Introduction à la théorie de l'information et à la théorie du codage

Sebastien.Kramm@univ-rouen.fr

IUT de Rouen, dept. SRC

2012-2013

Sommaire

UNIVERSITÉ DE ROUEN

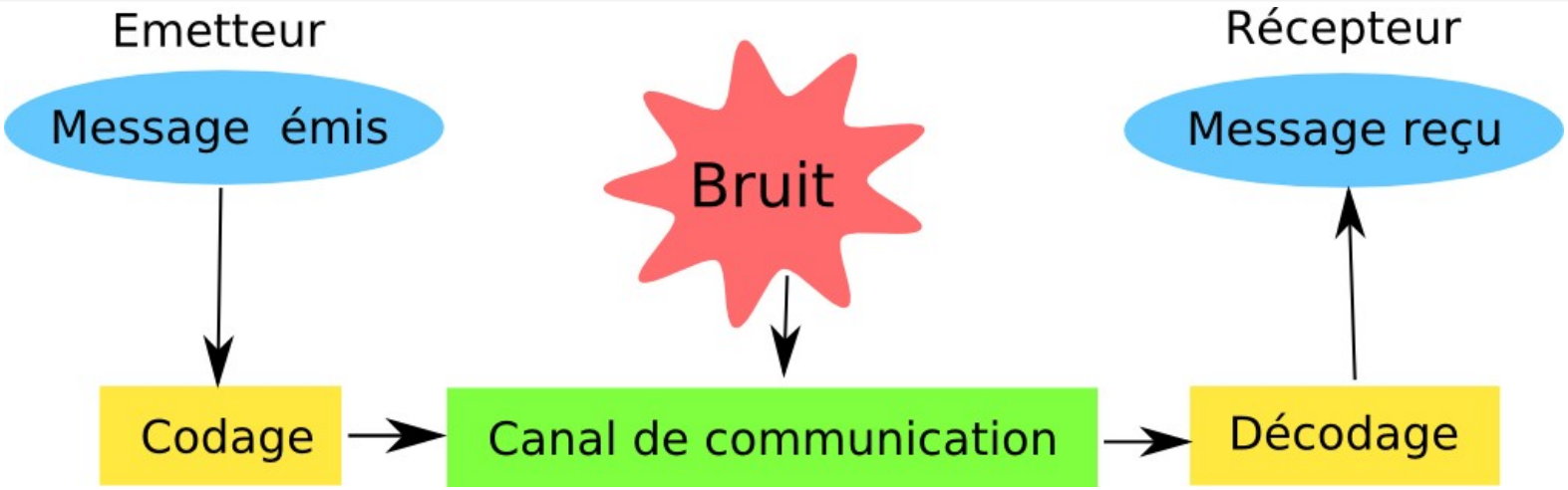
Sous-sommaire

UNIVERSITÉ DE ROUEN

Communication ?

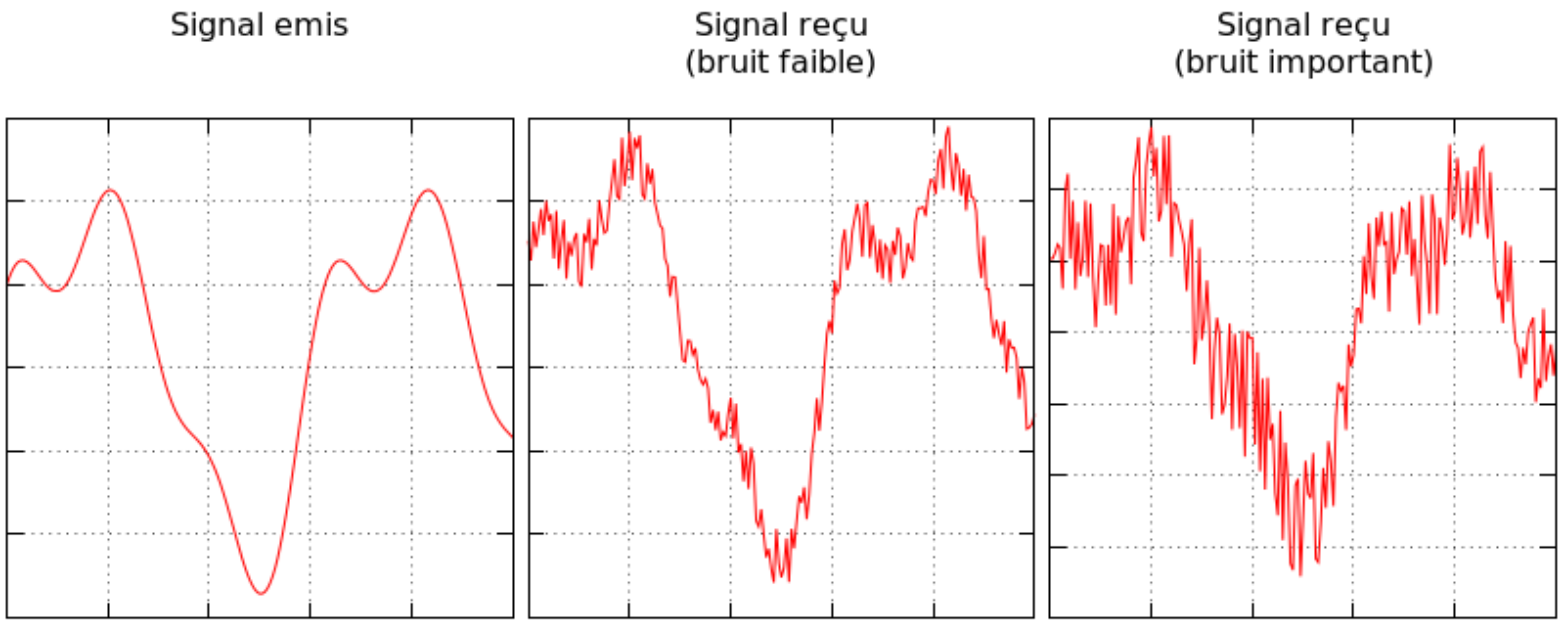
Définition

Echange d'information (sous forme de signal) entre un émetteur et un récepteur à l'aide d'un canal de communication.



Bruit de communication

- Lorsqu'on envoie un message de l'émetteur au récepteur, le message subit une déformation pendant le transport, c'est le bruit.



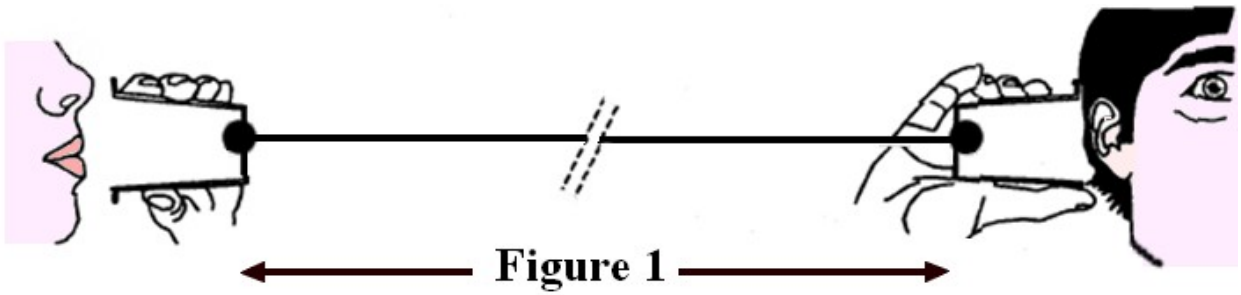
- Remarque : le bruit peut être déterministe ou aléatoire.

Exemple de canaux de communication

- Canal de communication :
  - Conversation humaine : air,
  - liaison informatique Wi-Fi : air puis câble réseau,
  - liaison informatique filaire : câble réseau,
- Codage de l'information, sous la forme d'un signal physique :
  - Conversation humaine : son de la voix (onde sonore),
  - liaison informatique Wi-Fi : onde radio-électrique,
  - liaison informatique filaire : trame ethernet (signal électrique sur un câble).

Exemple de canaux de communication

- Canal de communication :
  - Conversation humaine : air,
  - liaison informatique Wi-Fi : air puis câble réseau,
  - liaison informatique filaire : câble réseau,
- Codage de l'information, sous la forme d'un signal physique :
  - Conversation humaine : son de la voix (onde sonore),
  - liaison informatique Wi-Fi : onde radio-électrique,
  - liaison informatique filaire : trame ethernet (signal électrique sur un câble).



(schéma du yaourtophone)

Codage

- Le codage consiste à **adapter** le signal au canal, de façon à
  - Rendre la communication possible
  - Fiabiliser la communication (détection et/ou correction d'erreur)
  - Optimiser la communication (minimiser le coût, le temps, la complexité, ...)
- Pour les transmissions analogiques, le codage peut-être une modulation (radio...)
- Pour les transmission numériques, le codage utilise un **alphabet**.

UNIVERSITÉ DE ROUEN

Concept d'alphabet

- Un nombre désigne une quantité. On peut lui associer diverses représentations à l'aide de symboles, appartenant à des alphabets.
- Exemple : dix, X et 10 sont trois représentations du nombre 10 par des symboles.
  - dix : avec l'alphabet latin $A = \{a\,;\,b\,;\,c\,;\,d\,;\,e\,;\,f\,;\,g\,;\,h\,;\,i\,;\,j\,;\,k\,;\,l\,;\,m\,;\,n\,;\,o\,;\,p\,;\,q\,;\,r\,;\,s\,;\,t\,;\,u\,;\,v\,;\,w\,;\,x\,;\,y\,;\,z\}$;
  - X : avec l'alphabet des chiffres romains $A = \{I\,;\,V\,;X\,;\,L\,;C\,;D\,;M\}$;
  - 10 : avec l'alphabet des chiffres arabes $A = \{0\,;\,1\,;\,2\,;\,3\,;\,4\,;\,5\,;\,6\,;\,7\,;\,8\,;\,9\,\}$.
- Ces représentations sont des mots (des suites finies de symboles) définis sur les alphabets A.
- De ces trois écritures, la dernière est la plus pratique pour effectuer des calculs.

Sous-sommaire

UNIVERSITÉ DE ROUEN

Information ?

- L'information désigne à la fois le message à communiquer et les symboles utilisés pour l'écrire.
- Elle utilise un code de signes porteurs de sens, qu'on appelle un **alphabet**.
- Comment mesurer, quantifier l'information ?
  - Soit la phrase : "le soleil se couchera ce soir"
    $\rightarrow$ L'information contenue est faible...

UNIVERSITÉ DE ROUEN

Information ?

- L'information désigne à la fois le message à communiquer et les symboles utilisés pour l'écrire.
- Elle utilise un code de signes porteurs de sens, qu'on appelle un **alphabet**.
- Comment mesurer, quantifier l'information ?
  - Soit la phrase : "le soleil se couchera ce soir"
    $\rightarrow$ L'information contenue est faible...
  - Soit la phrase : "le soleil se couchera vers 20h"
    $\rightarrow$ L'information contenue est plus importante.

UNIVERSITÉ DE ROUEN

Information ?

- L'information désigne à la fois le message à communiquer et les symboles utilisés pour l'écrire.
- Elle utilise un code de signes porteurs de sens, qu'on appelle un **alphabet**.
- Comment mesurer, quantifier l'information ?
  - Soit la phrase : "le soleil se couchera ce soir"
    $\rightarrow$ L'information contenue est faible...
  - Soit la phrase : "le soleil se couchera vers 20h"
    $\rightarrow$ L'information contenue est plus importante.
  - Soit la phrase : "le soleil se couchera à 19h52"
    $\rightarrow$ L'information contenue est encore plus importante.

UNIVERSITÉ DE ROUEN

Information ?

- L'information désigne à la fois le message à communiquer et les symboles utilisés pour l'écrire.
- Elle utilise un code de signes porteurs de sens, qu'on appelle un **alphabet**.
- Comment mesurer, quantifier l'information ?
  - Soit la phrase : "le soleil se couchera ce soir"
    $\rightarrow$ L'information contenue est faible...
  - Soit la phrase : "le soleil se couchera vers 20h"
    $\rightarrow$ L'information contenue est plus importante.
  - Soit la phrase : "le soleil se couchera à 19h52"
    $\rightarrow$ L'information contenue est encore plus importante.

UNIVERSITÉ DE ROUEN

Information ?

- L'information désigne à la fois le message à communiquer et les symboles utilisés pour l'écrire.
- Elle utilise un code de signes porteurs de sens, qu'on appelle un **alphabet**.
- Comment mesurer, quantifier l'information ?
  - Soit la phrase : "le soleil se couchera ce soir"
    $\rightarrow$ L'information contenue est faible...
  - Soit la phrase : "le soleil se couchera vers 20h"
    $\rightarrow$ L'information contenue est plus importante.
  - Soit la phrase : "le soleil se couchera à 19h52"
    $\rightarrow$ L'information contenue est encore plus importante.

Intuition

Plus la source émet d'informations différentes, plus l'entropie (ou incertitude sur ce que la source émet) est grande.

UNIVERSITÉ DE ROUEN

Information ?

- L'information désigne à la fois le message à communiquer et les symboles utilisés pour l'écrire.
- Elle utilise un code de signes porteurs de sens, qu'on appelle un **alphabet**.
- Comment mesurer, quantifier l'information ?
  - Soit la phrase : "le soleil se couchera ce soir"
    $\rightarrow$ L'information contenue est faible...
  - Soit la phrase : "le soleil se couchera vers 20h"
    $\rightarrow$ L'information contenue est plus importante.
  - Soit la phrase : "le soleil se couchera à 19h52"
    $\rightarrow$ L'information contenue est encore plus importante.

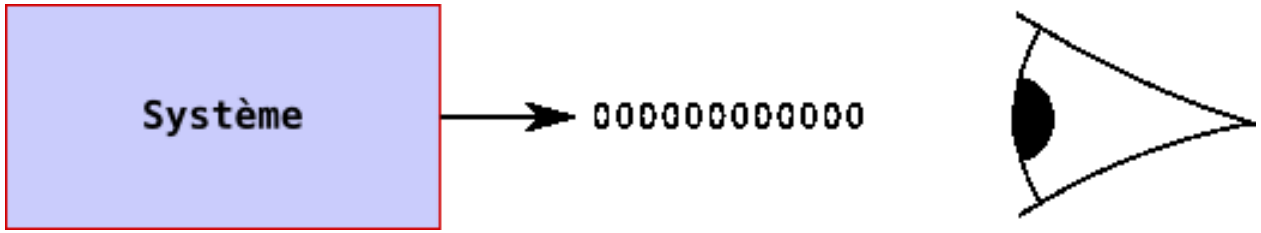
Intuition

Plus la source émet d'informations différentes, plus l'entropie (ou incertitude sur ce que la source émet) est grande.

- Tout ceci est formalisé dans la **théorie de l'information**, initiée par Claude Shannon (1916-2001), et basé sur la théorie des probabilités.

Exemple : système à un seul état

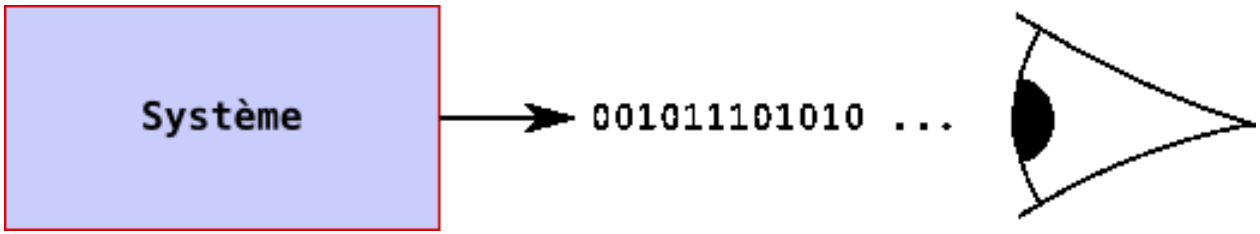
- Soit un système qui fournit en sortie une série de symboles identiques (des '0').
- La probabilité d'apparition d'un 0 en sortie de ce système est donc $p(0) = 1$.
- Ce système n'apporte aucune information car on sait toujours à coup sûr ce qui va sortir du système.



source : Jérôme Landré

Exemple : système à deux états équiprobables

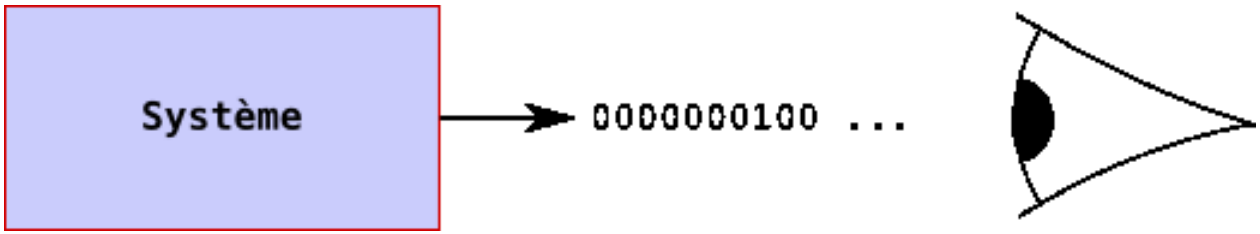
- Soit un système qui fournit en sortie un message composé de deux symboles (0 ou 1).
- On considère que les deux symboles ont autant de chance d'apparaître : on dit qu'ils sont **equiprobables**.
- On a donc $p(0) = 0.5$ et $p(1) = 0.5$
  (la somme des probabilités de tous les symboles doit être égale à 1)
- Lorsqu'on observe la sortie du système, on s'attend à voir autant de 0 que de 1.



source : Jérôme Landré

Exemple : système à deux états non-équiprobables

- On considère l'évènement "un 0 sort du système" plus probable que l'évènement "un 1 sort du système"
- On suppose $p(0) = 0.9$ et $p(1) = 0.1$. Lorsqu'on observe le message en sortie du système, on s'attend à voir plus souvent des 0 que des 1,
- Quand un 1 sort, il a une valeur plus importante, il apporte **plus d'information** que les 0.



source : Jérôme Landré

Quantité d'information

- Soit une source d'information S qui fournit une suite de symboles x dont la probabilité d'apparition de chacun d'eux est $p(x)$.
- La **quantité d'information** associée à chaque symbole x s'exprime en bits et s'écrit :

$$I(x) = \log_2 \frac{1}{p(x)} = -\log_2 p(x)$$

  - Si le symbole n'apparait presque jamais ($p(x) \sim 0$), alors $I(x) = \infty$ $\Rightarrow$ il porte un grande quantité d'information.
  - Si le symbole est "certain" ($p(x) \sim 1$), alors $I(x) = 0$ $\Rightarrow$ l'information transmise est nulle.

Quantité d'information

- Soit une source d'information S qui fournit une suite de symboles x dont la probabilité d'apparition de chacun d'eux est $p(x)$.
- La **quantité d'information** associée à chaque symbole x s'exprime en bits et s'écrit :

$$I(x) = \log_2 \frac{1}{p(x)} = -\log_2 p(x)$$

  - Si le symbole n'apparait presque jamais ($p(x) \sim 0$), alors $I(x) = \infty$ $\Rightarrow$ il porte un grande quantité d'information.
  - Si le symbole est "certain" ($p(x) \sim 1$), alors $I(x) = 0$ $\Rightarrow$ l'information transmise est nulle.

Rappel

$$\log_2(a) = \frac{\ln(a)}{\ln(2)} = \frac{\log(a)}{\log(2)}$$

Entropie

- L'entropie $H(x)$ d'une source S fournissant un message défini sur un alphabet à i symboles x_i s'exprime en bits, et vaut :

$$H(x) = \sum_i p(i) I(x_i) = -\sum_i p(i) \log_2[p(i)]$$

  avec $p(i)$ la probabilité d'apparition de chaque symbole.
- L'entropie correspond au nombre de bits minimal (au sens du **codage optimal**) nécéssaire pour coder un message avec cette source.

Exemples de calcul de l'entropie

- Soit une source fournissant les deux symboles 0 et 1 de façon équiprobable $\Rightarrow$ p(0)=1/2, p(1)=1/2

$$\begin{aligned} H(x) &= -p(0)\log_2[p(0)] - p(1)\log_2[p(1)] \\ &= -1/2 \cdot \log_2[1/2] - 1/2 \cdot \log_2[1/2] \\ &= -\log_2[1/2] = \log_2[2] \\ &= 1 \end{aligned}$$

$\Rightarrow$ Le nombre de bits nécessaire est de 1, ce qui correspond à l'intuition (2 états possibles, donc 1 bit).

Exemples de calcul de l'entropie

- Soit une source fournissant les deux symboles 0 et 1 de façon équiprobable $\Rightarrow$ p(0)=1/2, p(1)=1/2

$$\begin{aligned} H(x) &= -p(0)\log_2[p(0)] - p(1)\log_2[p(1)] \\ &= -1/2 \cdot \log_2[1/2] - 1/2 \cdot \log_2[1/2] \\ &= -\log_2[1/2] = \log_2[2] \\ &= 1 \end{aligned}$$

$\Rightarrow$ Le nombre de bits nécessaire est de 1, ce qui correspond à l'intuition (2 états possibles, donc 1 bit).

- Soit une source fournissant de façon equiprobable les 26 lettres de l'alphabet $\Rightarrow p(x_i) = 1/26$

$$H(x) = -26 \cdot 1/26 \cdot \log_2[1/26] = \log_2[26] = 4,27$$

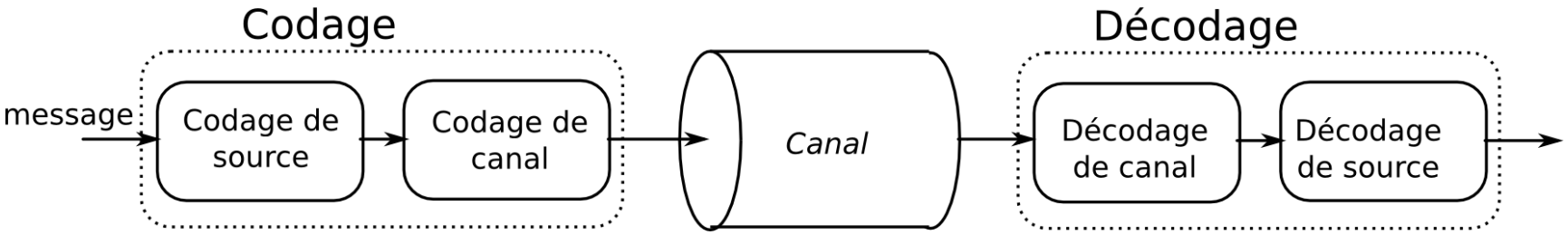
$\Rightarrow$ On obtient bien une valeur située entre 4, permettant de coder $2^4 = 16$ valeurs, et 5, permettant de coder $2^5 = 32$ valeurs.

Sous-sommaire

UNIVERSITÉ DE ROUEN

Codage de l'information

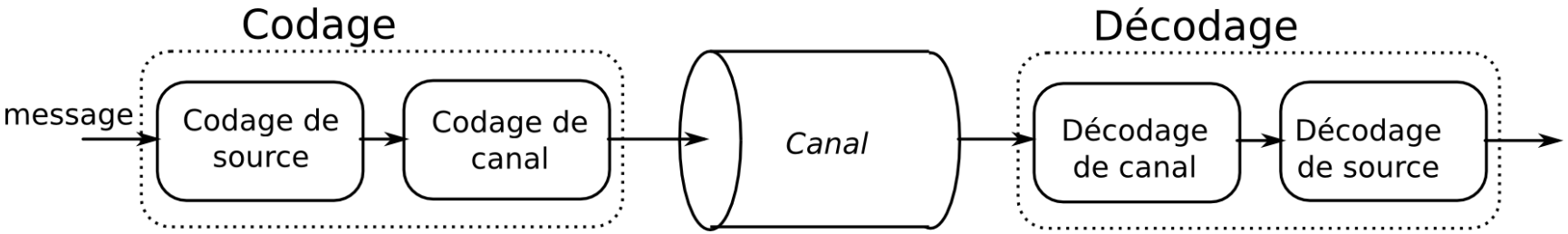
- Le codage doit permettre le transfert du message sur le canal, d'une façon **fiable**.



- On distingue deux étapes :
  1. Le **codage de source**, dont les objectifs sont :
     - encoder l'information sur un alphabet qui soit transmissible sur le canal,
     - éventuellement, minimiser le volume de l'information ("codage optimal").
  2. Le **codage de canal**, dont l'objectif est de permettre de détecter une erreur de transmission, voire de corriger le message à postériori. Ces codes introduisent de la **redondance** dans le message.

Codage de l'information

- Le codage doit permettre le transfert du message sur le canal, d'une façon **fiable**.



- On distingue deux étapes :
  1. Le **codage de source**, dont les objectifs sont :
     - encoder l'information sur un alphabet qui soit transmissible sur le canal,
     - éventuellement, minimiser le volume de l'information ("codage optimal").
  2. Le **codage de canal**, dont l'objectif est de permettre de détecter une erreur de transmission, voire de corriger le message à postériori. Ces codes introduisent de la **redondance** dans le message.
- Note : la compression de données, vue plus tard, peut-être vue comme une forme de codage de source optimal.

Sommaire

UNIVERSITÉ DE ROUEN

Sous-sommaire

UNIVERSITÉ DE ROUEN

Definitions préliminaires

- Un alphabet A est un ensemble fini non vide de symboles.
  Exemple :
  - binaire : $A = \{0, 1\}$, card(A)=2
  - alphabet latin : $A = \{a, b, c, d, e, f, ..., x, y, z\}$, card(A)=26
  - Morse : $A = \{., -, silence\}$, card(A)=3
- Un **mot** est une suite finie de symboles construite à partir d'un alphabet donné.
  Exemple : u=100011100011, u=bonjour
- La longueur du mot u est notée $|u|$.
  Exemple : u = 01000100010100101001, $|u| = 20$
- A^n est l'ensemble des mots de longueur n construits avec A.
- A^* est l'ensemble de tous les mots constructibles sur A $(\forall n)$.
- Un **langage** est toute partie de A^*.
  Exemple : si $A = \{0, 1\}$ et $n = 3$, $A^* = \{0, 100, 000, 111\}$ est un langage.

UNIVERSITÉ DE ROUEN

Definitions préliminaires - 2

- Le mot vide ϵ est le mot de longueur nulle.
- La **concaténation** de deux mots u et v est l'opération, notée . , consistant à mettre les symboles de u et v bout à bout.
  - Associativité :(u.v).w = u.(v.w) = u.v.w
  - Elément neutre : $u.\epsilon = \epsilon.u = u$
  - Longueurs :$|u.v| = |u| + |v|$

  Exemple : si m1=011 et m2=10, alors m1.m2=01110

Definitions préliminaires - 2

- Le mot vide ϵ est le mot de longueur nulle.
- La **concaténation** de deux mots u et v est l'opération, notée . , consistant à mettre les symboles de u et v bout à bout.
  - Associativité :(u.v).w = u.(v.w) = u.v.w
  - Elément neutre : $u.\epsilon = \epsilon.u = u$
  - Longueurs :$|u.v| = |u| + |v|$

  Exemple : si m1=011 et m2=10, alors m1.m2=01110
- Un mot u est **préfixe** d'un mot v ssi $\exists$ un mot w tel que v = u.w
  Exemple : 01 est préfixe de 010
- Un mot u est **suffixe** d'un mot v ssi $\exists$ un mot w tel que v = w.u
  Exemple : 10 est suffixe de 010

Codage et décodage

Définition

Un codage est l'association de chaque symbole d'un **alphabet source** $\mathcal{S}$ à un mot d'un **alphabet cible** noté $\mathcal{A}$.

- Mais ce n'est pas suffisant !

Codage et décodage

Définition

Un codage est l'association de chaque symbole d'un **alphabet source** $\mathcal{S}$ à un mot d'un **alphabet cible** noté $\mathcal{A}$.

- Mais ce n'est pas suffisant !
  Par exemple, avec les les alphabets $\mathcal{S} = \{A, B, C, D\}$ et $\mathcal{A} = \{0, 1\}$.
  Soit les codes c_1 et c_2 :

Codage et décodage

Définition

Un codage est l'association de chaque symbole d'un **alphabet source** $\mathcal{S}$ à un mot d'un **alphabet cible** noté $\mathcal{A}$.

- Mais ce n'est pas suffisant !
  Par exemple, avec les les alphabets $\mathcal{S} = \{A, B, C, D\}$ et $\mathcal{A} = \{0, 1\}$.
  Soit les codes c_1 et c_2 :

x	$c_1(x)$	$c_2(x)$
A	1	1
B	00	00
C	01	01
D	01	10

- c_1 n'est pas un code, car les symboles "C" et "D" sont codés par le même mot $\Rightarrow$ impossible de décoder !

Codage et décodage

Définition

Un codage est l'association de chaque symbole d'un **alphabet source** $\mathcal{S}$ à un mot d'un **alphabet cible** noté $\mathcal{A}$.

- Mais ce n'est pas suffisant !
  Par exemple, avec les les alphabets $\mathcal{S} = \{A, B, C, D\}$ et $\mathcal{A} = \{0, 1\}$.
  Soit les codes c_1 et c_2 :

x	$c_1(x)$	$c_2(x)$
A	1	1
B	00	00
C	01	01
D	01	10

- c_1 n'est pas un code, car les symboles "C" et "D" sont codés par le même mot $\Rightarrow$ impossible de décoder !
- c_2 n'est pas un code, car le mot "ABA" est codé par 1001 et "DC" est codé par 1001 $\Rightarrow$ Impossible de décoder !

Codage et décodage - 2

Un code c est valide (décodable) si deux conditions sont réunies :

1. Unicité du code : $\forall u, v \in S^*, u \neq v \Rightarrow c(u) \neq c(v)$
2. Unicité de la factorisation : Un langage C est un code s'il n'existe pas de mot ayant deux factorisations distinctes avec des mots de C.

- Comment déterminer l'unicité de la factorisation : pas facile...

Codage et décodage - 2

Un code c est valide (décodable) si deux conditions sont réunies :

1. Unicité du code : $\forall u, v \in S^*, u \neq v \Rightarrow c(u) \neq c(v)$
2. Unicité de la factorisation : Un langage C est un code s'il n'existe pas de mot ayant deux factorisations distinctes avec des mots de C.

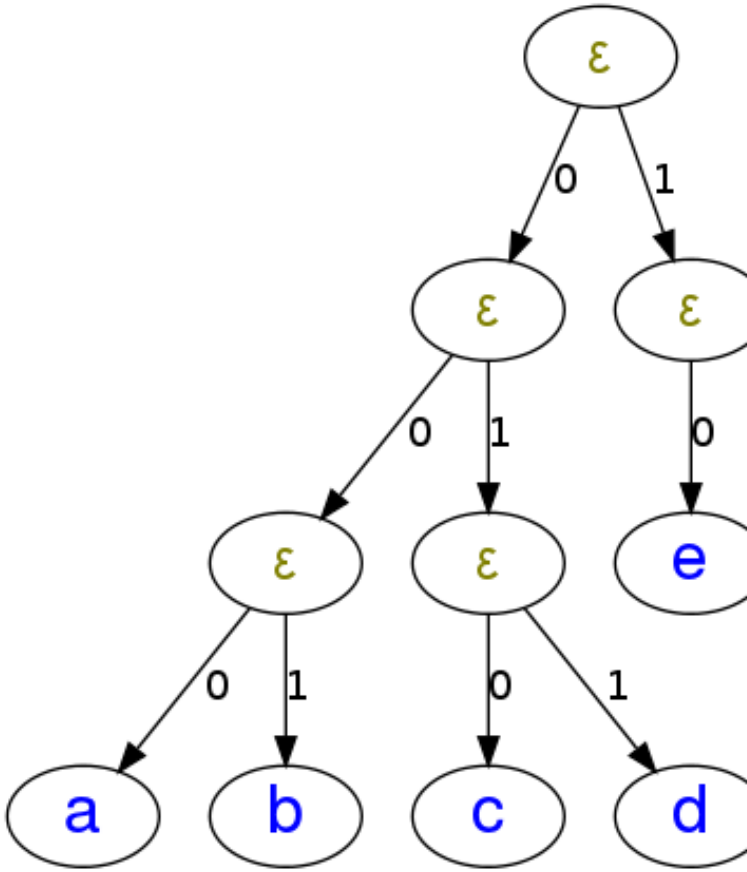
- Comment déterminer l'unicité de la factorisation : pas facile...
  Une méthode : algorithme de **Sardinas-Patterson** (pas traité dans ce cours).

UNIVERSITÉ DE ROUEN

Représentation graphique d'un code

- Un code peut-être représenté par un arbre :

- chaque nœud terminal représente un symbole de A,
- chaque arc représente un symbole de S dans le mot,
- le mot (code) associé à un symbole de A s'obtient en partant de la racine jusqu'à ce symbole.



⇒ Le symbole 'b' est codé par _____

Sous-sommaire

UNIVERSITÉ DE ROUEN

Codes particuliers

1. Code de longueur fixe : $|u| = Cte$. Par exemple :
   - le code $c = \{00, 01, 10, 11\}$,
   - le code ASCII.
2. Code à virgule
   - Soit le code $c = \{0, 01, 011, 0111\}$: les 4 mots commencent par le même symbole.
   - Toute concaténation de ces quatre mots ne peut etre obtenue que d'une seule manière, donc C est un code.
   - Le rôle joué par le symbole 0 est celui d'un **délimiteur** : soit au début, soit à la fin.
     1. sur l'alphabet $A = \{0, 1, \#\}$, $C = \{1\#, 00\#, 10\#, 01\#\}$ est un code.
     2. le Morse est un code a virgule, les silences en faisant office.
3. Code préfixe : code pour lequel aucun mot n'est préfixe d'un autre.
   - $c = \{0, 10, 110\}$ est préfixe.
   - $c = \{0, 01, 10\}$ n'est pas préfixe (car 0 est préfixe de 01).

UNIVERSITÉ DE ROUEN

Sous-sommaire

UNIVERSITÉ DE ROUEN

Codage optimal

- Un code optimal est un code dont la longueur moyenne des mots est la plus faible.
- Formellement, un code optimal est un code dont le **coût** est le plus faible.

Coût d'un code

Soit un code c défini sur un alphabet A, composé de $n = card(A)$ symboles, chacun d'eux ayant une probabilité d'apparition $p(x_i)$.

$$cout(C) = \sum_{i=1}^{n} p(x_i) \cdot |c(x_i)|$$

- Remarque : pour un code de longueur fixe, le coût sera : $|c| \cdot \sum_n p(x_i) = |c|$ (en effet, $\sum_n p(x_i) = 1$)

UNIVERSITÉ DE ROUEN

Codage optimal de longueur fixe

- Pour les codes de longueur fixe, un code optimal est un code qui utilise la plus petite longueur de mot permettant d'associer un mot à chaque symbole.
- Formellement, on doit choisir des mots de longueur n le plus petit possible, mais respectant $card(A)^n \geq card(S)$
- Exemples :
  - si S est composé de 8 symboles et A=$\{0,1\}$, alors on choisira des mots de 3 symboles ($2^3 = 8 \geq 8$)

Codage optimal de longueur fixe

- Pour les codes de longueur fixe, un code optimal est un code qui utilise la plus petite longueur de mot permettant d'associer un mot à chaque symbole.
- Formellement, on doit choisir des mots de longueur n le plus petit possible, mais respectant $card(A)^n \geq card(S)$
- Exemples :
  - si S est composé de 8 symboles et A=$\{0,1\}$, alors on choisira des mots de 3 symboles ($2^3 = 8 \geq 8$)
  - si S est composé de 9 symboles et A=$\{0,1\}$, alors on choisira des mots de 4 symboles ($2^4 = 16 \geq 9$)

Codage optimal de longueur fixe

- Pour les codes de longueur fixe, un code optimal est un code qui utilise la plus petite longueur de mot permettant d'associer un mot à chaque symbole.
- Formellement, on doit choisir des mots de longueur n le plus petit possible, mais respectant $card(A)^n \geq card(S)$
- Exemples :
  - si S est composé de 8 symboles et A=$\{0,1\}$, alors on choisira des mots de 3 symboles ($2^3 = 8 \geq 8$)
  - si S est composé de 9 symboles et A=$\{0,1\}$, alors on choisira des mots de 4 symboles ($2^4 = 16 \geq 9$)
  - si S est composé de 9 symboles et A=$\{x,y,z\}$, alors on choisira des mots de 2 symboles ($3^2 = 9 \geq 9$)

UNIVERSITÉ DE ROUEN

Codage optimal de longueur fixe

- Pour les codes de longueur fixe, un code optimal est un code qui utilise la plus petite longueur de mot permettant d'associer un mot à chaque symbole.
- Formellement, on doit choisir des mots de longueur n le plus petit possible, mais respectant $card(A)^n \geq card(S)$
- Exemples :
  - si S est composé de 8 symboles et A=$\{0,1\}$, alors on choisira des mots de 3 symboles ($2^3 = 8 \geq 8$)
  - si S est composé de 9 symboles et A=$\{0,1\}$, alors on choisira des mots de 4 symboles ($2^4 = 16 \geq 9$)
  - si S est composé de 9 symboles et A=$\{x,y,z\}$, alors on choisira des mots de 2 symboles ($3^2 = 9 \geq 9$)

UNIVERSITÉ DE ROUEN

Codage optimal de longueur fixe

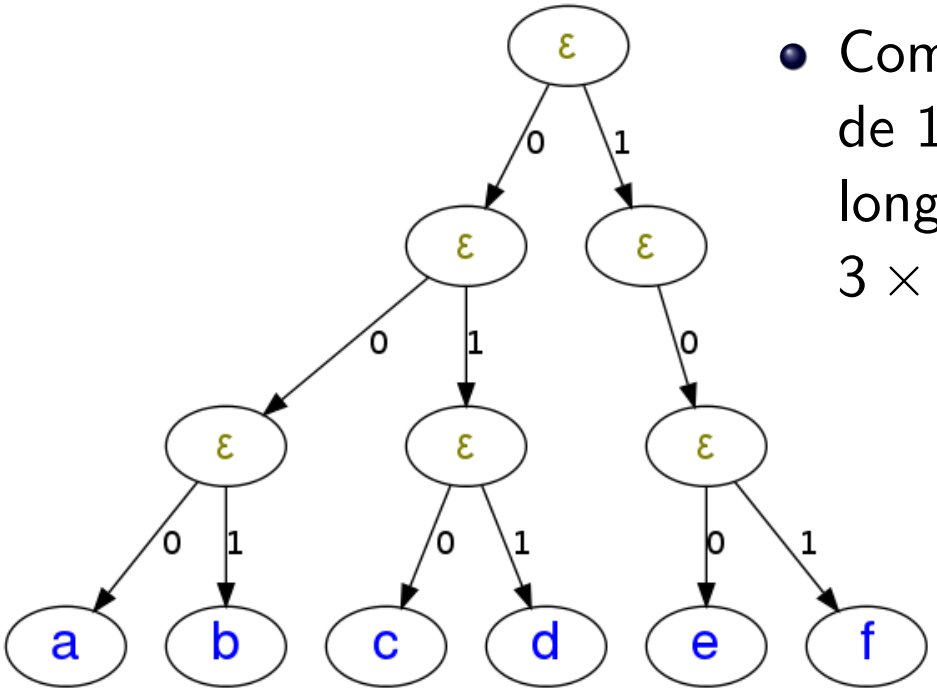
- Pour les codes de longueur fixe, un code optimal est un code qui utilise la plus petite longueur de mot permettant d'associer un mot à chaque symbole.
- Formellement, on doit choisir des mots de longueur n le plus petit possible, mais respectant $card(A)^n \geq card(S)$
- Exemples :
  - si S est composé de 8 symboles et A=$\{0,1\}$, alors on choisira des mots de 3 symboles ($2^3 = 8 \geq 8$)
  - si S est composé de 9 symboles et A=$\{0,1\}$, alors on choisira des mots de 4 symboles ($2^4 = 16 \geq 9$)
  - si S est composé de 9 symboles et A=$\{x,y,z\}$, alors on choisira des mots de 2 symboles ($3^2 = 9 \geq 9$)

Si S est composé de 59 symboles, et A=$\{0,1,2,3\}$, alors on choisira des mots de _____ symboles.

Codage optimal de longueur fixe : exemple

- Soit les alphabets $A = \{a, b, c, d, e, f\}$ et $S = \{0, 1\}$
- Soit le code suivant :

Symbole	a	b	c	d	e	f
Code	000	001	010	011	100	101
Probabilité	0,45	0,13	0,12	0,16	0,09	0,05

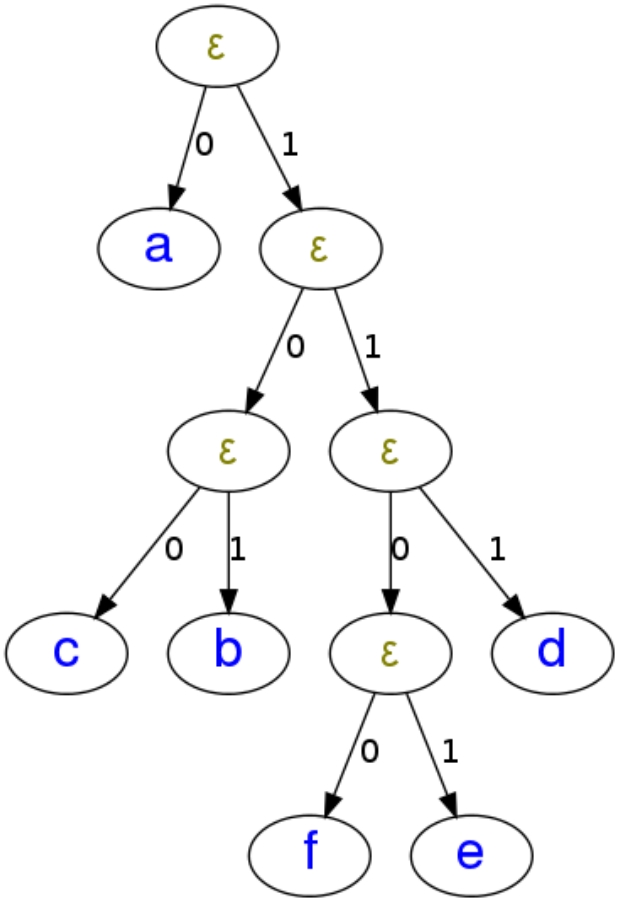


- Comme la somme des probabilités est de 1 et que tous les mots sont de longueur 3, le coût de ce code est de $3 \times 1 = 3$

Codage optimal de longueur variable : exemple

- Avec les mêmes alphabets, soit le code suivant :

Symbole	a	b	c	d	e	f
Code	0	101	100	111	1101	1100
Probabilité	0,45	0,13	0,12	0,16	0,09	0,05



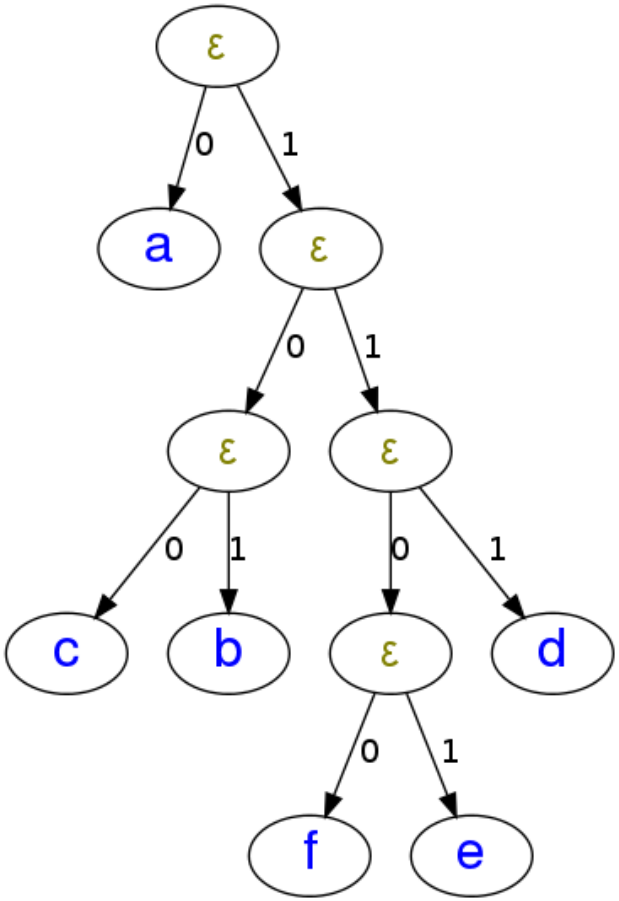
- Le coût de ce code s'écrit :

$$\begin{aligned}\text{coût} &= 0,45 \times 1 + 0,13 \times 3 \\ &\quad + 0,12 \times 3 + 0,16 \times 3 \\ &\quad + 0,09 \times 4 + 0,05 \times 4 \\ &= 2,24\end{aligned}$$

Codage optimal de longueur variable : exemple

- Avec les mêmes alphabets, soit le code suivant :

Symbole	a	b	c	d	e	f
Code	0	101	100	111	1101	1100
Probabilité	0,45	0,13	0,12	0,16	0,09	0,05



- Le coût de ce code s'écrit :

$$\begin{aligned} \text{coût} &= 0,45 \times 1 + 0,13 \times 3 \\ &\quad + 0,12 \times 3 + 0,16 \times 3 \\ &\quad + 0,09 \times 4 + 0,05 \times 4 \\ &= 2,24 \end{aligned}$$

Principe

Les symboles les **plus fréquents** sont codés avec les mots les **plus courts**.

▸ code 2

Codage de Huffman

- Ce type de code s'appelle **Codage de Huffman**
- Algorithme : on classe les symboles par probabilité décroissante.

Symbole	a	d	b	c	e	f
Probabilité	0,45	0,16	0,13	0,12	0,09	0,05

  Puis on itère l'algorithme suivant :
  - Ajouter 0 et 1 en suffixe des codes des symboles des de plus faibles probabilités
  - Regrouper ces deux symboles ensemble en additionnant leur probabilité, trier la table, et recommencer jusqu'à ce que tous les symboles aient un code d'assigné.

Illustration de l'algorithme de Huffman

1	Symbole	a	d	b	c	e	f
	Code					0	1
	Probabilité	0,45	0,16	0,13	0,12	0,09	0,05

Illustration de l'algorithme de Huffman

1	Symbole	a	d	b	c	e	f
	Code					0	1
	Probabilité	0,45	0,16	0,13	0,12	0,09	0,05

2	Symbole	a	d	ef	b	c
	Code				0	1
	Probabilité	0,45	0,16	0,14	0,13	0,12

Illustration de l'algorithme de Huffman

1	Symbole	a	d	b	c	e	f
	Code					0	1
	Probabilité	0,45	0,16	0,13	0,12	0,09	0,05

2	Symbole	a	d	ef	b	c
	Code				0	1
	Probabilité	0,45	0,16	0,14	0,13	0,12

3	Symbole	a	bc	d	ef
	Code			0	1
	Probabilité	0,45	0,25	0,16	0,14

Illustration de l'algorithme de Huffman

1	Symbole	a	d	b	c	e	f
	Code					0	1
	Probabilité	0,45	0,16	0,13	0,12	0,09	0,05

2	Symbole	a	d	ef	b	c
	Code				0	1
	Probabilité	0,45	0,16	0,14	0,13	0,12

3	Symbole	a	bc	d	ef
	Code			0	1
	Probabilité	0,45	0,25	0,16	0,14

4	Symbole	a	def	bc
	Code		0	1
	Probabilité	0,45	0,30	0,25

Illustration de l'algorithme de Huffman

1	Symbole	a	d	b	c	e	f
	Code					0	1
	Probabilité	0,45	0,16	0,13	0,12	0,09	0,05

2	Symbole	a	d	ef	b	c
	Code				0	1
	Probabilité	0,45	0,16	0,14	0,13	0,12

3	Symbole	a	bc	d	ef
	Code			0	1
	Probabilité	0,45	0,25	0,16	0,14

4	Symbole	a	def	bc
	Code		0	1
	Probabilité	0,45	0,30	0,25

5	Symbole	defbc	a
	Code	0	1
	Probabilité	0,55	0,45

Illustration de l'algorithme de Huffman

- Il ne reste qu'a reconstituer le code associé à chaque symbole :

Symbole	a	b	c	d	e	f
Code	1	010	011	000	0010	0011

Illustration de l'algorithme de Huffman

- Il ne reste qu'a reconstituer le code associé à chaque symbole :

Symbole	a	b	c	d	e	f
Code	1	010	011	000	0010	0011

- Remarque : le codage optimal peut ne pas être unique :
  - on peut inverser les symboles 0 et 1 dans l'algorithme,
  - En cas d'égalité de probabilité, on peut indifféremment assigner 0 ou 1.

▸ code 1

Sommaire

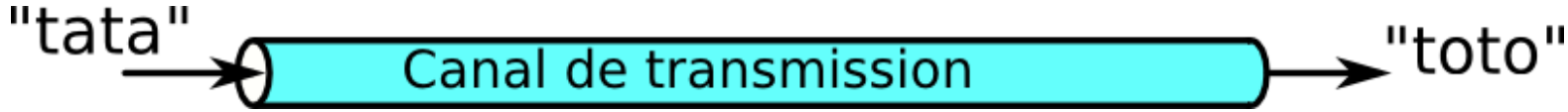
UNIVERSITÉ DE ROUEN

Sous-sommaire

UNIVERSITÉ DE ROUEN

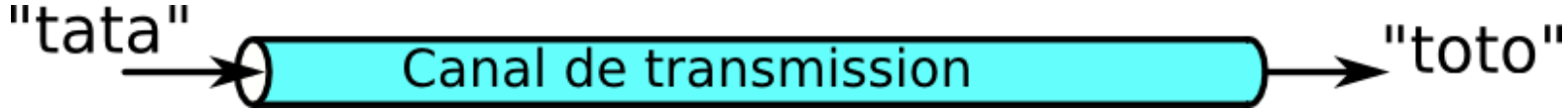
Modélisation d'un canal de transmission

- Un canal de transmission est bruité : certains symboles peuvent être transformés en d'autres symboles.

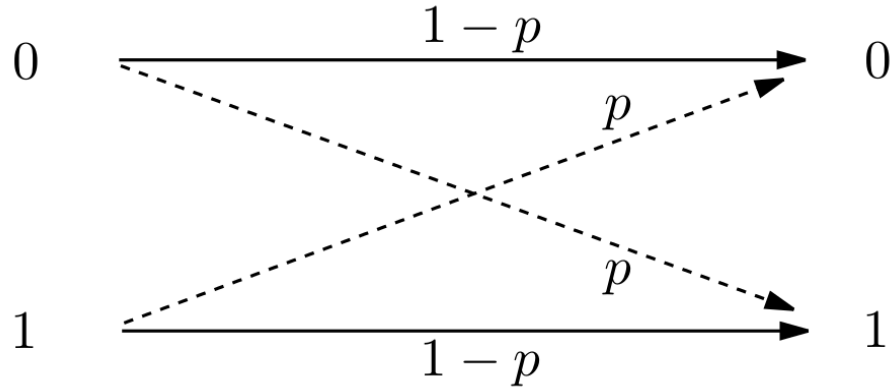


Modélisation d'un canal de transmission

- Un canal de transmission est bruité : certains symboles peuvent être transformés en d'autres symboles.



- Pour un canal transmettant du binaire, on peut modéliser les erreurs par un **taux** d'erreur p, avec $0 < p < 1$, qui représente la probabilité qu'un 1 soit transmis comme un 0 (ou vice-versa).



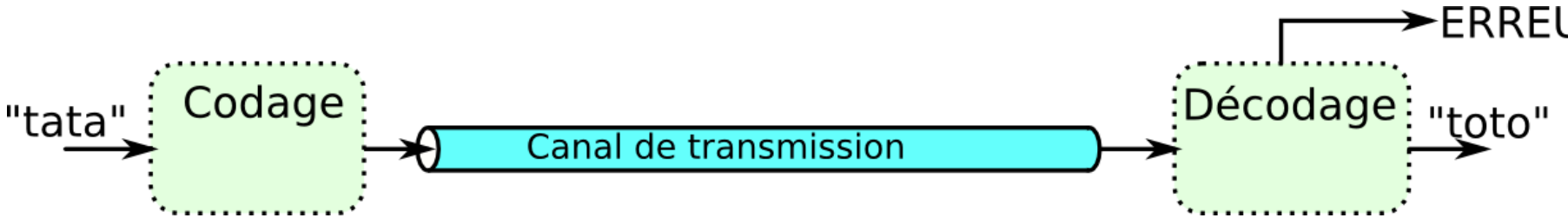
- Exemple de taux d'erreur : CD : 10^{-5} ; liaison téléphonique : de 10^{-4} à 10^{-7} ; ADSL : de 10^{-3} à 10^{-9} ; réseau informatique (filaire) : 10^{-12}

Exercice

- Dans l'exemple de la page précédente, en supposant que les caractères sont transmis en ASCII 8 bits, combien de bits ont été transmis de façon incorrecte ?
- A partir des 4 octets de ce message, estimer la valeur de p pour ce canal :

Codage de canal

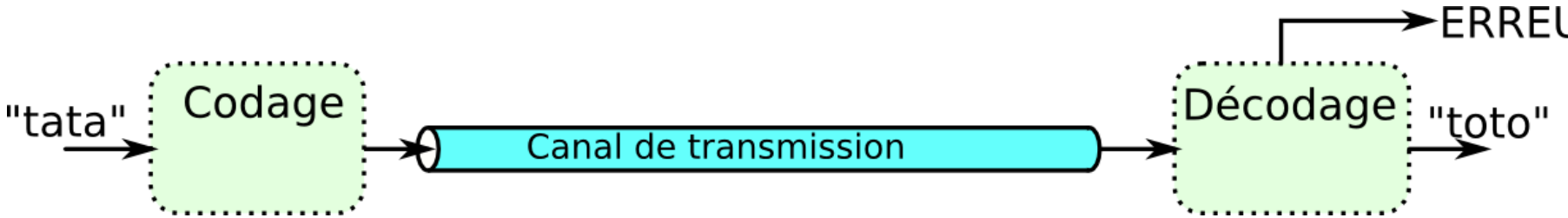
- Le codage de canal a pour objectif de fiabiliser la transmission en apportant une possibilité de **détection** d'erreur par le récepteur.



Celui-ci pourra alors soit demander la ré-émission du message (si c'est possible), ou ignorer les informations corrompues.

Codage de canal

- Le codage de canal a pour objectif de fiabiliser la transmission en apportant une possibilité de **détection** d'erreur par le récepteur.

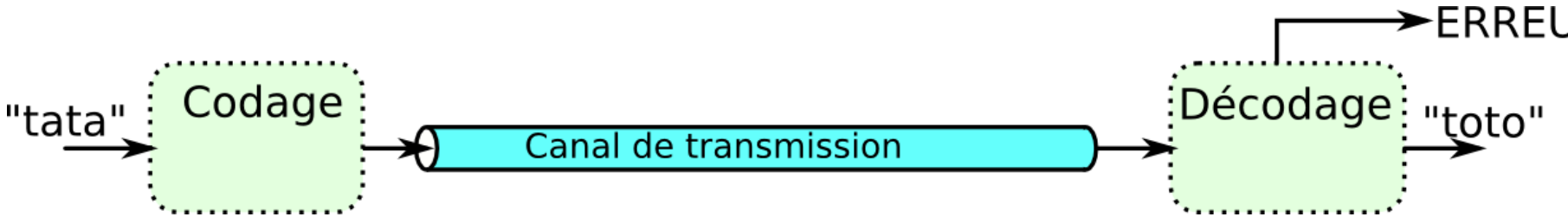


Celui-ci pourra alors soit demander la ré-émission du message (si c'est possible), ou ignorer les informations corrompues.

- Approche intuitive : alphabet aéronautique.
  Les communications radio prévoient l'usage d'un code destiné à limiter les erreurs de communication lors de la transmission de sigles.
  Par exemple, pour transmettre l'immatriculation d'un avion à la radio, plutot que de dire "F-MNBD", on dira

Codage de canal

- Le codage de canal a pour objectif de fiabiliser la transmission en apportant une possibilité de **détection** d'erreur par le récepteur.

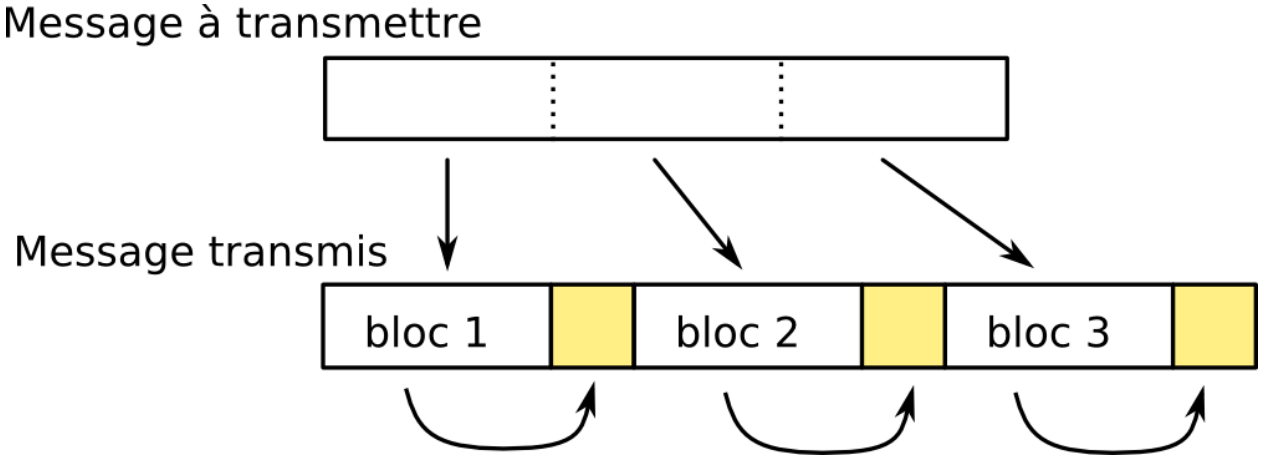


Celui-ci pourra alors soit demander la ré-émission du message (si c'est possible), ou ignorer les informations corrompues.

- Approche intuitive : alphabet aéronautique.
  Les communications radio prévoient l'usage d'un code destiné à limiter les erreurs de communication lors de la transmission de sigles.
  Par exemple, pour transmettre l'immatriculation d'un avion à la radio, plutot que de dire "F-MNBD", on dira
  "Fox Mike Novembre Bravo Delta"
  $\Rightarrow$ On a **ajouté** de l'information au message pour limiter les erreurs de transmission dues au bruit.

Codage de canal : principe

- Principe général : l'emetteur **découpe** le message en blocs, et ajoute à chaque bloc des bits supplémentaires.

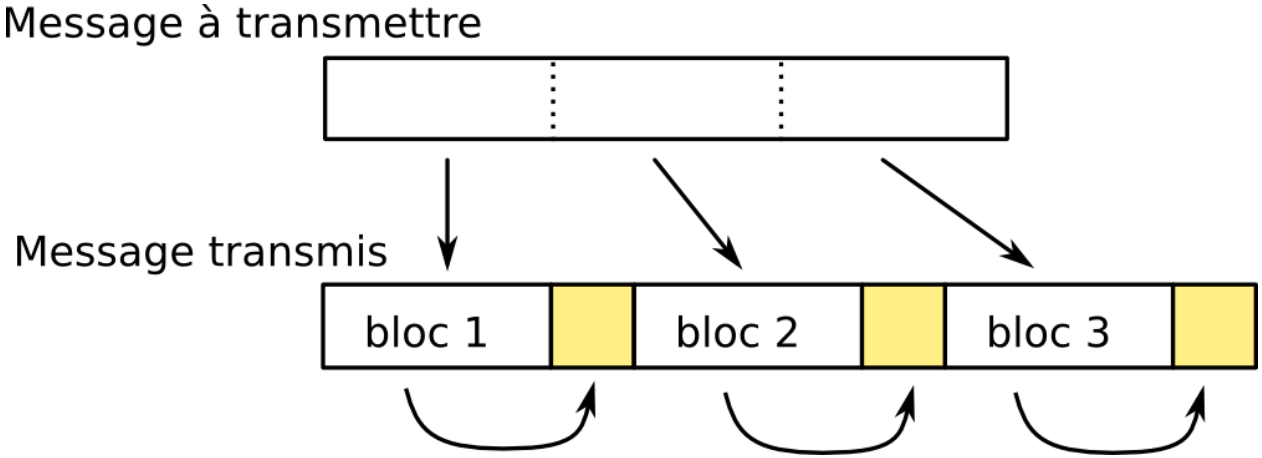


  $\Rightarrow$ On parle de **codage par blocs**.

- Les informations ajoutées sont calculées par l'émetteur à partir du contenu du bloc, d'après une règle connue de l'émetteur et du recepteur.

Codage de canal : principe

- Principe général : l'emetteur **découpe** le message en blocs, et ajoute à chaque bloc des bits supplémentaires.

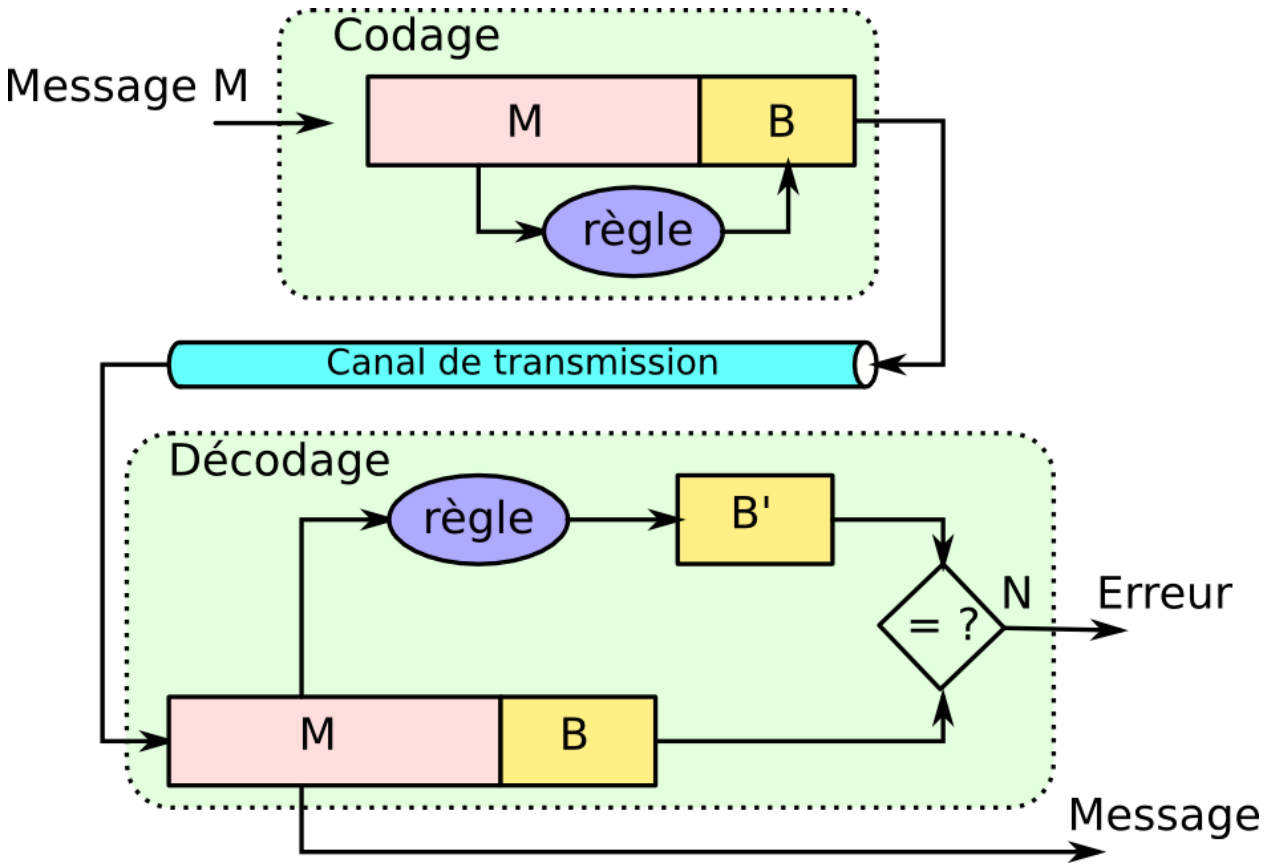


  $\Rightarrow$ On parle de **codage par blocs**.

- Les informations ajoutées sont calculées par l'émetteur à partir du contenu du bloc, d'après une règle connue de l'émetteur et du recepteur.
- Exemple au quotidien : clé du RIB, clé du NIR (n° INSEE), ...
  NIR : clé (sur 2 chiffres) = 97 - (NIR % 97)

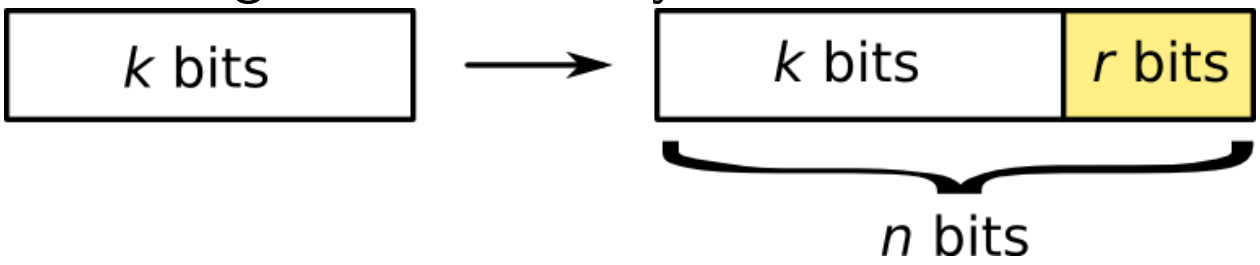
Codage de canal

- Le récepteur fait de même sur le message reçu, génère un bloc B', et compare B avec B'.
- Si B $\neq$ B', alors il y a une erreur de transmission.



Codage de canal : formalisme

- On découpe le message en blocs de k bits, auquels on ajoute de la redondance sous la forme de r bits.
- Le message est alors envoyé sous la forme de $n = k + r$ bits.



- On définit le rendement du codage : $\tau = k/n$

Codage de canal : deux approches

Détection d'erreur

- En cas d'erreur, le récepteur doit alors "redemander" la transmission du message.
- Avantage : mise en œuvre plus simple.
- Inconvénient : il faut avoir :
  1. le temps,
  2. un protocole de communication bidirectionnel.

  $\Rightarrow$ Pas toujours possible (canal unidirectionnel, distance élevée, etc.)

  Exemple : _____

Codage de canal : deux approches

Détection d'erreur

- En cas d'erreur, le récepteur doit alors "redemander" la transmission du message.
- Avantage : mise en œuvre plus simple.
- Inconvénient : il faut avoir :
  1. le temps,
  2. un protocole de communication bidirectionnel.

  $\Rightarrow$ Pas toujours possible (canal unidirectionnel, distance élevée, etc.)

  Exemple : _____

Correction d'erreur

- Permettent à la fois de détecter et de corriger les erreurs.
- Inconvénient : plus complexe, augmente le nombre de bits de contrôle.
- Avantage : évite d'avoir à redemander la transmission du message.

A retenir

- Certains codes peuvent **corriger** si l'erreur n'est pas trop importante, mais aussi **détecter** si l'erreur est plus importante.
- Aucun codage n'est infaillible ! Si le bruit est trop important, la communication sera **impossible**.
- Pas de code idéal. Chaque code a un coût (temps de calcul au codage et au décodage) et une performance en fonction d'un type de bruit donné.

UNIVERSITÉ DE ROUEN

Sous-sommaire

UNIVERSITÉ DE ROUEN

Principales techniques de détection d'erreurs

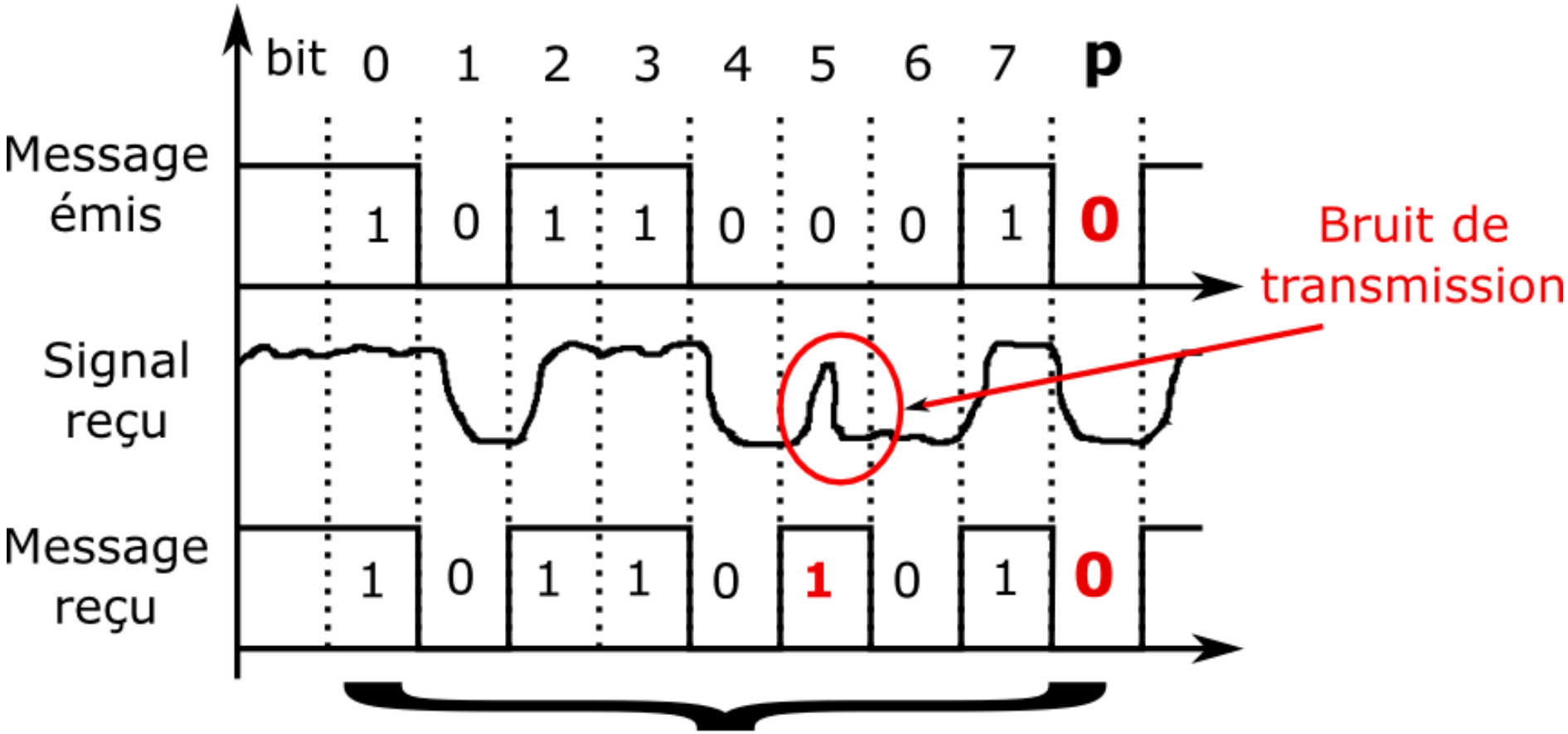
- Bit de parité : Ajout d'un bit de façon à ce que le poids du mot soit pair.
- Somme de contrôle : ajout d'une somme tronquée de tous les octets du message.
- Contrôle de redondance cyclique (CRC) : ajout du reste d'une division polynômiale.

1 - Bit de parité

- Principe : on ajoute à chaque "élement d'information" (en général, un octet) un bit appelé **bit de parité**.
  - le bit est positionné par l'emetteur en fonction du nombre de bits à '1' du message transmis ;
  - à la réception, on vérifie la règle ci-dessous.
- Règle (en mode "parité paire") : **le nombre de bits à '1' du message total (message + bit de parité) doit être pair**.
- Exemples (en mode "parité paire") :

octet à transmettre	bit de parité
0000 0000	
0100 0000	
0110 0000	
1101 0001	

Bit de parité : exemple



Inconvénient : ne détecte que les erreurs en nombre impair !

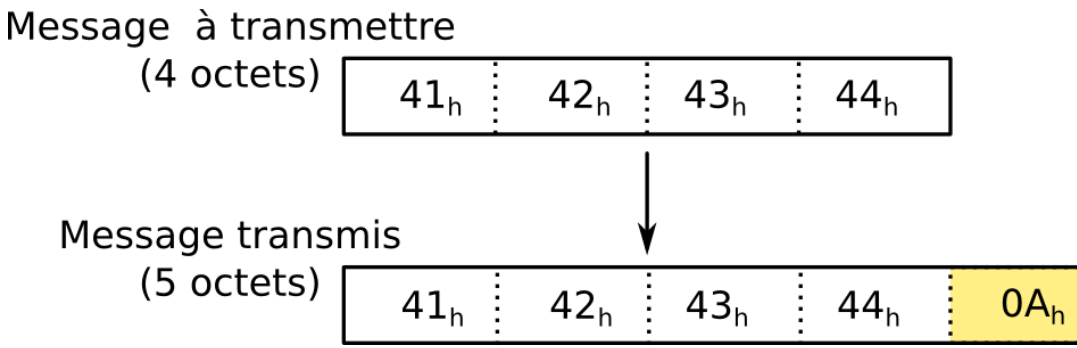
2 - Somme de contrôle

- Principe : on ajoute une somme des octets du message, tronquée aux r bits de poids faible.
- Par exemple : on décide d'ajouter 1 octet de somme de contrôle ($r = 8$) à chaque bloc de 4 octets ($k = 32$).
- Exemple de transmission :

2 - Somme de contrôle

- Principe : on ajoute une somme des octets du message, tronquée aux r bits de poids faible.
- Par exemple : on décide d'ajouter 1 octet de somme de contrôle ($r = 8$) à chaque bloc de 4 octets ($k = 32$).
- Exemple de transmission :

- On transmet le message "ABCD". La somme vaut $41 + 42 + 43 + 44 = (1)0A_h$
- Le récepteur recoit la suite d'octet (hexa) ACCD, la somme de contrôle calculée vaudra : $41 + 43 + 43 + 44 = (1)0B_h$
  $\Rightarrow$ l'erreur sera détectée.



Somme de contrôle : limitations

- Problème : si le message reçu est est ACBD, l'erreur ne sera pas détectée.
- idem si on reçoit DCBA ....
- Plus généralement, la somme de contrôle est invariante en cas de permutation de valeurs.
- Il faudrait que le bloc ajouté caractérise le message de façon unique... (concept des fonctions de hachage)
- La technique du contrôle de redondance cyclique permet de répondre à ce problème.

UNIVERSITÉ DE ROUEN

3 - Contrôle de redondance cyclique (CRC)

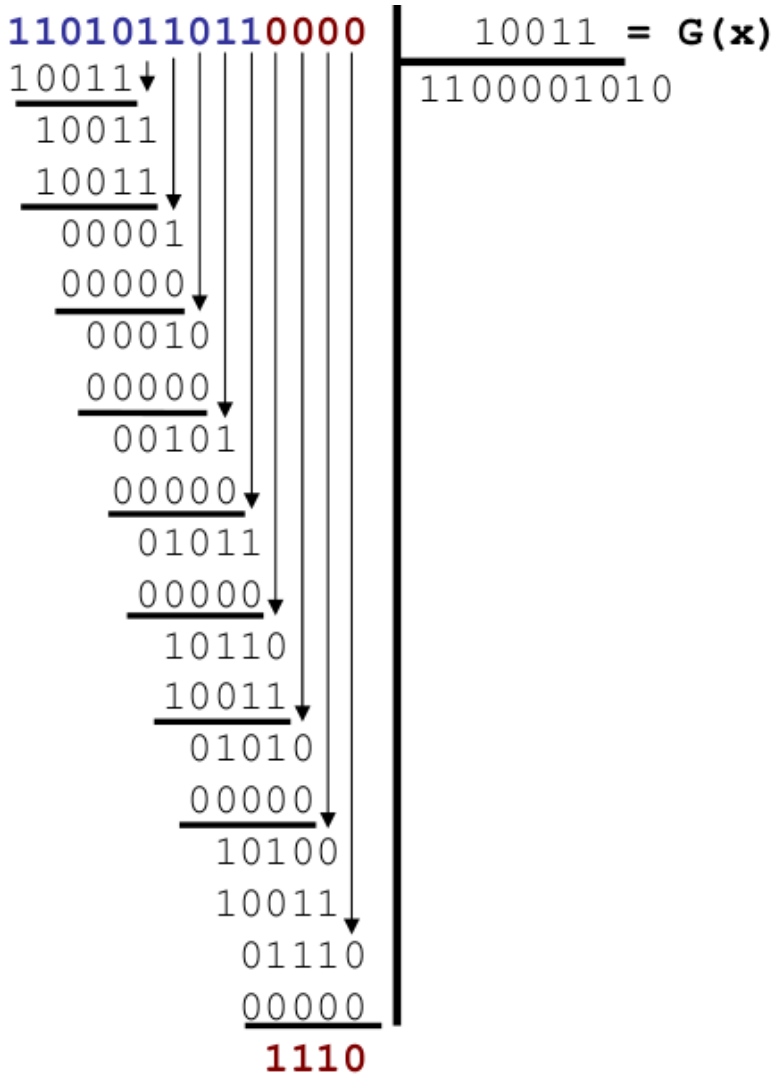
- En anglais : *Cyclic Redundancy Check*
- Le principe du CRC consiste à traiter les séquences de bits comme des polynômes binaires, et à calculer pour B un reste de division, obtenu en arithmétique modulo 2 (addition→)ou-exclusif)
- Par ex., la séquence binaire 110101001 peut être représentée sous la forme polynomiale suivante :
  $X^8 + X^7 + X^5 + X^3 + X^0 = X^8 + X^7 + X^5 + X^3 + 1$
- Une séquence de n bits constitue donc un polynôme de degré maximal n-1.
- On utilise un polynôme prédéfini, appelé polynôme générateur et noté G(X), connu de l'émetteur et du récepteur.

CRC : principe

- Emetteur et recepteur s'entendent sur un polynôme G(x)
- Pour chaque message M :
  - On ajoute à M un bloc B0 de d bits à 0, à droite :
  - On effectue la division de M.B0 par G (en *arithmétique modulo 2*), on obtient un reste R de d bits.
  - On transmet le message M'=M.R
  - Le récepteur fait la division de M' par G.
    $\Rightarrow$ Si M'/G=0, alors il n'y a pas eu d'erreur.

CRC : exemple

- On choisit un CRC sur 4 bits, avec le polynôme $G(x) = x^4 + x + 1$, soit 10011
- Message originel : 1101011011
- La division donne le reste 1110
- Le message transmis sera 1101011011 1110



(Source : S. Jean, IUT Valence)

CRC : choix du polynôme générateur

- Pour un CRC de r bits, alors il faut que G(x) soit au plus d'ordre r.
- G(x) doit avoir 1 comme dernier coefficient.
- Quelques polynômes usuels (la notation en hexa montre le polynôme de façon synthétique, en ne montrant que les coefficients non-nuls)

Nom	r	polynôme	hexa
CRC-12	12	$x^{12} + x^{11} + x^3 + x^2 + x + 1$	80F
CRC-16	16	$x^{16} + x^{15} + x^2 + 1$	8005
CRC-CCITT	16	$x^{16} + x^{12} + x^5 + 1$	1021
CRC-32	32	...	04C1 1DB7

Sous-sommaire

UNIVERSITÉ DE ROUEN

Codage par répétition

- Codage élémentaire : codage par répétition
- Principe : On ajoute à chaque bit du message ($k = 1$) deux bits identiques ($r = 2$).
- Exemple : le message 10110 sera encodé par 111.000.111.111.000
- La correction d'erreur se fait par **vote majoritaire** sur les groupes de 3 bits.
  - Réception 3 symboles identiques : pas d'erreur.
  - Réception de deux '1' et un '0' : c'est un '1'
  - Réception de deux '0' et un '1' : c'est un '0'
- Inconvénient : rendement très faible ($\tau = 1/3$)

UNIVERSITÉ DE ROUEN

Intermède : opérateur "Ou-Exclusif"

- On définit l'opération booléenne Ou-exclusif comme une addition "modulo-2" (pas de retenue)
- Table de vérité $S = a \oplus b$

a	b	S
0	0	0
0	1	1
1	0	1
1	1	0

- Propriétés : $a \oplus 0 = a$ $a \oplus 1 = \bar{a}$

Distance entre deux mots de code

- Soit un alphabet destination A, sur lequel on a deux mots m et m', avec $|m| = |m'|$.
- On appelle **distance de Hamming** le nombre de positions où m et m' ont des symboles différents

$$d(w, w') = \text{card}(i, m_i \neq m'_i)$$

- Cette distance peut s'interpréter comme
  - le nombre de symboles à modifier pour passer de m à m'.
  - le nombre d'erreurs nécessaires pour confondre m et m'.
- Exemples : d(1101, 1001) = 1, d(1101, 0000) = 3, d(IUT , DUT) = 1, d(toto, titi) = 2.

UNIVERSITÉ DE ROUEN

Distance d'un code

- On appelle distance de Hamming (ou simplement distance) d'un code C , et on note d(C), la plus petite distance entre deux mots distincts de C :

$$d(C) = \min_{x,y \in C; x \neq y} (d(x,y))$$

- Correspond au nombre minimum d'erreurs permettant de transformer un mot du code en un autre mot du code.
- Plus la distance de Hamming du code est grande, plus les mots du code sont "dispersés" dans A^n.
- On peut la calculer de façon algébrique :

$$d(C) = \min_{u,v \in C; u \neq v} w(u \oplus v)$$

Exemple de distance de code

1. Codage par répétition de longueur 3 : le code ne comprend que 2 mots : $c = \{000, 111\}$
   $\Rightarrow d(c) = 3$

Exemple de distance de code

1. Codage par répétition de longueur 3 : le code ne comprend que 2 mots : $c = \{000, 111\}$
   $\Rightarrow d(c) = 3$
2. Codage par bit de parité (paire) sur bloc de 2 bits :
   $c = \{000, 011, 101, 110\}$
   Table des distances

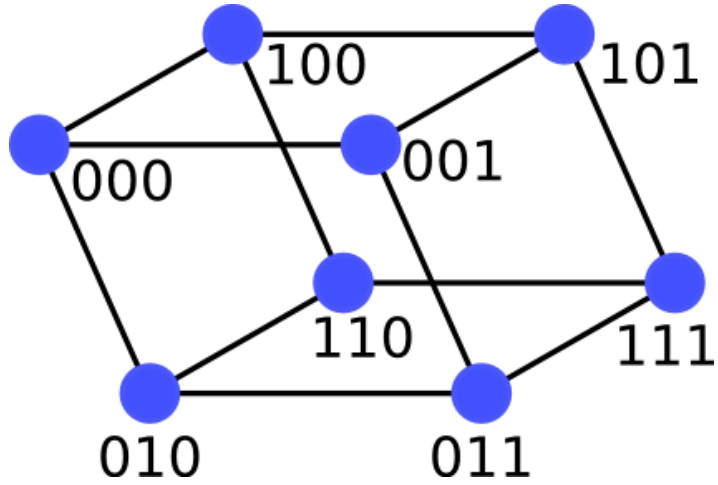
$x \backslash y$	000	011	101	110
000	0	2	2	2
011		0	2	2
101			0	2
110				0

   La plus petite distance de la table (en dehors du cas x=y) est 2
   $\Rightarrow d(c) = 2$

Représentation graphique d'un code

- Chaque mot correspond à un nœud, et est connecté à tous les autres nœuds dont il est distant de 1 bit.
- La distance $d(a, b)$ est le nombre d'arêtes qu'il faut longer pour se rendre de a à b en passant par le chemin le plus court.

- Exemple : code complet binaire sur 3 bits (8 mots) :

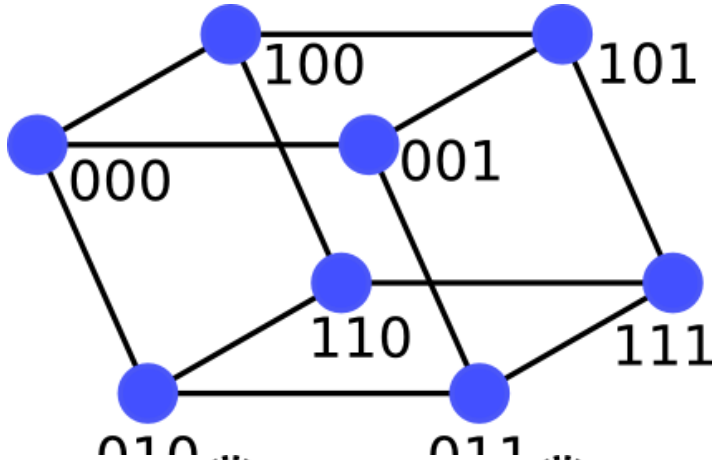


$\Rightarrow$ la distance entre 2 mots de code est de 1 bit.

Représentation graphique d'un code

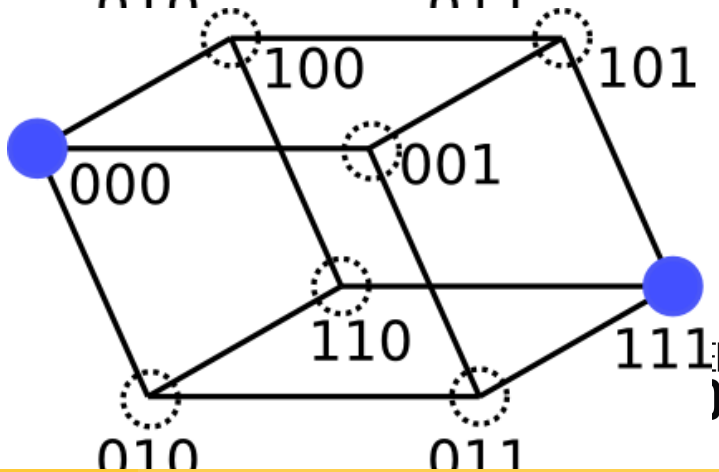
- Chaque mot correspond à un nœud, et est connecté à tous les autres nœuds dont il est distant de 1 bit.
- La distance $d(a, b)$ est le nombre d'arêtes qu'il faut longer pour se rendre de a à b en passant par le chemin le plus court.

- Exemple : code complet binaire sur 3 bits (8 mots) :



⇒ la distance entre 2 mots de code est de 1 bit.

- Exemple : code par répétition de longueur 3 :



⇒ on voit que la distance minimale pour passer d'un mot à l'autre est de 3 bits.

Correction d'erreur et distance d'un code

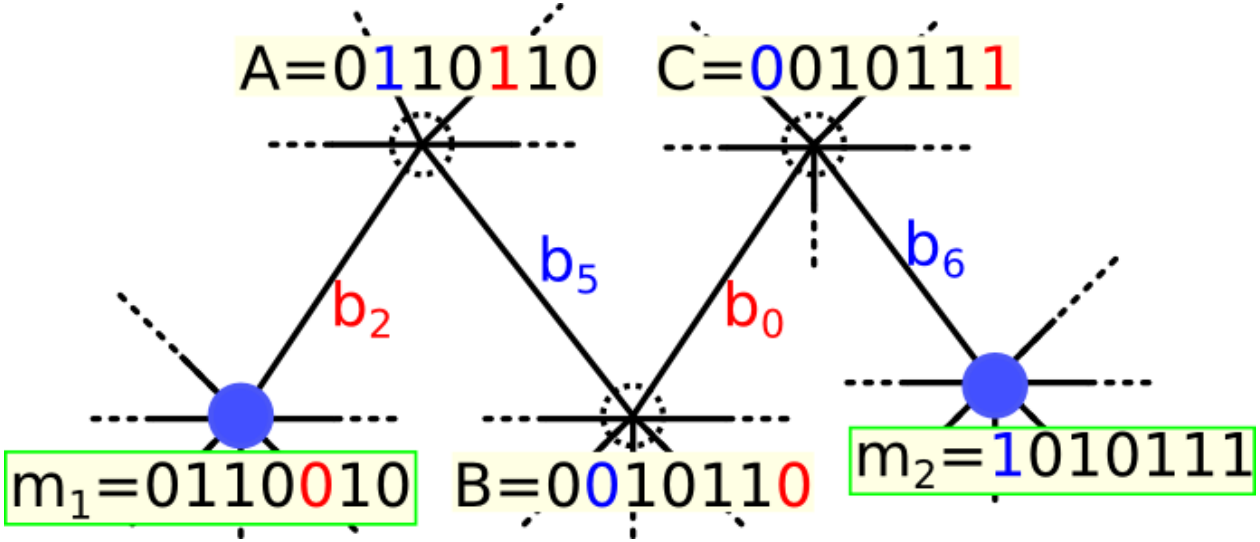
Plus un code a une distance élevée, et plus il est performant en détection et correction.

- En cas de réception d'un mot n'appartenant pas au code, on peut **corriger** en le remplaçant par le mot du code le **plus proche**.
- Exemple d'école : soit le code de 3 mots $c = \{00000, 00100, 11111\}$
  - Si on reçoit 00011, on va corriger en 00000 (d=2).
  - Si on reçoit 01111, on va corriger en 11111 (d=1).
  - Si on reçoit 00111, on **détecte** une errreur, qu'on ne peut pas corriger (égale distance de 00100 et 11111).
- Généralisation à un code quelconque :
  - Un code de distance n permettra de **détecter** des erreurs de $n-1$ bits.
  - Un code de distance n permettra de **corriger** des erreurs de $(n-1)/2$ bits.

UNIVERSITÉ DE ROUEN

Exemple de code correcteur sur 7 bits

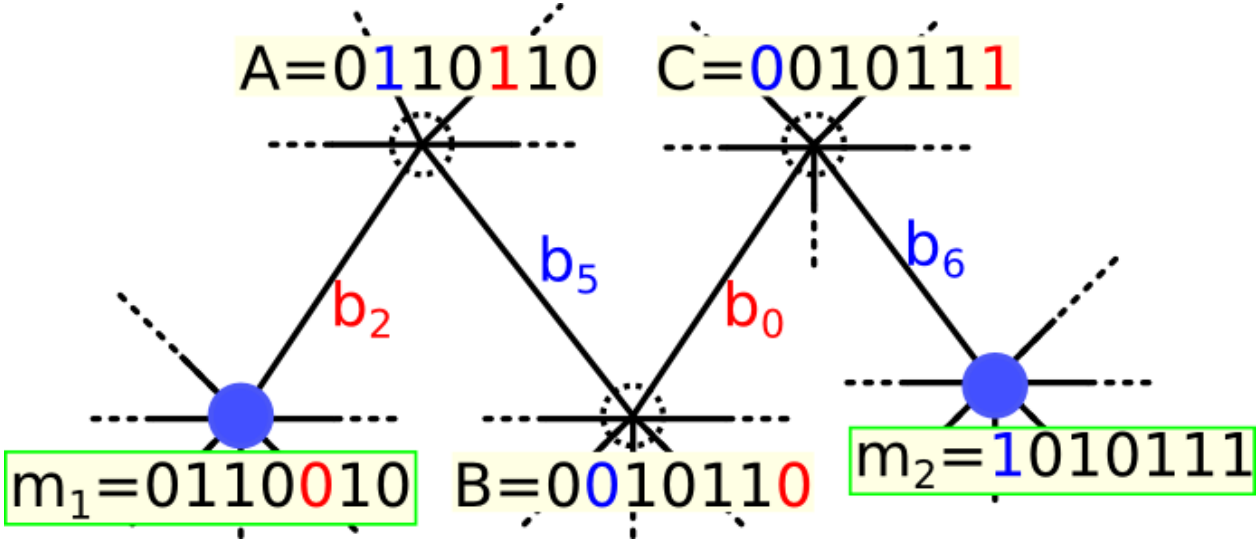
- Soit un code incomplet sur 7 bits, dont on donne une représentation partielle, montrant les mots m1=0110010 et m2=1010111, ainsi que le chemin entre eux (d=4), passant par les noeuds A, B et C.



  - Si le récepteur reçoit le mot A=0110110, il va corriger vers le mot m1
  - Si le récepteur reçoit le mot C=0010111, il va corriger vers le mot m2

Exemple de code correcteur sur 7 bits

- Soit un code incomplet sur 7 bits, dont on donne une représentation partielle, montrant les mots m1=0110010 et m2=1010111, ainsi que le chemin entre eux (d=4), passant par les noeuds A, B et C.



  - Si le récepteur reçoit le mot A=0110110, il va corriger vers le mot m1
  - Si le récepteur reçoit le mot C=0010111, il va corriger vers le mot m2
  - Si le récepteur reçoit le mot B=0010111, il va détecter l'erreur, mais sera **incapable** de la corriger.