



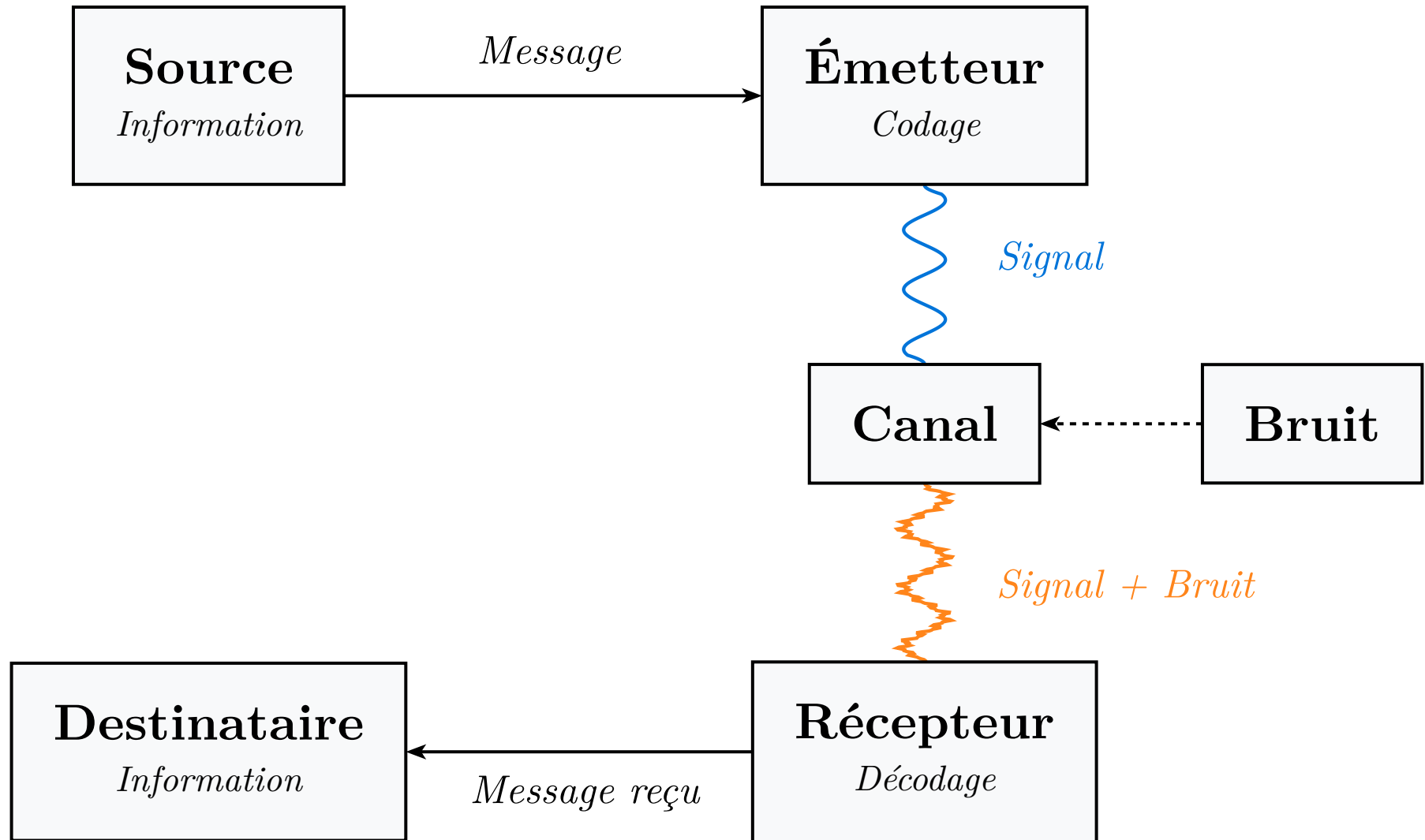
Codes LDPC

Anthony PERRONI

n°49871

2025 - 2026

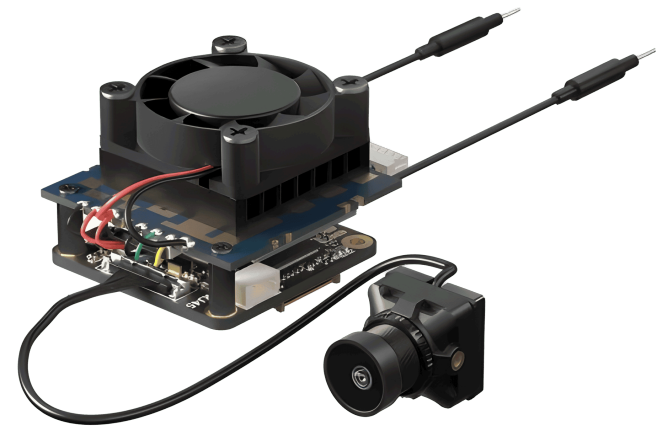
Introduction : Communication Numérique



Introduction : Utilisation



Athena-Fidus



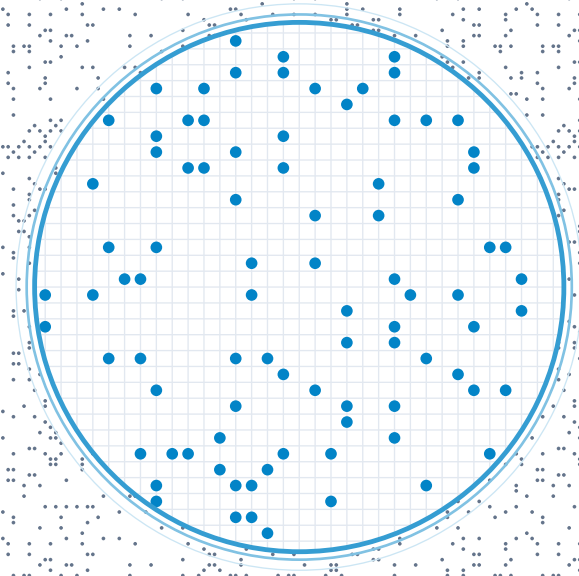
Module OpenIPC

Problématique

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

Plan

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



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

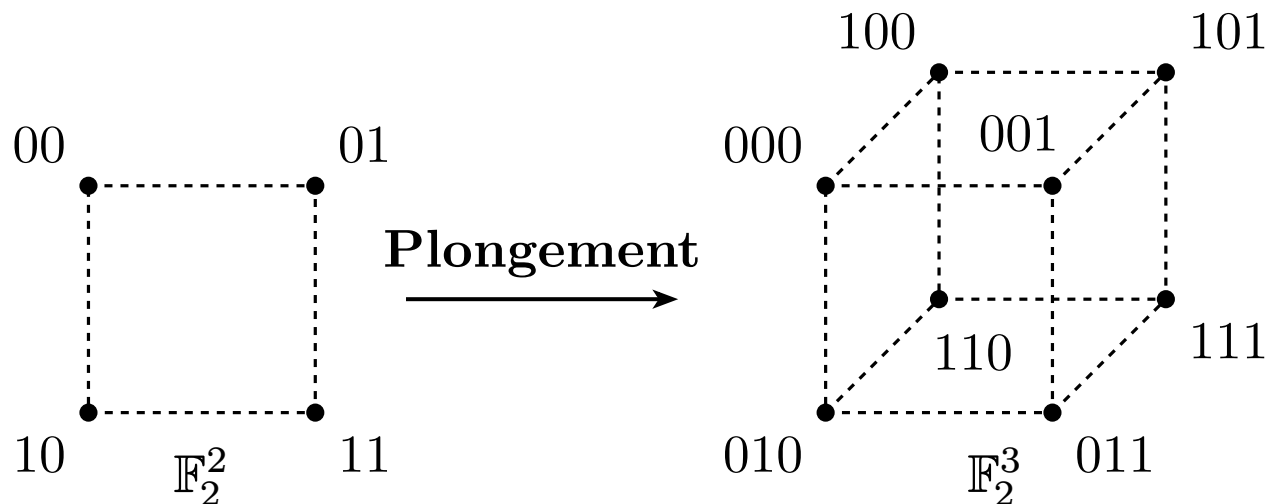
Code $(n, k) \in \mathbb{N}^2$

\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

$$G = \begin{bmatrix} I_k & P \end{bmatrix}$$

- Pour $u \in \mathbb{F}_2^k$, $u \odot G = \begin{bmatrix} u & u \odot P \end{bmatrix}$
- $P \in \mathcal{M}_{k,n-k}(\mathbb{F}_2)$ 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'un code linéaire

Structure systématique de H :

$$H = \begin{bmatrix} P^\top & I_3 \end{bmatrix}$$

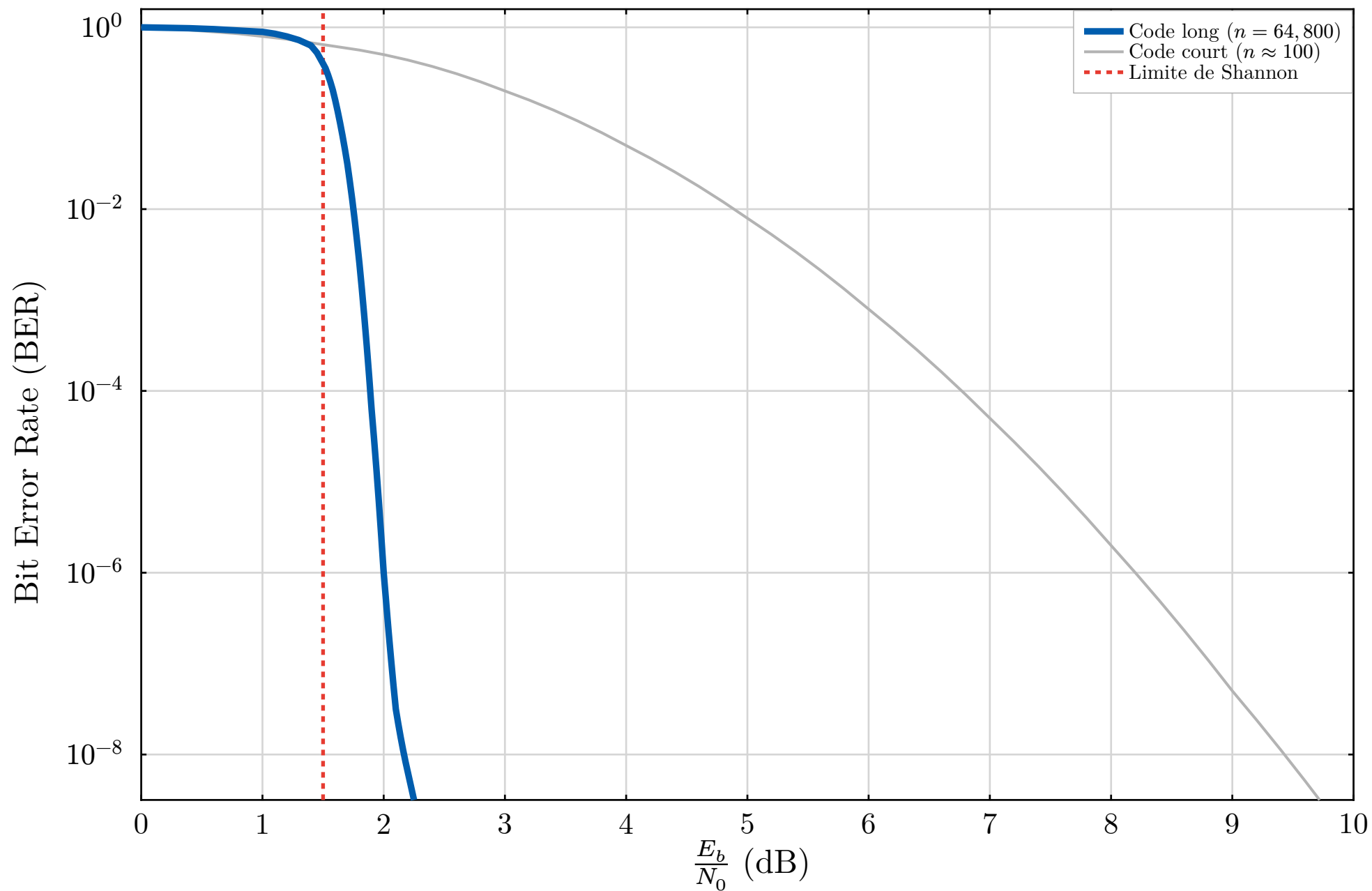
Ainsi :

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

Vérification du mot de code $c = (1, 1, 1, 0, 1)$:

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

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

Codes LDPC Réguliers

Code linéaire en bloc avec une matrice de contrôle H 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

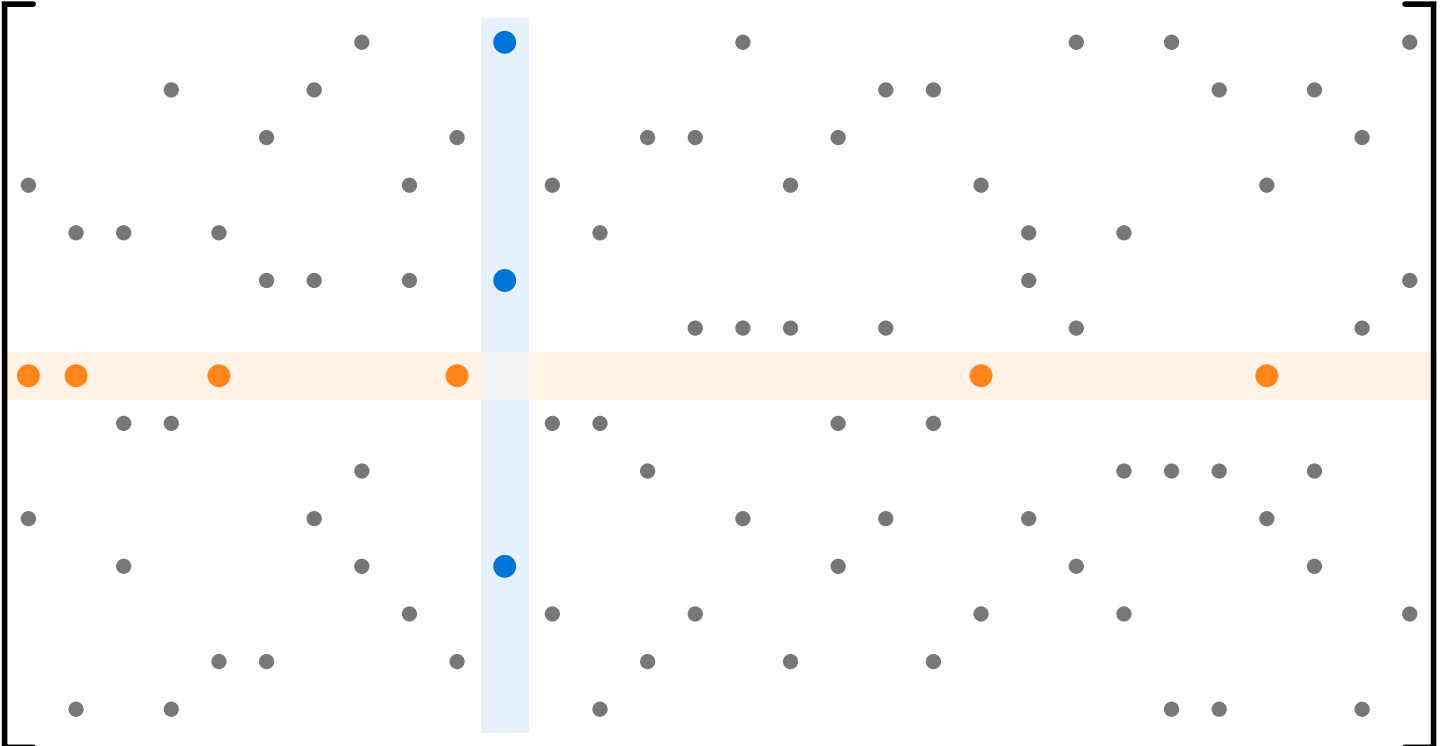
$$R = \frac{n - \text{rg}(H)}{n} \geq 1 - \frac{m}{n}$$

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

$$H = \begin{bmatrix} \text{---} & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \text{---} \\ \bullet & & & & & & & \\ & \bullet & & & & & & \\ & & \bullet & & & & & \\ & & & \bullet & & & & \\ & & & & \bullet & & & \\ & & & & & \bullet & & \\ & & & & & & \bullet & \\ & & & & & & & \bullet \\ & & & & & & & & \bullet \\ & & & & & & & & & \bullet \\ & & & & & & & & & & \bullet \\ & & & & & & & & & & & \bullet \\ & & & & & & & & & & & & \bullet \\ & & & & & & & & & & & & & \bullet \\ & & & & & & & & & & & & & & \bullet \\ & & & & & & & & & & & & & & & \bullet \\ & & & & & & & & & & & & & & & & \bullet \\ & & & & & & & & & & & & & & & & & \bullet \\ & & & & & & & & & & & & & & & & & & \bullet \\ & & & & & & & & & & & & & & & & & & & \bullet \\ & \bullet \\ & \bullet \\ & \bullet \end{bmatrix} \begin{bmatrix} r_0 \\ r_1 \\ \vdots \\ r_{29} \end{bmatrix}$$

Mot reçu $r \in \mathbb{F}_2^{30}$

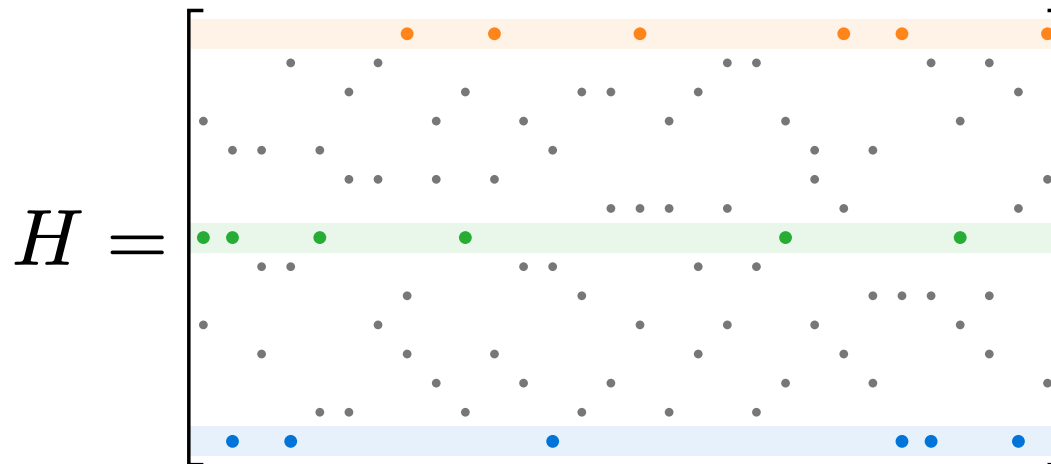
- Chaque ligne j de H définit une équation de parité f_j .
- Pour r , on vérifie le syndrome : $Hr^\top = 0$.

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 :

$$\begin{cases} r_7 \oplus r_{10} \oplus r_{15} \oplus r_{22} \oplus r_{24} \oplus r_{29} = 0 & (f_0) \\ r_0 \oplus r_1 \oplus r_4 \oplus r_9 \oplus r_{20} \oplus r_{26} = 0 & (f_7) \\ r_1 \oplus r_3 \oplus r_{12} \oplus r_{24} \oplus r_{25} \oplus r_{28} = 0 & (f_{14}) \end{cases}$$

- r_{24} : Surveillé simultanément par f_0 , f_7 et f_{14} .
- Si $\forall j \in \{0, 7, 24\}$, $f_j = 1$, alors le bit est comme suspect.

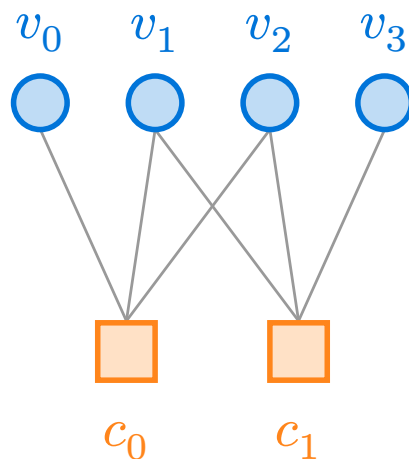
Graphe de Tanner : Définition

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

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

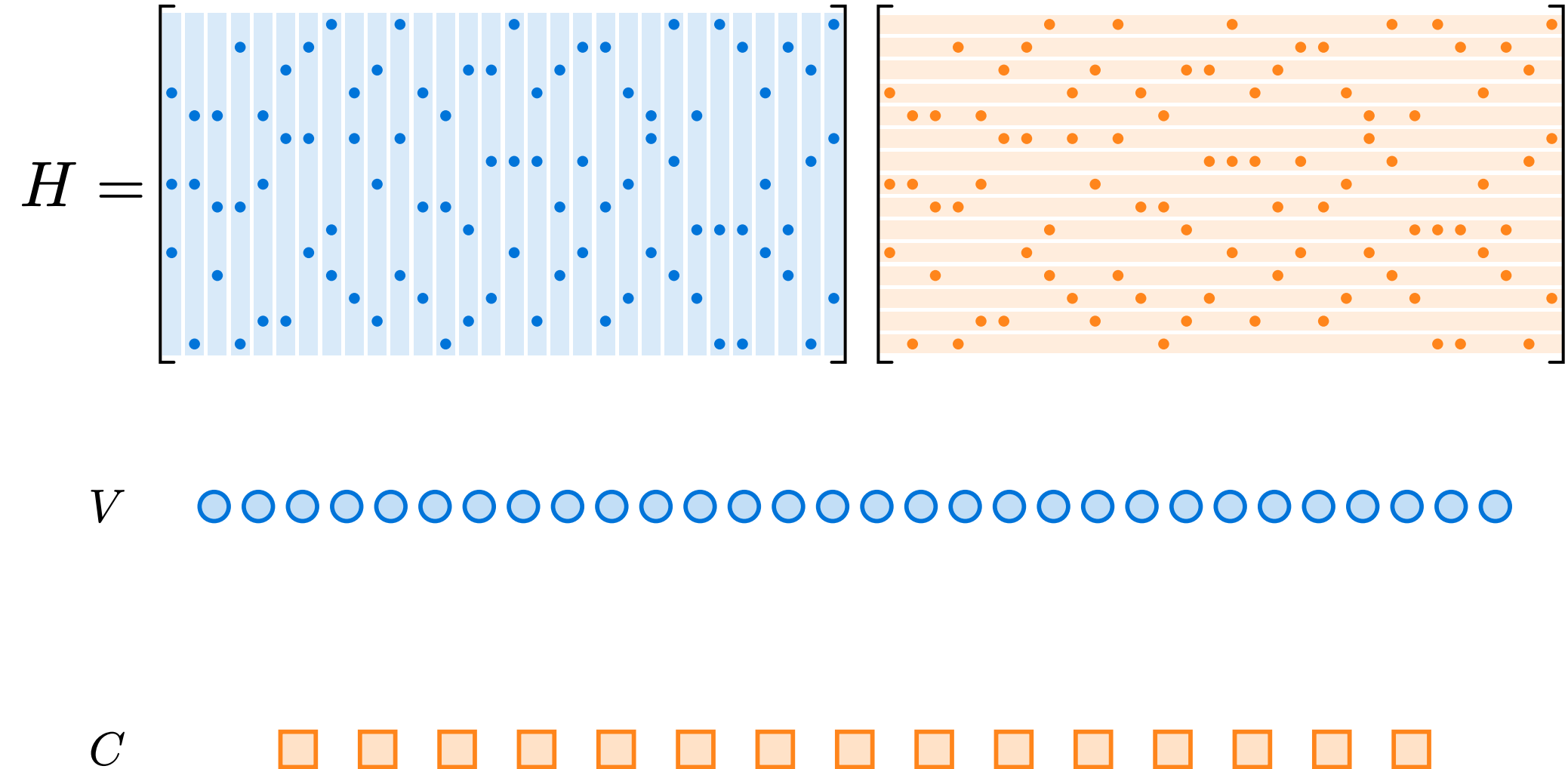
$$(v_j, c_i) \in A \iff H_{i,j} = 1$$

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

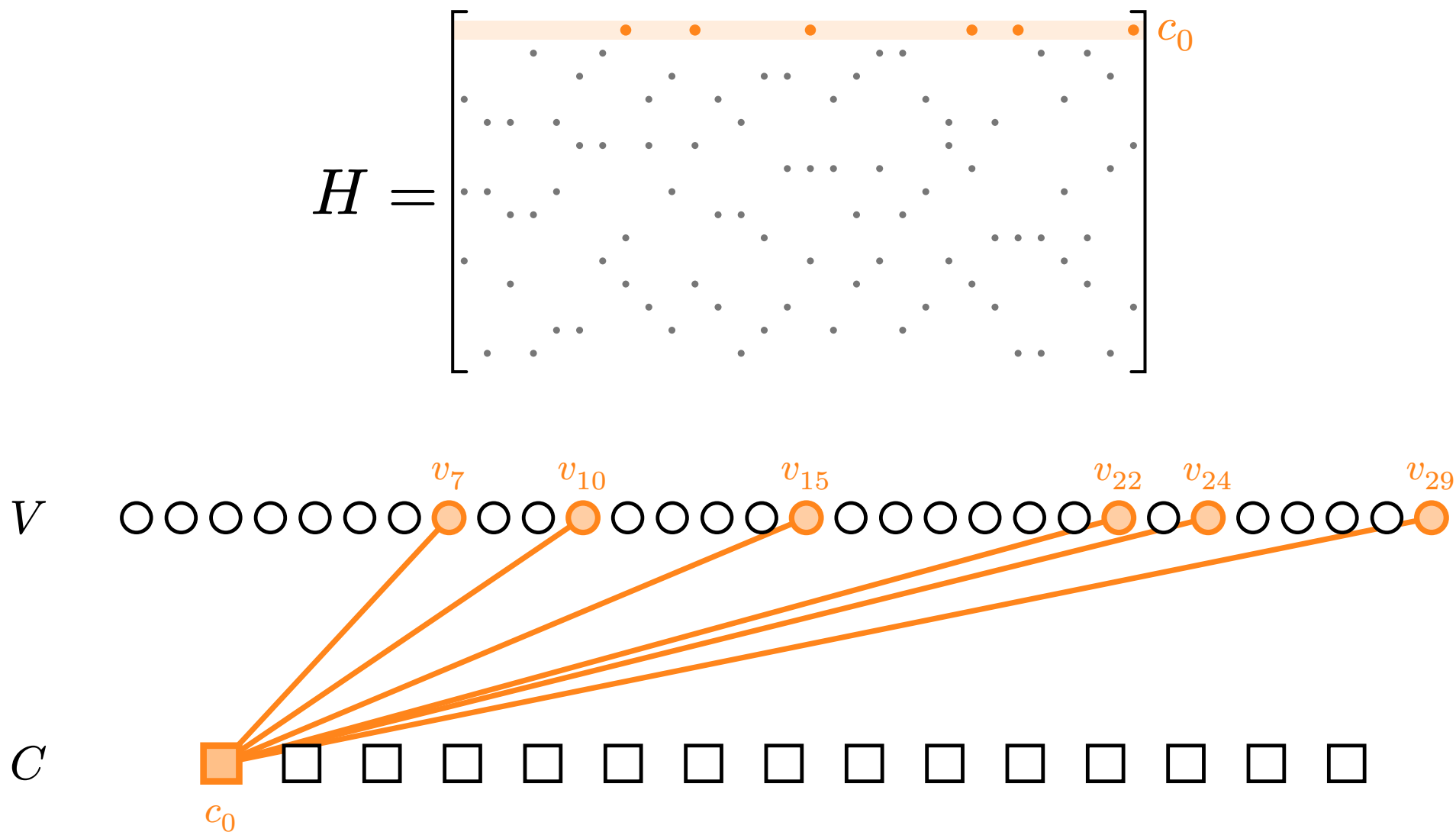


Exemple $n = 4, m = 2$

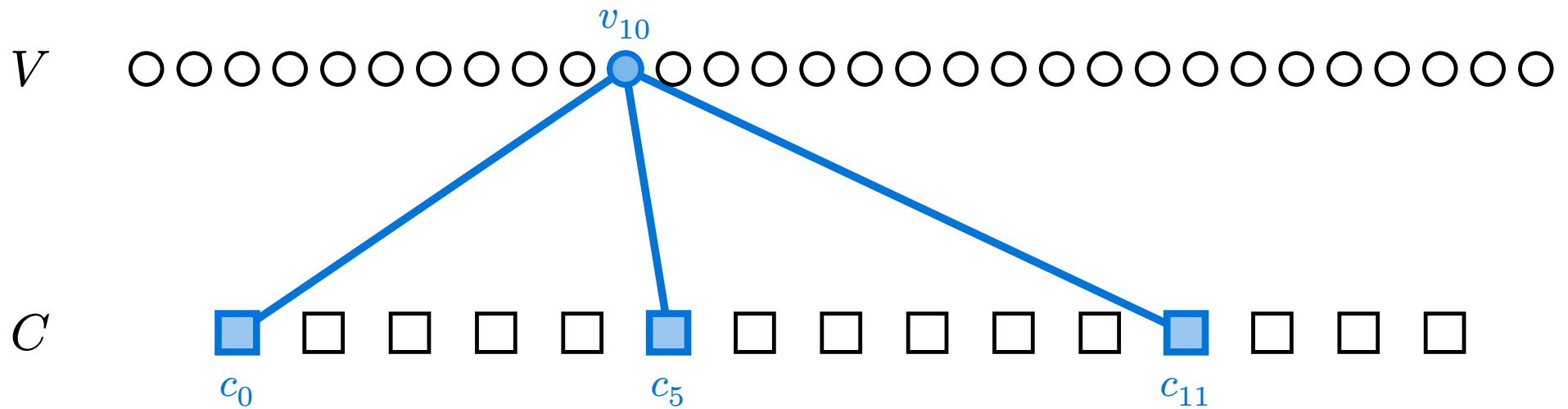
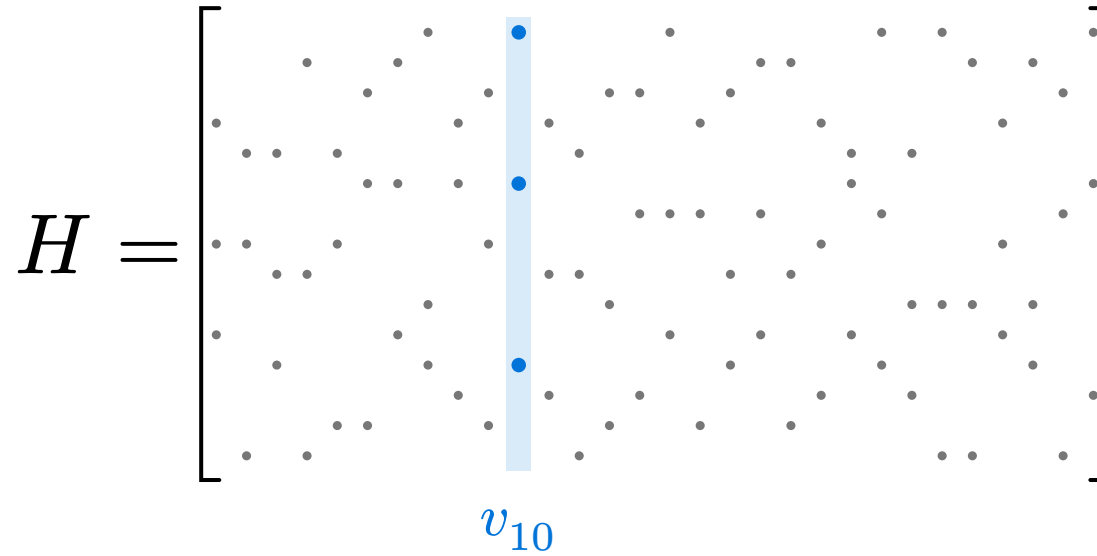
Construction du Graphe : Les Nœuds



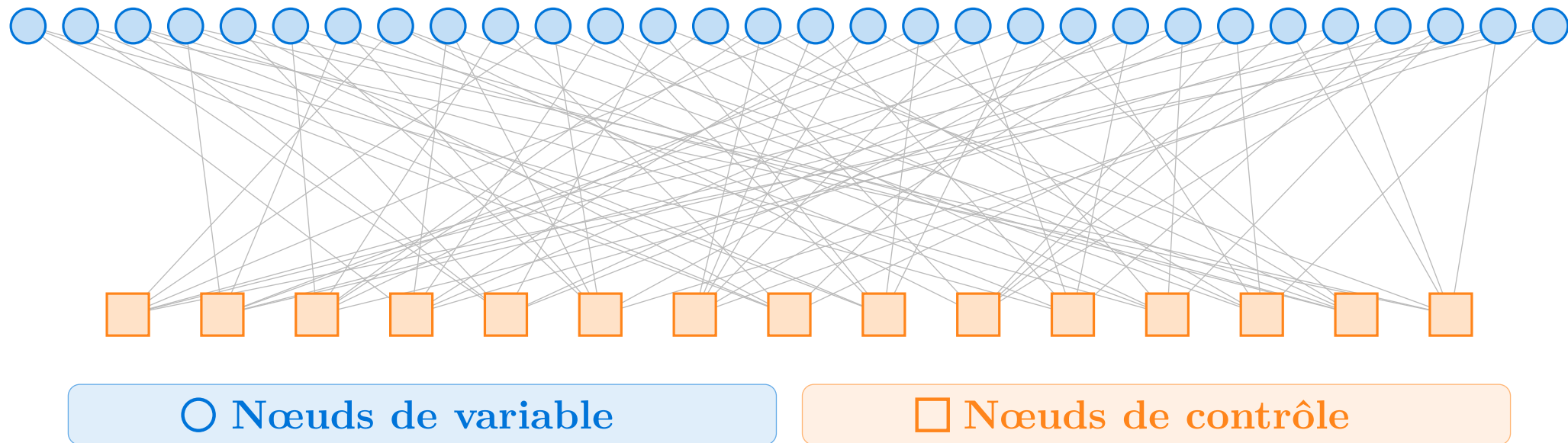
Construction du Graphe : Nœud de Contrôle



Construction du Graphe : Nœud de Variable



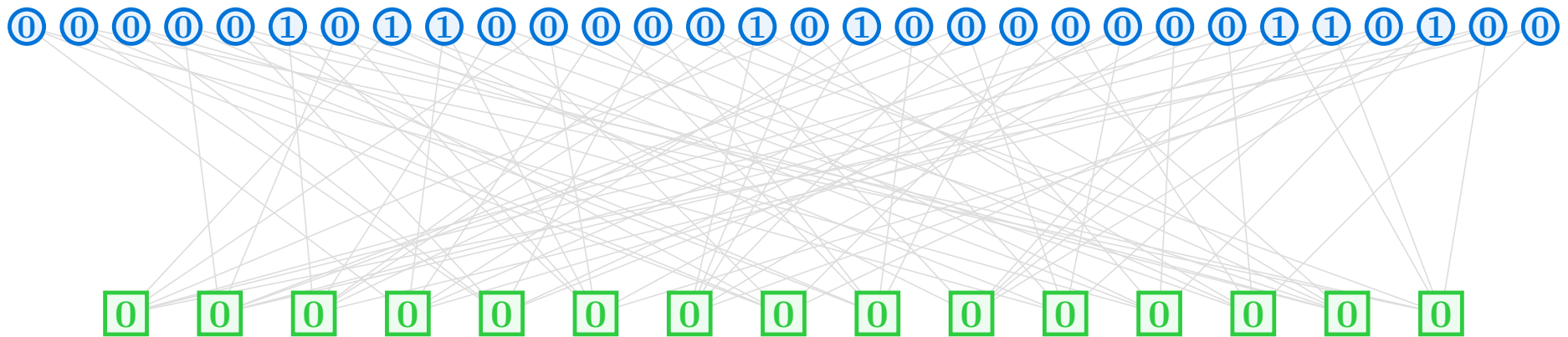
Graphe de Tanner Final




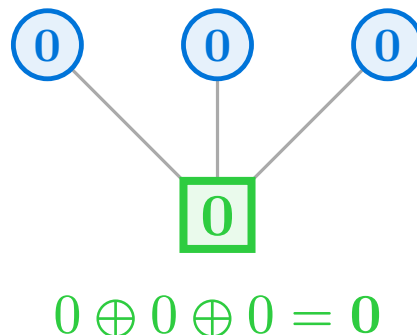
La Contrainte de Somme Nulle

Vision Graphe

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



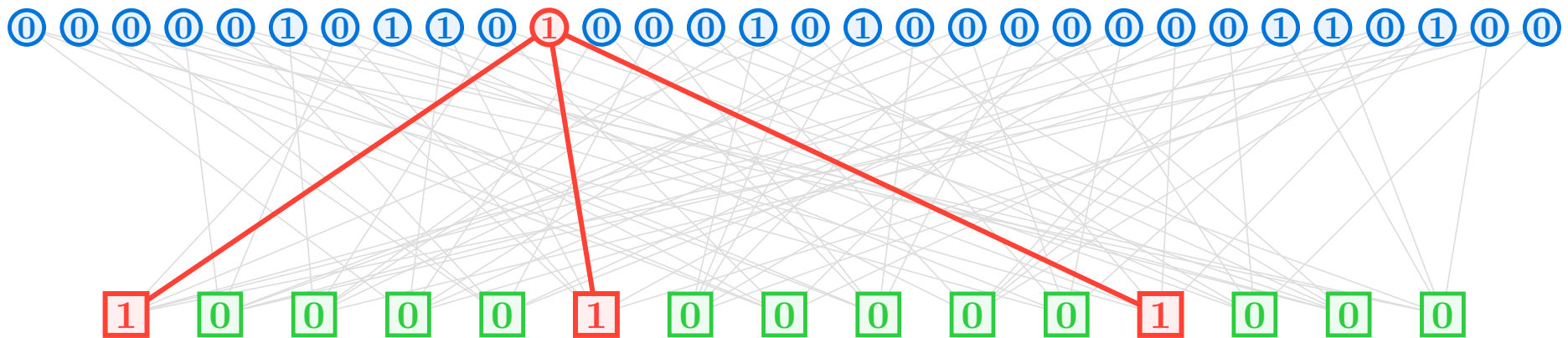
Chaque  calcule le xor de ses voisins  : $f_i = \bigoplus_{j \in \mathcal{N}(c_i)} v_j$



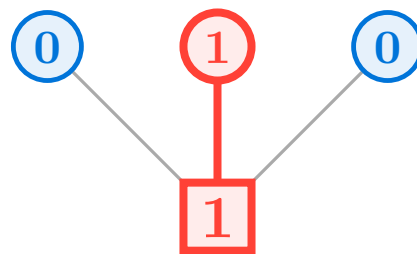
La Contrainte de Somme Nulle

Détection d'Erreur

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



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



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

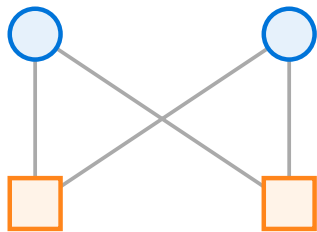
La Topologie de H : Le Girth

Définition : Le Girth (La Maille)

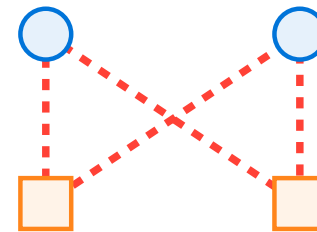
Longueur du plus court cycle dans le graphe de Tanner

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

Girth élevé \Rightarrow Meilleure diffusion de l'information.



Graphe de Tanner



4-Cycle

Gallager

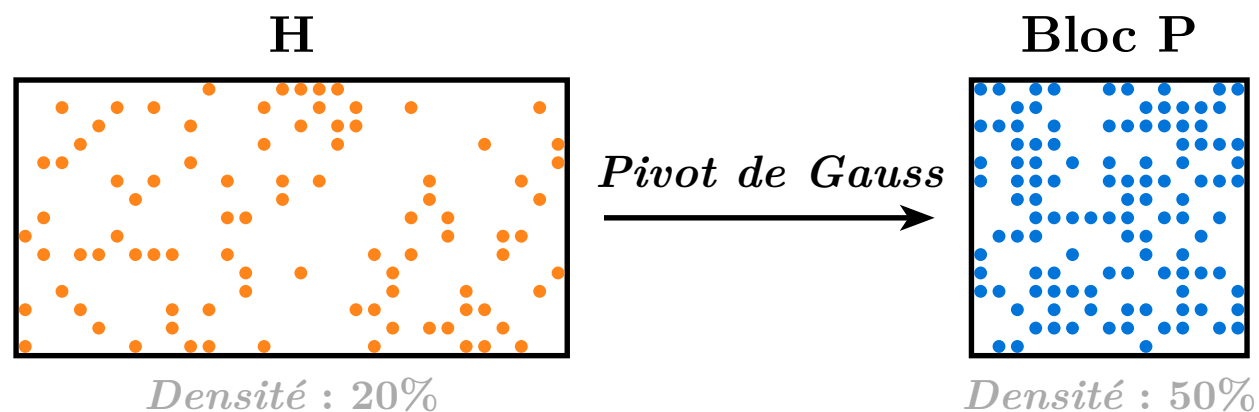
Mackay-Neal

Progressive Edge-growth

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.

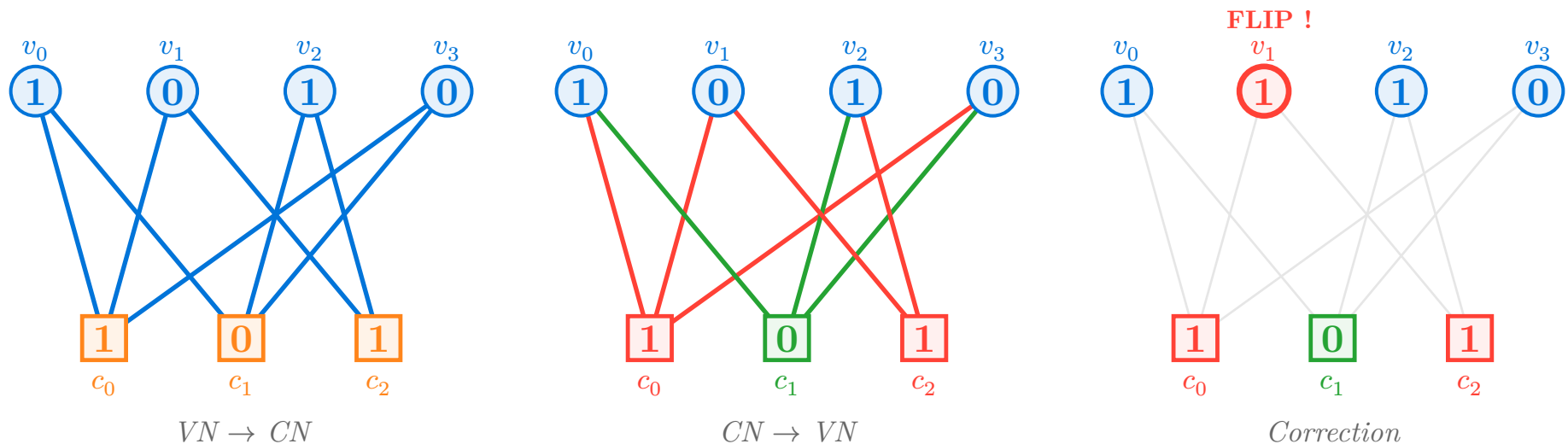
Décodage : Bit-Flipping

Décision Stricte (Hard Decision)

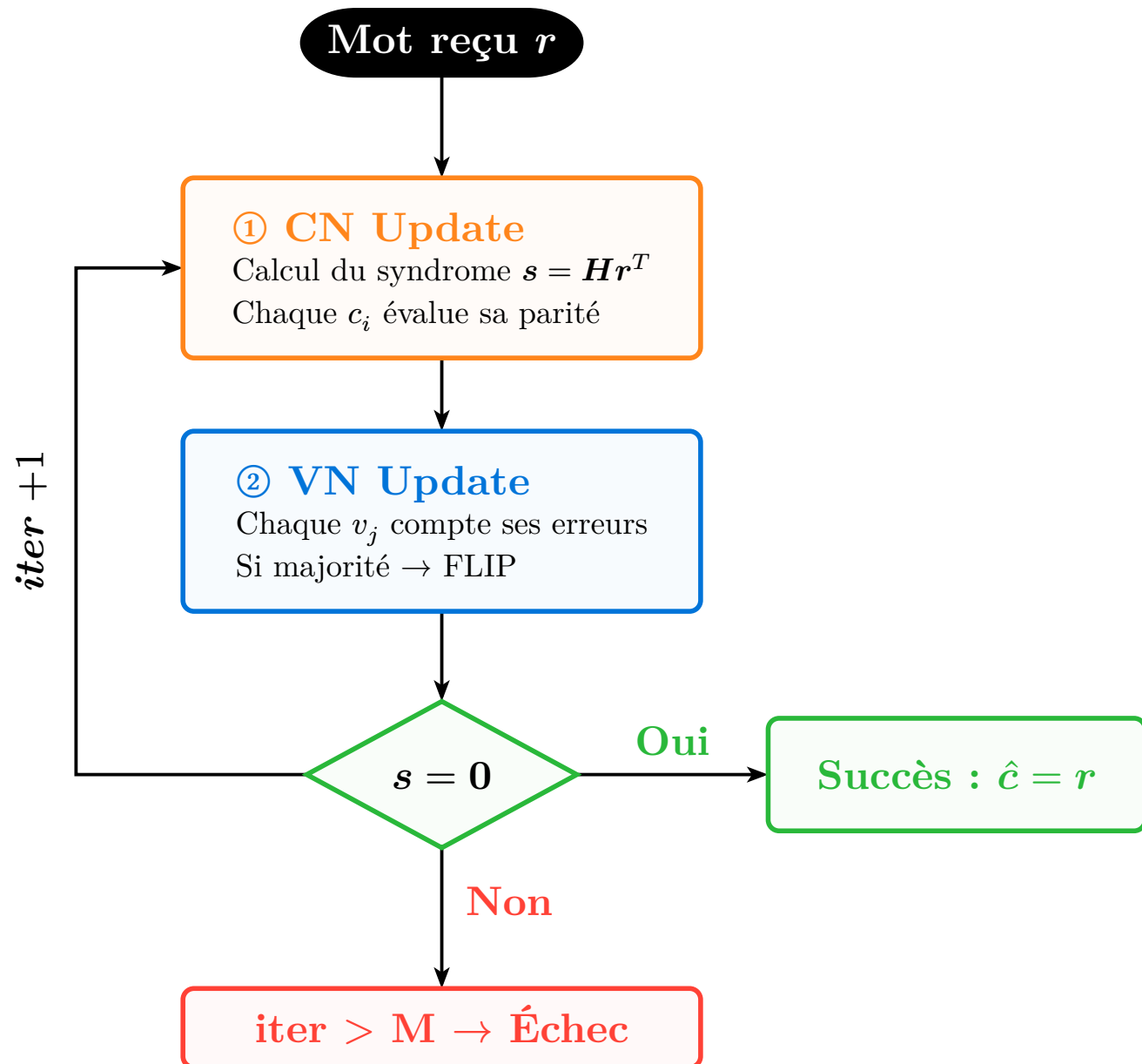
Algorithme **itératif** : les nœuds **échangent des bits** pour localiser les erreurs.

Message Passing

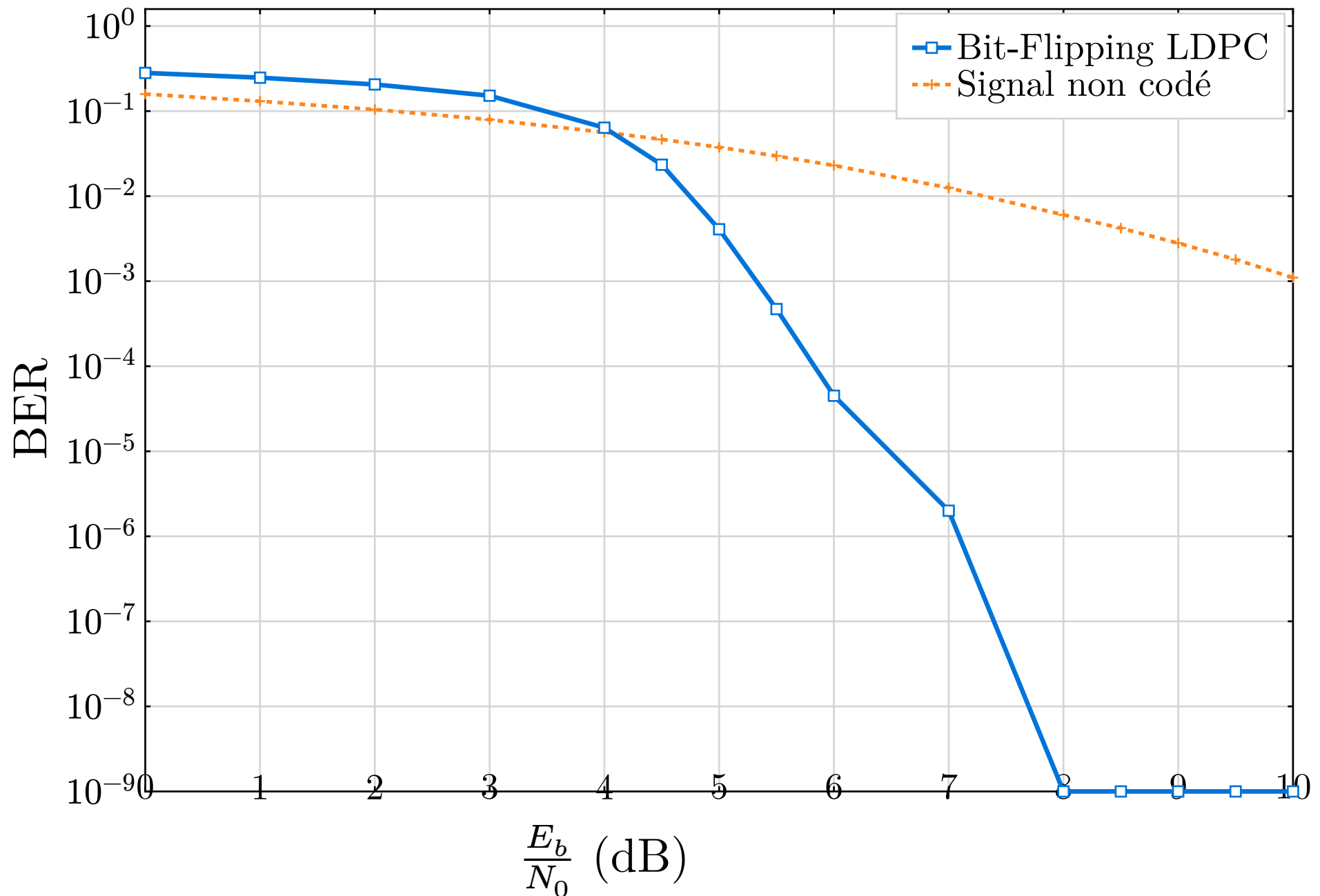
-  envoie son bit courant à ses voisins 
-  renvoie son **verdict de parité** (0 ou 1)
- Si v_j participe à **trop d'équations non satisfaites** \Rightarrow on l'inverse.



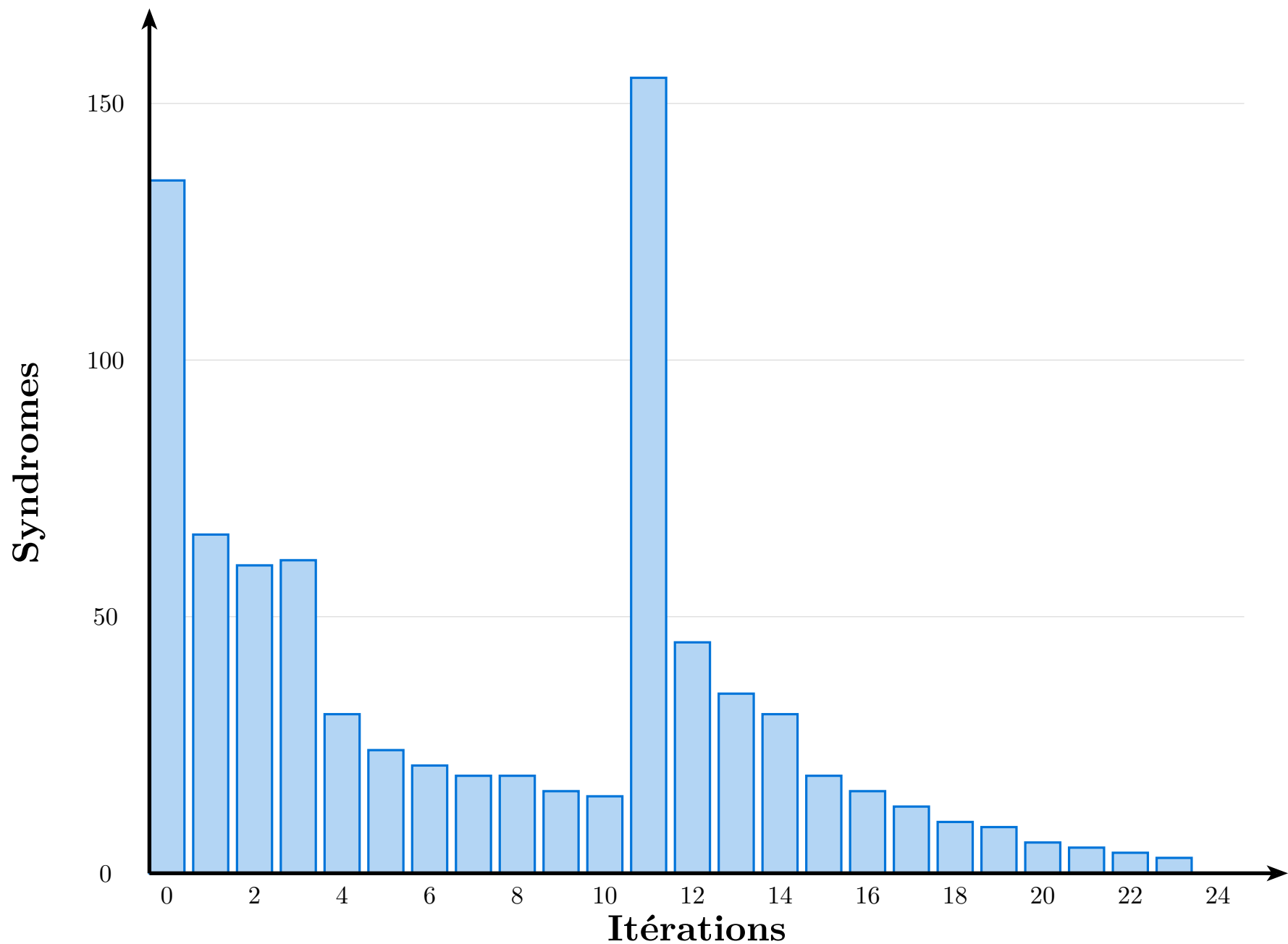
Bit-Flipping : Graphe de flot de contrôle



Waterfall : LDPC (3, 6) $n = 1296$, $k = 648$, $R = \frac{1}{2}$



Résultats : Convergence syndrome



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

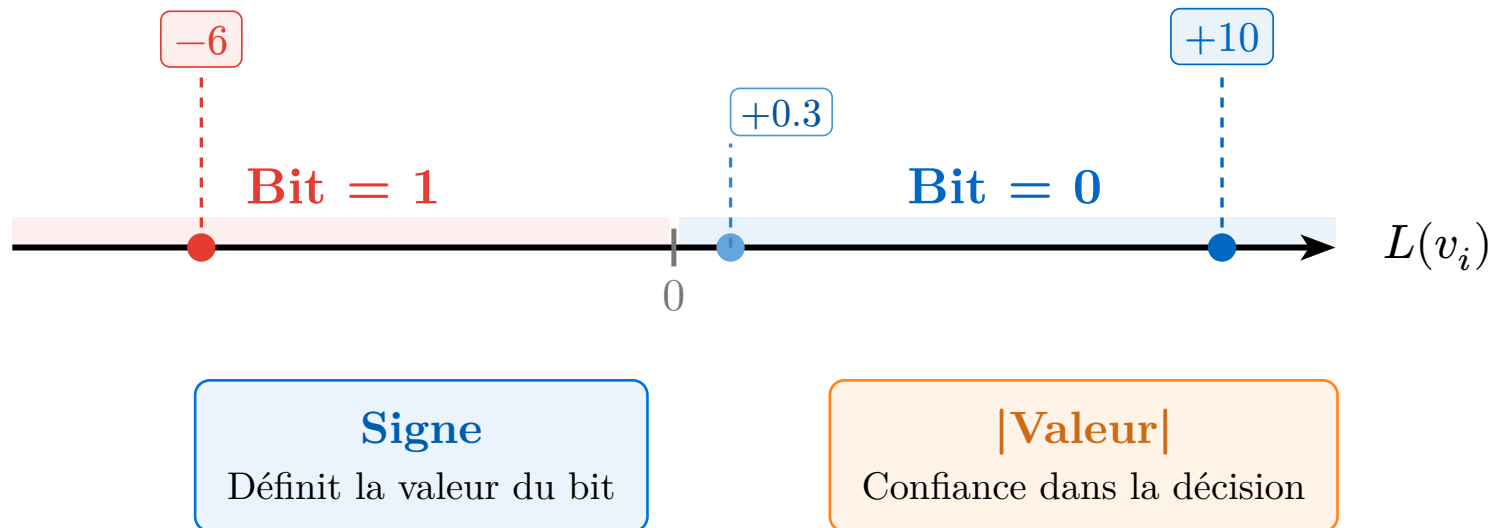
Décodage Soft : Le LLR

Signal

On reçoit une valeur y_i (ex: +4.5V ou -0.2V).

Log-Likelihood Ratio (LLR)

$$L(v_i) = \ln \left(\frac{P(v_i = 0 \mid y_i)}{P(v_i = 1 \mid y_i)} \right)$$



Sum-Product : Belief Propagation

Décodage Optimal

Échange itératif de croyances (LLR) entre les nœuds du graphe

Information Extrinsèque

Exclure l'avis du destinataire pour éviter l'auto-influence

Mise à jour CN

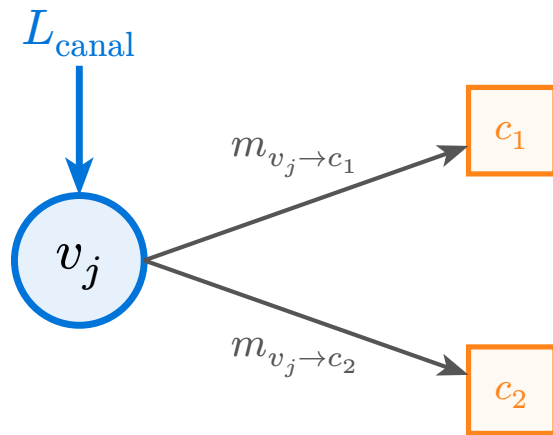
$$m_{c \rightarrow v} = 2 \tanh^{-1} \left(\prod_{u \in \mathcal{N}(c) \setminus \{v\}} \tanh \left(\frac{m_{u \rightarrow c}}{2} \right) \right)$$

Mise à jour VN

$$m_{v \rightarrow c} = L_{v\text{canal}} + \sum_{c' \in \mathcal{N}(v) \setminus \{c\}} m_{c' \rightarrow v}$$

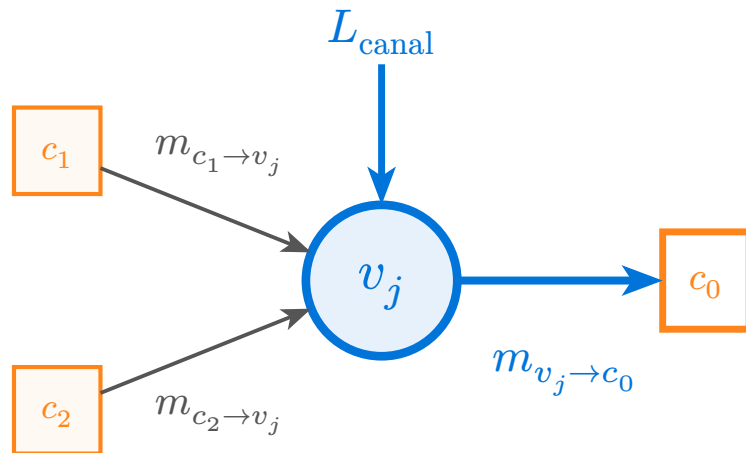
Sum-Product

Initialisation



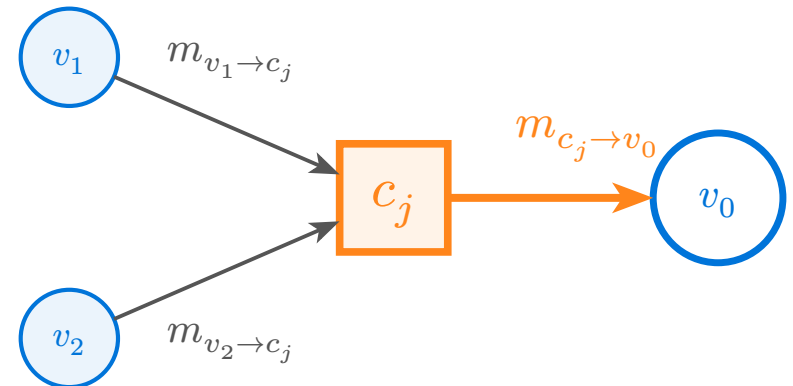
$$m_{v_j \to c_i} = L_{\text{canal}}$$

Échange VN

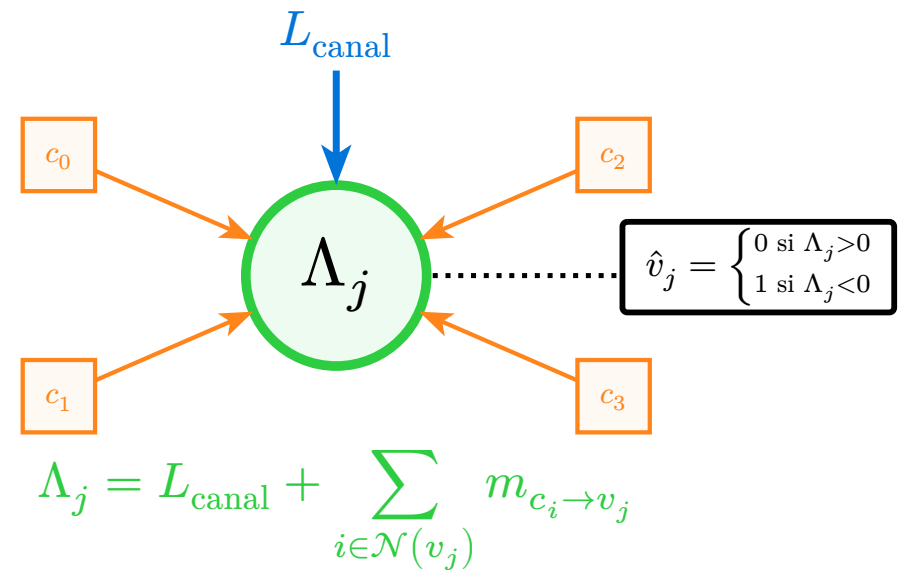


Itérations
 $i = 1, \dots, I_{\max}$

Échange CN

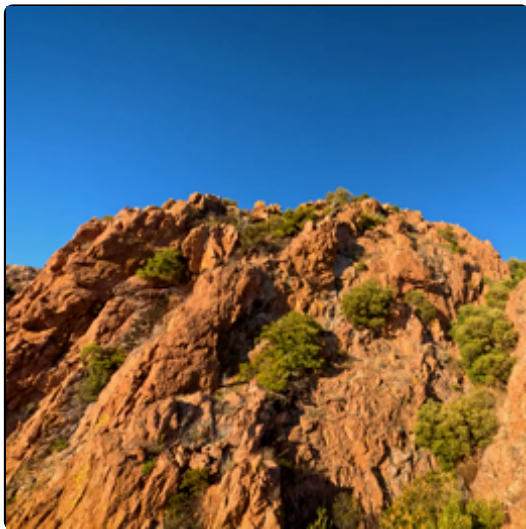


Décision Finale



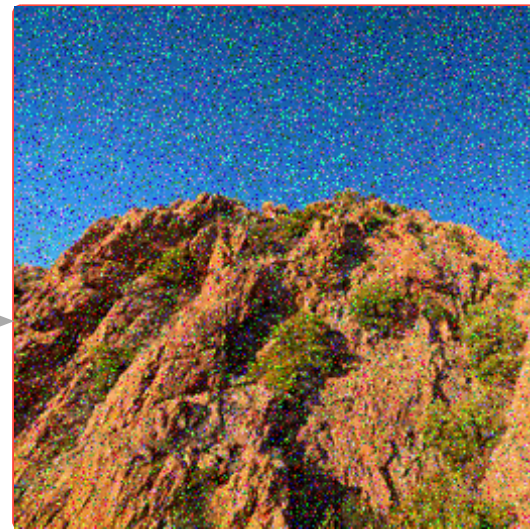
Transmission d'image

Originale

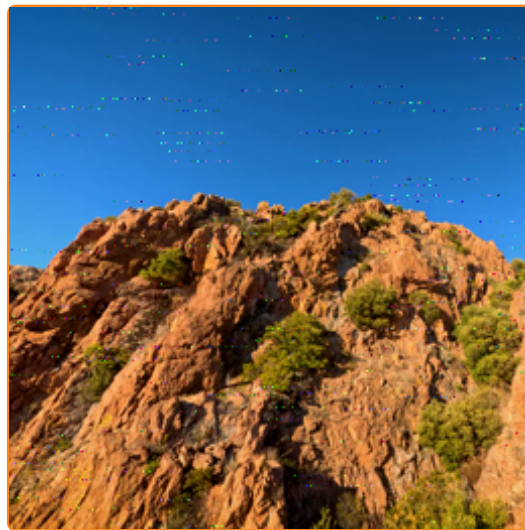


AWGN (2.3 dB)

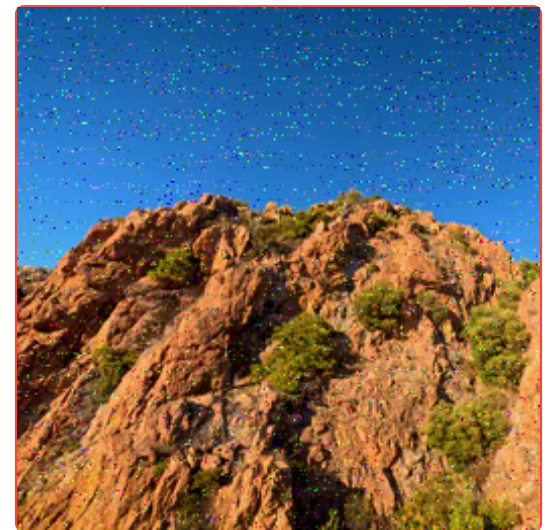
Reçue (Bruité)



$$R = \frac{1}{2}$$



$$R = \frac{2}{3}$$



$$R = \frac{3}{4}$$

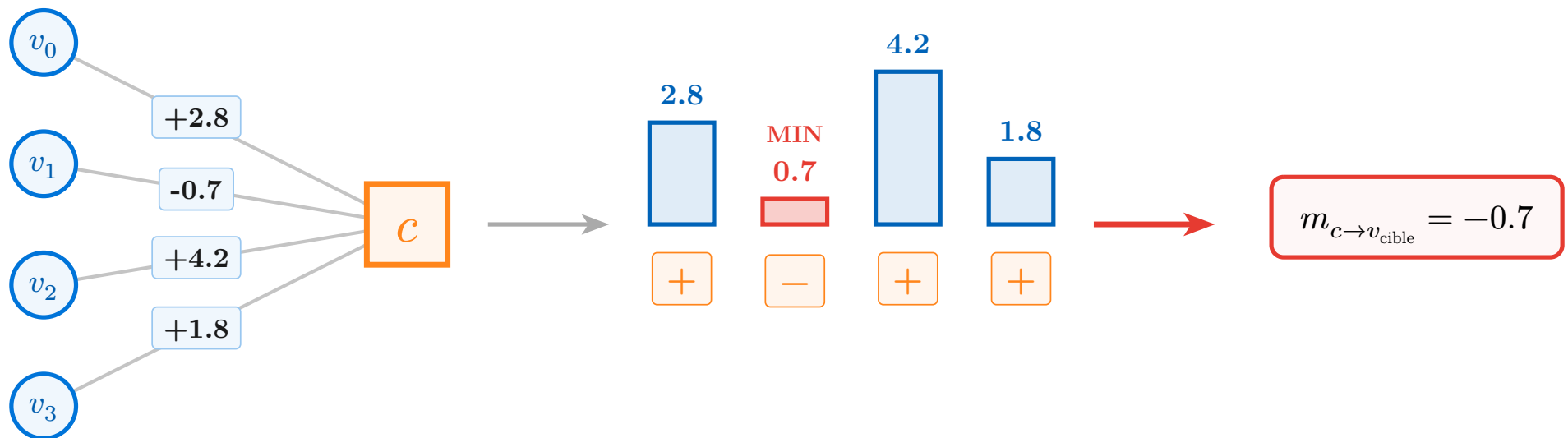
Min-Sum

Avantage Matériel

- **Comparateurs** pour le minimum
- **XOR** pour le produit des signes

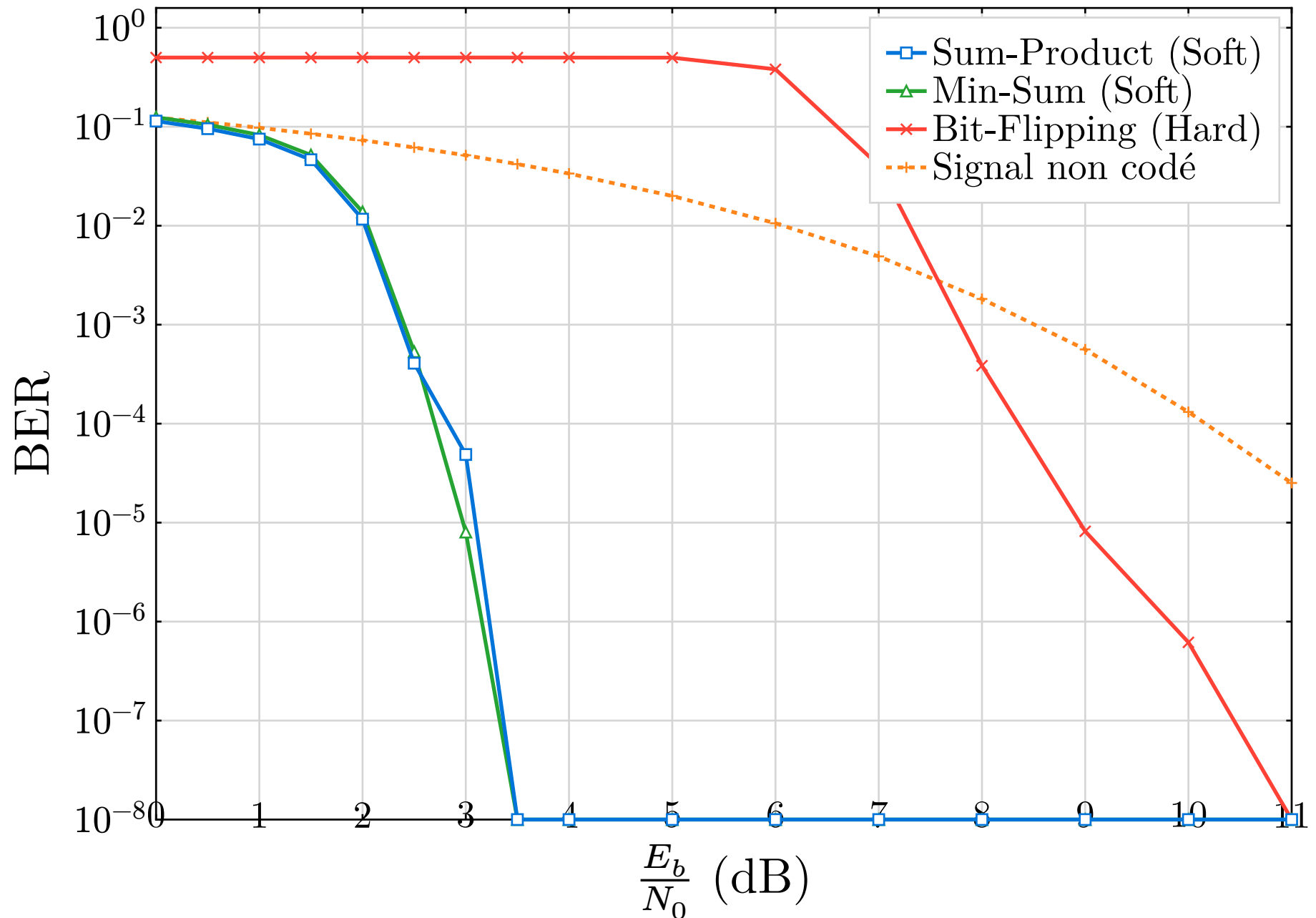
Mise à jour du CN

$$m_{c \rightarrow v_i} = \prod_{j \neq i} \text{sgn}(m_{v_j \rightarrow c}) \times \min_{j \neq i} |m_{v_j \rightarrow c}|$$

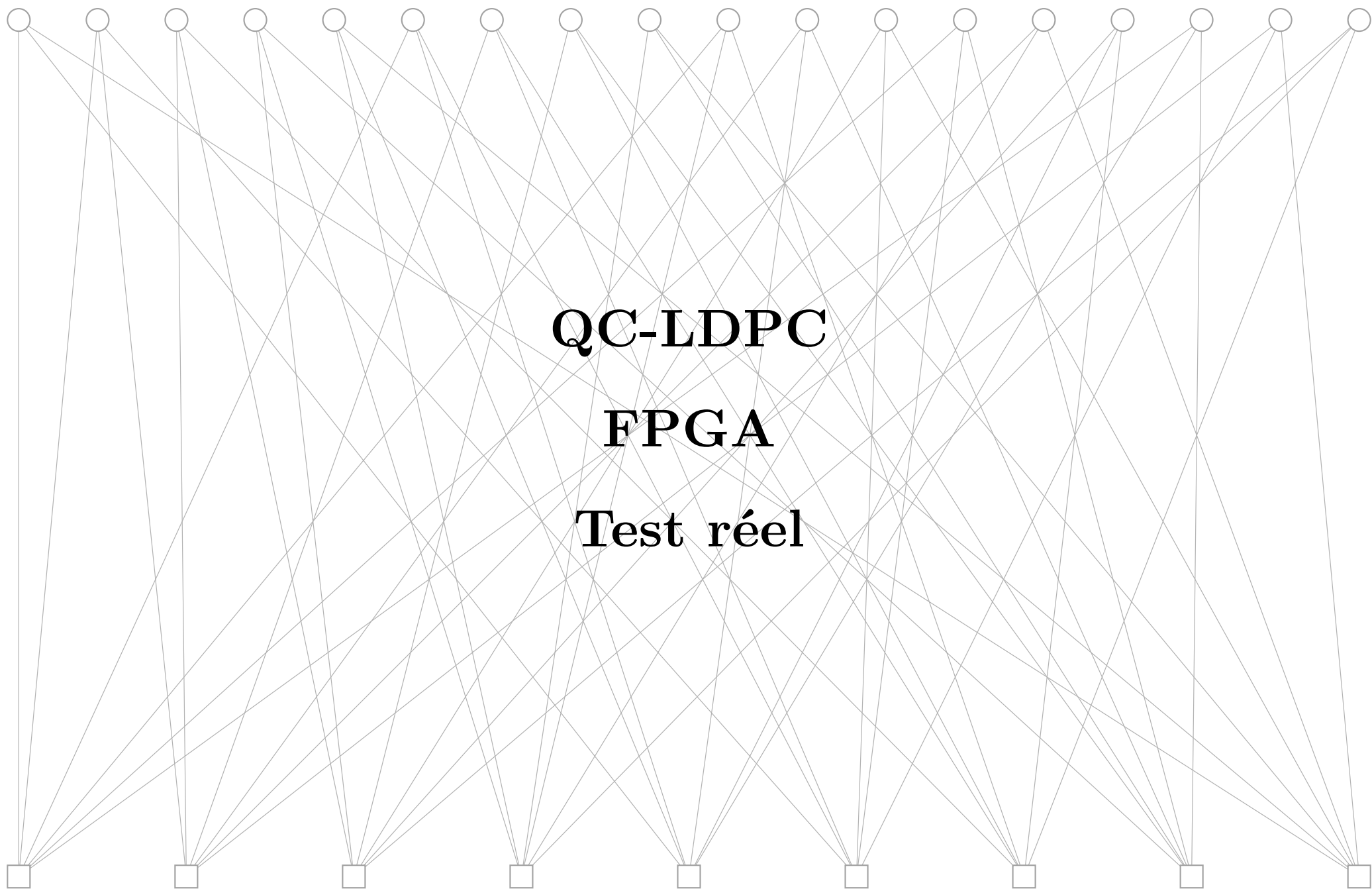


$$\text{Pour les VN : } m_{v \rightarrow c} = L_{\text{canal}} + \sum_{c' \neq c} m_{c' \rightarrow v}$$

Waterfall : LDPC (3, 9) $n = 1296$, $k = 864$, $R = \frac{2}{3}$



Conclusion



Annexe



Théorie derrière la définition des codes linaires

Poser les notations algébriques etc...

Définition du BER et SFR

Decodage par maximum de vraisemblance

Expliquer, quelle distance ? etc

Code LDPC non régulier

CN Update : Formalisme probabiliste

Soit $(V_u)_{u \in \mathcal{N}(c)}$ une famille de variables aléatoires mutuellement indépendantes à valeurs dans \mathbb{F}_2 . On cherche à déterminer la loi du message $R_{c \rightarrow v}(b)$ envoyé par le nœud de contrôle c .

Conditionnement de l'événement de parité

Le message $R_{c \rightarrow v}(b)$ correspond à la probabilité conditionnelle :

$$R_{c \rightarrow v}(b) = P \left(\bigoplus_{u \in \mathcal{N}(c)} V_u = 0 \mid V_v = b \right)$$

Par linéarité du XOR, cette condition est équivalente à :

$$P \left(\bigoplus_{u \in \mathcal{N}(c) \setminus \{v\}} V_u = b \right)$$

Par le théorème des probabilités totales appliqué au système complet d'événements associé aux configurations $x \in \mathbb{F}_2^{d_c-1}$ des voisins :

$$R_{c \rightarrow v}(b) = \sum_{\substack{x \in \mathbb{F}_2^{d_c-1} \\ \bigoplus_{u \in \mathcal{N}(c) \setminus \{v\}} x_u = b}} \prod_{u \in \mathcal{N}(c) \setminus \{v\}} P(V_u = x_u)$$

CN Update (1) : Probabilités

En utilisant les messages entrants du graphe, on définit la probabilité locale $P_{u \rightarrow c}(x_u) = P(V_u = x_u)$.

Le message sortant devient :

$$R_{c \rightarrow v}(b) = \sum_{\substack{x \in \mathbb{F}_2^{d_c-1} \\ \bigoplus_{u \in \mathcal{N}(c) \setminus \{v\}} x_u = b}} \prod_{u \in \mathcal{N}(c) \setminus \{v\}} P_{u \rightarrow c}(x_u)$$

Exacte mais complexité est exponentielle en d_c . On utilise alors une transformation pour simplifier le calcul (Lemme de Gallager).

CN Update (2) : Lemme de Gallager

Lemme de Gallager – Soient (X_1, \dots, X_n) des variables de Bernoulli indépendantes sur \mathbb{F}_2 .

$$P\left(\bigoplus_{i=1}^n X_i = 0\right) = \frac{1}{2} \left(1 + \prod_{i=1}^n E[(-1)^{X_i}]\right)$$

On notes :

$$E[(-1)^{V_u}] = \delta_{u \rightarrow c} = P(V_u = 0) - P(V_u = 1) = 1 - 2P_{u \rightarrow c}(1)$$

On en déduit les probabilités marginales conditionnelles pour le nœud c :

$$R_{c \rightarrow v}(0) = \frac{1}{2} \left(1 + \prod_{u \in \mathcal{N}(c) \setminus \{v\}} \delta_{u \rightarrow c}\right), \quad R_{c \rightarrow v}(1) = \frac{1}{2} \left(1 - \prod_{u \in \mathcal{N}(c) \setminus \{v\}} \delta_{u \rightarrow c}\right)$$

CN Update (3) : Passage aux LLR

Log-Likelihood Ratio (LLR)

Le message LLR entrant $m_{u \rightarrow c}$ au nœud de contrôle est défini par :

$$m_{u \rightarrow c} = \ln \left(\frac{P(V_u = 0)}{P(V_u = 1)} \right)$$

$\delta_{u \rightarrow c}$ s'exprime alors :

$$\delta_{u \rightarrow c} = \tanh \left(\frac{m_{u \rightarrow c}}{2} \right)$$

On en déduit le LLR du message sortant :

$$m_{c \rightarrow v} = \ln \left(\frac{R_{c \rightarrow v}(0)}{R_{c \rightarrow v}(1)} \right) = 2 \tanh^{-1} \left(\prod_{u \in \mathcal{N}(c) \setminus \{v\}} \tanh \left(\frac{m_{u \rightarrow c}}{2} \right) \right)$$

CN Update (4) : Algorithmes

Sum-Product – Mise à jour :

$$m_{c \rightarrow v} = 2 \tanh^{-1} \left(\prod_{u \in \mathcal{N}(c) \setminus \{v\}} \tanh \left(\frac{m_{u \rightarrow c}}{2} \right) \right)$$

Min-Sum – Approximation :

$$m_{c \rightarrow v} \approx \left(\prod_{u \in \mathcal{N}(c) \setminus \{v\}} \operatorname{sgn}(m_{u \rightarrow c}) \right) \times \min_{u \in \mathcal{N}(c) \setminus \{v\}} |m_{u \rightarrow c}|$$

