Codes détecteurs et correcteurs d'erreurs

Malika More
(malika.more@iut.u-clermont1.fr)
Alex Esbelin
(Alex.Esbelin@math.u-bpclermont.fr)

IREM de Clermont-Ferrand

Formation ISN

30 Juin 2011

Plan du cours

1. Généralités
   - Transmission et stockage de l'information
   - Codage par blocs
2. Premiers Exemples
   - Code de parité
   - Codage par répétition
3. Risques d'erreurs
   - Rappels de probabilités
   - Le pari de la détection des erreurs
   - Le pari de la correction des erreurs
4. Distance
   - Différences entre deux mots
   - Distance minimale du code

Canaux de transmission et supports de stockage

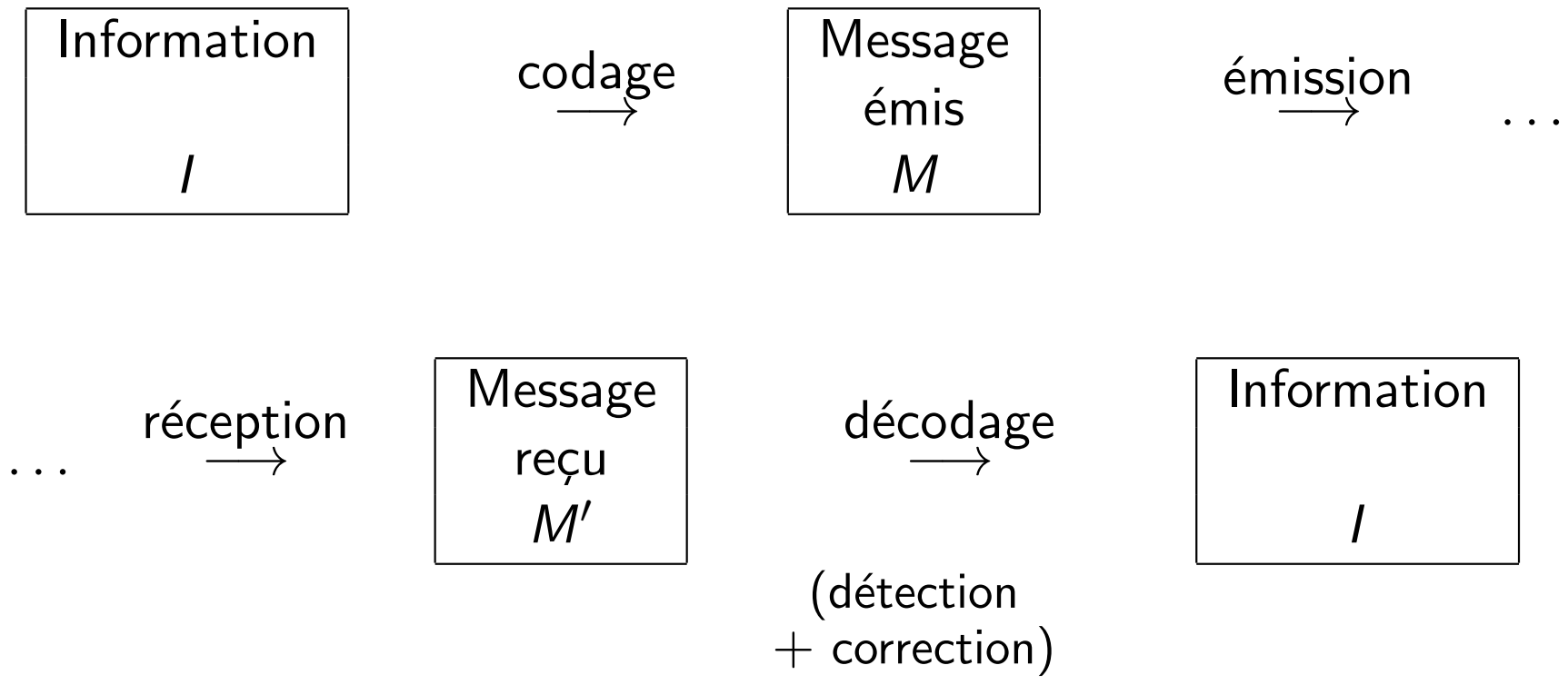
Ils sont forcément imparfaits !

- Lors du transfert et/ou du stockage d'information, des erreurs surviennent.
  - Un 1 devient un 0 ou vice-versa
  - (Pour simplifier, on ne s'occupe pas des bits perdus ou ajoutés)
- On souhaite détecter les erreurs de transmission et/ou de stockage et les corriger.
- C'est le rôle des codes détecteurs et correcteurs d'erreurs.
- On utilise une stratégie de surcodage
  - Plus efficace que de recommencer la transmission ou de stocker plusieurs fois l'information

Codage et décodage

- Avant émission :
  - L'information est surcodée par ajout de *bits de contrôle*
  - Le mot binaire obtenu est le *message émis*
  - C'est le *codage*
- Après réception :
  - À partir du message reçu, en utilisant les bits de contrôle
  - On *détecte* les erreurs
  - On *corrige* le message reçu pour retrouver l'information initiale.
  - C'est le *décodage*

Résumé


Information
I
codage
Message
émis
M
émission
...
...
réception
Message
reçu
M'
décodage
(détection
+ correction)
Information
I

Code de type $\mathcal{C}_{n,m}$

- L'information est découpée en blocs de longueur fixe :

codage

mot d'information : $\boxed{m \text{ bits}}$ $\longrightarrow$ mot de code : $\boxed{n \text{ bits}}$ = m bits d'information + r bits de contrôle

- Redondance : $r = n - m =$ nombre de bits de contrôle
- Rendement : $\rho = \frac{m}{n} < 1$

Messages possibles vs. mots de code

- Bon sens :

  mots d'information $\neq$ $\longrightarrow$ mots de code $\neq$

- Mots d'information de m bits :

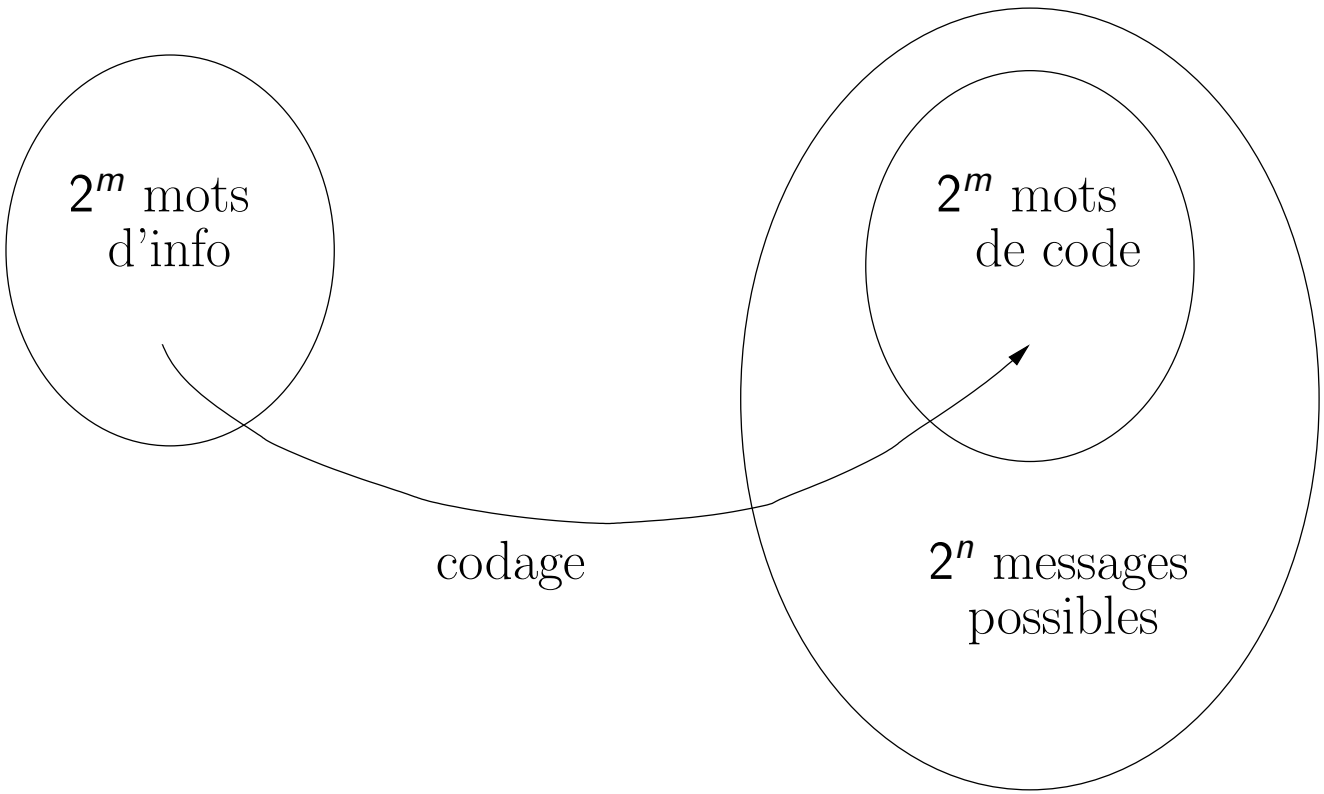
  2^m mots d'information $\longrightarrow$ 2^m mots de code

- Mots de code de n bits :

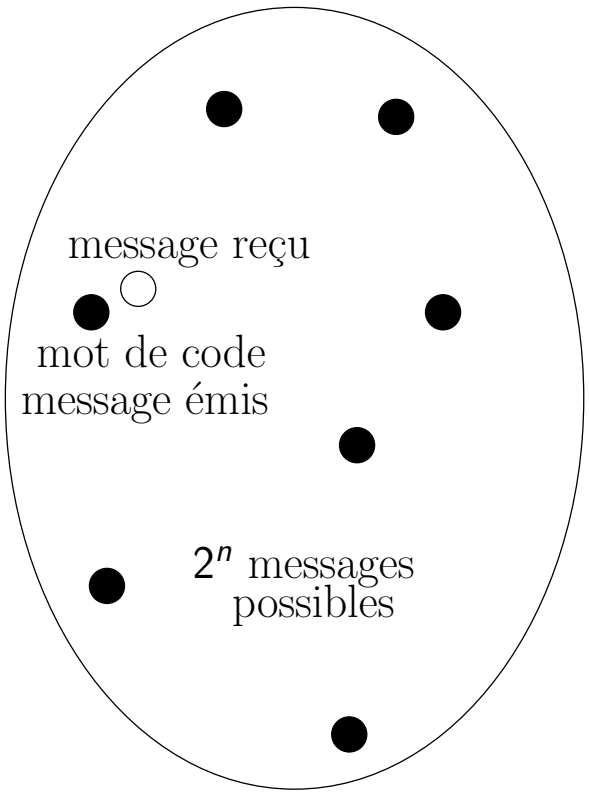
  $\boxed{2^n \text{ messages possibles}}$ parmi lesquels seulement $\boxed{2^m \text{ mots de code}}$

Messages possibles vs. mots de code

- C'est ce qui va rendre possible la détection et la correction d'erreurs



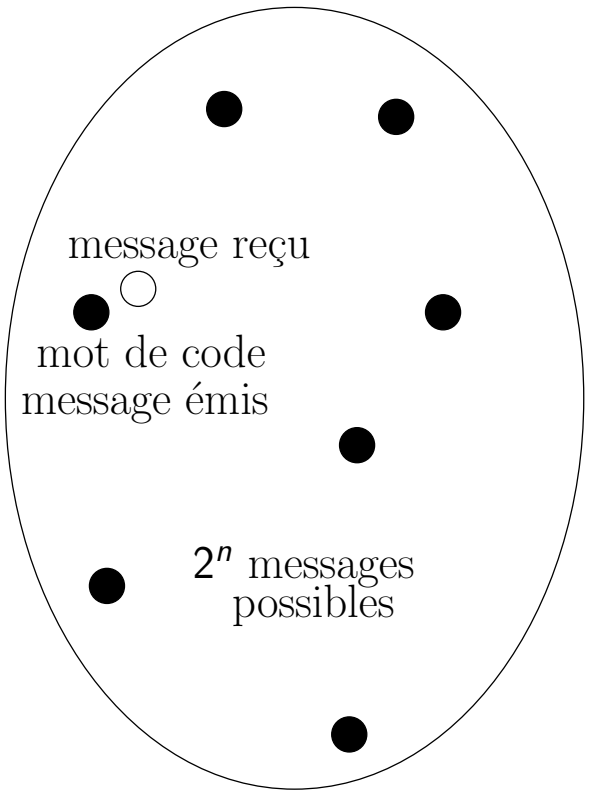
Détection d'erreurs



- S'il y a des erreurs alors le message reçu n'est pas un mot de code
- Conséquence :

Détection

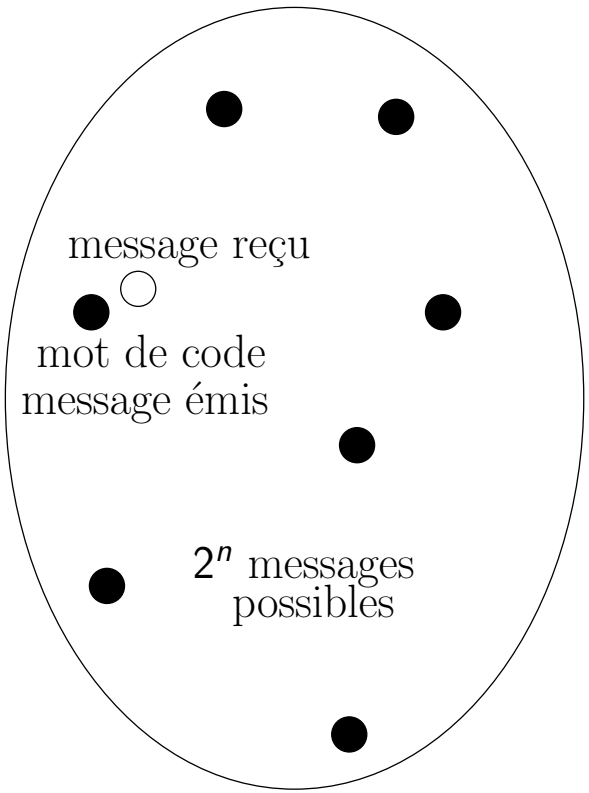
Détection d'erreurs



- S'il y a des erreurs - *jusqu'à un certain point* - alors le message reçu n'est pas un mot de code
- Conséquence :

  Détection. . . *imparfaite*

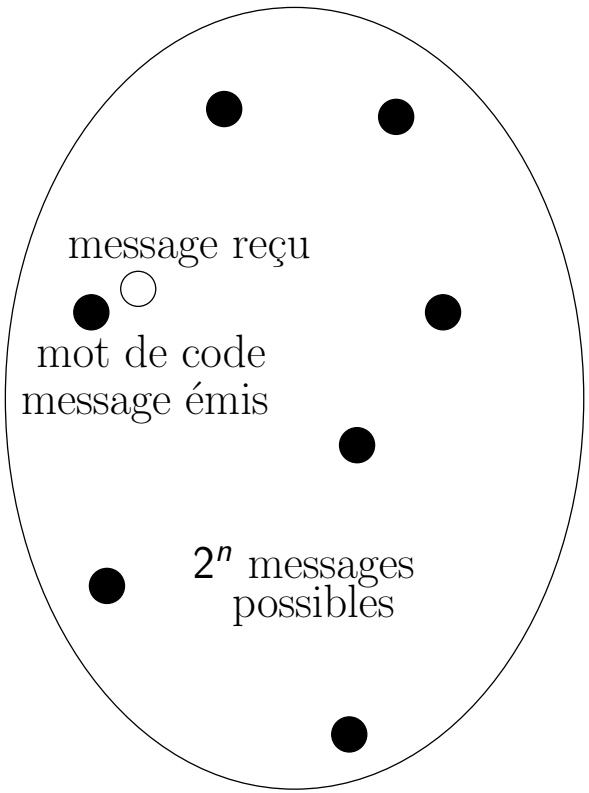
Correction d'erreurs



- Remplacer le message reçu par le mot de code *le plus proche*
- Conséquence :

Correction

Correction d'erreurs



- Remplacer le message reçu par le mot de code *le plus proche*
- Conséquence :

  Correction. . . imparfaite

Efficacité

Pour le choix d'un code :

- On fixe la longueur m des mots d'information.
- On veut un code qui détecte/corrige bien les erreurs :
  - Redondance élevée $\longrightarrow$ r grand
- On veut un code qui est rapide/court :
  - Rendement élevé $\longrightarrow$ ρ grand
- On doit faire un compromis entre :
  - Efficacité de la détection/correction vs. efficacité de la transmission
  - Redondance r vs. rendement ρ

Codage systématique vs. codage entrelacé

- Mot d'information : $iiiiii$
- Mot de code : bits d'information $iiiiii$ + bits de contrôle ccc
  - *Codage systématique* : les bits d'information et les bits de contrôle sont groupés, bits d'information en premier ou l'inverse

    $iiiiiiccc$ ou $ccciiiiii$
  - *Codage entrelacé* : les bits d'information et les bits de contrôle sont mélangés

    $iicicciii$
  - Pour simplifier, on se limitera au cas du codage systématique, même si pour une pratique avancée, les codages entrelacés sont utiles

Définition

- Lors de la transmission de caractères de texte, on utilise le code ASCII.
- Chaque caractère est codé sur 7 bits plus 1 bit de parité (bit de contrôle) en général placé avant les bits d'information
- Le bit de parité est calculé de telle sorte que le nombre total de 1 soit toujours pair (ou impair).

Paramètres

Exemple

$\mathcal{C}_{8,7}$ parité paire

- Mot d'information : 100 1101 $\rightsquigarrow$ Mot de code : 0 100 1101
- Mot d'information : 110 0111 $\rightsquigarrow$ Mot de code : 1 110 0111

- Longueur des mots d'information : $m = 7$
- Longueur des mots de code : $n = 8$
- Redondance : $r = n - m = 1$ (i.e. 1 bit de contrôle)
- Rendement : $\rho = \frac{7}{8} = 0,875$

Détection et correction

Exemple

(Message émis)	Message reçu	Contrôle	Correction
(1110 1000)	1110 1000		
(1110 1000)	1110 1001		
(1110 1000)	0100 1100		
(1110 1000)	0011 0000		

-
-

Détection et correction

Exemple

(Message émis)	Message reçu	Contrôle	Correction
(1110 1000)	1110 1000	OK	
(1110 1000)	1110 1001	anomalie	
(1110 1000)	0100 1100	anomalie	
(1110 1000)	0011 0000	OK	

- Le code de parité permet de détecter une anomalie lorsqu'il y a un nombre impair de bits erronés.
-

Détection et correction

Exemple

(Message émis)	Message reçu	Contrôle	Correction
(1110 1000)	1110 1000	OK	
(1110 1000)	1110 1001	anomalie	?
(1110 1000)	0100 1100	anomalie	?
(1110 1000)	0011 0000	OK	

- Le code de parité permet de détecter une anomalie lorsqu'il y a un nombre impair de bits erronés.
- Le code de parité ne permet pas de corriger les erreurs.

Définition et paramètres

- Transmission bit par bit :

mot d'info	mot de code
0	000
1	111

- Type $\mathcal{C}_{3,1}$
  - Longueur des mots d'information : $m = 1$
  - Longueur des mots de code : $n = 3$
  - Redondance : $r = n - m = 2$ (i.e. 2 bits de contrôle)
  - Rendement : $\rho = \frac{1}{3} = 0,333...$

Détection et correction

Exemple

(Message émis)	Message reçu	Contrôle	Correction
(000)	000		
(000)	100		
(000)	010		
(000)	001		
(000)	110		
(000)	101		
(000)	011		
(000)	111		

Détection et correction

Exemple

(Message émis)	Message reçu	Contrôle	Correction
(000)	000	OK	
(000)	100	anomalie	
(000)	010	anomalie	
(000)	001	anomalie	
(000)	110	anomalie	
(000)	101	anomalie	
(000)	011	anomalie	
(000)	111	OK	

- Le codage par répétition permet de détecter une anomalie lorsqu'il y a un ou deux bits erronés, mais pas lorsqu'il y en a trois.

Détection et correction

Exemple

(Message émis)	Message reçu	Contrôle	Correction
(000)	000	OK	
(000)	100	anomalie	
(000)	010	anomalie	
(000)	001	anomalie	
(000)	110	anomalie	
(000)	101	anomalie	
(000)	011	anomalie	
(000)	111	OK	

- Rappel : On corrige par le mot de code *le plus proche* (avec le moins de bits différents).

Détection et correction

Exemple

(Message émis)	Message reçu	Contrôle	Correction
(000)	000	OK	
(000)	100	anomalie	000
(000)	010	anomalie	000
(000)	001	anomalie	000
(000)	110	anomalie	111
(000)	101	anomalie	111
(000)	011	anomalie	111
(000)	111	OK	

- Le codage par répétition permet de corriger une erreur portant sur un seul bit mais ne permet pas de corriger correctement une erreur portant sur deux bits.

Modèle mathématique

- Schéma de Bernouilli
  - Taux d'erreur du canal de transmission $p < 0,5$
    - $p \approx$ nombre de bits erronés/nombre de bits transmis
  - Transmission d'un message de n bits
  - Hypothèses :
    - symétrie des erreurs sur les 0 et les 1
    - erreurs indépendantes sur chaque bit
    - même si...
- Loi binomiale de paramètres n et p
  - Variable aléatoire $X =$ nombre de bits erronés
    - Les valeurs possibles pour X sont $0 \dots n$
  - Probabilité de recevoir un message avec k bits erronés
    - $\mathcal{P}(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$
    - où $\binom{n}{k} = \frac{n!}{k!(n-k)!}$

Calcul des coefficients binomiaux

Rappel : $n! = n \times (n-1) \times (n-2) \times \ldots \times 2 \times 1$

- $\binom{n}{k} = \frac{n!}{k!(n-k)!}$
- $\binom{n}{0} = \frac{n!}{0!(n-0)!} = 1$
- $\binom{n}{1} = \frac{n!}{1!(n-1)!} = n$
- $\binom{n}{2} = \frac{n!}{2!(n-2)!} = \frac{n(n-1)}{2}$
- etc.

Exemple 1

Un bit reçu sur mille est faux i.e. $p = 0,001$	Code de parité paire $\mathcal{C}_{8,7}$ i.e. $n = 8$

- Probabilité de recevoir un message avec k bits erronés
  - $\mathcal{P}(X = k) = \binom{8}{k} 0,001^k (0,999)^{8-k}$
- Probabilité que le message reçu ne comporte pas d'erreur
  - $\mathcal{P}(X = 0) = 0,999^8 \approx 99,203\%$
- Probabilité que le message reçu comporte une erreur
  - $\mathcal{P}(X = 1) = 8 \times 0,001 \times 0,999^7 \approx 0,794\%$
- Probabilité que le message reçu comporte deux erreurs
  - $\mathcal{P}(X = 2) = \frac{8 \times 7}{2} \times 0,001^2 \times 0,999^6 \approx 0,003\%$
- etc. (jusqu'à 8 erreurs)

Exemple 2

Un bit reçu sur cent est faux i.e. $p = 0,01$	Code par répétition $\mathcal{C}_{3,1}$ i.e. $n = 3$

- Probabilité de recevoir un message avec k bits erronés
  - $\mathcal{P}(X = k) = \binom{3}{k} 0,01^k (0,99)^{3-k}$
- Probabilité que le message reçu ne comporte pas d'erreur
  - $\mathcal{P}(X = 0) = 0,99^3 \approx 97,030\%$
- Probabilité que le message reçu comporte une erreur
  - $\mathcal{P}(X = 1) = 3 \times 0,01 \times 0,99^2 \approx 2,940\%$
- Probabilité que le message reçu comporte deux erreurs
  - $\mathcal{P}(X = 2) = \frac{3\times 2}{2} \times 0,01^2 \times 0,99 \approx 0,030\%$
- Probabilité que le message reçu comporte trois erreurs
  - $\mathcal{P}(X = 3) \approx 0,0001\%$

Message sans erreur détectée

- Il est possible qu'il soit quand même erroné
  - "sans erreur détectée" signifie que le message reçu est un mot de code
  - Le message envoyé était aussi un mot de code, mais on ne peut pas savoir si c'est le même
  - Peut-être que les erreurs qui se sont produites ont transformé le message envoyé en un autre mot de code
- On fait le pari qu'il n'y a effectivement pas eu d'erreurs
  - Autrement dit que le message reçu est identique au message envoyé
  - Est-ce un pari risqué ?
  - On calcule le taux de messages erronés détectés

Exemple 1

Un bit reçu sur mille est faux i.e. $p = 0,001$	Code de parité paire $\mathcal{C}_{8,7}$ i.e. $n = 8$

- Probabilité que le message reçu comporte une ou des erreurs
  - $1 - \mathcal{P}(X = 0) \approx 0,797\%$
- Probabilité qu'un message erroné soit détecté
  - $\mathcal{P}(X = 1) + \mathcal{P}(X = 3) + \mathcal{P}(X = 5) + \mathcal{P}(X = 7) \approx 0,794\%$
- Taux des messages erronés détectés
  - $\frac{\text{Proba erreurs détectées}}{\text{Proba erreurs}} \approx 99,651\%$

Exemple 2

Un bit reçu sur cent est faux i.e. $p = 0,01$	Code par répétition $\mathcal{C}_{3,1}$ i.e. $n = 3$

- Probabilité que le message reçu comporte une ou des erreurs
  - $1 - \mathcal{P}(X = 0) \approx 2,97010\%$
- Probabilité qu'un message erroné soit détecté
  - $\mathcal{P}(X = 1) + \mathcal{P}(X = 2) \approx 2,97000\%$
- Taux des messages erronés détectés
  - $\frac{\text{Proba erreurs détectées}}{\text{Proba erreurs}} \approx 99,997\%$

Message reçu reconnu erroné puis corrigé

- Il est possible qu'il soit en fait mal corrigé
  - "corrigé" signifie que le message reçu erroné est remplacé par le mot de code le plus proche
  - Le message envoyé était aussi un mot de code, mais on ne peut pas savoir si c'est le même
  - Peut-être que les erreurs puis la correction ont transformé le message envoyé en un autre mot de code
- On fait le pari que la correction a atteint son but
  - Autrement dit que le message corrigé est identique au message envoyé
  - Est-ce un pari risqué ?
  - On calcule le taux de messages reconnus erronés bien corrigés

Exemple 2

Un bit reçu sur cent est faux i.e. $p = 0,01$	Code par répétition $\mathcal{C}_{3,1}$ i.e. $n = 3$

- Probabilité qu'un message soit erroné et détecté
  - $\mathcal{P}(X = 1) + \mathcal{P}(X = 2) \approx 2,97\%$
- Probabilité qu'un message soit erroné et bien corrigé
  - $\mathcal{P}(X = 1) \approx 2,94\%$
- Taux de messages reconnus erronés bien corrigés
  - $\frac{\text{Proba erreurs bien corrigées}}{\text{Proba erreurs détectées}} \approx 99\%$

Résumé

Dans tous les exemples réalistes

- Pour $0 \leq k < k' \leq n$, il est beaucoup plus probable de recevoir un message avec k erreurs qu'avec k' erreurs
- Quand on ne détecte aucune erreur dans le message reçu, il est très probable qu'il ne contient effectivement aucune erreur
- Quand on corrige un message erroné par le mot de code le plus proche, il est très probable qu'il soit corrigé correctement

Vecteur de différence

M_1 et M_2 mots de n bits

- $M_1 = 01010110$
- $M_2 = 11010010$

Vecteur de différence entre M_1 et M_2

- Ou exclusif
- $D = M_1 \oplus M_2 = 10000100$
- Position des bits différents

Remarque

- On a $D = M_1 \oplus M_2$
- Mais aussi $M_1 = M_2 \oplus D$ (propriété du "ou exclusif")

Distance de Hamming

M_1 et M_2 mots de n bits

- $M_1 = 01010110$
- $M_2 = 11010010$

Distance de Hamming entre M_1 et M_2

- Nombre de bits dont ils diffèrent, ici 2
- Poids (nombre de 1) du vecteur de différence
- $w(M_1 \oplus M_2) = w(10000100) = 2$

Remarque

- C'est une distance au sens algébrique : les axiomes usuels sont satisfaits ($dist(x, y) = 0$ ssi $x = y$, $dist(x, y) = dist(y, x)$, inégalité triangulaire $dist(x, z) \leq dist(x, y) + dist(y, z)$)

Cas des codes

$M_1 = 01010110$ message émis
$M_2 = 11010010$ message reçu

- Vecteur d'erreur $E = M_1 \oplus M_2 = 10000100$
  - Position des bits erronés
- Distance de Hamming $w(M_1 \oplus M_2) = 2$
  - Nombre de bits erronés

Remarque

- On a dit que quand on détecte un message erroné, on le corrige en le remplaçant par le mot de code "le plus proche".
- On est maintenant en mesure de préciser : "le plus proche" selon la distance de Hamming
- C'est-à-dire avec le moins de bits à modifier

Définition

Distance minimale du code (notée d)

- Plus petite distance de Hamming entre deux mots de code différents
- Plus petit nombre de bits dont diffèrent deux mots de code différents

Exemples

- Le code de parité paire $\mathcal{C}_{8,7}$
  -
  -
- Le code par répétition $\mathcal{C}_{3,1}$
  -
  -

Exemples

- Le code de parité paire $\mathcal{C}_{8,7}$
  - Tous les mots de 8 bits qui ont un nombre pair de 1 sont des mots de code et seulement ceux-là
  - $d = 2$
- Le code par répétition $\mathcal{C}_{3,1}$
  - Les seuls mots de code sont 000 et 111
  - $d = 3$

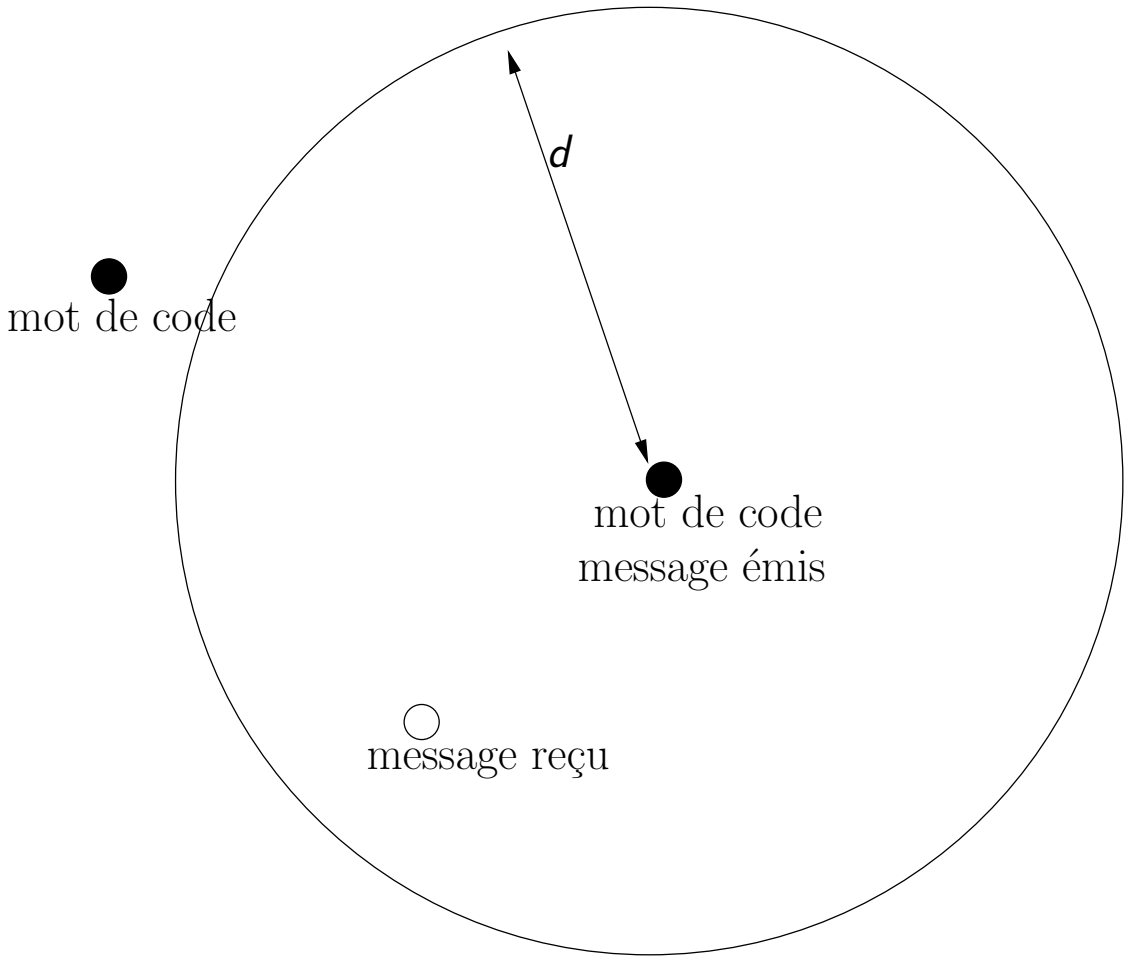
Cas général

- Code $\mathcal{C}_{n,m}$: il faudrait prendre toutes les paires de mots de code différents, calculer à chaque fois leur distance de Hamming, et garder la plus petite valeur obtenue... Il y a 2^m mots de code...

  C'est long

- On verra qu'on peut aller plus vite pour les codes linéaires

Messages erronés avec strictement moins de d erreurs : TOUS DÉTECTÉS

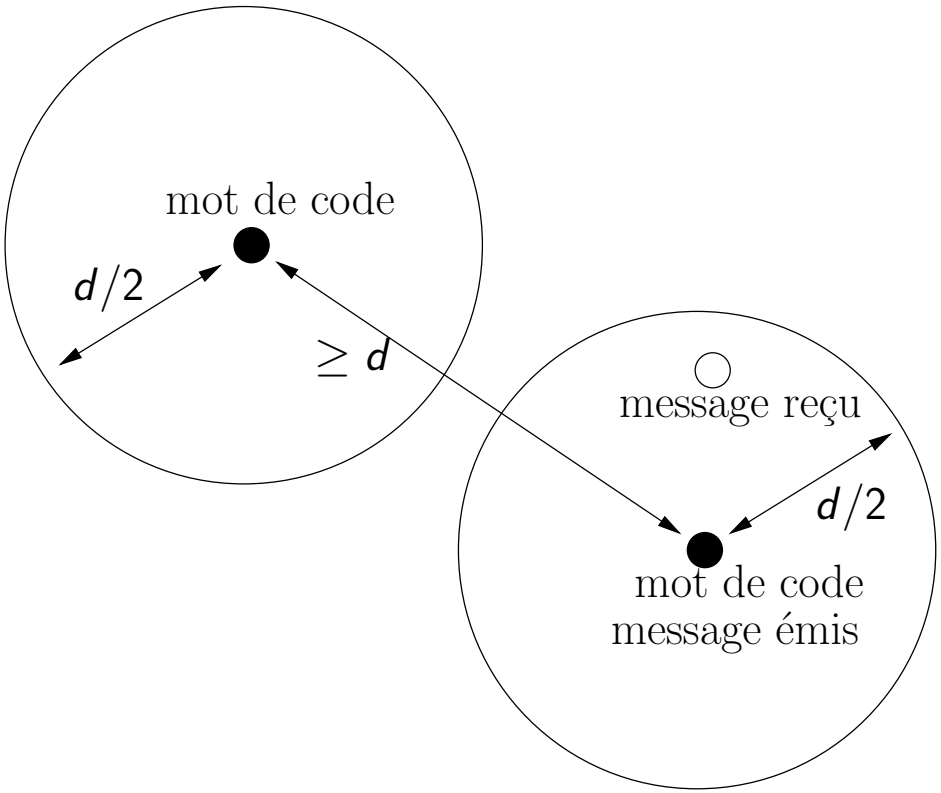


Messages erronés avec strictement moins de d erreurs : TOUS DÉTECTÉS

- Le code de parité paire $\mathcal{C}_{8,7}$: on a vu $d = 2$
  - Les messages avec exactement 1 bit erroné ont une parité impaire, donc ils sont bien détectés
  - Les messages avec 3, 5 ou 7 bits erronés sont aussi détectés, mais les messages avec 2, 4, 6 ou 8 bits erronés ne sont pas détectés

- Le code par répétition $\mathcal{C}_{3,1}$: on a vu $d = 3$
  - Les messages avec 1 ou 2 bits erronés n'ont pas tous leurs bits égaux, donc ils sont bien détectés
  - Les messages avec 3 bits erronés ne sont pas détectés

Messages erronés avec strictement moins de $d/2$ erreurs : TOUS BIEN CORRIGÉS



Messages erronés avec strictement moins de $d/2$ erreurs : TOUS BIEN CORRIGÉS

- Le code de parité paire $\mathcal{C}_{8,7}$: on a $d = 2$, donc $d/2 = 1$
  - Les messages avec strictement moins de 1 erreur sont les messages sans erreur, ils sont évidemment bien corrigés
  - Les messages avec au moins une erreur ne sont pas corrigés

- Le code par répétition $\mathcal{C}_{3,1}$: on a $d = 3$, donc $d/2 = 1,5$
  - Les messages avec 1 bit erroné sont correctement corrigés
  - Les messages avec 2 bits erronés sont corrigés, mais incorrectement

Notion essentielle

Mesurer l'efficacité du code

- Messages erronés avec

  Strictement moins de d erreurs : TOUS DÉTECTÉS

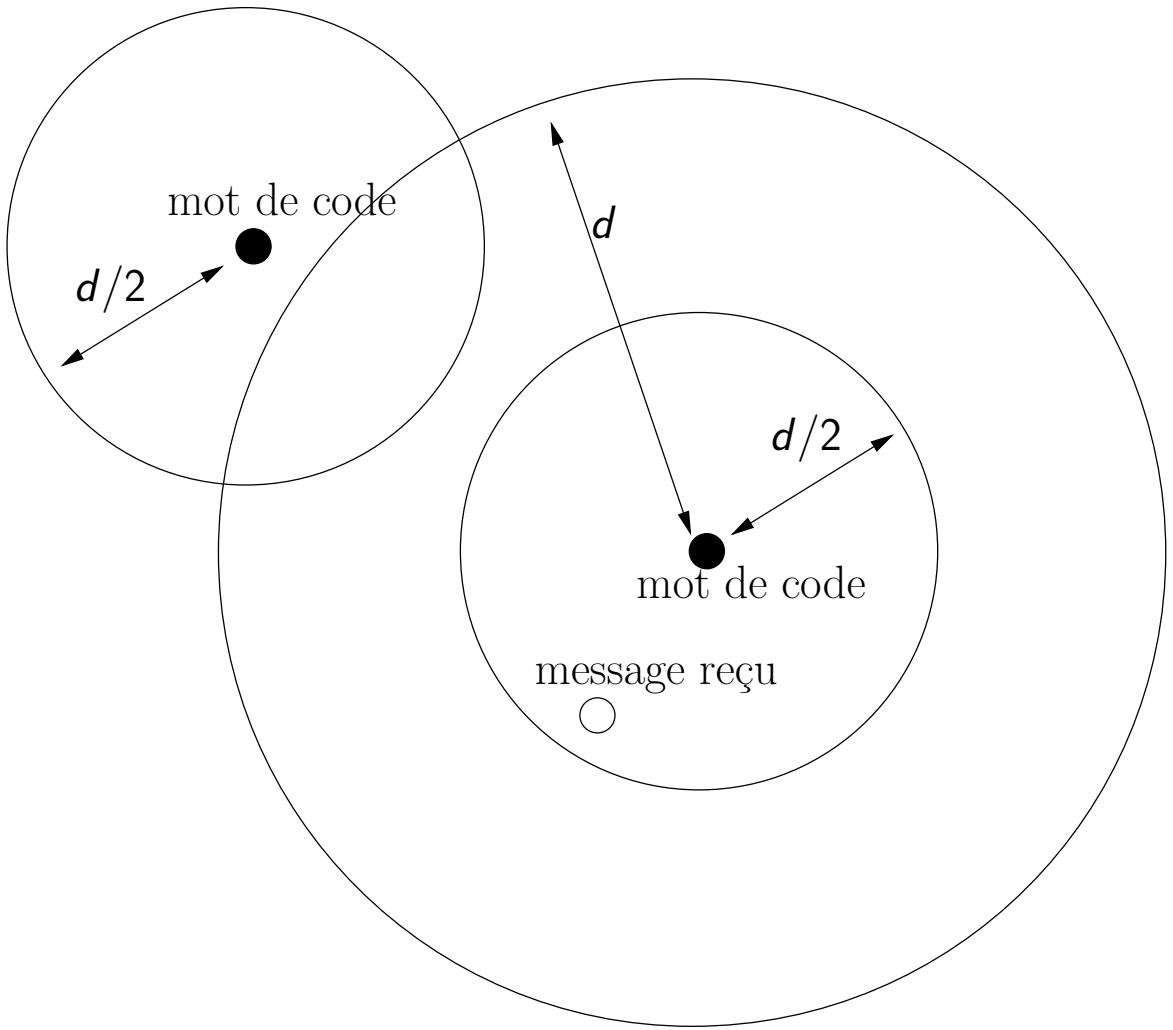
  Les autres : certains pas détectés

- Messages erronés avec

  Strictement moins de $d/2$ erreurs : TOUS BIEN CORRIGÉS

  Les autres : certains mal corrigés

Résumé



Retour sur le type des codes

La distance minimale du code joue un rôle tellement important en théorie et en pratique qu'on l'introduit dans la description du type des codes :

Code de type $\mathcal{C}_{n,m,d}$

- Code de parité : $\mathcal{C}_{8,7,2}$
- Code par répétition : $\mathcal{C}_{3,1,3}$

FIN