

## Texte sur la détection/correction

[1] Nous allons voir les principes généraux sur lesquels reposent les phases de détection et de correction de tous les codes détecteurs et correcteurs d'erreurs.

[2] Commençons par la détection. On rappelle que les mots de code ne constituent qu'une petite partie des messages reçus possibles. C'est ce qui va permettre la détection des erreurs.

[3] On représente plus exactement les mots de code de façon dispersée dans l'ensemble des messages reçus possibles, par des points noirs [4].

L'un d'eux correspond au message émis [5].

Bien que la plupart des messages soient transmis sans erreurs, plaçons-nous dans le cas où le message reçu est erroné, c'est-à-dire qu'il n'est pas identique au message émis, et voyons comment on peut détecter cette anomalie.

[6] Le message erroné reçu est représenté par un point blanc. Le destinataire ne connaît pas le message émis, mais lorsque, comme sur ce schéma, le message reçu ne correspond pas à un mot de code, il a la certitude que quelque chose ne va pas : l'erreur est détectée.

[7] Remarquons cependant que ceci n'est valable que jusqu'à un certain point car rien n'empêche que, par malchance, l'ensemble des erreurs pendant la transmission ait transformé le message émis en un mot de code différent. Cela signifie que la détection des erreurs est par nature imparfaite !

[8] C'est pourquoi, quand le message reçu est bien un mot de code, le destinataire fait le pari qu'il s'agit effectivement du message émis, c'est-à-dire qu'il ne contient pas d'erreur. C'est un pari raisonnable car en réalité, il est très très probable que ce message reçu sans erreur détectée soit effectivement le message émis. Un document annexe comportant quelques calculs des probabilités précise ce qu'on entend par « très très probable ».

[9] Une fois une anomalie détectée, il s'agit de tenter de corriger le message erroné. Plaçons-nous de nouveau dans le cas où le message reçu n'est pas un mot de code.

[10] Le principe de la correction consiste simplement à remplacer le message reçu erroné par le mot de code *le plus proche*, ici représenté en rouge, c'est-à-dire celui nécessitant de modifier le moins de bits possible.

[11] Remarquons que, de même que pour la détection, la correction peut être mise en défaut car l'ensemble des erreurs pendant la transmission pourrait avoir donné un message reçu plus éloigné du message émis que d'un autre mot de code.

[12] Dans ce cas, la correction proposée est incorrecte. Ainsi, la correction des erreurs est elle aussi par nature imparfaite.

[13] Le destinataire fait encore une fois un pari sur la correction, en s'appuyant sur le même principe probabiliste qui assure qu'en réalité il est très peu probable qu'il se trompe lors de la correction en appliquant la règle du minimum de modifications.

[14] Pour conclure, nous avons vu que la détection des erreurs consiste à vérifier si le message reçu est bien un mot de code. Lorsque ce n'est pas le cas, la correction consiste à remplacer le message erroné reçu par le mot de code le plus proche, obtenu en modifiant le moins de bits possible. Tant que les messages reçus ne contiennent qu'un petit nombre d'erreurs, ce qui arrive le plus souvent, ces deux principes permettent de détecter et de corriger correctement les erreurs. Cependant, nous avons vu aussi qu'un code capable de détecter et de corriger parfaitement toutes les erreurs ne peut pas exister.

[15]