

## Semaine 12 : Disques et (surtout) réseaux [Solutions]

### 1 Synthèse musicale

#### 1.1 Besoin de structure

*Retrouvez l'amplitude du signal sonore au temps 0.01 s dans l'information suivante stockée sur le disque :*

C'est évidemment **impossible**!...

Il faut la **structure** de ces informations pour les décoder. C'est justement le but du reste de l'exercice.

#### 1.2 Echantillonnage du signal

*Ce signal peut-il être reconstitué exactement à partir des échantillons du CD ?*

**Oui sans souci** : la bande passante est de  $5 \times 440 = 2200$  Hz qui est bien inférieur à  $44100/2 = 22'050$  Hz.

*Sinon, que devrait-on faire ?*

Sinon on devrait **filtrer les hautes fréquences**, celles supérieures à 22'050 Hz.

#### 1.3 Codage du signal

*Quelle précision a-t-on alors relativement à l'amplitude maximale ?*

Sur 16 bits on peut stocker 65'536 valeurs, on a donc une précision relative de... environ **1/32768**.

Et oui ! Il ne faut pas oublier que les amplitudes peuvent être « négatives » (au sens : « *inférieures à la moitié de l'écart entre l'amplitude maximale et l'amplitude minimale* »).

Plus précisément, si le signal varie entre l'amplitude minimale  $a_{\min}$  et l'amplitude maximale  $a_{\max}$ , l'erreur absolue sera alors de  $\frac{a_{\max} - a_{\min}}{2^{16} - 1}$  (et donc l'erreur relative par rapport à  $a_{\max}$  de  $\frac{a_{\max} - a_{\min}}{(2^{16} - 1) a_{\max}}$ ). On peut par exemple choisir de coder  $a_{\min}$  par 0000000000000000, et  $a_{\max}$  par 1111111111111111.

Si l'on considère par exemple le cas symétrique où  $a_{\min} = -a_{\max}$ , on obtient alors une précision relative de  $\frac{2}{2^{16} - 1} \simeq \frac{1}{2^{15}}$  comme indiqué plus haut.

### Notes :

- si par « amplitude maximale » vous avez compris « amplitude totale », c.-à-d.  $a_{\max} - a_{\min}$ , alors la réponse est  $\frac{1}{2^{16} - 1}$  (puisque vous avez déjà intégré le facteur 2 dans la définition d'« amplitude maximale ») ;
- un signal (au sens du cours : « somme de sinusoïdes ») n'est pas nécessairement symétrique (par rapport à 0). Par exemple pour  $\sin(t + \frac{\pi}{2}) + \sin(2t + \frac{\pi}{2})$ ,  $a_{\max} = 2$  et  $a_{\min} = -1.125$ .

*Est-ce possible de comprimer à 50% ?*

On parle *bien sûr* ici de compression sans perte et sans préfixe (pour décoder), donc le th. de Shannon s'applique : on **ne peut pas** compresser en dessous de 10.5 bit par valeur, soit une compression de  $1 - 10.5/16 = 34\%$ .

*Quel taux de compression obtient-on avec Huffman (ordre de grandeur) ?*

Toujours par le th. de Shannon :  $H \leq L \leq H + 1$  donc on a entre 10.5 et 11.5 bit par valeur en moyenne, soit entre 28% et 34% de compression. Disons **30%**.

## 1.4 Stockage sur disque : contenu du fichier

*Combien de blocs (environ) utilise notre fichier ?*

1 seconde de musique donne 44'100 échantillons, soit environ 485'100 bits (longueur moyenne précédente), soit 60'637 octets, soit environ 60 Kio, c'est-à-dire 15 blocs.

Pour 2 s, il faut donc **30 blocs**.

*Nous avons un disque de 512 Gio : combien de blocs contient-il ?*

$$512 \text{ Gio} = 2^9 \cdot 2^{10} \cdot 2^{10} \text{ Kio} = 2^{29} \text{ Kio} = 2^{27} \text{ blocs}$$

*Combien de bits sont nécessaires pour l'adresse des blocs ?*

Il faut donc (au moins) **27 bits** (nombre minimum de bits pour représenter  $2^{27}$  valeurs).

*Quelle est l'adresse du bloc correspondant à l'amplitude du signal sonore au temps 0.01 s ?*

0.01 s correspond au 441<sup>e</sup> échantillon, c'est-à-dire à 4851 bits, soient 606 octets : c'est donc dans le 1<sup>er</sup> bloc.

On doit donc décoder la 1<sup>re</sup> adresse, c'est-à-dire la 2<sup>e</sup> ligne, soit 000000000000000000000000101011, qui représente **43** en binaire.

## 1.5 Stockage sur disque : nom absolu du fichier

*Combien de blocs utilise alors le répertoire **Musique** ?*

La taille de la table de **Musique** est de  $(1 + 30) \cdot 32$  bits (1 pour la taille et 30 adresses (pour les 30 blocs de nos 2 secondes de musique)), soient 992 bits. **1 bloc** suffit donc.

(La table de **Musique** ne contient que les adresses des blocs qui contiennent les données et non les blocs eux-mêmes.)

*Quel est le début du fichier représentant le répertoire **musche** ?*

Il indique la taille (1 bloc) et l'adresse du premier bloc (qui ici est aussi le seul bloc) :

```
000000000000000000000000000000001 <<- taille = 1
00000000000000000000000000000000101010 <<- 42
```

## 1.6 Décodage du « message » de départ

*Quelle est la valeur de l'amplitude de notre signal au temps  $2/44100$  s ?*

On décode : en 42 on trouve la table de notre fichier de musique. Le temps  $2/44100$  s correspond au 3<sup>e</sup> échantillon, qui est donc dans le 1<sup>er</sup> bloc, dont l'adresse est 43 (décodée plus haut).

On doit donc décoder le bloc 43, qui se décompose comme suit en utilisant le code de Huffman : 11001100110, 10101010, 0001110001

ce qui correspond donc aux amplitudes (relatives, codées sur 16 bits) : 0, 5929 et 11056.

Celle qui nous intéresse est 11056, qui nous donne au final une amplitude (physique) de  $4 \cdot 11056 / 32768 = 1.35$ .

## 2 Couches Internet

Dans l'Internet, « IP » est un protocole de la couche niveau 3 (réseau) utile pour une fonction de routage.

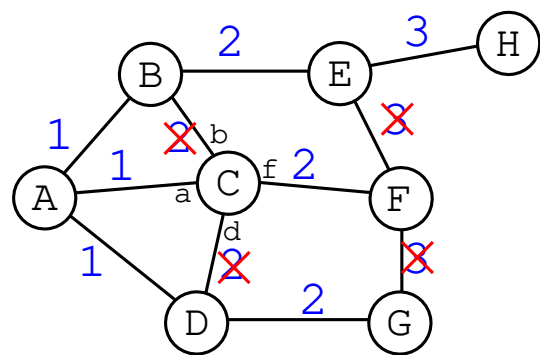
Dans l'Internet, « TCP » est un protocole de la couche niveau 4 (transport) utile pour une fonction de connexion entre machines/services.

Dans l'Internet, « HTTP » est un protocole de la couche niveau 5 (application) utile pour une fonction de communication de contenus formatés (pages Web).

Dans l'Internet, « DNS » est un protocole de la couche niveau 5 (application) utile pour une fonction de

résolution de noms en adresses ( $\simeq$  bottin).

### 3 Routage IP

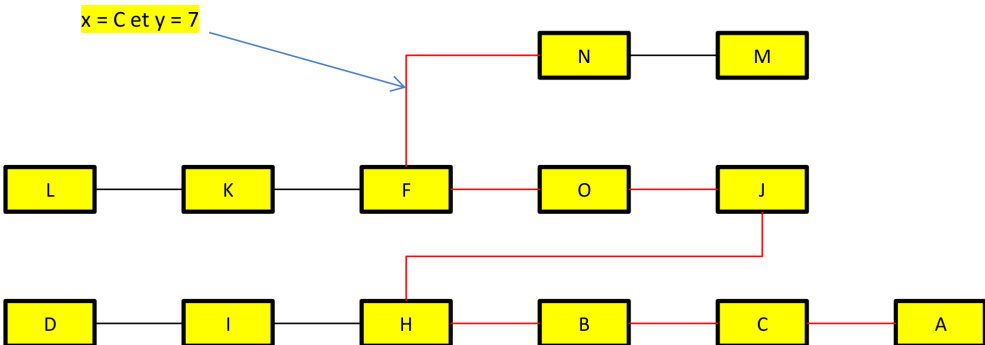


Par exemple (solution non unique) :

table de C

A	1	a
B	1	b
D	1	d
E	2	b
F	1	f
G	2	d
H	3	f

### 4 Routage encore



## 5 Routage toujours

a)

C			D			E		
dest.	dir.	dist.	dest.	dir.	dist.	dest.	dir.	dist.
A	A	1	A	C	2	A	C	2
K	E	4	K	E	4	K	G	3

F			G			H			I		
dest.	dir.	dist.	dest.	dir.	dist.	dest.	dir.	dist.	dest.	dir.	dist.
A	E	3	A	E	3	A	G	4	A	G	4
K	G	3	K	I	2	K	I	2	K	K	1

b)

C			D			E		
dest.	dir.	dist.	dest.	dir.	dist.	dest.	dir.	dist.
K	E	5	K	F	4	K	F	4

F			G			H			I		
dest.	dir.	dist.	dest.	dir.	dist.	dest.	dir.	dist.	dest.	dir.	dist.
			A	F	4	A	F	4	A	G	5

c)

C			D			E		
dest.	dir.	dist.	dest.	dir.	dist.	dest.	dir.	dist.
K	D	5	K	F	4			

F			G			H			I		
dest.	dir.	dist.	dest.	dir.	dist.	dest.	dir.	dist.	dest.	dir.	dist.
A	D	3	A	F	4	A	F	4	A	G	5

## 6 Encore un peu plus de routage ?

oui et la longueur est 4 :

