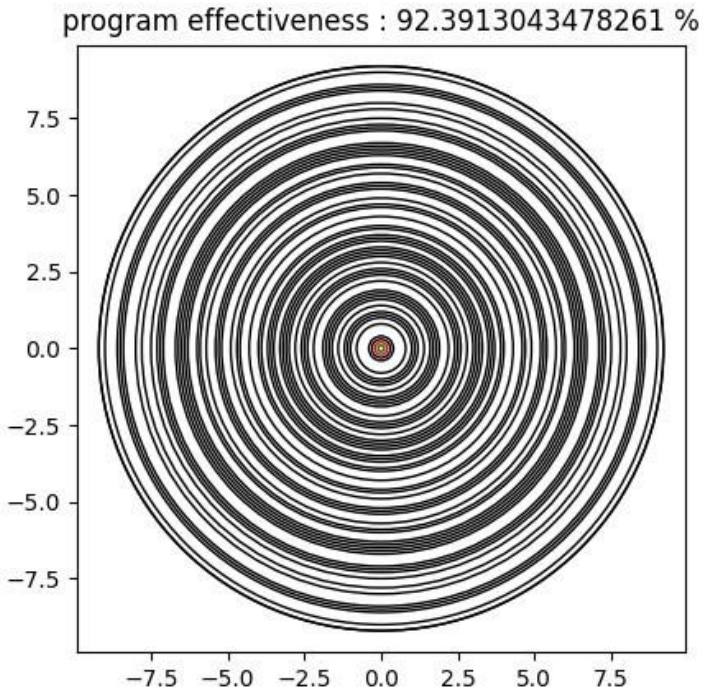
DOSSI MOUNDELE GLORIA GODVIE

LARIOL NY AINA SEHENOSOA

UNIVERSITÉ Clermont Auvergne

PROJET DU RÉSEAU II



Introduction

Dès l'annonce du sujet de Travaux Pratiques Réseaux nous avons essayé de trouver un moyen original de pouvoir transmettre une information, mais être original dans le domaine de l'informatique n'est pas chose aisée. La première idée qui nous est venue était de transmettre l'information autour d'un cercle. Un cercle composé de 8 divisions égales, ce qui correspondait bien avec la composition d'un octet, nous trouvions donc le lien assez intéressant. Nous avons donc commencé à coder cette idée et nous avons rapidement avancée. Nous aurions également pu coder plus d'informations en utilisant des couleurs, mais nous trouvons cette méthode trop contraignante

pour sa lecture, les aléas sont trop nombreux, c'est pourquoi nous essaierons de passer le plus possible de couleurs.

Programmation

Lancement du programme

Pour la programmation, on a décidé de faire ce projet en **Python (version 3)** car c'est l'un des langages qu'on maîtrise le plus, mais aussi car c'est un langage très polyvalent qui nous permet d'avoir beaucoup de libertés et possède une communauté très active. On était sûr de ne pas rester bloquer au milieu du projet en faisant ce choix.

Pour mener à bien ce projet nous avons donc utilisé plusieurs bibliothèques, nécessaires notamment pour la partie graphique. Il est donc impératif de les posséder sur son ordinateur avant d'exécuter le programme :

- svgwrite (https://pypi.org/project/svgwrite/)
- numpy (https://numpy.org/install/)
- pillow (https://pillow.readthedocs.io/en/stable/installation.html)
- resizeimage(https://pypi.org/project/python-resize-image/)
- math et time, inclus dans **Python3**

La plupart des installations ont été faite avec **pip3** .
Le lancement du programme se fait simplement avec la commande :

python3 projet_reseau2.py

Il suffit ensuite de suivre le menu affiché et le programme effectuera la demande.

Construction de la matrice

Pour la construction de la matrice, on a choisi d'utiliser le format Longueur-Donnée Redondance-Fin dans un premier temps. L''utilisateur entre alors la chaîne de caractère qu'il veut coder et à ce moment-là le programme peut calculer la longueur du message, cette information sera la première de notre matrice. On a établi la limite maximum de cette longueur à 255. Il traite ensuite le message et transforme chaque caractère par son code ASCII. Grâce à la matrice qu'on a actuellement, on peut calculer et ajouter la redondance d'information (détaillé dans le paragraphe suivant) et ajouté une donnée de fin arbitrairement choisie.

Une fois notre matrice d'entiers obtenue il peut transformer ces entiers en binaire. Attention, les valeurs des données de base n'excédant pas 128 peuvent se coder sur 8 bits mais la

redondance, elle, se code sur 16 bits. on repère donc les octets de redondance et les divise en deux octets de 8 bits.

Avantages :
La forme circulaire permet la lecture dans n'importe quel sens.
Un peut ajouter un logo assez facilement.
On ne fait pas la différence entre les bits d'informations et les bits de correction donc nous n'avons pas besoin de plus de bit pour coder cela.

Inconvénients :
On perd beaucoup d'espace car notre protocole augmente le rayon des cercles même s'il ne trace rien.
Le pourcentage de l'espace utilisé varie selon le nombre de bit et selon l'information, pour cela on vous laisse voir l'efficacité directement sur le code.

Interface :

Les deux premiers bits correspondent a des bits du protocole et non à l'information.
Sur cette version du protocole on utilise ces deux bits pour donner la version par exemple mais aussi le plus important c'est de donner le départ et l'écart de lecture pour pouvoir lire ce protocole.

Ensuite pour chaque bit dans notre tableau d'information et de correction si c'est un 1 le protocole trace un cercle et montre le rayon d'une valeur donné qu'on va appeler z qui correspond à l'écart entre chaque bit.
Si le bit est 0 le protocole ne trace rien mais augmente le rayon de z.
Le protocole répète cette manœuvre jusqu'au dernier bit.

Lecture :

La lecture est très similaire à l'écriture. Pour lire le protocole toutes les informations sont disponibles dans le protocole. Il commence par repérer le cercle le plus au centre et il le lie les deux premières bit (pour le moment on utilise seulement deux bit, l'idéal c'est de mettre 16 bits) on récupère la version et en lisant ces deux bits on calculs l'écart entre les deux pour pouvoir lire la suite, on va appeler cet écart Z.

Ensuite à chaque étape on saute de Z et s'il y'a un cercle on ajoute un à notre tableau d'information, s'il n'y a rien on ajoute 0.
L'information qui consiste à corriger l'erreur se trouve sur toutes les positions de puissances de 2.

L'autre bit c'est ceux de l'information. Si on détecte trop d'erreur on peut relire le protocole.

Information supplémentaire :

Les informations supplémentaires sont les bit de parité du code correcteur ainsi que 2 bits pour la version et pour récupérer la taille de l'écart pour lire.

Efficacité :

L'efficacité dépend de l'information

Robustesse :

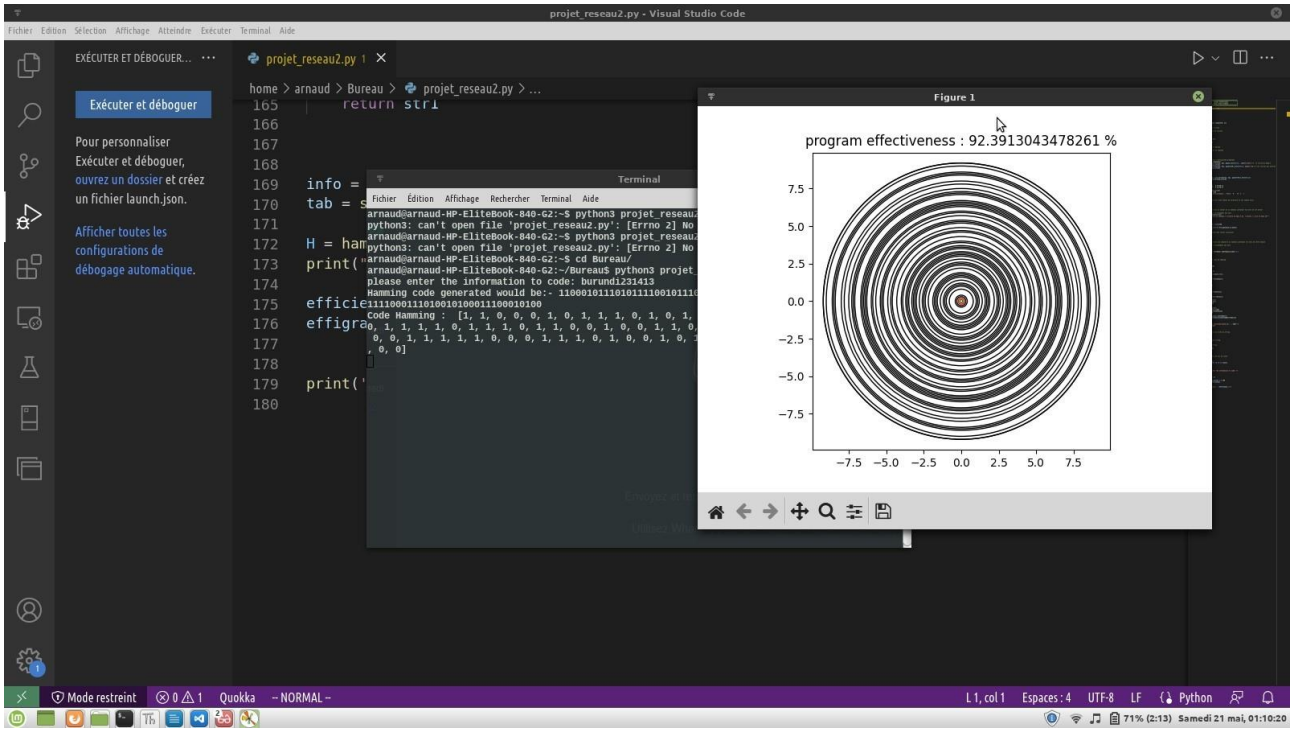
D'une certaine manière ce code est assez robuste car même si on perd une très grande partie de l'image générée par le protocole, par exemple où elle se déchire nous pourrons quand même lire le protocole même s'il ne reste qu'un dixième de l'image.

Contrôle d'erreur :

Le contrôle d'erreur de fait après avoir récupéré les bits de parité dans les positions des puissances de deux.

Esthétique :

Le protocole est assez esthétique lorsque nous travaillons avec quelques dizaines de bits, en revanche lorsque le nombre de bits deviennent trop élevés le protocole commence à ressembler à un gros cercle noir et sont illisibles sauf si on zoom.



La figure ci-dessus nous montre l’image et une partie de la matrice après avoir entré la chaîne de caractère à coder.

Conclusion

Pour conclure, on a bien aimée travailler sur ce projet car la vision de sa progression est très motivante et satisfaisante. On a pu découvrir des bibliothèques inconnues et améliorer notre niveau dans le langage Python qui nous parait être un langage indispensable à maîtriser. On a également été très curieuse sur le domaine du réseau et de la transmission de données mais aussi du lien avec les Mathématiques pour la sécurisation des données.