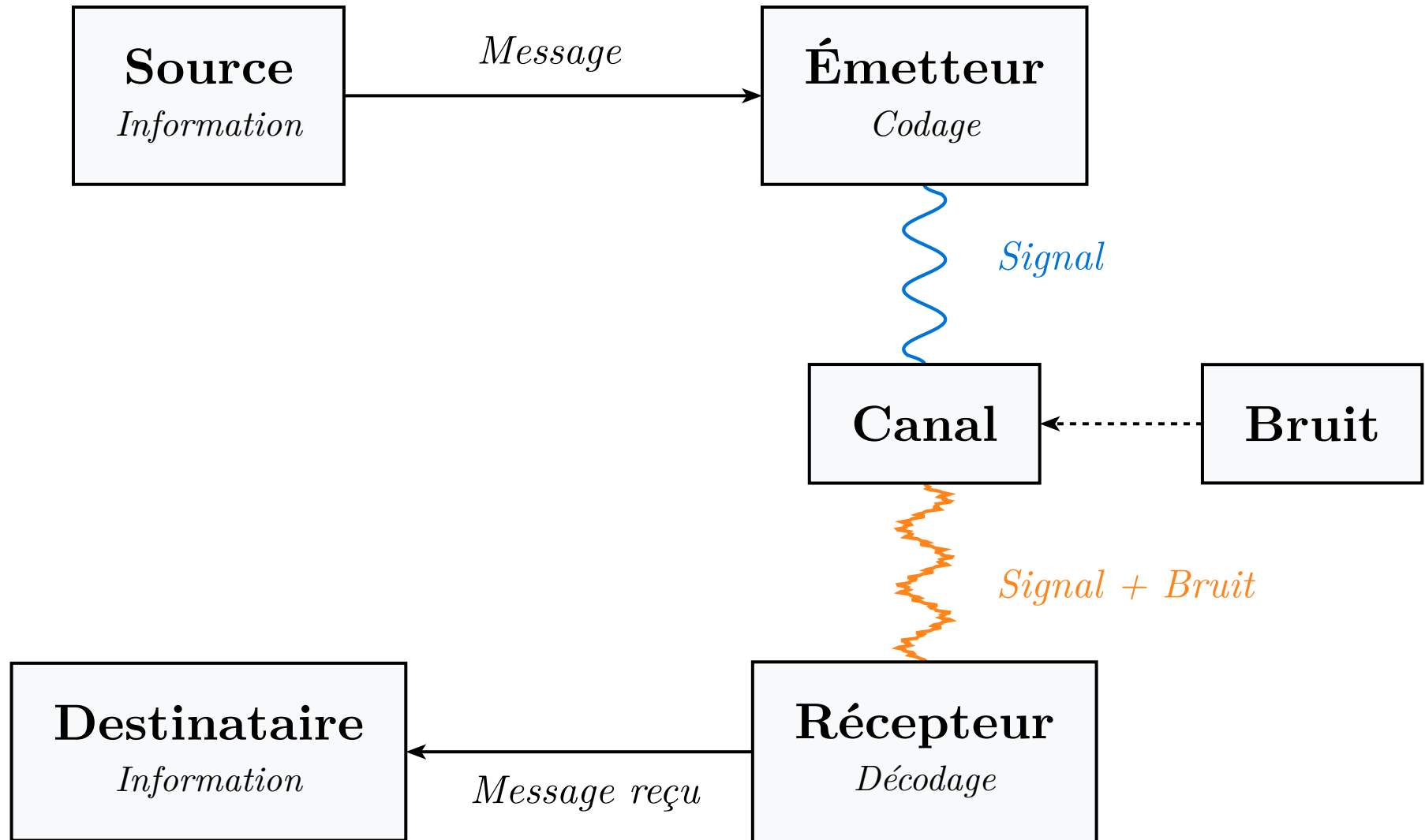
2025 - 2026

Plan

- Introduction
- Codes linéaires
- LDPC
- Codage
- Décodage
- Analyse



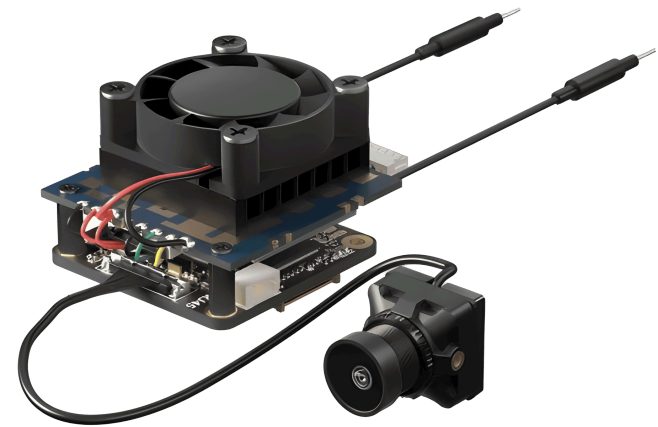
Introduction : Communication Numérique



Introduction : Utilisation



Athena-Fidus



Module OpenIPC

Problématique

Comment utiliser les codes LDPC pour garantir la fiabilité d'une transmission en présence de bruit ?

Définition : Codes Linéaires en Bloc

Code $(n, k) \in \mathbb{N}^2$

\mathcal{C} sous-espace vectoriel de dimension k de \mathbb{F}_2^n

- k : longueur du message original
- n : longueur du mot de code
- $m = n - k$: nombre de bits de parités

Encodage

$\Phi : \mathbb{F}_2^k \rightarrow \mathbb{F}_2^n \in \mathcal{L}(\mathbb{F}_2^k, \mathbb{F}_2^n)$

Définition : Codes Linéaires en Bloc

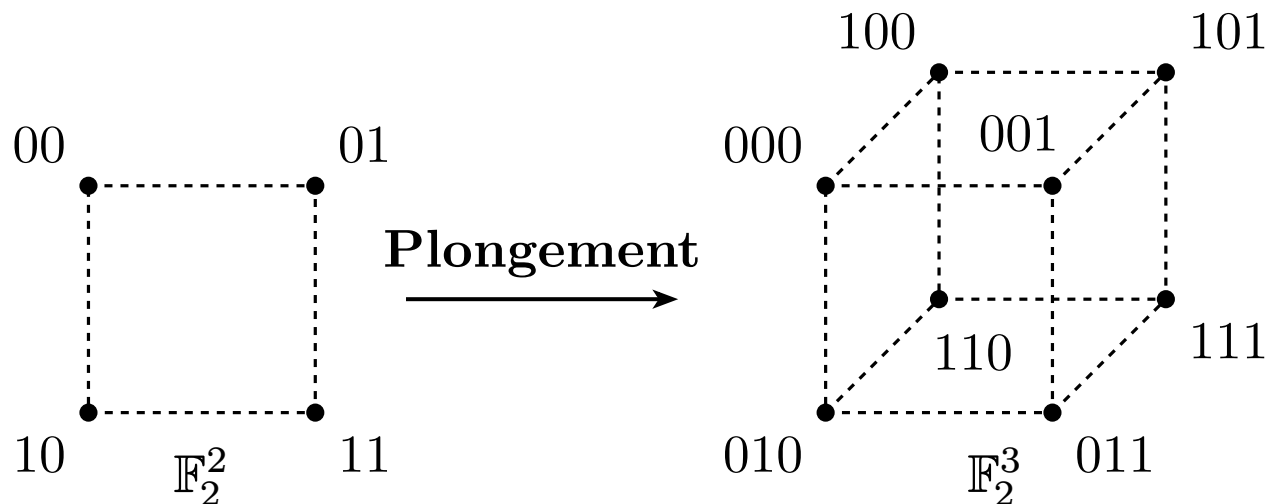
Code $(n, k) \in \mathbb{N}^2$

\mathcal{C} sous-espace vectoriel de dimension k de \mathbb{F}_2^n

- k : longueur du message original
- n : longueur du mot de code
- $m = n - k$: nombre de bits de parités

Encodage

$\Phi : \mathbb{F}_2^k \rightarrow \mathbb{F}_2^n \in \mathcal{L}(\mathbb{F}_2^k, \mathbb{F}_2^n)$



Définition : Matrice Génératrice

Matrice Génératrice

$G \in \mathcal{M}_{k,n}(\mathbb{F}_2)$ dont les lignes sont une base de \mathcal{C}

Encodage

Pour un message $u \in \mathbb{F}_2^k$ le mot de code $c \in \mathcal{C}$ est :

$$c = \Phi(u) = u \odot G$$

Forme systématique

$$G = \left[\begin{array}{c|c} I_k & P \end{array} \right]$$

- Pour $u \in \mathbb{F}_2^k$, $u \odot G = \left[\begin{array}{c|c} u & u \odot P \end{array} \right]$
- $P \in \mathcal{M}_{k,n-k}(\mathbb{F}_2)$ matrice de parité

Définition : Matrice de Contrôle

Matrice de Contrôle

$$H = \begin{bmatrix} P^\top & I_{n-k} \end{bmatrix}$$

- $\mathcal{C} = \ker(H) = \{v \in \mathbb{F}_2^n \mid H \odot v^\top = 0\}$
- $G \odot H^\top = 0$

Syndrome

Pour un vecteur reçu $r = c + e$, $s \in \mathbb{F}_2^{n-k}$

$$s = Hr^\top = Hc^\top + He^\top = 0 + He^\top$$

- Si $s = 0$, r est un mot de code valide
- Sinon s donne la signature de l'erreur e

Exemple d'un code linéaire

Exemple d'un code (5, 2)

- On choisit la matrice de parité P :

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

- Alors la matrice génératrice G est :

$$G = \left[\begin{array}{cc|ccc} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{array} \right]$$

- Message $u = [1 \ 1]$
- Mot de code $c = uG$:

$$c = [1 \ 1] \left[\begin{array}{cc|ccc} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{array} \right] = [1 \ 1 \ 1 \ 0 \ 1]$$

Exemple d'une code linéaire

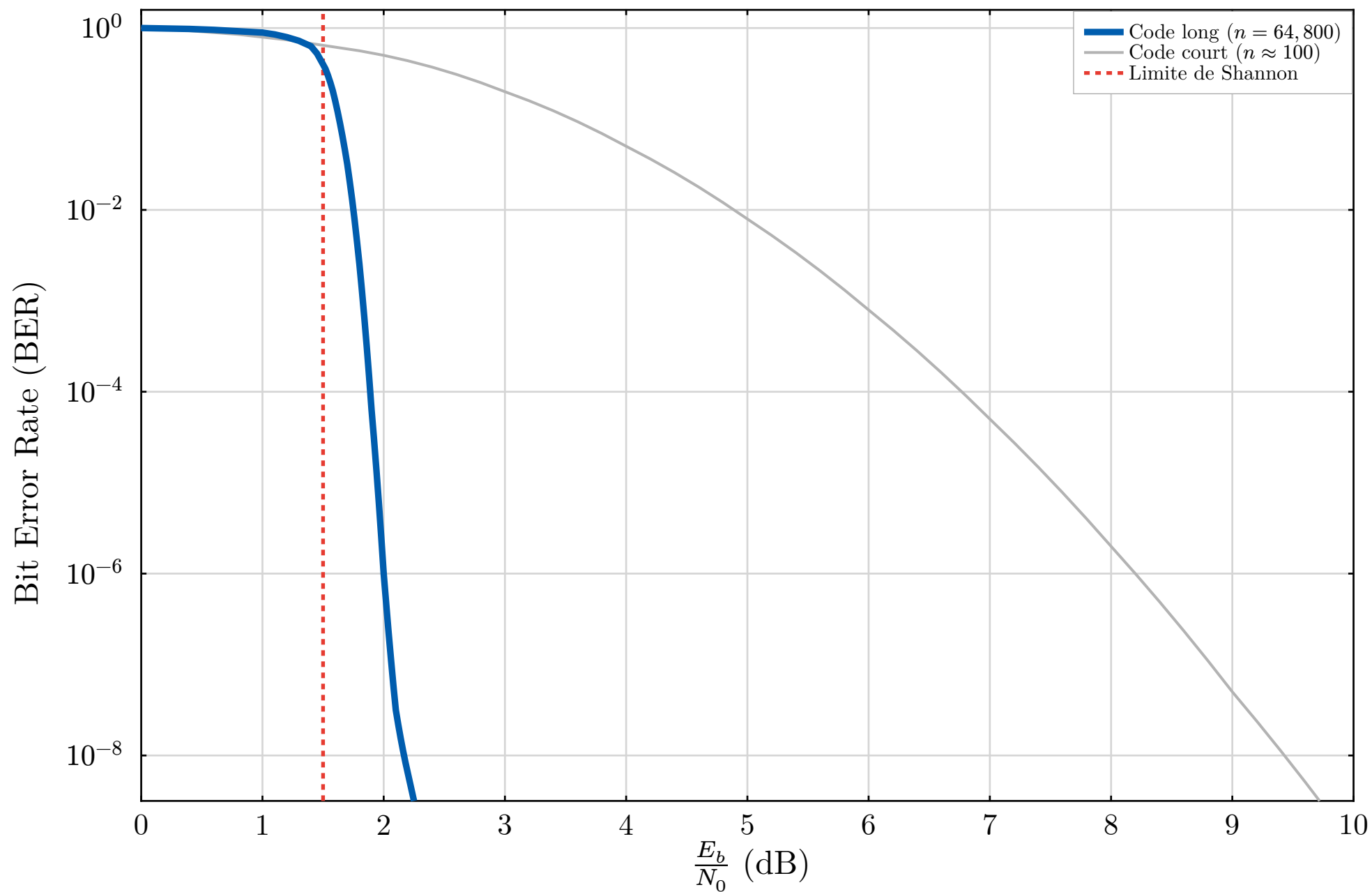
Enfin

$$H = \left[\begin{array}{cc|ccc} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{array} \right]$$

Vérification du mot de code $c = [1 \ 1 \ 1 \ 0 \ 1]$

$$Hc^{\top} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} [1 \ 1 \ 1 \ 0 \ 1]^{\top} = \begin{bmatrix} 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \\ 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \\ 0 \oplus 1 \oplus 0 \oplus 0 \oplus 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Approcher la Limite de Shannon



Le Mur de la Complexité

Décodage par Maximum de Vraisemblance (MDL)

Chercher le mot de code $\mathbf{c} \in \mathcal{C}$ le plus probable sachant \mathbf{r} reçu :

$$\hat{\mathbf{c}} = \arg \min_{\mathbf{c} \in \mathcal{C}} d_H(\mathbf{r}, \mathbf{c})$$

- Équivalent à chercher l'erreur \mathbf{e} de poids minimal tel que $\mathbf{H}\mathbf{e}^\top = \mathbf{s}$.

Le Problème du décodage par Syndrome

NP-Difficile et pour H quelconque : $\mathcal{O}(2^k)$

- Pour $k = 100$ bits, $2^{100} \approx 10^{30}$ opérations nécessaires.

Définition des Codes LDPC

Formalisation des Codes LDPC Réguliers

Code linéaire en bloc avec une matrice de contrôle \mathbf{H} est clairsemée.

- Poids de Colonne w_c
- Poids de Ligne w_r

Conditions de Faible Densité

$$w_c \ll n - k \qquad w_r \ll n$$

Rendement

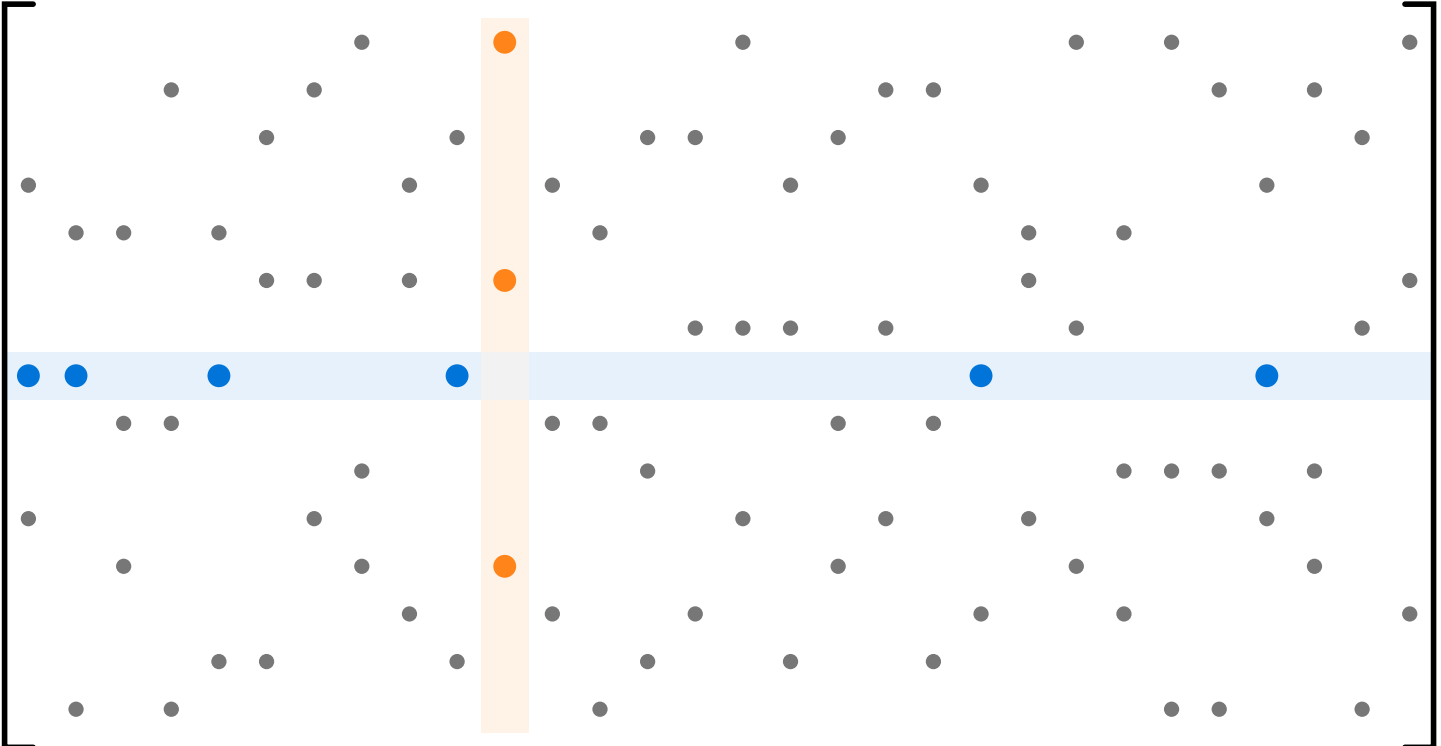
$$R = \frac{n - \text{rg}(\mathbf{H})}{n} \geq 1 - \frac{m}{n}$$

Matrice de contrôle

Code LDPC (6, 3)

$$mw_r = nw_c \text{ donc } H \in \mathcal{M}_{15,30}(\mathbb{F}_2) \text{ et } R = 1 - \frac{m}{n} = \frac{1}{2}$$

$w_c = 3$

$H =$ 

$w_r = 6$

De la Matrice aux Équations de Parité

$$H = \begin{bmatrix} \text{Matrix of points} \end{bmatrix} \begin{bmatrix} r_0 \\ r_1 \\ \vdots \\ r_{29} \end{bmatrix}$$

Mot re\u0219u $r \in \mathbb{F}_2^{30}$

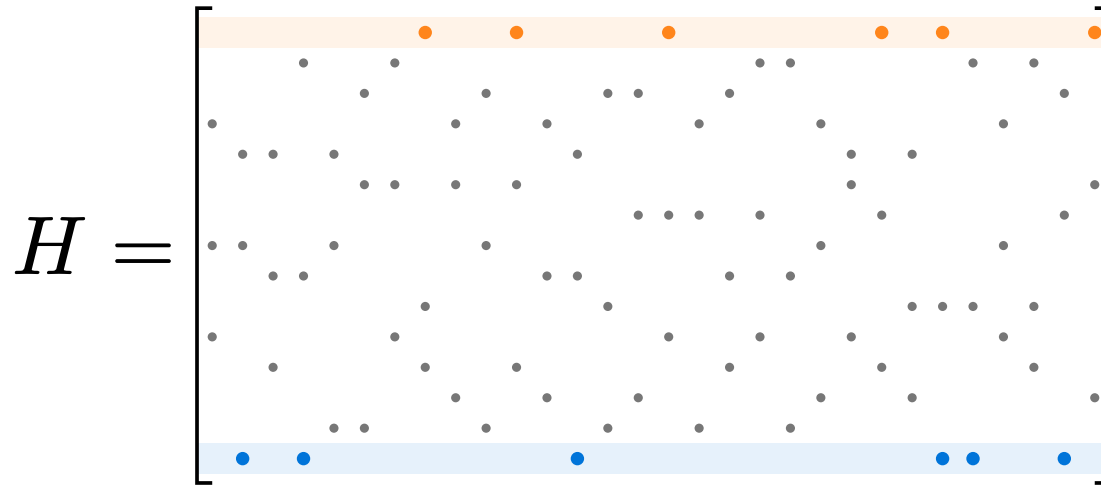
- Chaque ligne j de H définit une équation de parité f_j .
- Pour r , on vérifie le syndrome : $Hr^\top = 0$.

Équations de Parité

$$f_0 : r_7 \oplus r_{10} \oplus r_{15} \oplus r_{22} \oplus r_{24} \oplus r_{29} = 0$$

- Si $f_j = 1$, un nombre impair de bits a été inversé par le canal.

L'Entrelacement des Contraintes



- Chaque bit r_i participe à $w_c = 3$ équations distinctes

$$\begin{cases} r_7 \oplus r_{10} \oplus r_{15} \oplus r_{22} \oplus r_{24} \oplus r_{29} = 0 \\ \vdots \\ r_1 \oplus r_3 \oplus r_{12} \oplus r_{24} \oplus r_{25} \oplus r_{28} = 0 \end{cases}$$

- r_{24} : Surveillé par f_0 et f_{14} .
- Si $f_0 = 1$ et $f_{14} = 1$, r_{24} est suspect

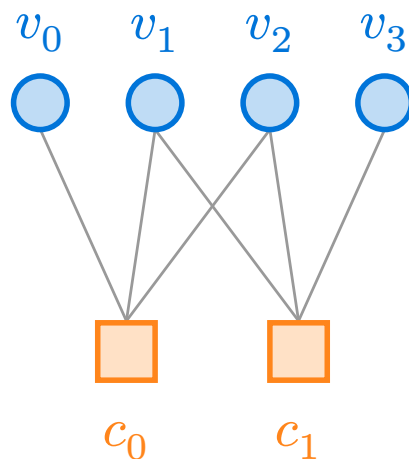
Graphe de Tanner : Définition

Graphe de Tanner $\mathcal{G}(H)$

Graphe bipartite $\mathcal{G} = (V \sqcup C, E)$:

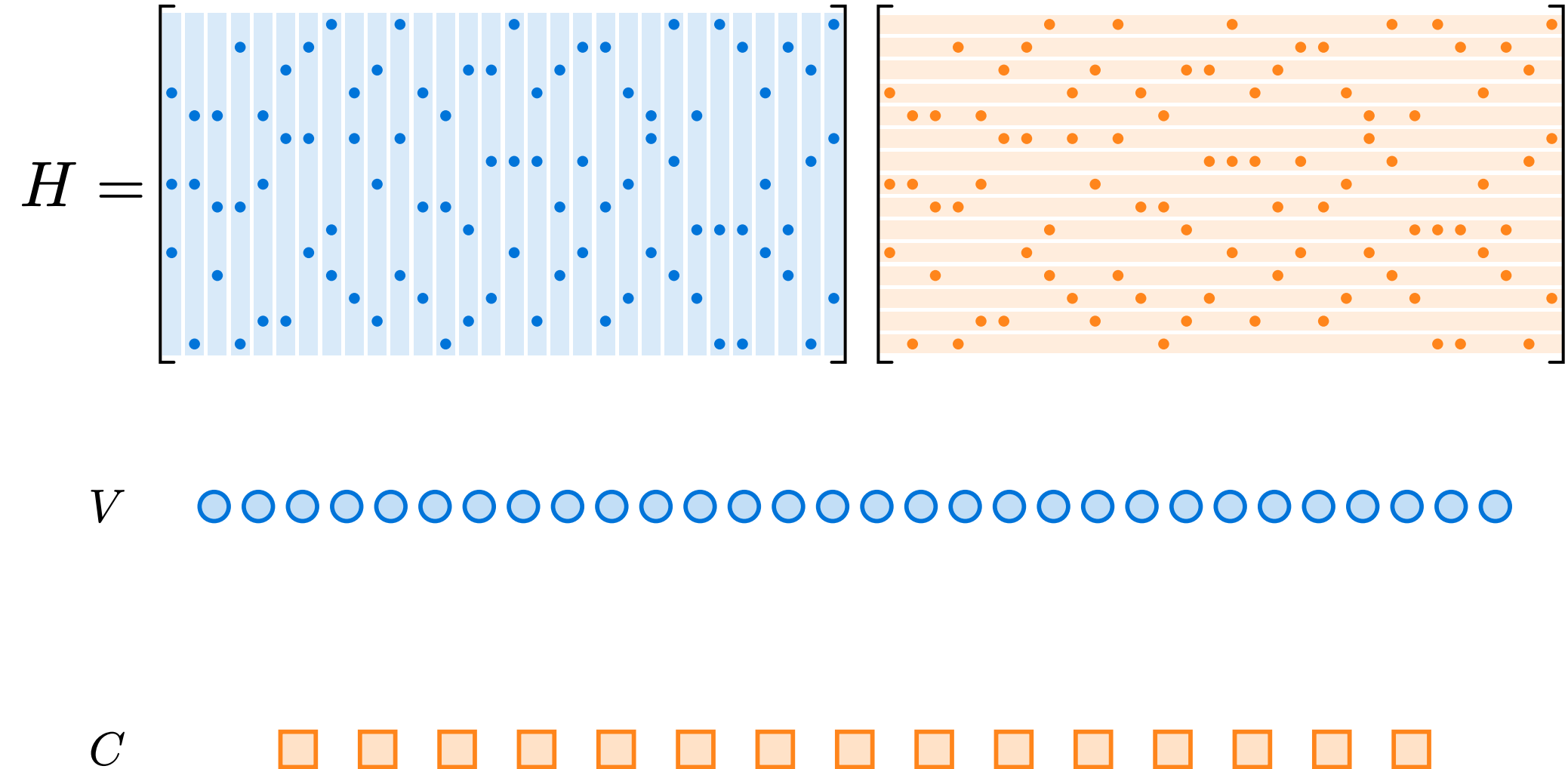
$$(v_j, c_i) \in E \iff H_{i,j} = 1$$

- $V = \{v_0, \dots, v_{n-1}\}$ nœuds de **variable**
- $C = \{c_0, \dots, c_{m-1}\}$ nœuds de **contrôle**
- $\deg(v_j) = w_c$
- $|E| = n \cdot w_c = m \cdot w_r$
- $H \cong \mathcal{G}$
- $\deg(c_i) = w_r$

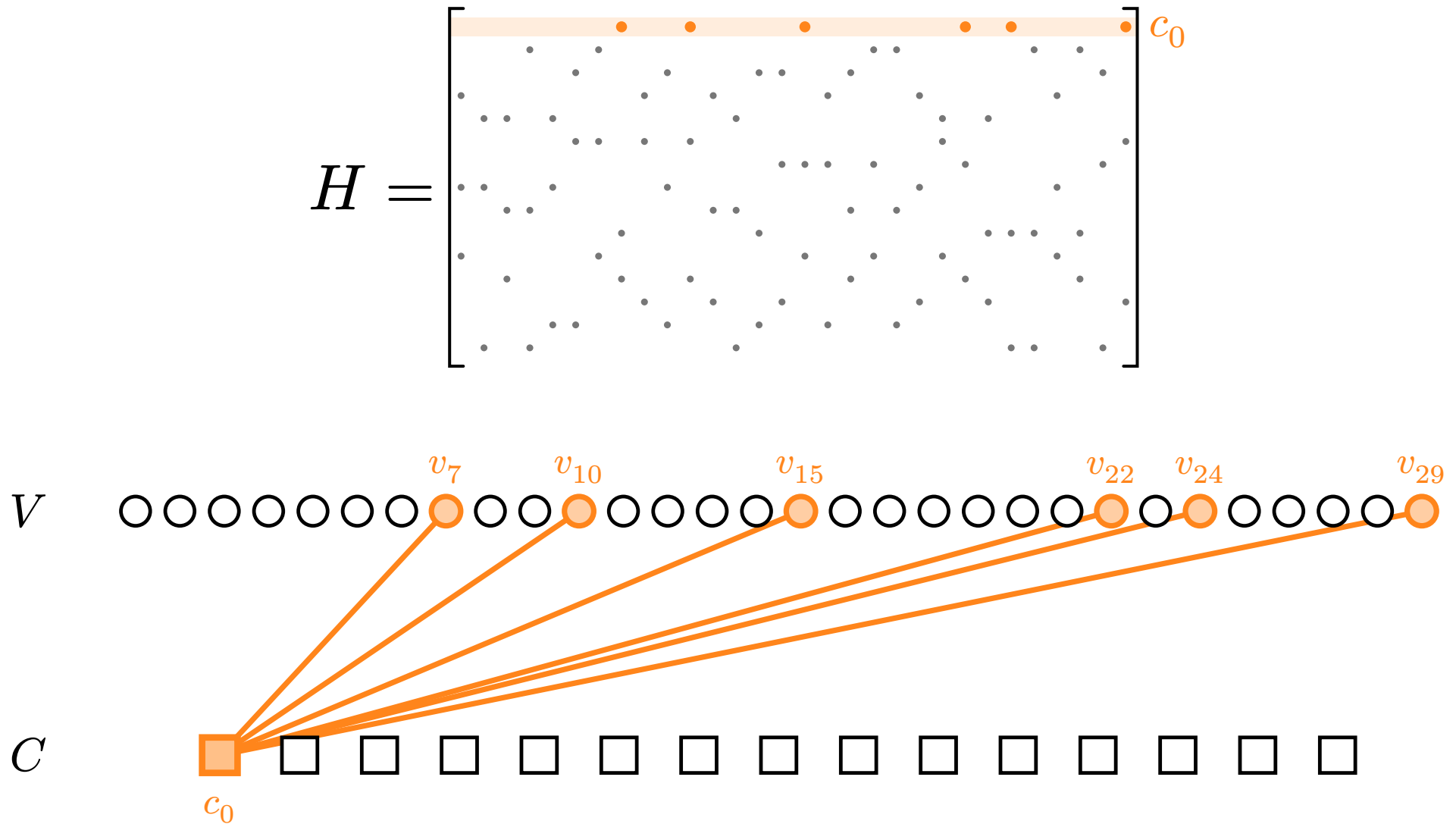


Exemple $n = 4, m = 2$

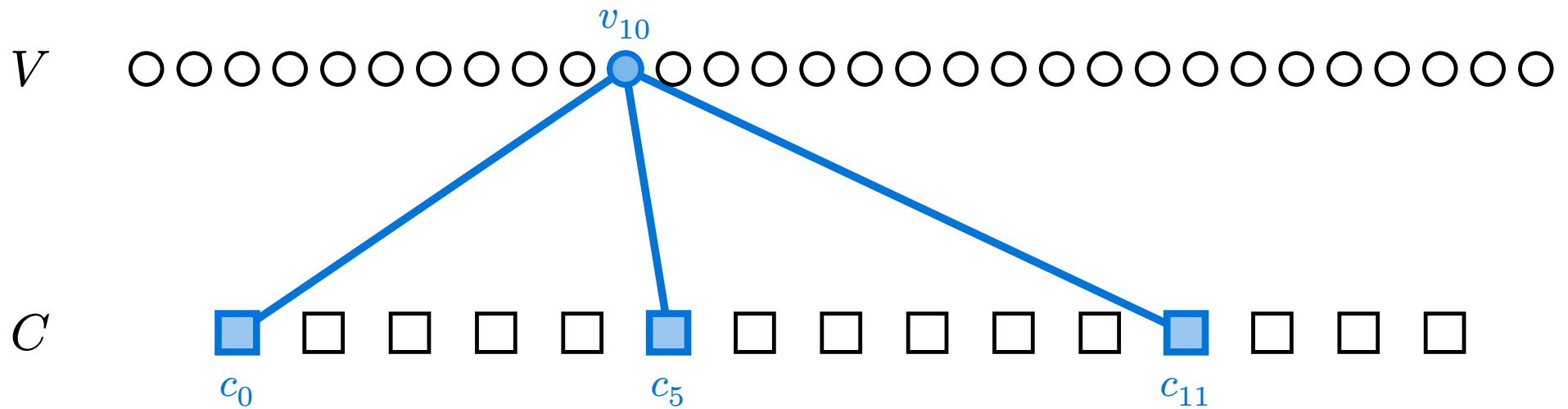
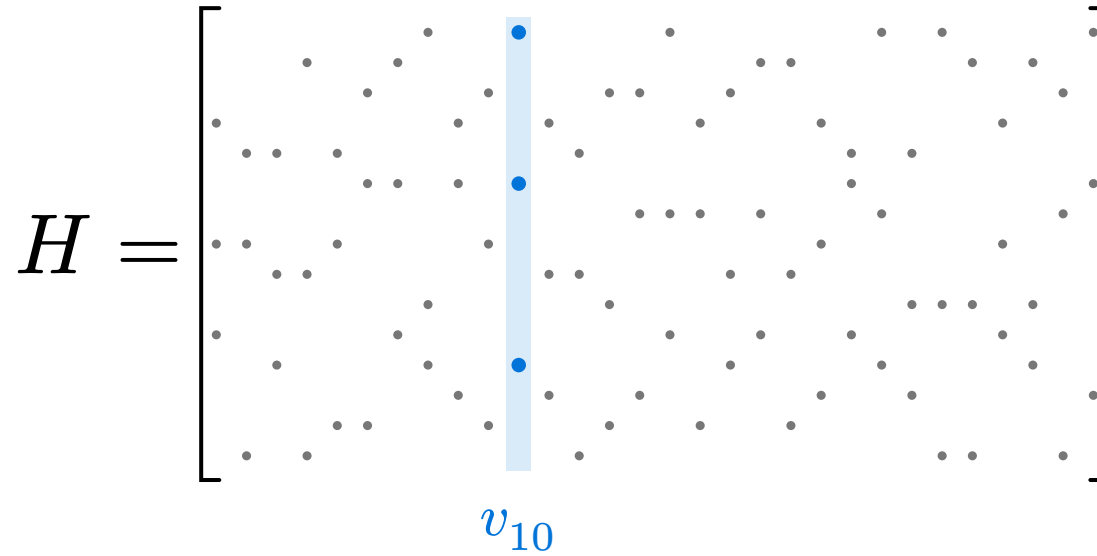
Construction du Graphe : Les Nœuds



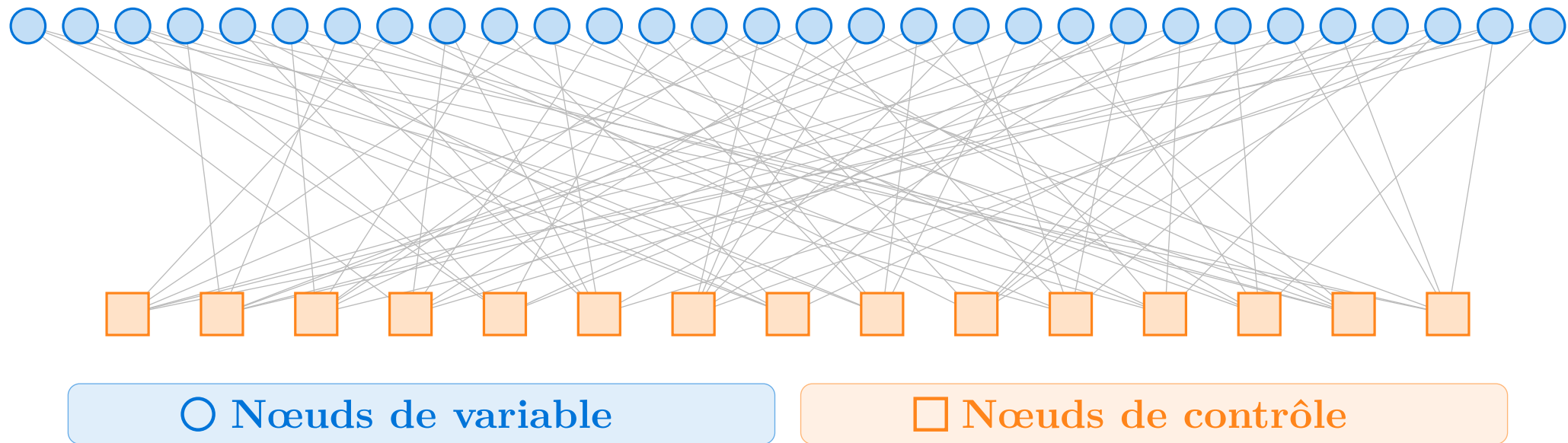
Construction du Graphe : Nœud de Contrôle



Construction du Graphe : Nœud de Variable



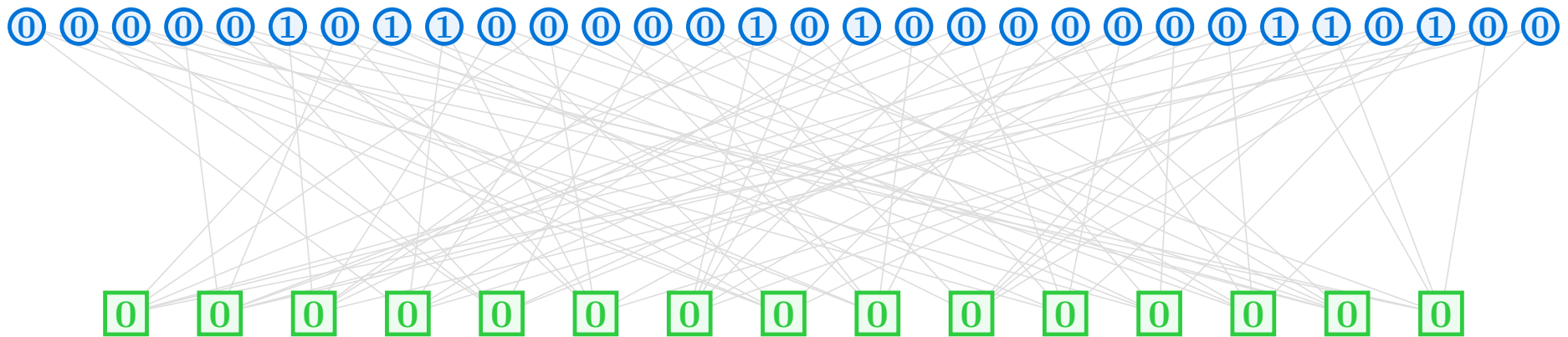
Graphe de Tanner Final




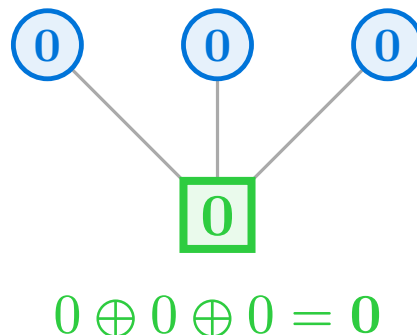
La Contrainte de Somme Nulle

Vision Graphe

Si $s = 0$ alors que chaque nœud de contrôle est localement satisfait

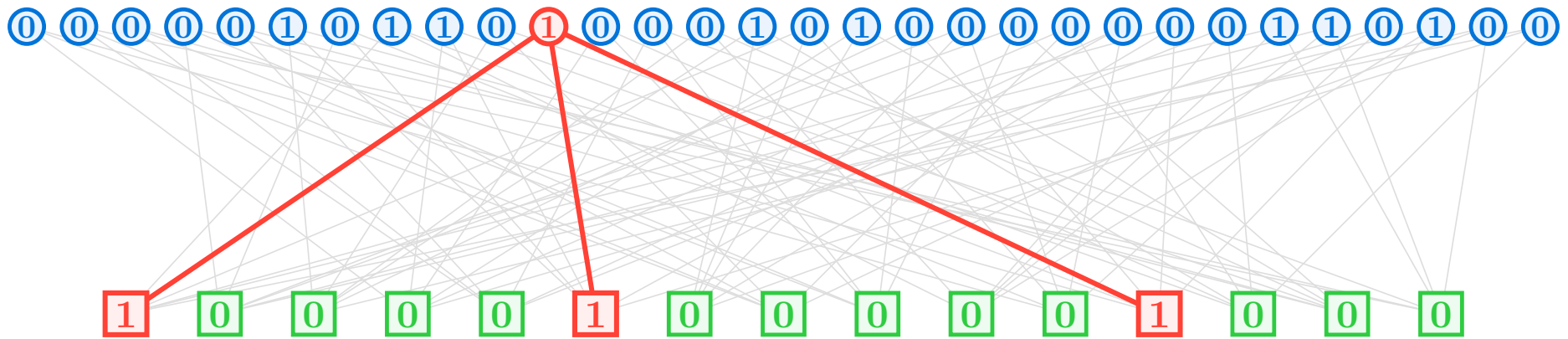


Chaque  calcule le xor de ses voisins  : $f_i = \bigoplus_{j \in \mathcal{N}(c_i)} v_j$

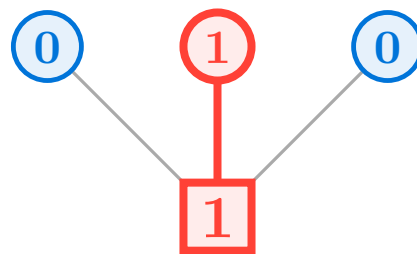


Détection d'Erreur

Si un bit est inversé, toutes les contraintes associées sont à 1



$$0 \oplus 1 \oplus 0 = 1 \rightarrow \text{Erreur détectée}$$



$$0 \oplus 1 \oplus 0 = 1$$

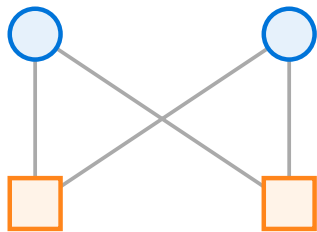
La Topologie de H : Le Girth

Définition : Le Girth (La Maille)

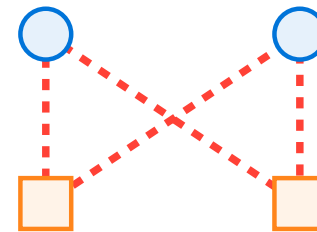
Longueur du plus court cycle dans le graphe de Tanner

- Le girth est **pair**
- La valeur minimale est $g = 4$.

Girth élevé \Rightarrow Meilleure diffusion de l'information.



Graphe de Tanner

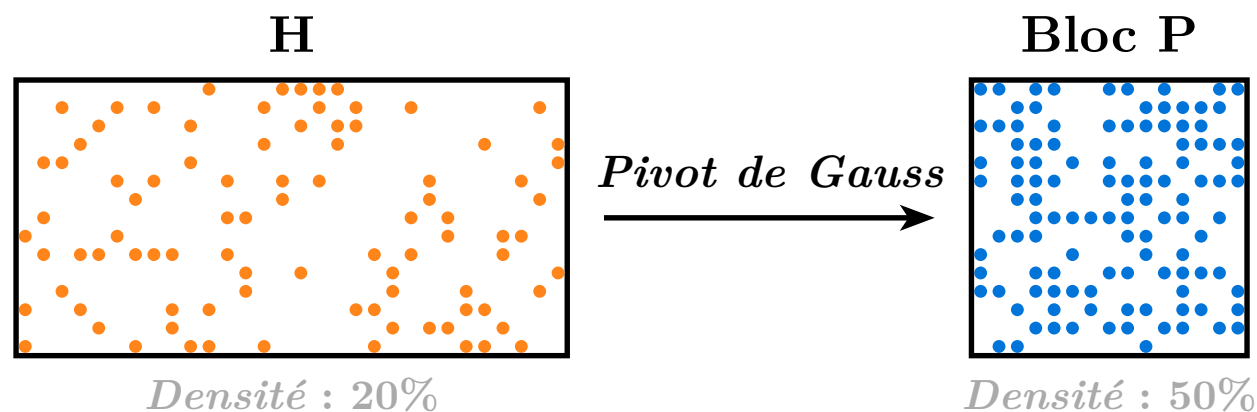


4-Cycle

Encodage LDPC : Calcul de G

Encodage

Mot de code \mathbf{c} généré à partir d'un message \mathbf{u} : $\mathbf{c} = \mathbf{u}\mathbf{G}$



- Forme Systématique : Par élimination de Gauss sur \mathbf{H} , on obtient

$$\mathbf{H} = \begin{bmatrix} \mathbf{P}^\top & \mathbf{I}_{n-k} \end{bmatrix} \longrightarrow \mathbf{G} = \begin{bmatrix} \mathbf{I}_k & \mathbf{P} \end{bmatrix}$$




- La matrice \mathbf{G} devient dense \Rightarrow encodage en $\mathcal{O}(n^2)$

Décodage : Bit-Flipping

Décision Stricte (Hard Decision)

Algorithme **itératif** : les nœuds **échangent des bits** pour localiser les erreurs.

Message Passing




-  envoie son bit courant à ses voisins 
 -  renvoie son **verdict de parité** (0 ou 1)
-
- Si v_j participe à **trop d'équations non satisfaites** \Rightarrow on l'inverse.

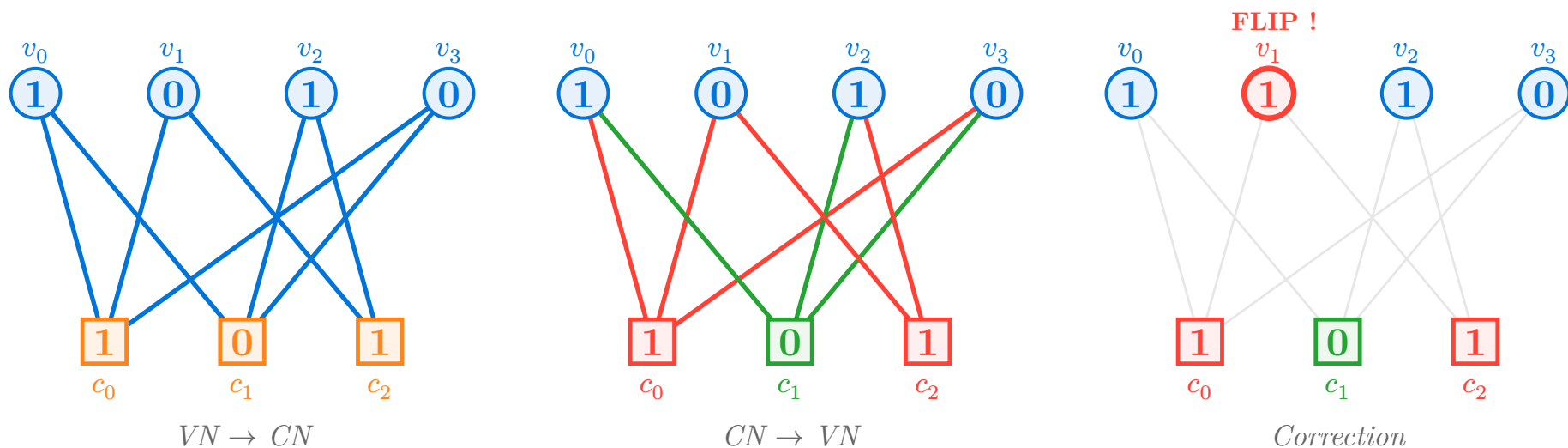
Décodage : Bit-Flipping

Décision Stricte (Hard Decision)

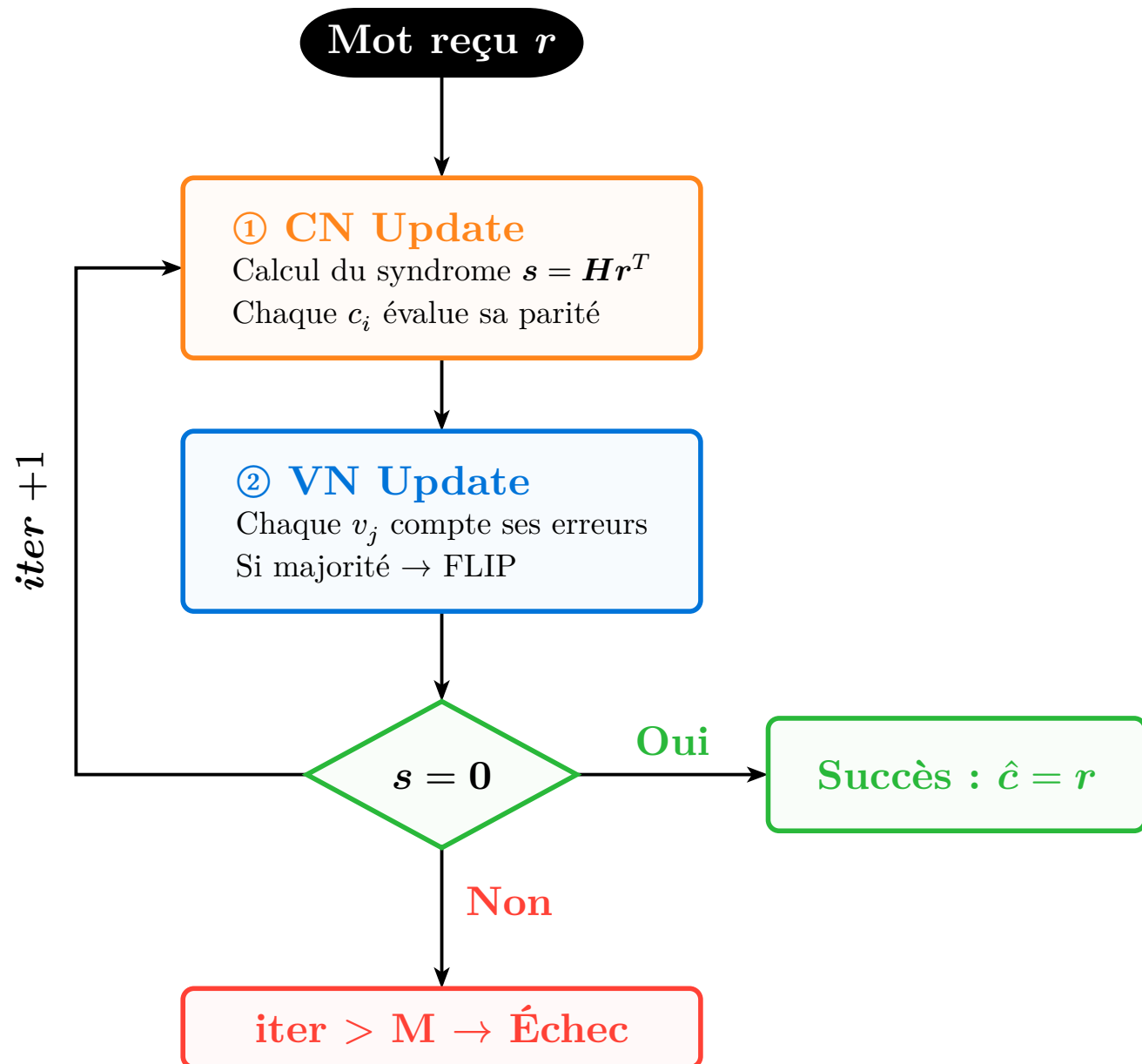
Algorithme **itératif** : les nœuds **échangent des bits** pour localiser les erreurs.

Message Passing

-  envoie son bit courant à ses voisins 
 -  renvoie son **verdict de parité** (0 ou 1)
- Si v_j participe à **trop d'équations non satisfaites** \Rightarrow on l'inverse.



Bit-Flipping : Graphe de flot de contrôle



Exemple implementation Bit-Flipping rust

ex + canal d'étude bruit AWGN avec ce qu'il se passe dans les radio / cable etc
+ tension

Bit-Flipping : Analyse

Avantages

- **Complexité** : simples XOR et compteurs — $\mathcal{O}(n)$ par itération

Limite

- Ignore la **confiance** du récepteur physique dans le signal
- Un bit reçu à 0.51 V est traité comme 0

Soft decoding

belief propagation, log ou virgule fixe, explication resultat meilleur

Implementation

Irl hackrf, test de diff de debit avec des paquets

Image

Test de transmission d'image avec différent ldpc non opti et opti (le H)

Annexe



Théorie derrière la définition des codes linaires

Poser les notations algébriques etc...

Decodage par maximum de vraisemblance

Expliquer, quelle distance ? etc

Code LDPC non régulier