Barret Alexandre - Mizoules Vincent

Compte rendu - TP3

Usage

Pour lancer notre programme :

```
make && ./tp3
```

Pour nettoyer le projet

```
make clean
make clear
```

Il est possible de désactiver l'affichage des informations de debug des fonctions en modifiant les `define` présents au début du programme (remplacer `true` par `false`).

Réponses aux questions

1)

Pour pouvoir mieux analyser le codage nous avons utilisé un fichier `input2.txt` qui contient moins de texte. Cela permet une analyse plus aisée.

Le codage est réalisé en prenant chaque bitset contenu dans le vector issu de la fonction `readFile`. Nous lui appliquons la fonction `bitsetHammingEncoding` qui opère un produit matriciel (c=M.G).

Remarque : il est possible d'utiliser le fichier d'origine (renommé `input1.txt`)

2)

Le décodage suit le même principe que le codage. Nous appliquons la fonction `bitsetHammingDecoding` a chaque bitset du vector.

3)

Nous simulons la transmission des données, en rajoutant des erreurs via la fonction `injectError`. Cette dernière ajoute un nombre d'erreur pour chaque bitset. Ce nombre peut être géré via la constante `MAX_ERROR_PER_BITSET`. Les erreurs sont placées aléatoirement dans le bitset et sont affichées dans la console lorsque le debug est actif.

4)

Pour calculer la distance de Hamming entre deux mots, nous partons de la valeur de chaque mot sous forme `Int`. En les comparant et itérant, nous obtenons la distance de ces deux mots.

Afin d'obtenir la distance minimale du code de Hamming (7,4), nous générons l'alphabet complet du code de base. Puis nous calculons la distance entre chacun des mots convertis dans le code de Hamming (7,4). Lors de ces itérations nous retenons la distance minimale et l'affichons à la fin de l'exécution du programme.

Ici, nous obtenons bien une distance minimale de **3**.