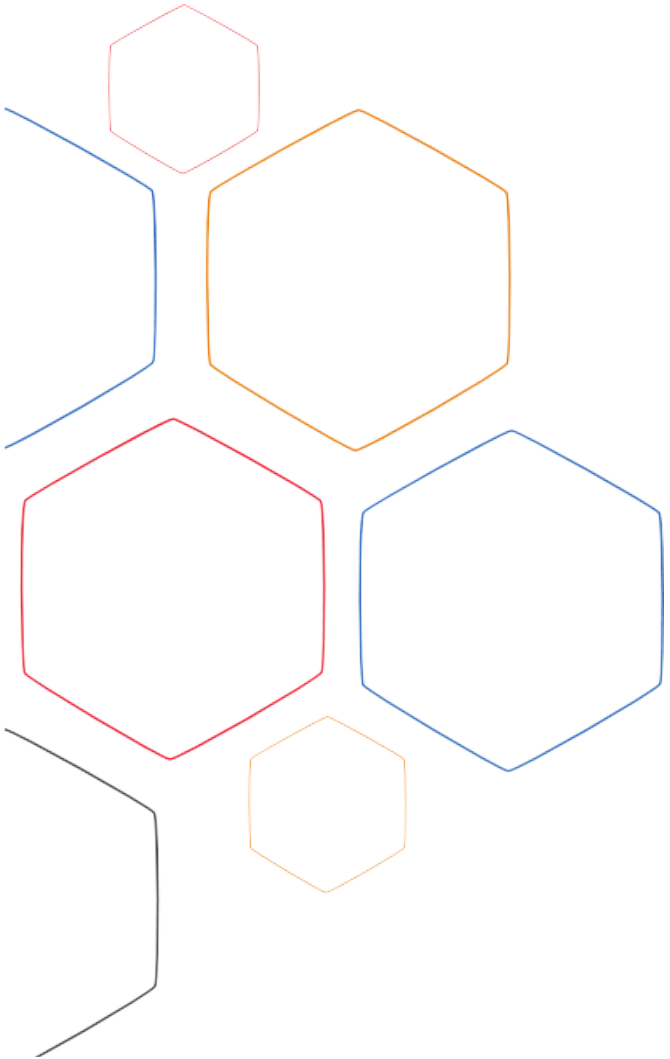
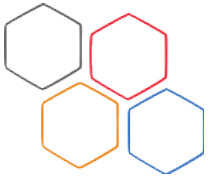


Cours de codes 1 sur 3

Eleonora Guerrini



A code: What for ?

- Recover faulty transmitted data
- Distributed Data Storage
- Conceive Fault Tolerant Algorithms

(ALGO DISTRIBUÉE)

A code: what is it ?

Definition

Un code correcteur est un ensemble de vecteurs (mots) et un couple d'algorithmes (Enc, Dec) qui gèrent la trasmission des mots sur un canal bruité.

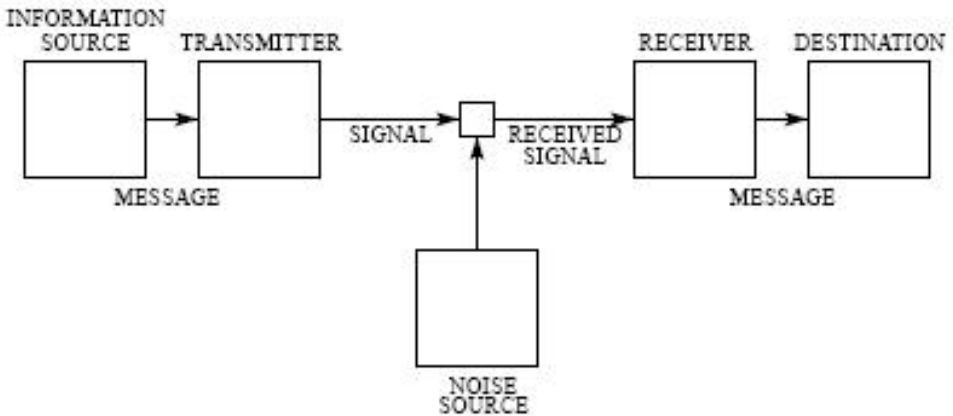
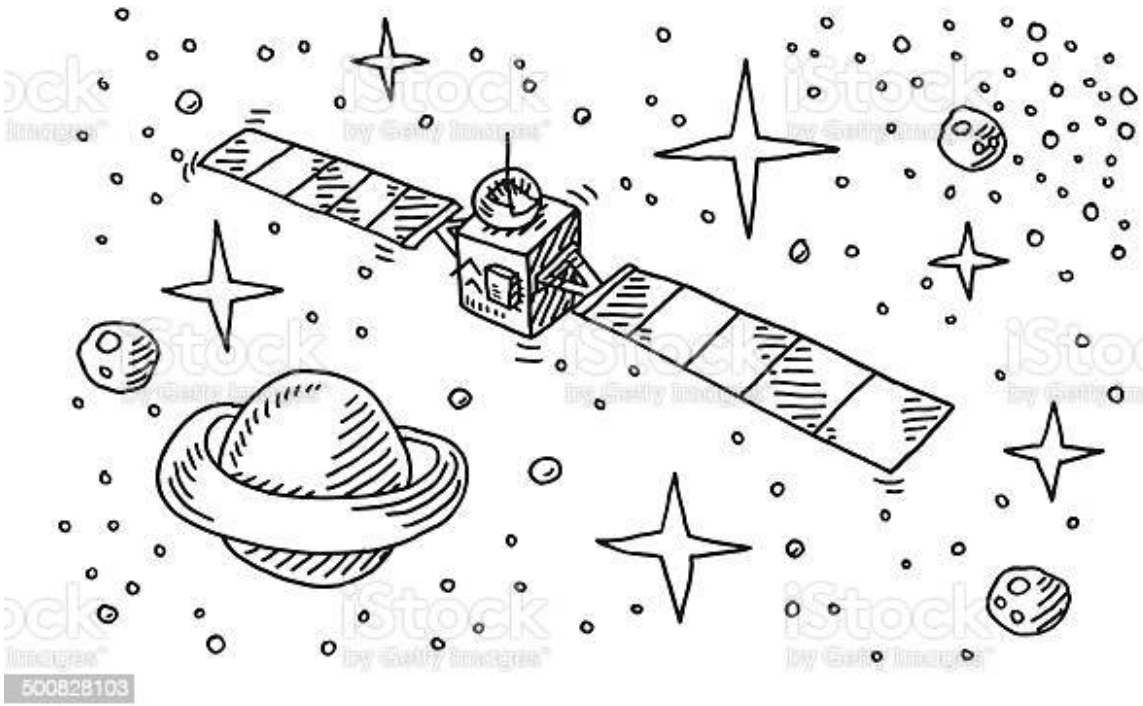


Fig. 1—Schematic diagram of a general communication system.

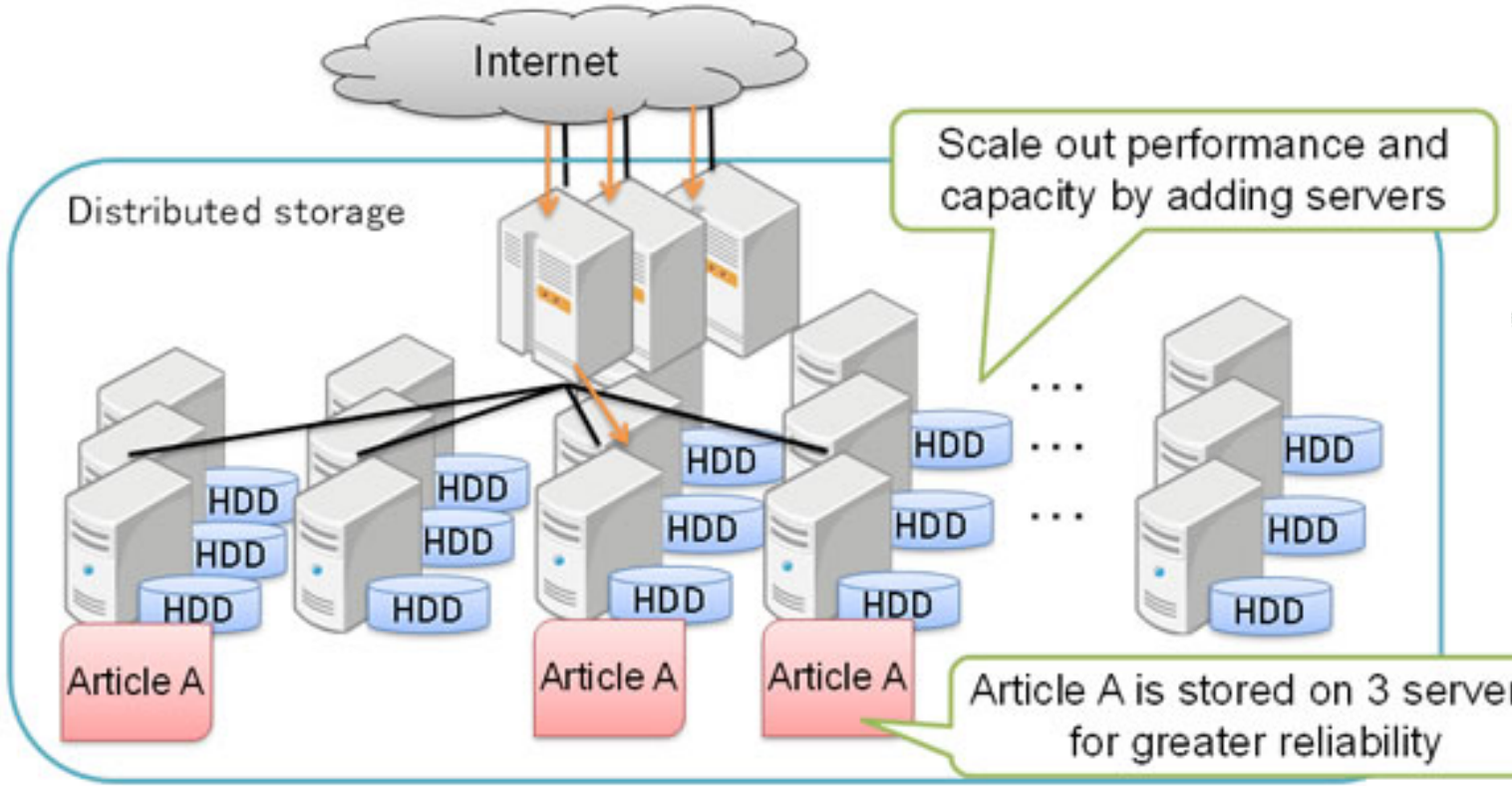
Codes correcteurs pour les transmissions



Dans les cas des Transmissions couteuses, il n'est pas possible de re-transmettre le message en cas de perturbation (ex. Satellites)

⊕ Besoin de corriger même si couteux

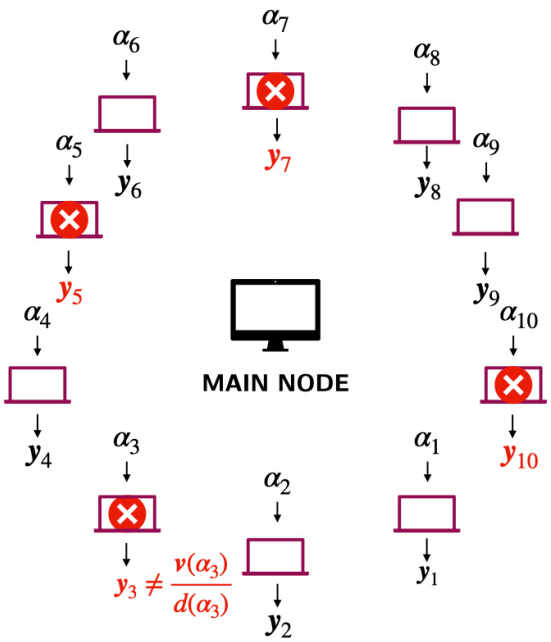
Codes correcteurs dans le stockage d'information



Impossible de re-demander le message endommagé

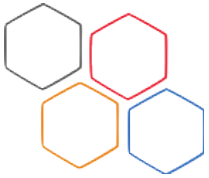
Si un serveur de stockage meurt, il faut avoir la possibilité de recuperer la donnée autrement

Codes correcteurs pour les algorithmes tolerants aux fautes



- cas algo distribuée
- Noeud defectueux
- Noeud malveillant

[ex : Produit matriciel en parallel]



Les questions fondamentales du cours 1

- Modèle: le code, le canal, le bruit
- Algorithmes: Encoder et decoder
- Parametres :Distance d'un code et Theorème de Shannon
- Exemples :code d'Hamming et decodage

Modèle de Shannon: le code, le canal, le bruit

Enc : Encodage $m \in (\mathbb{F}_2)^k \longrightarrow c \in (\mathbb{F}_2)^k$

Dec : Décodage Hamming 4 $\longrightarrow$ k'' $n = 7$

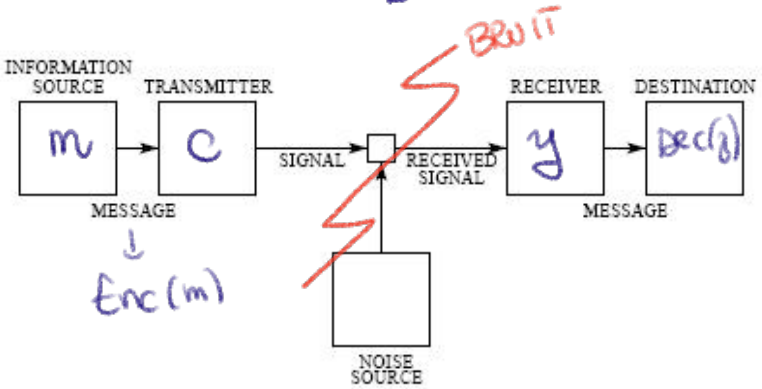


Fig. 1—Schematic diagram of a general communication system.

Bruit: Additif

y reçu $\exists \vec{e} \in (\mathbb{F}_2)^n$ tq

$y = c + e$

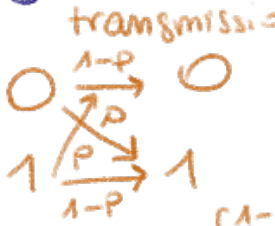
ex $c = (0,0,0,0,0,0,0)$

$y = (1,0,0,0,0,0,0)$

$e = (1,0,0,0,0,0,0)$

$Dec(y) = Enc(m)$

transmission



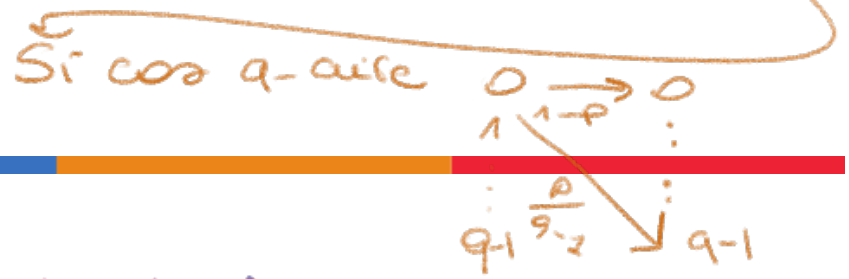
$\begin{bmatrix} 1-p & p \\ p & 1-p \end{bmatrix}$

MATRICE de transition

CANAL :

- BSC (Binary Symmetric Channel)
  - Sans mémoire $P_r(y_i \neq c_i)$ est indep de $P_r(y_{i-1} \neq c_{i-1})$ autrement dit $Pr(y_i | c_i)$ est indep de $Pr(y_{i-1} \neq c_{i-1})$
  - P : Proba qu'il y a un flip de bit $0 < P < 1/2$ (sinon on inverse)
  - Symmetrique $P_r(c_i \text{ flip de } 0 \to 1) = P_r(c_i \text{ flip de } 1 \to 0)$

Encoding and Decoding



$Enc(m) : (\mathbb{F}_2)^k \longrightarrow (\mathbb{F}_2)^n$

$\{0,1\}^k \longrightarrow \{0,1\}^n$ $n > k$ redondance

- Injective
- $e \in (\mathbb{F}_2)^n$

↳ $Dec(m)$ et $Enc(m)$ on le choisit linéaire pour "efficacité"

$Enc(m)$ linéaire $\Rightarrow$ Algo d'algèbre linéaire pour encoder m

↳ C est un sous-esp. vect. de $(\mathbb{F}_2)^n$ de dim k

Theorème de Shannon

"GRAAL de la théorie de Codes correcteur

p loi de proba (binaire) $H(p) = -p\log_2(p) - (1-p)\log_2(1-p)$

Formulation "simple" du theore

- tout canal est toute CAPACITÉ (max inform. qu'on peut transmettre de façon fiable)

- $C_{sp} = 1 - H(p)$ (correction)
- R taux de transmission (redondance)
- $\frac{k}{n}$ rendement du code (rate)

dim esp mess → k ; longueur code → n

code Hamming [n=7, k=4 → 4/7 ? corrige 1 err.]

th Shannon

C_{sp} fixé $= 1 - H(p)$, p proba d'erreur du canal

$0 < p < 1/2$

$\varepsilon \leq 1/2 - p$

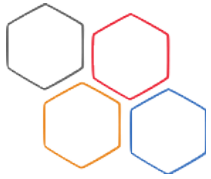
→ Pour $n \gg 0$

Pour envoyé et y reçu.

1. $\exists \delta > 0$ (Enc, Dec)

si $k \leq \lfloor (1 - H(p) + \varepsilon) \cdot n \rfloor$ alors

$- Pr(Dec(y_{reçu}) \neq 0) \leq 2^{-\delta n}$



2. Pour $k > \lceil (1-H(p)+\varepsilon)n \rceil$ pour tout (Enc, Dec) $\exists c \in \{0,1\}^n$ tq $P_n(Dec(y) \neq c) > 1/2$

Definition Code et Parametres

- Soit Σ un alphabet, $(\Sigma)^k$ l'espace des messages et k et n des entiers naturels tels que $k \leq n$.
- $C \subset (\Sigma)^n$:
- Linearité:
- On appelle k la dimension du code
- Rate:
- Distance (?) Hamming :

$Enc(m) : (\mathbb{F}_2)^k \longrightarrow (\mathbb{F}_2)^n$

• $Enc(0) = 0$ • $Enc(m_1 + m_2) = Enc(m_1) + Enc(m_2)$

$Enc((\mathbb{F}_2)^k)$ est esp. vect de $(\mathbb{F}_2)^n$

k/n

$x = (x_1 \dots x_n)$ $y = (y_1 \dots y_n)$

$d_H(x,y) = \sum (x_i + y_i)$

- $d_H(x,x) = 0$
- $d_H(x,y) = d_H(y,x)$
- $d_H(x,y) + d(y,z) \geq d_H(x,z)$

- Rapp. avec une base, G matrice base
- $Enc(m) = m \cdot G$

Hamming $k = 4$ $(\mathbb{F}_2)^4$

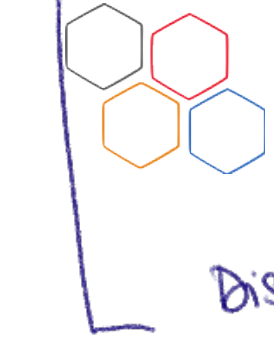
$mG = (1,0,1,1,0,1,0) \in (\mathbb{F}_2)^7$

mot de code de Hamming

$m = (1,0,1,1)$; $G = \begin{bmatrix} 1&0&0&0&1&1&0 \\ 0&1&0&0&1&0&1 \\ 0&0&1&0&1&1&1 \\ 0&0&0&1&0&1&1 \end{bmatrix}$

$x = (0,0,1)$

$d_H(x,0) = 2$



y = (1,0,0)
z = (1,0,1)
(preuve vue)

• Correction Détection Effacements

Distance (d'Hamming) d'un code C

$$\min_{x_1 \neq x_2} \left(d(x_1, x_2) \mid x_1, x_2 \in C \right)$$

Quantifions les erreurs avec la distance et la définition du code

Detection et Correction: Modèle de hamming

$y = (y_1, \dots, y_n)$
$c = (c_1, \dots, c_n)$

Philosophie de décodage MLD
Maximun Likelihood Decoding (maximum de vraisemblance)

Problematique et Algorithmes: Rôle de la distance

Si y reçu, on veut $Dec(y) = c$

$P_r\{y \text{ reçu} \mid c \text{ envoyé}\} = \prod_{i=1}^{n} P_r\{y_i \text{ reçu} \mid c_i \text{ envoyé}\}$

$P_r(y \text{ reçu} \mid c \text{ envoyé})$ plus grande possible

si $y_i = c_i$: $n - d(y,c)$ fois
$y_i \neq x_i$: $d(y,c)$ fois

$n - d(y,c) = \#\{i \mid y_i = c_i\}$

$(1-p) \cdot p^{d(y,c)}$

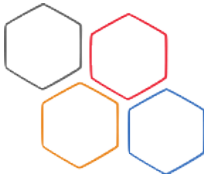
$c = 1\ 0\ 0$
$y = 0\ 1\ 0$

$p^{d(c,y)}$

Algorithme de decodage naif : MLD Algorithme

$(1-p)^n \cdot \left(\frac{p}{1-p}\right)^{d_H(y,c)}$

plus grand possible $\Rightarrow$ $d_H(y,c)$ plus petite possible



MLD decoder : bons et mauvais cotés

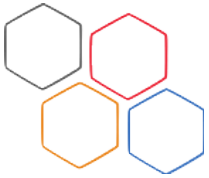
MLD Problème NP-hard

In : $y \in \mathbb{F}_2^n$, $\mathcal{C}$ code ($\mathcal{C} \in (\mathbb{F}_2)^n$ de dim k)

Out : $c \in \mathcal{C}$ tq $d(y,c) = \min\{d(y,c) \mid c \in \mathcal{C}\}$

(même si code lin)

On se restraint à un cas plus precis où on met une borne sur ce qu'on peut corriger



Modèles de Décodage

BDD Decoder

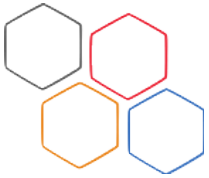
Bounded Decoding Distance

In : y reçu, $\mathcal{C}$ code, t borne sur les erreurs qu'on veut (peut-être) corriger

Out : c tq $d_H(c, y) \leq t$

BMD Decoder

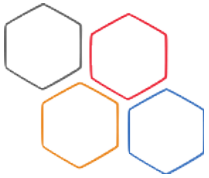
FIN COURS 1-



Code de repetition et Code de parité

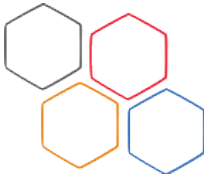
- Code de repetition :

- Codes de parité :

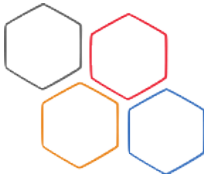


Ex : Code de repetition et parity check

- dimension du code
- longueur du code
- rendement du code : ratio k/n

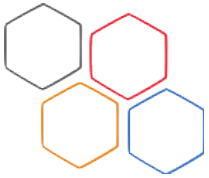


Décodeurs



Exo

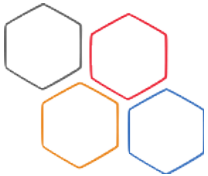
- Code de repetition peut corriger 1 erreur
- Code de parité peut detecter un nombre impair d'erreurs



Distance et Correction: Exo

Given a code C de longueur n et dimension k, les assertions suivantes sont équivalents

1. *C a distance minimale* $d \geq 2$,
2. *si d est impair, C peut corriger* $\frac{(d-1)}{2}$ *errors.*
3. *C peut detecter* $d - 1$ *errors.*
4. *C peut corriger* $d - 1$ *effacement.*



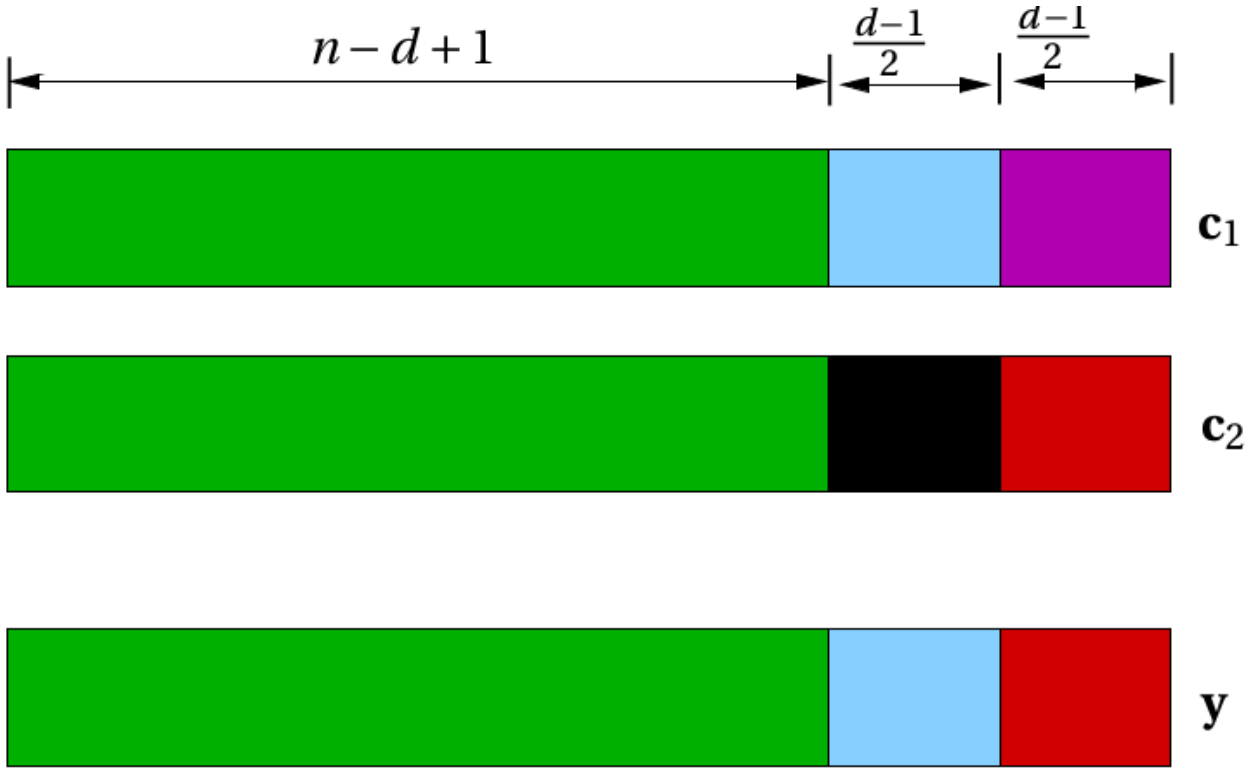
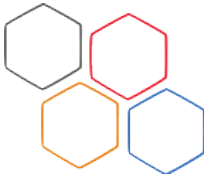
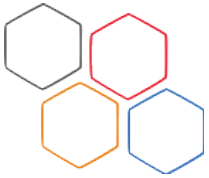


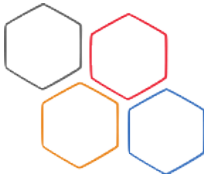
Figure 1.3: Bad example for unique decoding.



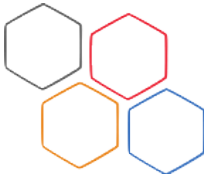
Qu'est-ce qu'on peut esperer comme rate



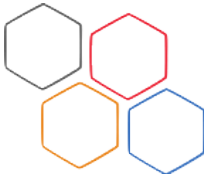
Code d'Hamming



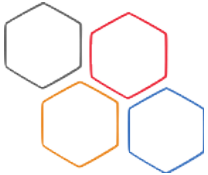
Codes linéaires



Matrice generatrice et de parité



Décodage d'un code d'Hamming



Exemple de Décodage