

Introduction

[1] Les codes détecteurs et correcteurs d'erreurs sont utilisés par de nombreuses fonctionnalités informatiques. Nous n'en n'avons pas conscience car ils sont transparents pour l'utilisateur. Et pourtant de tels codes interviennent chaque fois que des données numériques sont stockées ou transmises. Nous allons voir pourquoi ils sont nécessaires et comment ils fonctionnent. Partons d'une situation courante. [2] Lorsque nous parlons avec un interlocuteur à l'aide de nos téléphones mobiles, des données sont émises et reçues. À l'intérieur des téléphones, [3] les données sont traitées sous forme numérique. [4] Ce sont des suites de 0 et de 1 appelés des bits, pour *binary digits* en Anglais. Pour cette raison, on parle aussi de données binaires.

La transmission peut être perturbée par divers phénomènes physiques. [5] Ces perturbations sont d'origines très diverses : [6] des ondes électromagnétiques, [7] des interférences dues à des changements de l'environnement... Elles altèrent les données binaires échangées. Il est également possible que des modifications physiques, [8] comme par exemple une micro-rayure sur un CD, altèrent directement les données stockées sur un support numérique. Nous ne nous intéresserons qu'au problème de la transmission, mais les mêmes solutions s'appliquent au cas du stockage. [9] Les perturbations provoquent de temps en temps aléatoirement le changement d'un 0 en 1 ou inversement d'un 1 en 0. Ces modifications des données sont appelées des erreurs. [10] Les erreurs provoquées par ces phénomènes sont imprévisibles et inévitables compte tenu de la quantité de bits échangés et de l'instabilité de l'environnement. Il peut finalement paraître étonnant que la transmission de données numériques soit tout simplement possible ! La solution à ce problème est précisément fournie par l'utilisation massive des codes détecteurs et correcteurs d'erreurs.

Nous allons maintenant décrire leur fonctionnement. [11] Les bits des données initiales à transmettre sont appelés des bits d'information. Chez l'expéditeur, une opération appelée *codage* consiste à leur adjoindre des bits supplémentaires, appelés des bits de contrôle. Ainsi, le codage produit une nouvelle suite de bits : c'est le message émis. [12] Les bits de contrôle ne sont pas choisis n'importe comment : ils sont calculés à partir des bits d'information selon des règles spécifiques à chaque code. Comme les bits de contrôle sont construits à partir des données initiales, ils ne contiennent pas d'information nouvelle, on dit qu'ils sont *redondants*.

[13] Chez le destinataire, après réception, le message reçu subit d'abord une phase de détection et de correction des erreurs éventuelles. Ces opérations obéissent elles aussi à des règles spécifiques à chaque code et prennent en compte l'ensemble des bits reçus. [14] Ensuite, le décodage consiste à supprimer les bits de contrôle afin d'obtenir les données reçues. Cet ensemble de procédures est le plus souvent efficace, c'est-à-dire que les données reçues sont finalement identiques aux données initiales tant que les perturbations n'ont pas été trop importantes.

[15] En conclusion, nous avons vu pourquoi envoyer et recevoir des données binaires utilisables nécessite de passer par un code détecteur et correcteur d'erreurs. Les différentes phases du processus sont les suivantes : chez l'expéditeur, avant l'émission du message, le codage consiste à adjoindre des bits de contrôle et chez le destinataire, après réception, il y a d'abord une phase de détection et de correction des erreurs avant le décodage qui permet de récupérer les données reçues.

