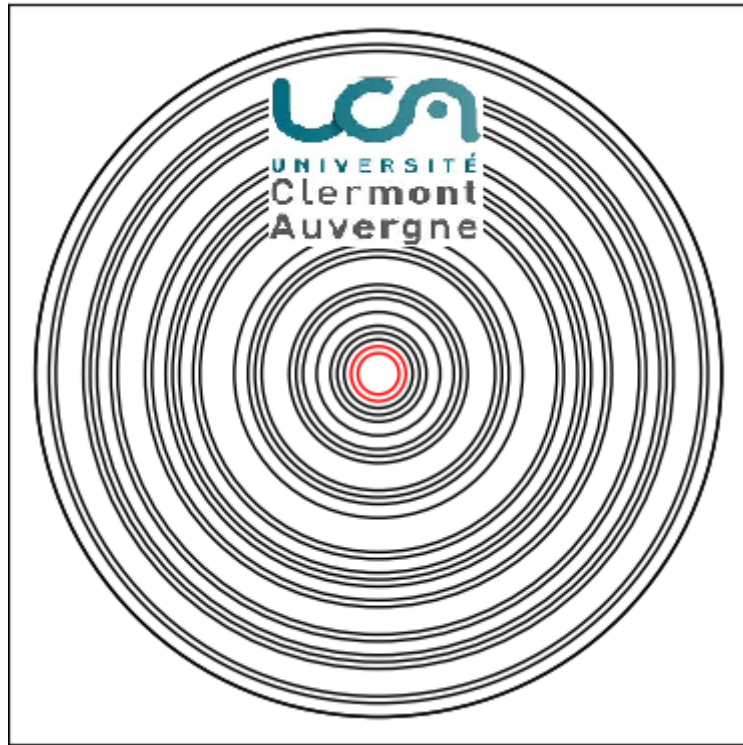


Adouani Mohtadi

Samuel Da Silva

Sybill Rault



Dans le cadre d'un TP en cours de réseau , la consigne est de créer un nouveau protocole graphique qui pourrait contenir de l'information.

Avec mes collaborateurs sur le projet nous avons choisi un protocole en forme de cercle afin de faciliter beaucoup de choses sur notre protocole.

Ce fichier est accompagné d'un programme python qui illustre notre projet. Nous allons étudier ici comment marche ce protocole, ses avantages et ses faiblesses.

Choix :

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.w

Inconvénients :

On perd beaucoup d'espace car notre protocole augmente le rayon des cercles même si 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.

Traitement de données :

Le code python demande de saisir une chaîne de caractères lorsqu'il est lancé.

La première étape est de modifier cette chaîne de caractères.

Nous utilisons la logique afin d'obtenir chaque bit pour le mettre sous sa forme entière dans un tableau.

Une fois que nous avons un tableau d'entiers composé de 0 et 1 qui représentent les bit de l'information.

Le deuxième traitement consiste à ajouter un protocole de contrôle et de correction d'erreur.

Nous avons fait le choix de choisir le code de Hamming pour deux raisons :

-Pour le moment nous ne transmettons pas beaucoup d'informations , le code de Hamming semble être plus efficace, car notre protocole est plus facile à lire lorsque il y'a peu d'informations.

Interface :

Les deux premiers bits correspondent a des bit du protocole et non a l'information.

Sur cette version du protocole on utilise ces deux bit 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'un 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 similaires 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 lis les deux premières bit (pour le moment on utilise seulement deux bit , l'idéal c'est de mettre 16 bit) on récupère la version et en lisant ces deux bit on calcule l'écart entre les deux pour pouvoir lire la suite, on va appeler cet écart Z.

Ensuite a chaque étape on saute de Z et si il y'a un cercle on ajoute un a notre tableau d'information, si 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.

Les autres 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 bit pour la version et pour récupérer la taille de l'écart pour lire.

Efficacité :

L'efficacité dépend de l'information alors on vous laisse la voir sur le code.

Robustesse :

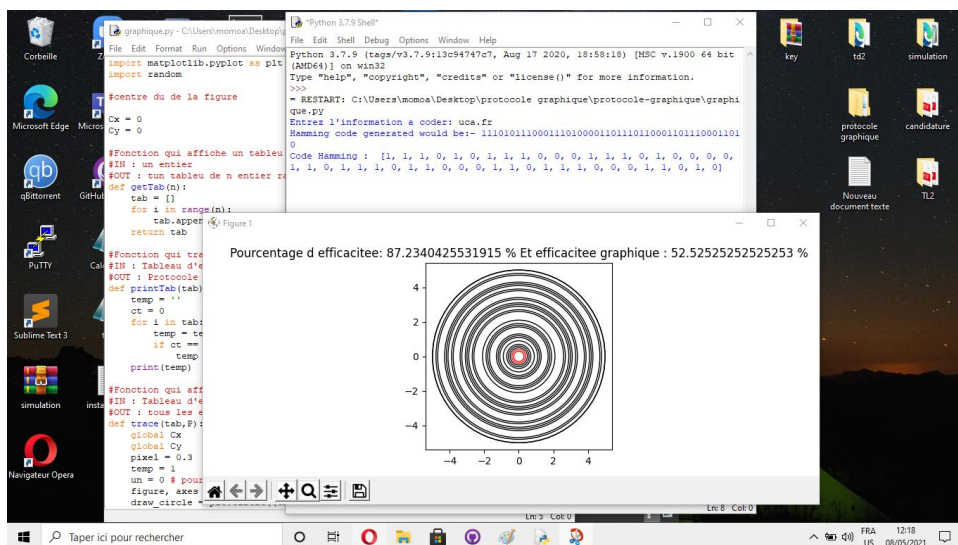
D'une certaine manière ce code est assez robuste car même si on perd une très grande partie de l'image générée par le protocole, avec du café par exemple où elle se déchire nous pourrions quand même lire le protocole même si il ne reste qu'un dixième de l'image.

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 bit devient trop élevé le protocole commence à ressembler à un gros cercle noir et sont illisibles sauf si on zoom.



Orientation :

Avec ce protocole nous pouvons lire de tout les côté car c'est un cercle.

En regardant le protocole de n'importe quel angle c'est possible de récupérer l'information.

Il suffit d'un seul anglais ou on voit les cercles pour récupérer l'information.

Logo :

Nous pouvons mettre facilement un logo sur notre protocole car celui-ci a besoin d'un seul angle pour lire l'image.

