



Détection et correction d'erreurs...

... au sein de la couche 2 : liaison de données

Introduction

- Les erreurs de transmission sont :
 - **rare**s sur des supports numériques (fibre optique),
 - beaucoup plus **fréquentes** sur les boucles locales du RTC (paires torsadées analogiques) ou les réseaux sans fil.
- Ces erreurs se produisent la plupart du temps en **rafale** :
 - **Avantage** (par rapport aux erreurs isolées, ie 1 seul bit) : en moyenne, moins de blocs de bits sont affectés. Exemple :
 - Soit la taille des blocs = 1000 bits, le taux d'erreur = 1/1000
 - Erreurs isolées : la plupart des blocs contiennent 1 erreur
 - Erreurs en rafale de 100 ou plus : 1 ou 2 blocs sur 100 seulement
 - **Inconvénient** : beaucoup plus difficiles à détecter et à corriger

Il faut donc apprendre à vivre avec les erreurs...

- Deux stratégies ont été développées. L'émetteur inclut dans le bloc de données :
 1. **suffisamment** de redondance pour que le récepteur puisse **reconstituer** les données originales.
 - ⇒ utilise des **codes correcteurs d'erreur**
 - ⇒ plutôt pour des canaux non fiables comme le sans fil
 2. **juste assez** de redondance pour que le récepteur puisse **détecter** les erreurs et **demander une retransmission**.
 - ⇒ utilise des **codes détecteurs d'erreur**
 - ⇒ plutôt pour des canaux fiables comme la fibre optique

Principe

- On souhaite envoyer **m** bits de données.
- On y rajoute **r** bits de redondance selon un certain « algorithme » ou « codage ».
⇒ Ainsi, la longueur de la trame envoyée est **n = m + r**.
(cette séquence de **n** bits \equiv un mot du code)
- A la réception, en fonction du « codage », on pourra détecter (2.) et/ou corriger (1.) des erreurs.
- En général, l'approche (1.) induit davantage de redondance, ce qui diminue le débit utile du canal...

Généralités sur les codes

- Un **code C** de longueur **n** est un ensemble de **mots** (séquences) de **n** bits.

Ex : $n = 3$, $C = \{ 110, 101, 011 \}$

- Parmi toutes les séquences possibles de n bits (2^n) :
 - celles qui appartiennent à C sont valides.
 - celles qui n'appartiennent pas à C sont invalides.

Ex : 111 est invalide
 101 est valide

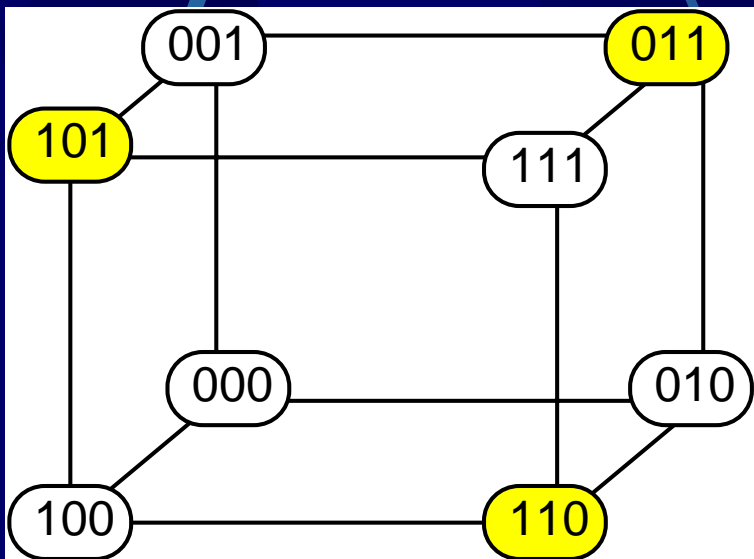
A la réception d'une séquence S de n bits...

- Soit S est invalide (n'appartient pas au code C).
 - Il y a forcément eu une (ou plusieurs) erreur de transmission.
 - Le récepteur corrige (1.) ou demande une retransmission (2.).
- Soit S est valide (appartient au code C).
 - La séquence S est « **considérée** » comme correcte et acceptée par le récepteur.
 - Remarque : « **considérée** » seulement, car...
 - si l'émetteur envoie **011** et le récepteur reçoit **101**,
aucun moyen de détecter qu'il y a eu des erreurs (2 ici)...
car **101** est valide !

Distance de Hamming d'un code

- C'est le critère qui permet d'évaluer le pouvoir détecteur d'un code ainsi que son pouvoir correcteur.
- Distance de Hamming **entre 2 mots** (noté d_h)
= nbre de positions qui ont des valeurs distinctes.
Ex : $d_h(110011, 101010) = 3$
(Astuce = nbre de 1 du OU exclusif)
- Distance de Hamming **d'un code C** (noté $D_H(C)$)
= le minimum des distances entre 2 mots du code
Ex : $D_H(\{110, 101, 011\}) = 2$
 $D_H(\{0011, 0101, 1001, 0110, 1010, 1100\}) = 2$

Représentation graphique



- Sommets : ts les mots de n bits
- Liens : entre les mots tq $d_h = 1$
- $D_H(C) =$ longueur du plus court chemin entre 2 mots valides

Pouvoir détecteur d'un code

- Déf : On parle d'**erreur d'ordre k** lorsqu'un mot émis u diffère par k bits du mot reçu v , ie $d_h(u,v) = k$.

- Pour **détecter** une **erreur d'ordre 1**, quelle doit être la distance de Hamming du code ?

Réponse : $D_H(C) = 2$

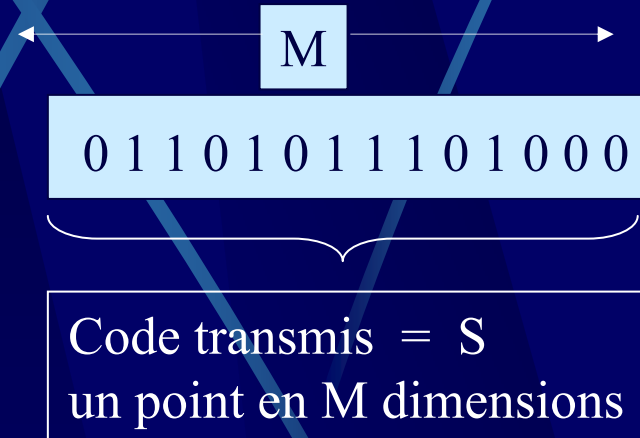
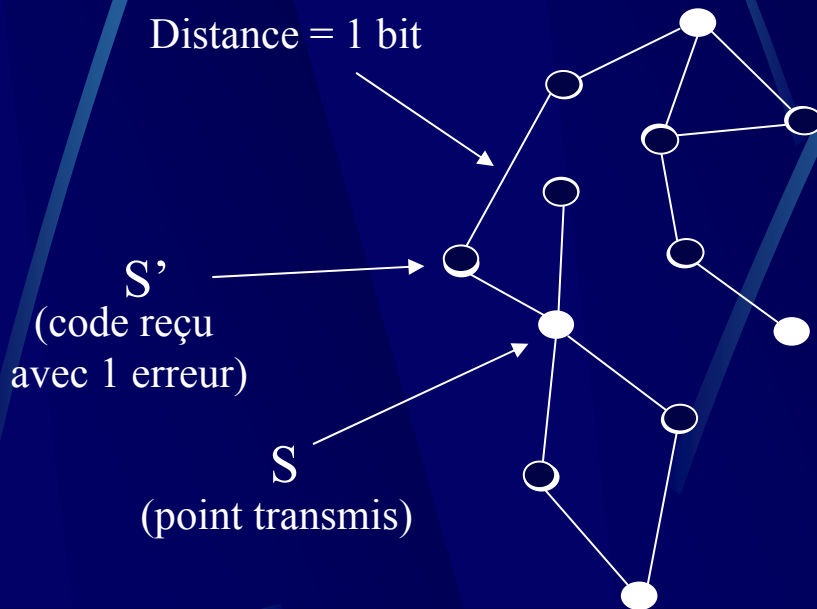
en effet, dans ce cas, 1 erreur simple ne peut pas changer un mot du code en un autre mot du code.

- Même question pour une **erreur d'ordre k** ?

Réponse : $D_H(C) = k+1$

Pouvoir correcteur d'un code

- mot de code valide
- point dans l'espace



Au décodeur:
 S est toujours le mot de code
le plus près du mot reçu S'
→ on décode sans erreur ...

Pouvoir correcteur d'un code

- Pour pouvoir **corriger** une **erreur d'ordre 1**, une distance de Hamming $D_H(C) = 2$ est-elle suffisante ?

Réponse : non... il faut **$D_H(C) = 3$**

en effet, si $D_H(C) = 2$, 1 erreur simple peut nous produire un mot exactement « au milieu » de 2 mots du code, à distance 1 de chacun !

si $D_H(C) = 3$, 1 erreur simple produit un mot qui reste le plus proche du mot transmis : on peut donc le retrouver !

- Pour corriger une **erreur d'ordre k**, il faut **$D_H(C) = 2k+1$** .

Résumé pouvoir détecteur et correcteur

- Un code C peut
 - **détecter** des erreurs d'ordre $D_H(C) - 1$
 - **corriger** des erreurs d'ordre $(D_H(C) - 1)/2$ (partie entière)

Distance de Hamming du code	Ordre maximal des erreurs détectables	Ordre maximal des erreurs corrigeables
1	-	-
2	1	-
3	2	1
4	3	1
5	4	2
6	5	2



Exemples de codes détecteurs

Détection des erreurs

- Permet de vérifier l'intégrité d'une trame au récepteur
- Retransmission des trames corrompues
- Plus efficace que la correction
 - requiert moins de bits de redondance
- Parité Verticale
- Parité Horizontale
- Parité Verticale et Horizontale
- **CRC** : “Cyclic Redundancy Code”
 - codes de détection couramment utilisés

Parité Verticale (Exo 2.1)

- Exemple : envoi de 7 caractères de longueur 3 ($m=3$).

<u>Info utile</u> :	000	111	000	110	101	011	010
<u>Info envoyée</u> :	000 0	111 1	000 0	110 0	101 0	011 0	010 1

- Propriétés :
 - distance de Hamming est 2 : détecte les erreurs simples
 - détecte les erreurs qui sont d'ordre impair
 - ignore les erreurs doubles et toutes celles qui sont d'ordre pair

Parité Horizontale (Exo 2.2)

- Exemple : envoi de 7 caractères de longueur 3 ($m=3$).

<u>Info utile :</u>	000	111	000	110	101	011	010	
<u>Info envoyée :</u>	000	111	000	110	101	011	010	010

Parité Verticale et Horizontale (Exo 2.3)

- Exemple : envoi de 7 caractères de longueur 3 (m=3).

Info utile : 000 111 000 110 101 011 010
Info envoyée : 000**0** 111**1** 000**0** 110**0** 101**0** 011**0** 010**1** **1010**

0	1	0	1	1	0	0	1
0	1	0	1	0	1	1	0
0	1	0	0	1	1	0	1
0	1	0	0	0	0	1	0

- Propriétés :
 - détecte les erreurs d'ordre 3 et corrige les erreurs simples
 - détecte les erreurs qui sont d'ordre impair

Codes Polynomiaux : CRC

- Codes de blocs particuliers
- Facilement implémentables de façon matérielle
- Excellents résultats
- Principe :
 - Toute séquence de n bits peut être représentée par un polynôme à coefficients binaires.
 - Opérations d'addition et de multiplication modulo 2.
 - $G(X)$ un polynôme de degré r appelé polynôme générateur.
 - Code polynomial $C_{G,n}$:
 - Ensemble des séquences de longueur n , dont le polynôme associé est multiple de $G(X)$.

CRC: Propriétés

- Peut détecter 1 erreur isolée si G contient au moins 2 termes
- Peut détecter 2 erreurs (si G ne divise pas x^k+1 , avec k =fenêtre)
- Peut détecter *tous les patrons d'erreurs impairs*
→ si $(x + 1)$ est un facteur de $G(x)$
- Peut détecter r erreurs consécutives s'il est d'ordre r ("bursts")
- Trois polynômes standards :
 - CRC-12 : $x^{12} + x^{11} + x^3 + x^2 + x + 1$
 - CRC-16 : $x^{16} + x^{15} + x^2 + 1$
 - CRC-CCITT : $x^{16} + x^{12} + x^5 + 1$