
Introduction aux Télécommunications

PROJET

Etude et implantation de la couche physique d'une chaîne de transmission par satellite, répondant au standard DVB-S, pour transmettre une image compressée au format jpg.

Nathalie Thomas
Département Sciences du Numérique, Première année
2018 - 2019

Transmission via la chaine DVB-S d'une image compressée

Objectifs - Organisation

→ Objectifs :

→ Objectif 1 :

- Simuler la couche physique DVB-S : intégration du codage canal à la chaine QPSK déjà réalisée en TP (canal AWGN).
- Visualiser l'impact du codage canal sur les performances de la chaine de transmission en comparant le TEB sans codage et le TEB avec différentes configurations de codage canal (un seul code, codes concaténés, codes concaténés + entrelaceur).
- Etre capable d'expliquer le principe de fonctionnement et l'intérêt du codage canal dans la chaine de transmission.

→ Objectif 2 :

- Programmer la compression/décompression jpeg (codage source) d'une image noir et blanc.
- Etre capable d'expliquer le principe de fonctionnement et l'intérêt du codage source dans une chaine de transmission.

→ Objectif 3 :

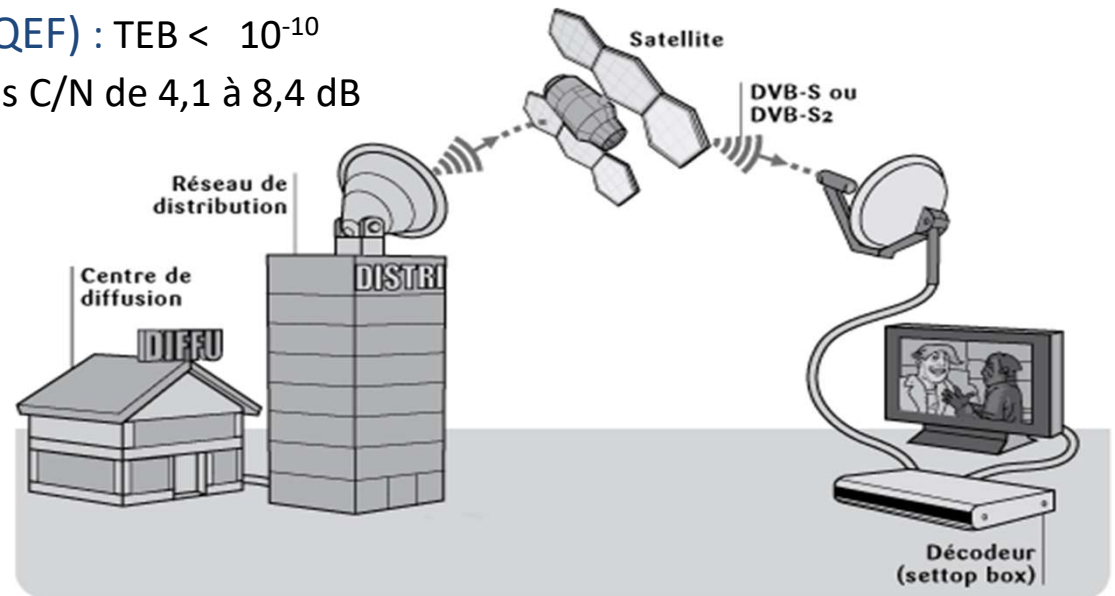
- Réaliser la transmission d'une image compressée en jpeg en utilisant la modulation QPSK sur canal AWGN afin de visualiser l'impact du bruit dans ce contexte.
- Intégrer le codage canal et évaluer son apport dans le même contexte que précédemment.

→ Organisation/Déroulement du projet :

- Travail par groupe de 4,
- 6 séances prévues à l'emploi du temps,
- Encadrement « allégé » : un enseignant pour deux groupes.
- Evaluation : suivi, compte-rendu + codes à fournir pour un groupe de 4.

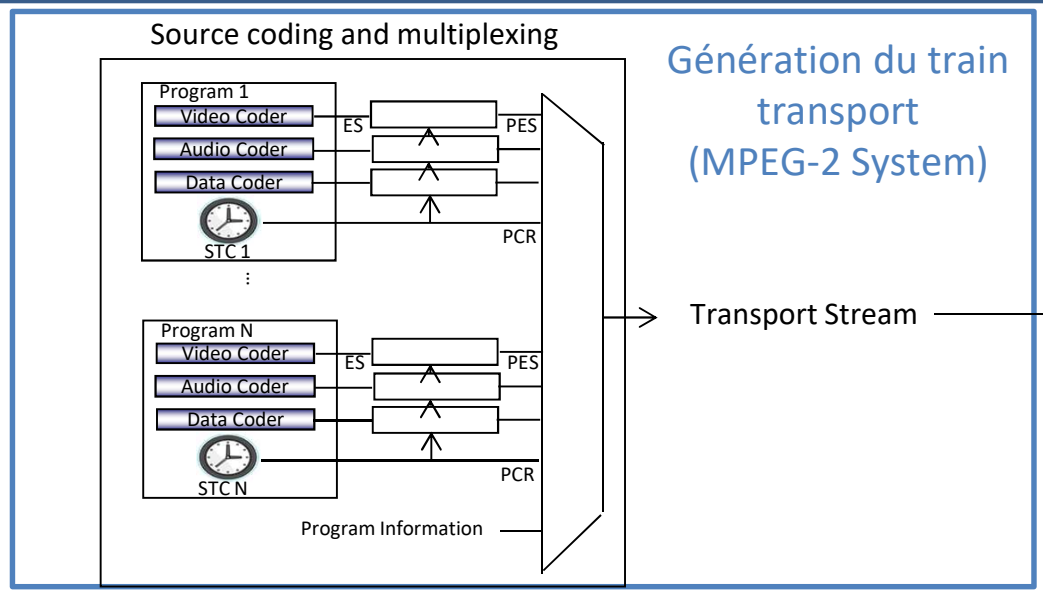
Transmission de type DVB-S (Digital Video Broadcasting by Satellite)

- Pour diffuser du contenu multimédia via le satellite
- Transmission fixe, ligne de vue directe => canal stationnaire de type AWGN
 - Propagation dans les bandes Ku (10.7-12.45 GHz) et C (4-6 GHz, zones tropicales)
 - Fortes atténuations : absorption, diffusion due aux gaz et aux molécules d'eau dans l'atmosphère, aux nuages, à la pluie, grande distance entre antennes
 - Bruit (supposé additif, blanc et Gaussien) :
 - Bruit externe = signaux reçus en plus du signal utile :
 - Sources naturelles : atmosphère (orage, foudre), terre, ciel (soleil, voie lactée)
 - Sources artificielles : activité humaine, interférences avec d'autres utilisateurs
 - Bruit interne = dispositifs électroniques du récepteur : amplificateurs, antennes, etc.
 - Non linéarités.
- Bande passante : 33 MHz (transpondeurs classiques)
- Transmission « Quasi Error Free » (QEF) : $TEB < 10^{-10}$
- Débits : 23,754 à 41,570 Mbps pour des C/N de 4,1 à 8,4 dB



Satellite Digital Video Broadcasting : DVB-S

Couche physique



Une transmission TV numérique doit être « Quasi Error Free » (QEF) : $TEB < 10^{-10}$

Couche Physique

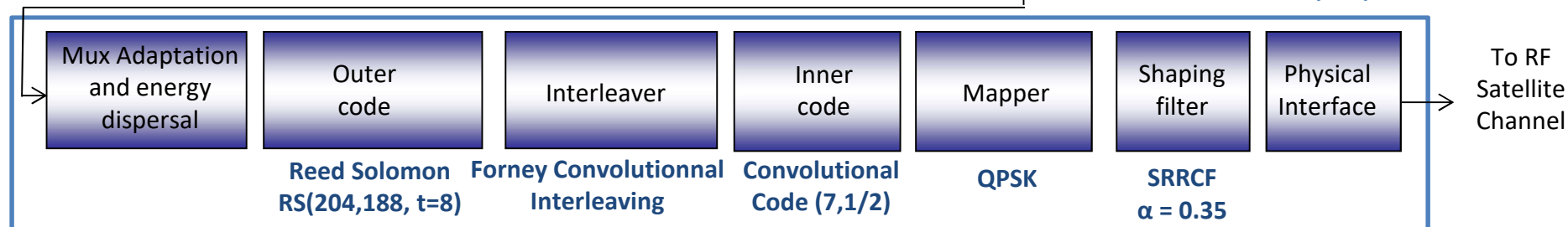


Table D.1: Example of System performance over 33 MHz transponder

Débits binaires

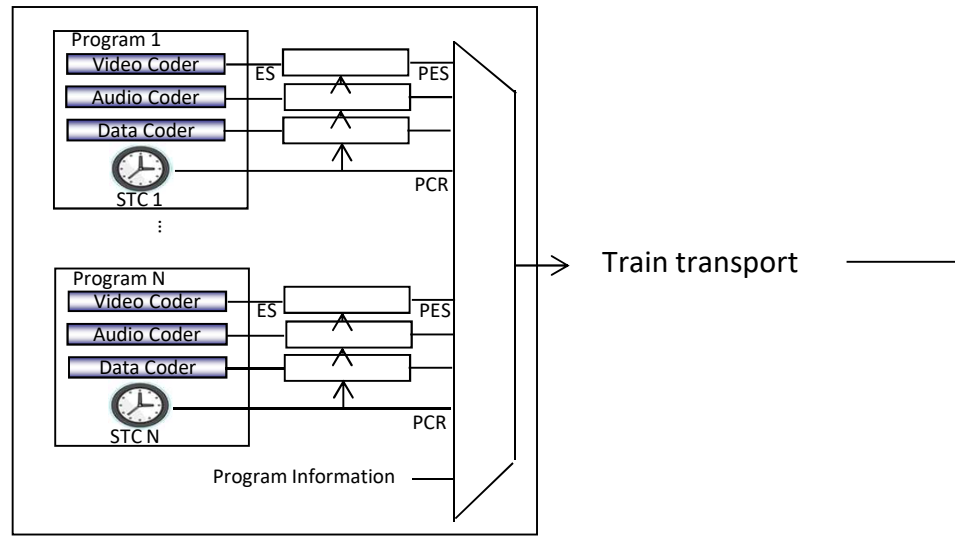
Bit Rate R_u (after MUX) [Mbit/s]	Bit Rate R'_u (after RS) [Mbit/s]	Symbol Rate [Mbaud]	Convolut. Inner Code Rate	RS Outer Code Rate	C/N (33 MHz) [dB]
23,754	25,776	25,776	1/2	188/204	4,1
31,672	34,368	25,776	2/3	188/204	5,8
35,631	38,664	25,776	3/4	188/204	6,8
39,590	42,960	25,776	5/6	188/204	7,8
41,570	45,108	25,776	7/8	188/204	8,4

SNRs nécessaires

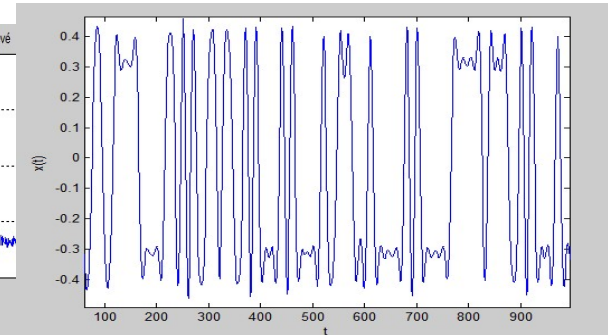
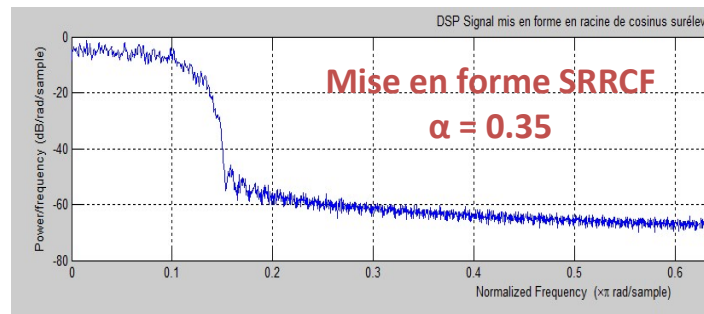
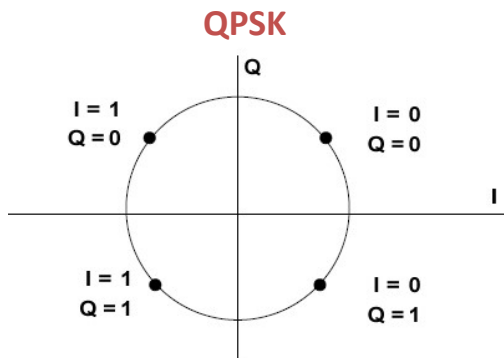
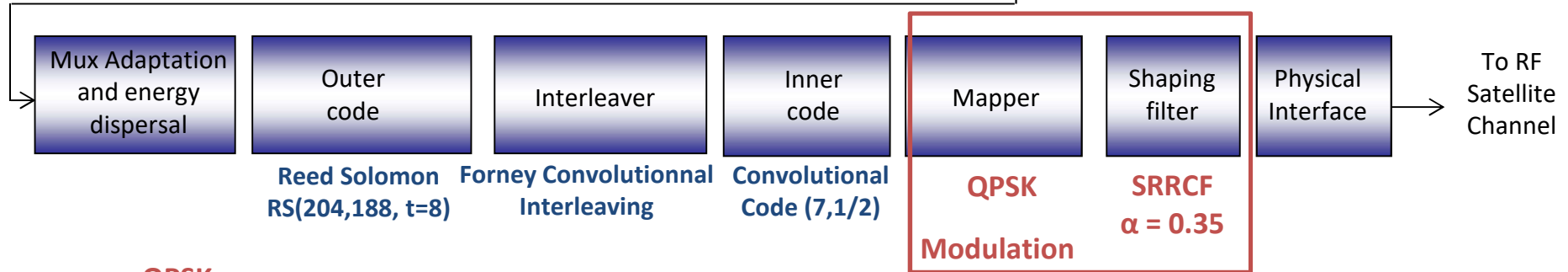
Satellite Digital Video Broadcasting : DVB-S

Modulation

Codage source et multiplexage

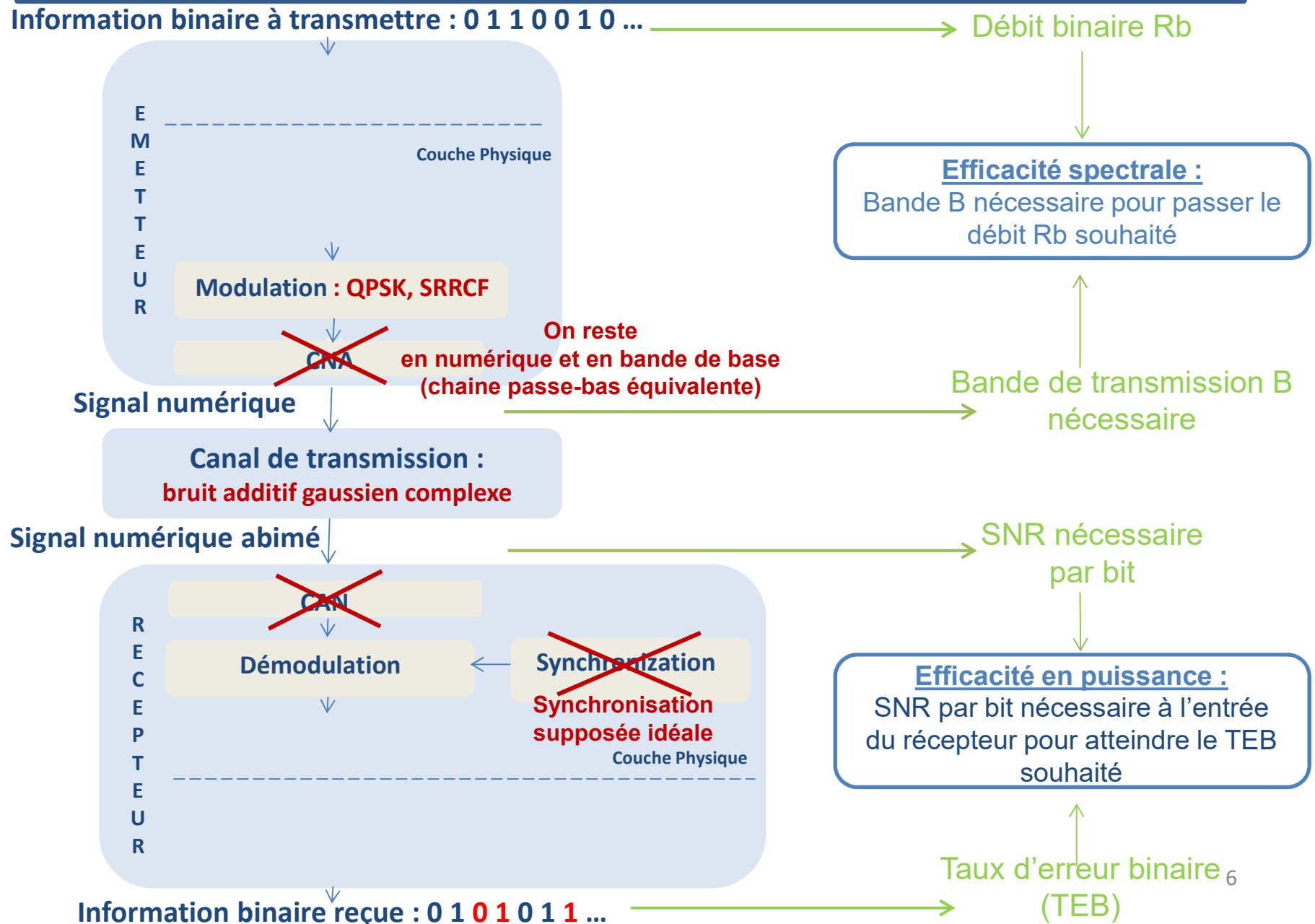


Canal AWGN
Avec présence de non linéarités



Satellite Digital Video Broadcasting : DVB-S

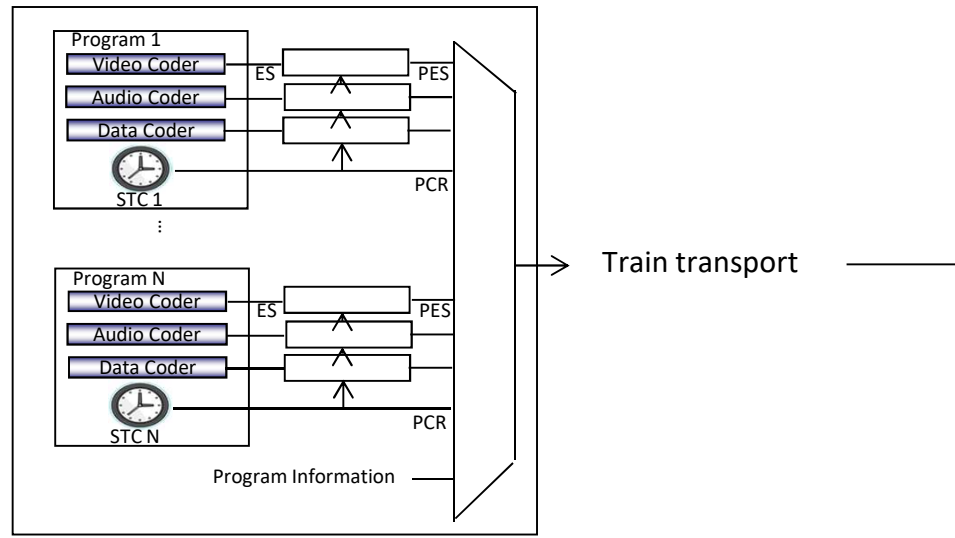
Modulation/Démodulation : FAIT EN TP



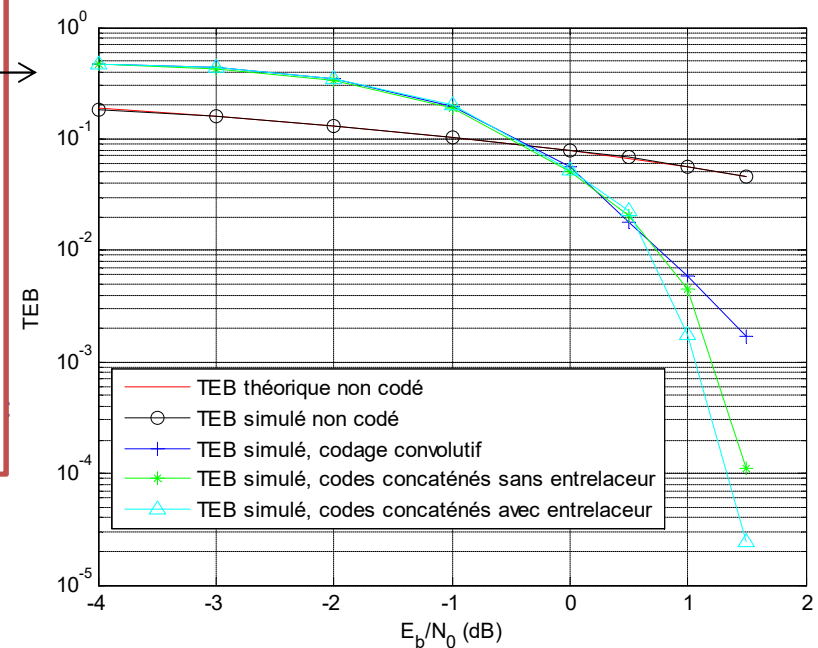
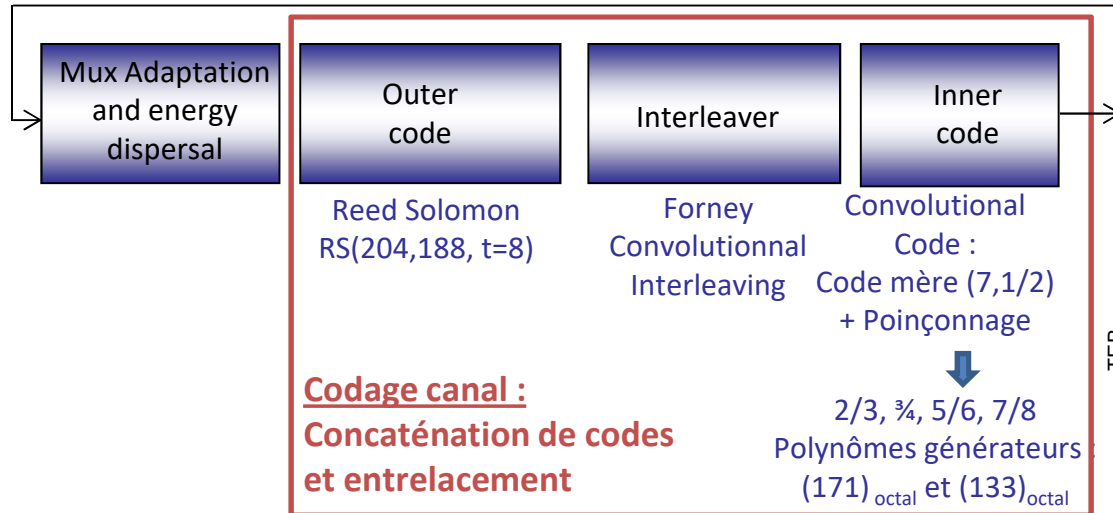
Satellite Digital Video Broadcasting : DVB-S

Codage canal

Codage source et multiplexage



Une transmission TV numérique doit être « Quasi Error Free » (QEF) : $TEB < 10^{-10}$



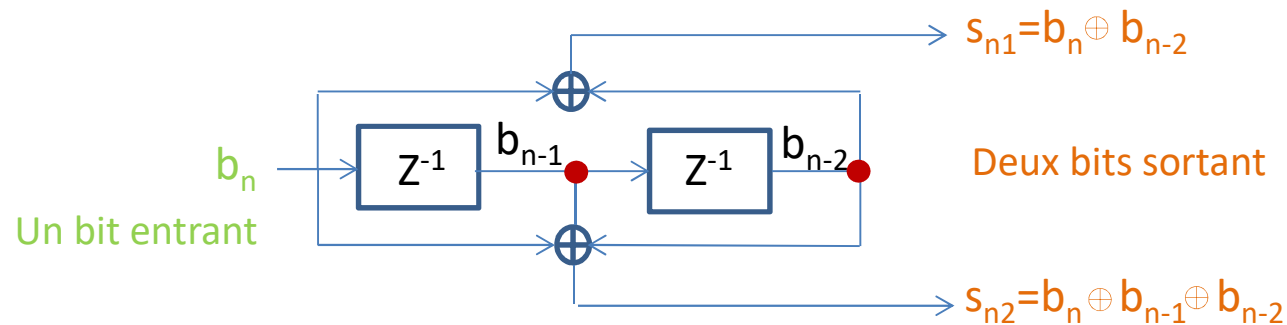
Satellite Digital Video Broadcasting : DVB-S

Codage canal : code convolutif

Exemple : code convolutif (3,1/2)

Longueur de contrainte

Rendement



● états du codeur

Initialisation : $b_{n-1} = b_{n-2} = 0$

k bits entrants/n bits sortant \Rightarrow rendement = $k/n = 1/2$

Polynômes générateurs :

$g_1 = [1 \ 0 \ 1]$ (pour générer s_{n1}) / $g_2 = [1 \ 1 \ 1]$ (pour générer s_{n2}) ou 5/7 en octal

$$s_{n1} = 1 \times b_n \oplus 0 \times b_{n-1} \oplus 1 \times b_{n-2}$$

$$s_{n2} = 1 \times b_n \oplus 1 \times b_{n-1} \oplus 1 \times b_{n-2}$$

Mémoire du code :

Nombre de bascule dans le registre à décalage = 2

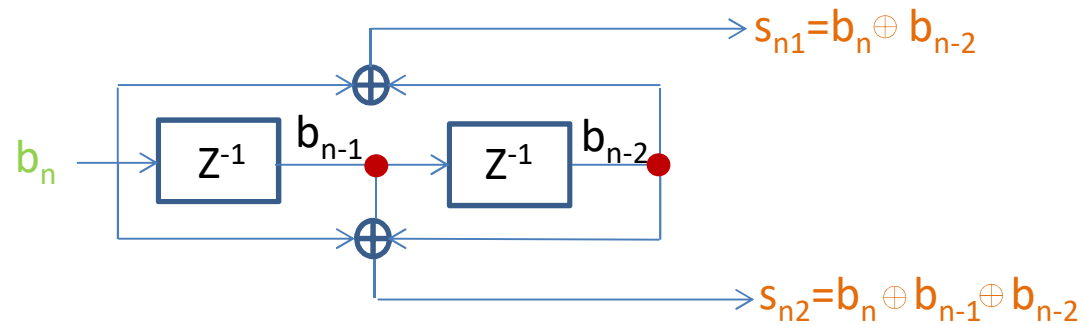
Longueur de contrainte :

$k(\text{mémoire du code} + 1) = 3$

Satellite Digital Video Broadcasting : DVB-S

Codage canal : code convolutif

Exemple : code convolutif (3,1/2)



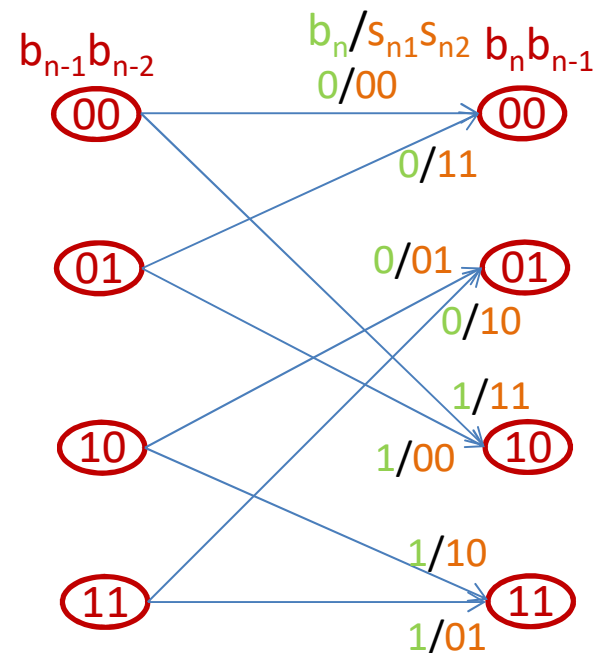
● états du codeur

Initialisation : $b_{n-1} = b_{n-2} = 0$

(poly2trellis.m)

b_n	Etat présent $b_{n-1}b_{n-2}$	Etat suivant b_nb_{n-1}	Sorties $s_{n1}s_{n2}$
0	00	00	00
1	00	10	11
0	01	00	11
1	01	10	00
0	10	01	01
1	10	11	10
0	11	01	10
1	11	11	01

Trellis associé :



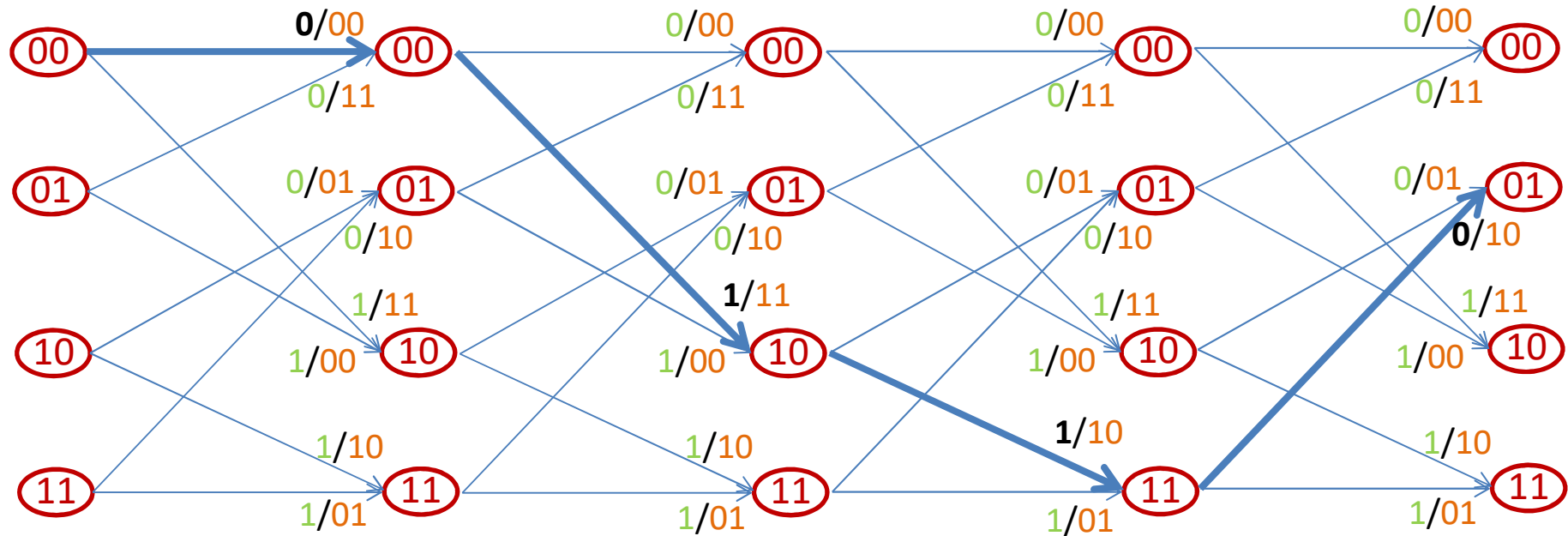
Satellite Digital Video Broadcasting : DVB-S

Codage canal : code convolutif

Exemple : code convolutif (3,1/2)

Une séquence d'entrée = un chemin dans le treillis

Exemple : 0 1 1 0



Satellite Digital Video Broadcasting : DVB-S

Codage canal : code convolutif

Exemple : code convolutif (3,1/2)

Décodage par maximum de vraisemblance - Algorithme de Viterbi

- Décodage par maximum de vraisemblance

Etant donné une séquence reçue de longueur N : $Y = [y_1, y_2, \dots, y_N]$, trouver la séquence émise la plus vraisemblable parmi toutes les séquences possibles de longueur N :

$$\hat{X} = \arg \max_X P(Y/X)$$

<=> trouver la séquence la plus proche parmi toutes les séquences possibles au sens d'une certaine distance.

Cas d'un canal à bruit additif Gaussien :

$$y_i = x_i + w_i, \quad w_i \text{ échantillon de bruit Gaussien } \sim \mathcal{N}(0, \sigma_w^2)$$

$$P(Y|X) = \prod_{i=0}^N p(y_i/x_i) \propto \prod_{i=0}^N e^{-\frac{(y_i - x_i)^2}{2\sigma_w^2}} \propto e^{-\sum_{i=0}^N \frac{(y_i - x_i)^2}{2\sigma_w^2}}$$

En passant au log : $\hat{X} = \arg \min_X \sum_{i=0}^N (y_i - x_i)^2$

Minimisation de la distance euclidienne

- Nécessité d'un algorithme

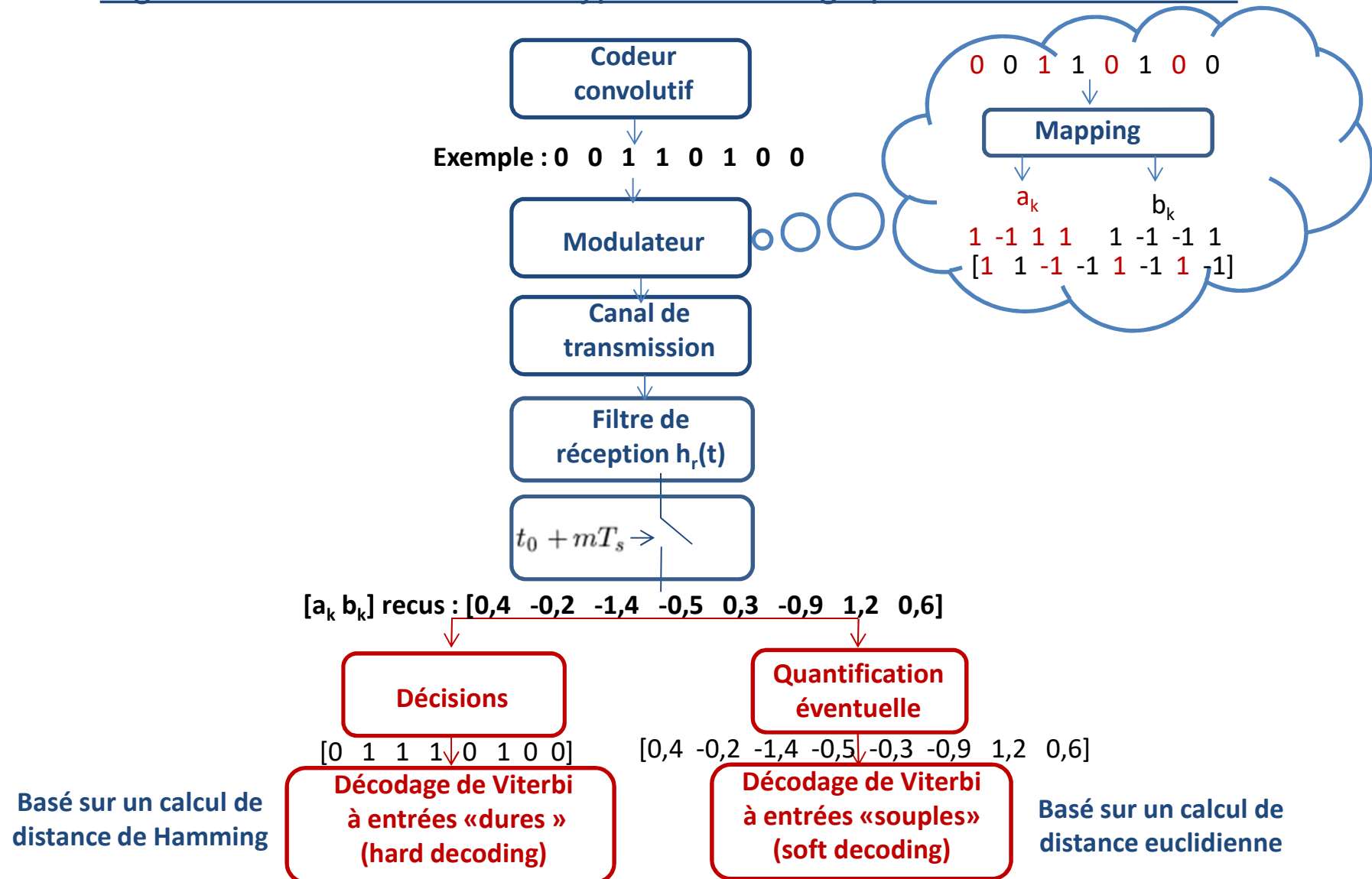
Pour une séquence de longueur N il y a (nombre d'états $\times 2^N$) chemins possible dans le treillis, c'est-à-dire séquences possibles à tester. Exemple N=20, 4 états : 4194304 séquences possibles !!

➡ **algorithme de Viterbi**

Satellite Digital Video Broadcasting : DVB-S

Codage canal : code convolutif

Algorithme de Viterbi - Deux type de décodage possibles : hard et soft



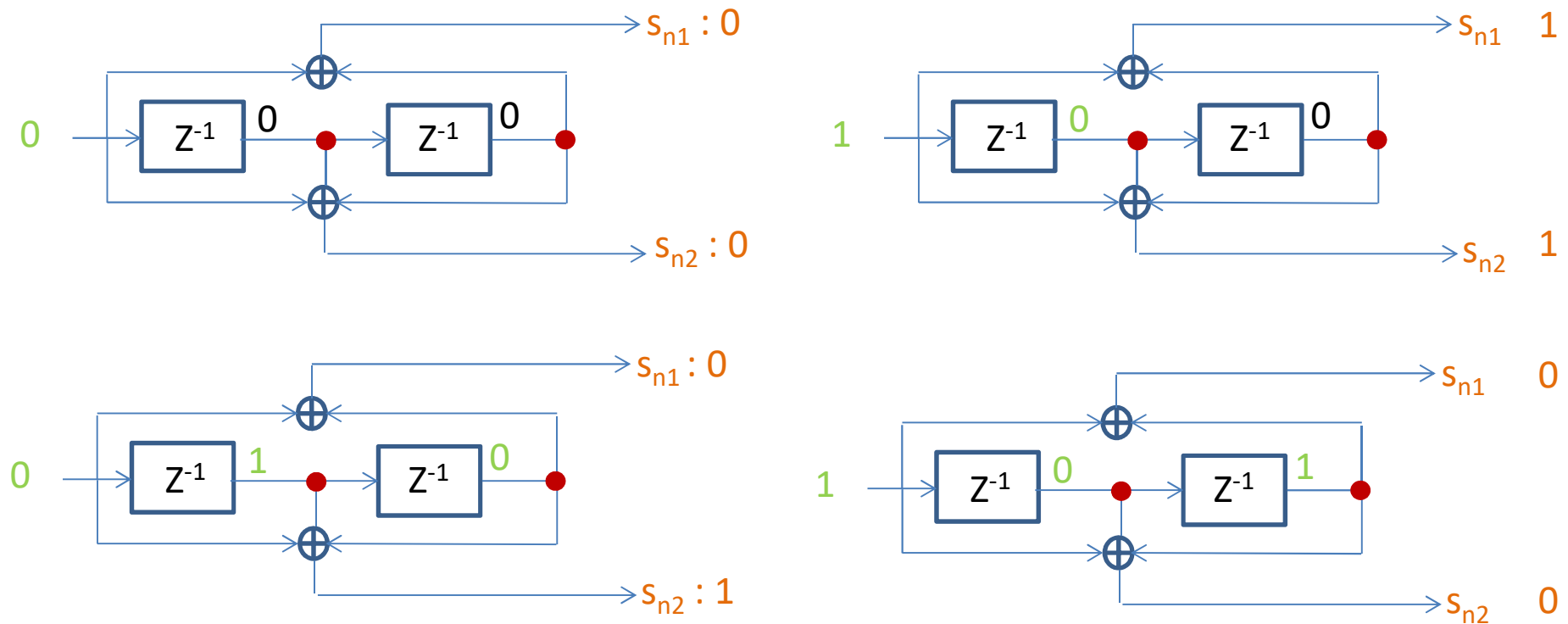
Satellite Digital Video Broadcasting : DVB-S

Codage canal : code convolutif

Exemple : code convolutif (3,1/2) / décodage de Viterbi

Séquence de bits à transmettre : 0 1 0 1

(convenc.m)



Séquence en sortie du codeur : 0 0 1 1 0 1 0 0

[a_k b_k] en sortie du mapping : [+1 +1 -1 -1 +1 -1 +1 +1]

Satellite Digital Video Broadcasting : DVB-S

Codage canal : code convolutif

Exemple : code convolutif (3,1/2)

Exemple de décodage par algorithme de Viterbi

(vitdec.m)

$[a_k \ b_k]$ en sortie du mapping : [+1 +1 -1 -1 +1 -1 +1 +1]

Séquence $[a_k \ b_k]$ en sortie de l'échantillonneur :

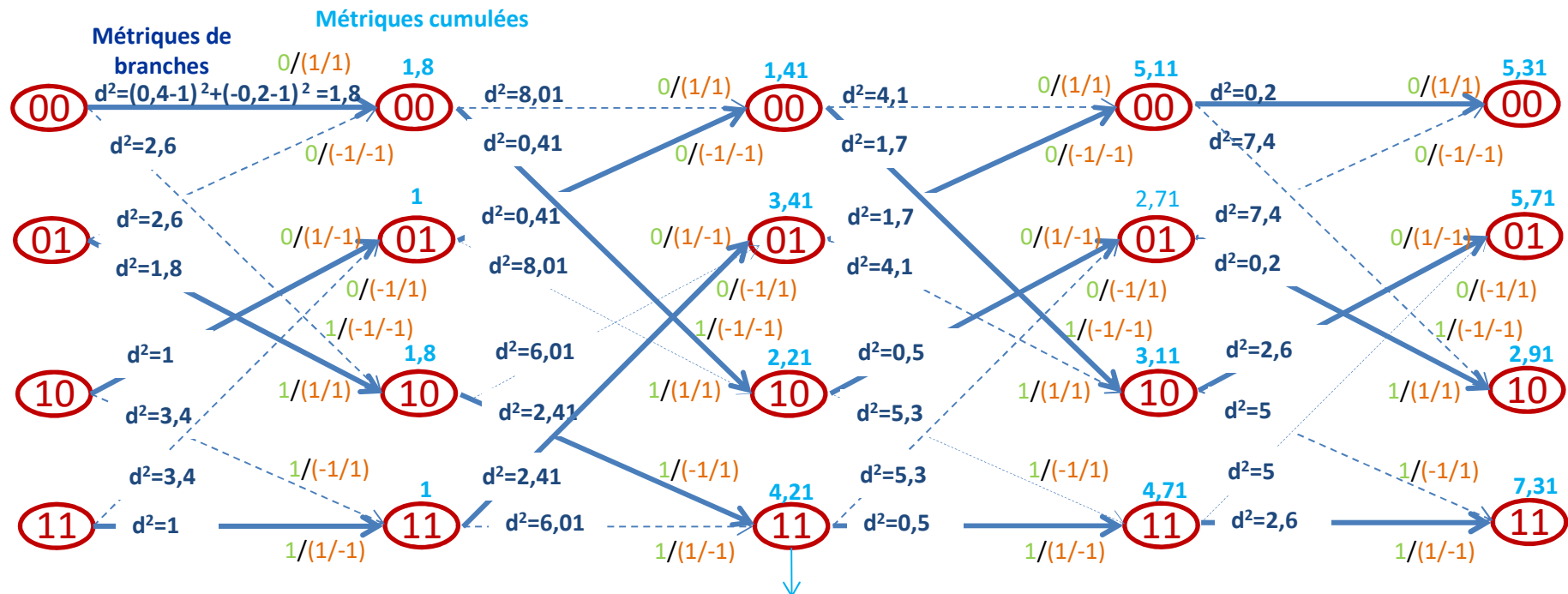
0,4/-0,2

-1,4/-0,5

0,3/-0,9

1,2/0,6

Algorithme de Viterbi :



Satellite Digital Video Broadcasting : DVB-S

Codage canal : code convolutif

Exemple : code convolutif (3,1/2)

Exemple de décodage par algorithme de Viterbi

a_k b_k en sortie du mapping : +1 +1 -1 -1 +1 -1 +1 +1

Séquence reçue en sortie de l'échantillonneur :

0,4/-0,2

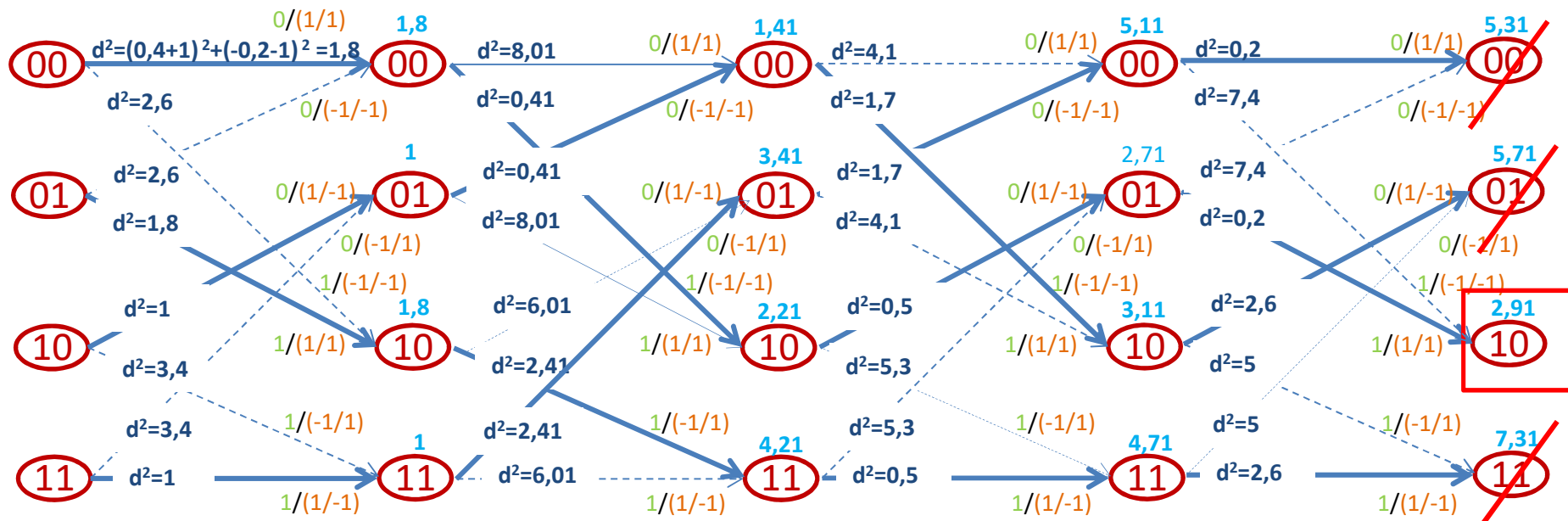
-1,4/-0,5

0,3/-0,9

1,2/0,6

Algorithme de Viterbi :

Choix de la métrique
Cumulée la plus faible



Satellite Digital Video Broadcasting : DVB-S

Codage canal : code convolutif

Exemple : code convolutif (3,1/2)

Exemple de décodage par algorithme de Viterbi

a_k b_k en sortie du mapping : +1 +1 -1 -1 +1 -1 +1 +1

Séquence reçue en sortie de l'échantillonneur :

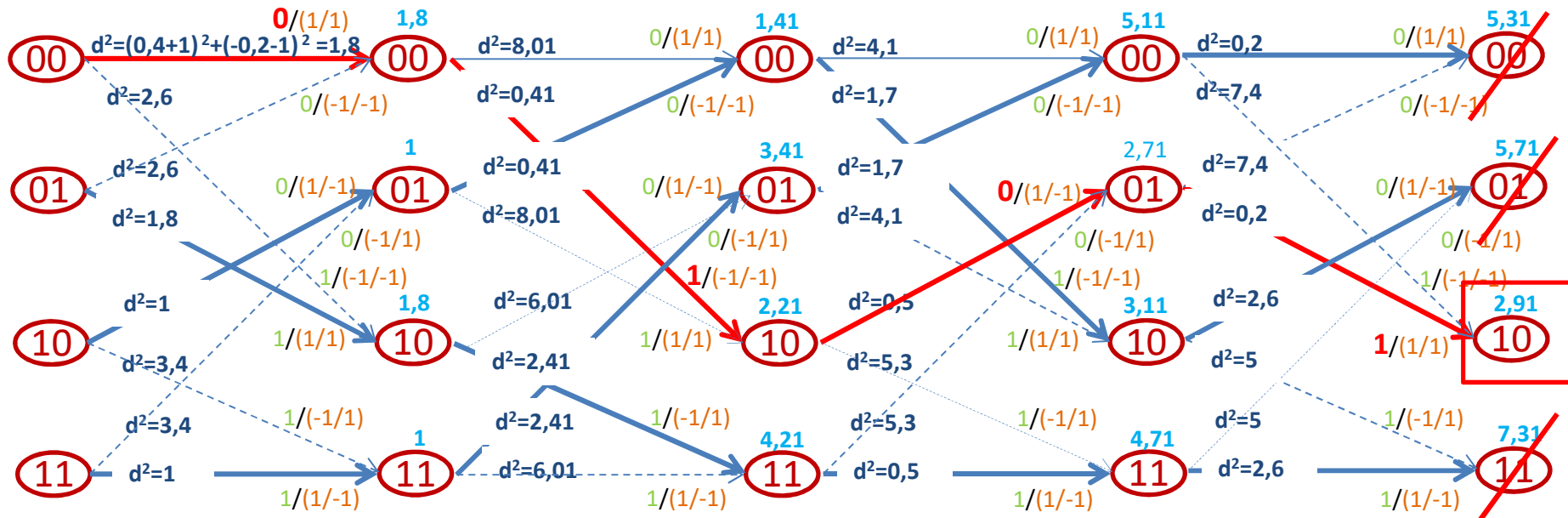
0,4/-0,2

-1,4/-0,5

0,3/-0,9

1,2/0,6

Algorithme de Viterbi :



On remonte le chemin => séquence émise 0 1 0 1

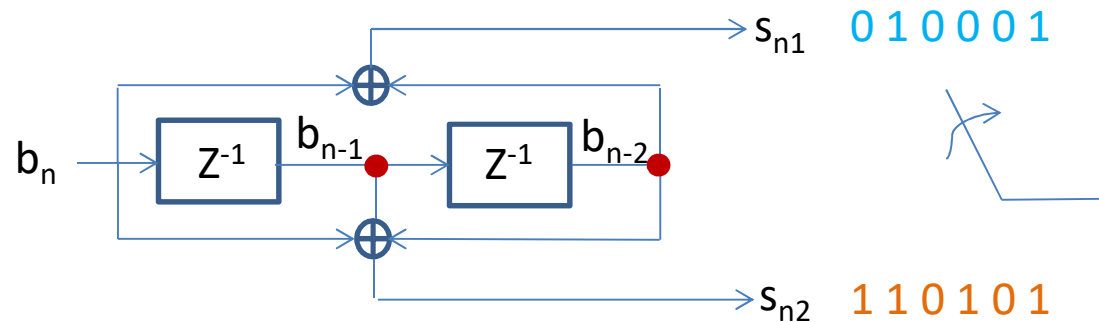
Distance de retour (traceback) pour la décision : environ 5 fois la longueur de contrainte du code

Temps de calcul pour une séquence de longueur $N = N_x(2 \times \text{nombre d'états}) < (2^N \times \text{nombre d'états})$

Satellite Digital Video Broadcasting : DVB-S

Codage canal : code convolutif

Exemple :



● états du codeur
Initialisation : $b_{n-1} = b_{n-2} = 0$

Poinçonnage

Rendement $\frac{1}{2}$:

(pour 1 bit entrant : 2 bits sortant)

0 1 1 1 0 0 0 1 0 0 1 1

Rendement $\frac{2}{3}$:

(pour 2 bits entrant : 3 bits sortant)

0 1 1 1 0 0 0 1 0 0 1 1

Rendement $\frac{3}{4}$:

(pour 3 bits entrant : 2 bits sortant)

0 1 1 1 0 0 0 1 0 0 1 1

k bits entrants/n bits sortant => **rendement** = k/n

$$R_{b_u} : \text{Débit utile} \rightarrow R_{b_u} = \frac{k}{n} R_{b_{tx}} \leftarrow R_{b_{tx}} : \text{Débit transmis}$$

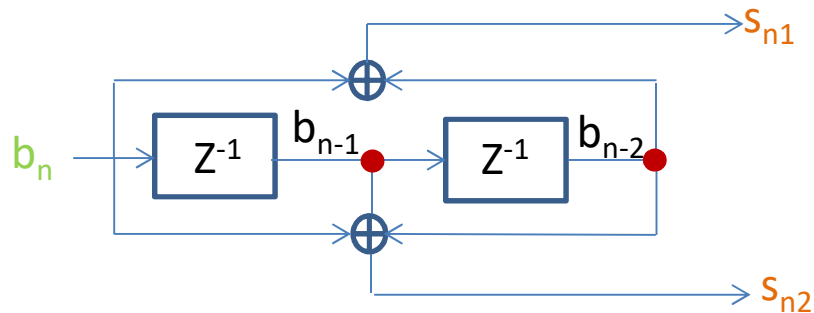
Original code			Code rates									
			1/2		2/3		3/4		5/6		7/8	
K	G1 (X)	G2 (Y)	P	dfree	P	dfree	P	dfree	P	dfree	P	dfree
7	171 _{OCT}	133 _{OCT}	X: 1 Y: 1 I=X1 Q=Y1	10	X: 1 0 Y: 1 1 I=X1 Y2 Y3 Q=Y1 X3 Y4	6	X: 1 0 1 Y: 1 1 0 I=X1 Y2 Q=Y1 X3	5	X: 1 0 1 0 1 Y: 1 1 0 1 0 I=X1 Y2 Y4 Q=Y1 X3 X5	4	X: 1 0 0 1 0 1 Y: 1 1 1 1 0 1 0 I=X1 Y2 Y4 Y6 Q=Y1 Y3 X5 X7	3

NOTE: 1 = transmitted bit
0 = non transmitted bit

Satellite Digital Video Broadcasting : DVB-S

Codage canal : code convolutif

Exemple : code convolutif (3,1/2)



● états du codeur

Initialisation : $b_{n-1} = b_{n-2} = 0$

1 bits entrants/2 bits sortant => **rendement** = 1/2

Polynômes générateurs :

$g_1 = [1 \ 0 \ 1] = 5_{\text{octal}} (s_{n1})$ / $g_2 = [1 \ 1 \ 1] = 7_{\text{octal}} (s_{n2})$

Mémoire du code :

Nombre de bascule dans le registre à décalage = 2

Longueur de contrainte :

$1 \times (\text{mémoire du code} + 1) = 3$

Exemple de code matlab (avec poinçonnage) :

```
%PARAMETRES DU CODE
% Longueur de contrainte
K=3;
% Premier polynôme générateur
g1=5;
% Deuxième polynôme générateur
g2=7;
% Génération du treillis du code convolutif
trellis=poly2trellis(K,[g1 g2]);
%Matrice de poinçonnage pour obtenir un rendement 2/3
P=[1 1 0 1];

%Génération d'un flux de Nb bits
Nb=100;
bits=randi(2,1,Nb)-1;

%CODAGE CONVOLUTIF
bits_codes=convenc(bits,trellis,P);

%mapping de type BPSK
symboles=1-2*bits_codes;

%DECODAGE DE VITERBI
bits_decodes=vitdec(symboles,trellis,5*K,'trunc','unquant',P);
```

Satellite Digital Video Broadcasting : DVB-S

Codage canal : code de Reed Solomon

(comm.RSEncoder.m
comm.RSDecoder.m
step.m)

Code RS(n,k,t)

n : nombre de symboles du mot de code

k : nombre de symboles d'information

t : capacité de correction du code en nombre de symboles = $\frac{d_{\min}-1}{2}$

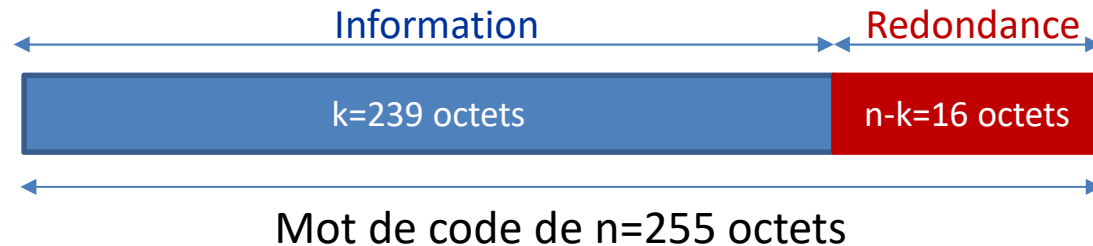
RS(255,239,t=8)

Un symbole = un octet

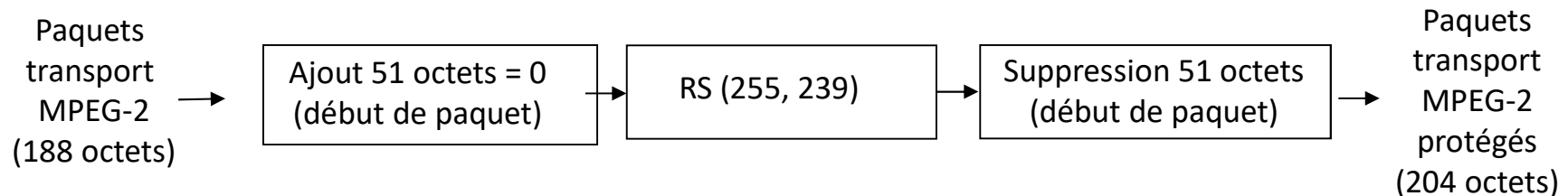
$n=2^8-1=255$ symboles (octets) dans un mot de code

$k=n-2t=239$ symboles d'information pour une capacité de correction de 8 octets par mot de code

Capacité de correction du code de $t=8$ symboles



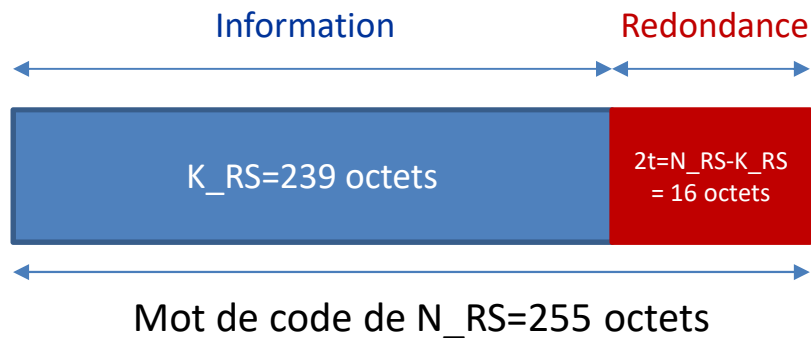
DVB-S : code raccourci RS(204,188,t=8)



Satellite Digital Video Broadcasting : DVB-S

Codage canal : code de Reed Solomon

Exemple : RS(255,239,t=8)



Un symbole = un octet

$n=2^8-1=255$ symboles dans un mot de code

$k=n-2t=239$ symboles d'information

Capacité de correction du code de $t=8$ symboles

Exemple de code matlab :

```
%PARAMETRES DU CODE
% nombre de bits par symbole
Nb_bits_symb = 8;
% capacite de correction du code
t = 8;
% nombre de symboles du mot de code RS (apres codage)
N_RS = 2^Nb_bits_symb-1;
% nombre de symboles du mot d'info RS
K_RS = N_RS-2*t;

%Génération de bits
%!! Le nombre de bits générés doit être un multiple de K_RS pour
%que les programmes de codage/décodage RS fonctionnent
%On génère Nb_paquets_RS de taille K_RS*Nb_bits_symb bits
Nb_paquets_RS=64;
Nb_bits=Nb_paquets_RS*K_RS*Nb_bits_symb;
bits=randi(2,1,Nb_bits)-1;

%CODAGE RS
H = comm.RSEncoder(N_RS,K_RS,'BitInput',true);
bits_code_RS=step(H,bits.'.');

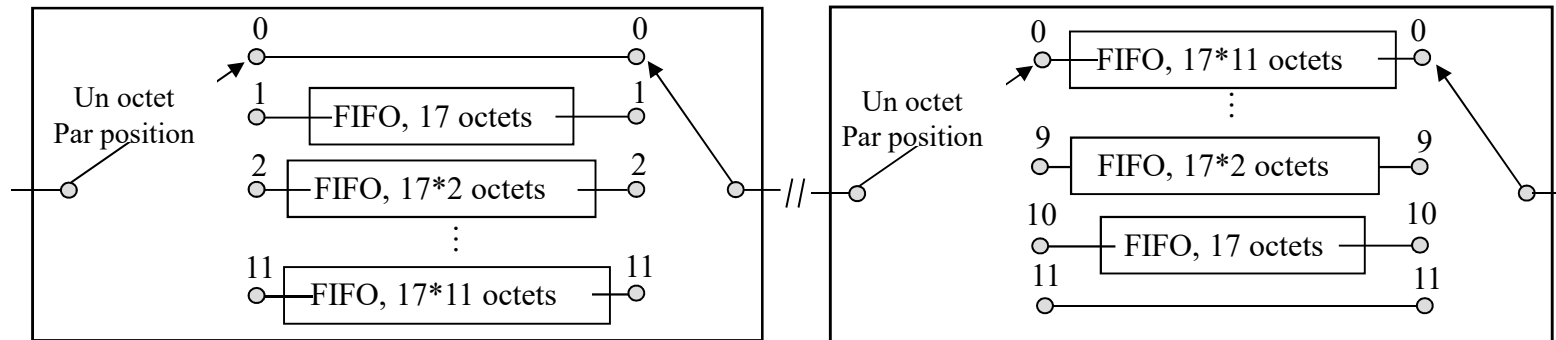
%DECODAGE RS
H = comm.RSDecoder(N_RS,K_RS,'BitInput',true);
bits_decodes_RS = step(H,bits_code_RS.'.');
```

Satellite Digital Video Broadcasting : DVB-S

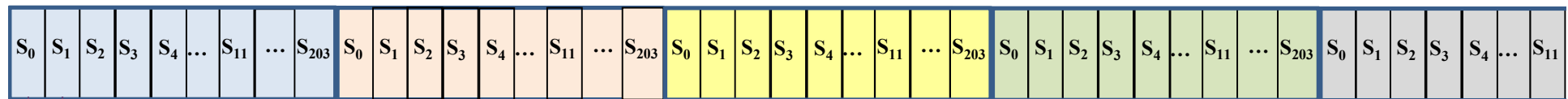
Codage canal : entrelacement

Entrelaceur convolutif de Forney

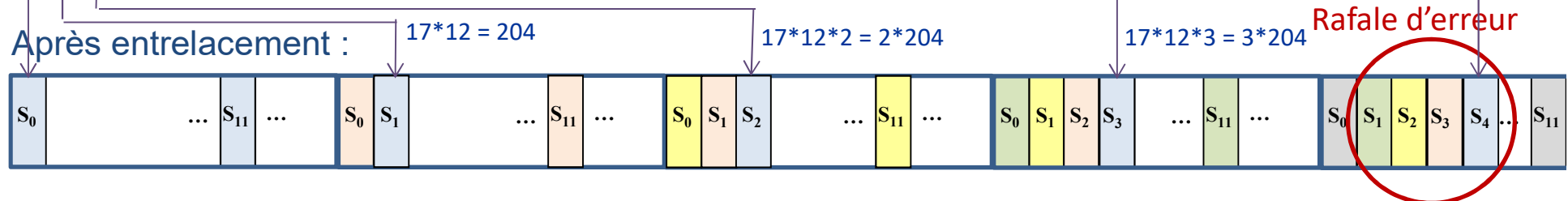
(Convintrlv.m, Convdeintrlv.m)



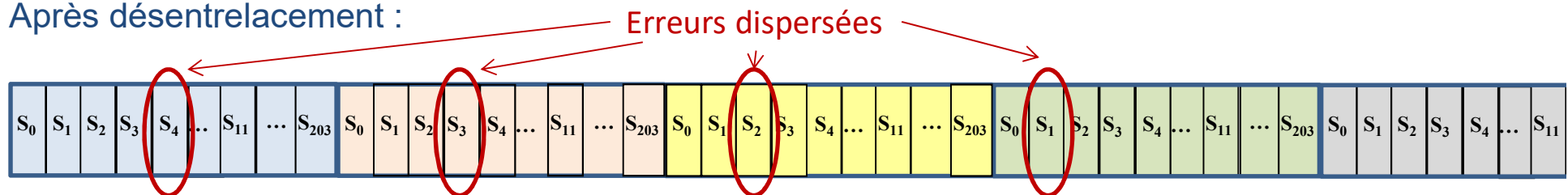
Avant entrelacement :



Après entrelacement :



Après désentrelacement :



Satellite Digital Video Broadcasting : DVB-S

Codage canal : entrelacement

Exemple d'entrelacement convolutif de Forney (code Matlab)

```
%Paramètres de l'entrelaceur
nrows=3; %Nombre de registres (FIFO)
slope=2;  %Taille des registres (FIFO)

%Délai introduit
Delai=nrows*(nrows-1)*slope;

%Génération des bits
Nb_bits=10;
bits= randi([0,1],1,Nb_bits) ;
bits_paddes=[bits zeros(1,Delai)];

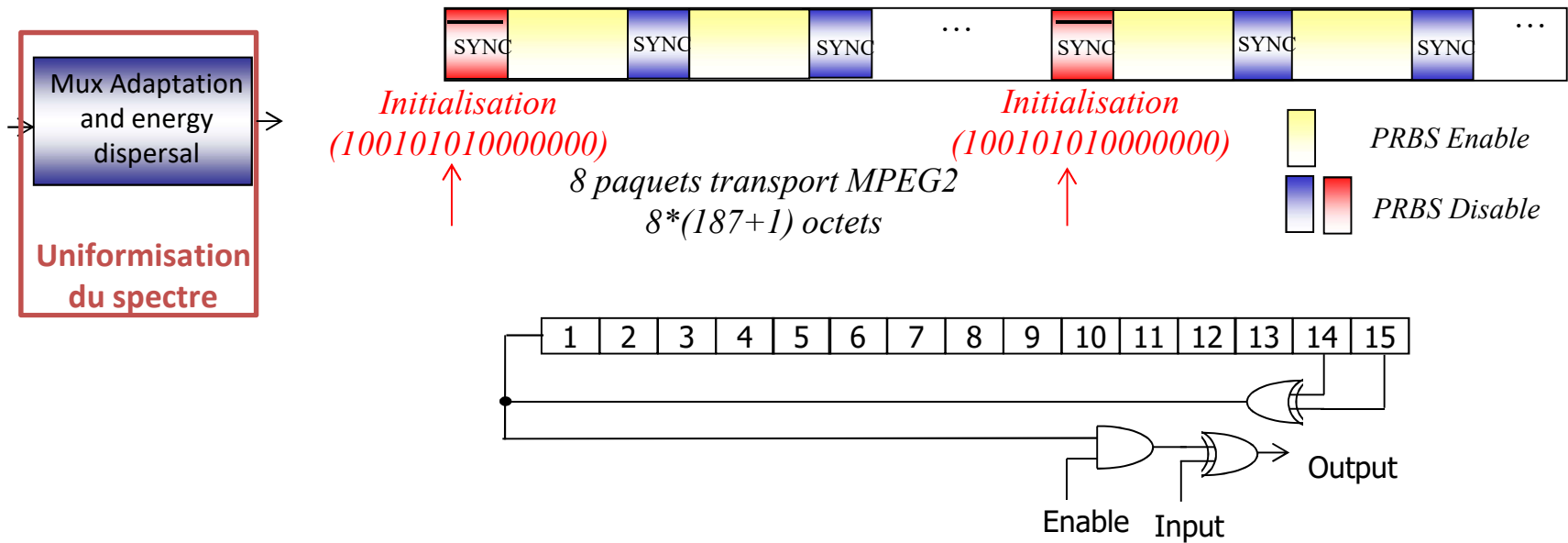
%Entrelacement
bits_entrelaces=convintrlv(bits_paddes,nrows,slope);

%Désentrelacement
bits_desentrelaces = convdeintrlv(bits_entrelaces, nrows, slope);

%Suppression du retard
Bits_retrouves=bits_desentrelaces(Delai+1:end);
```

Satellite Digital Video Broadcasting : DVB-S

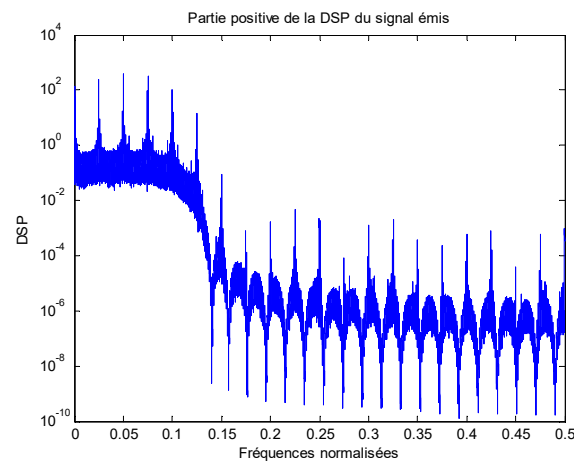
Dispersion d'énergie



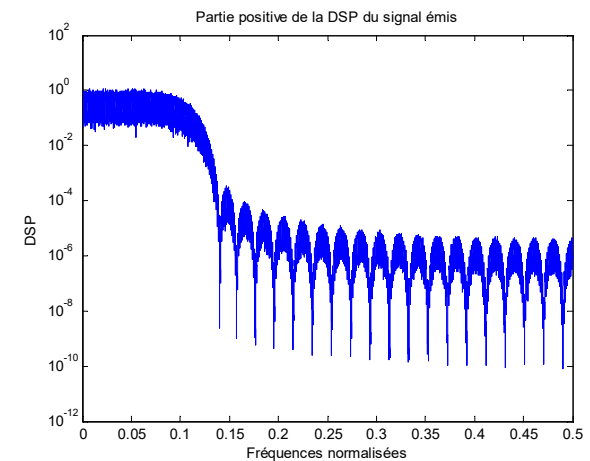
Exemple sur une image



DSP du signal associé sans embrouillage

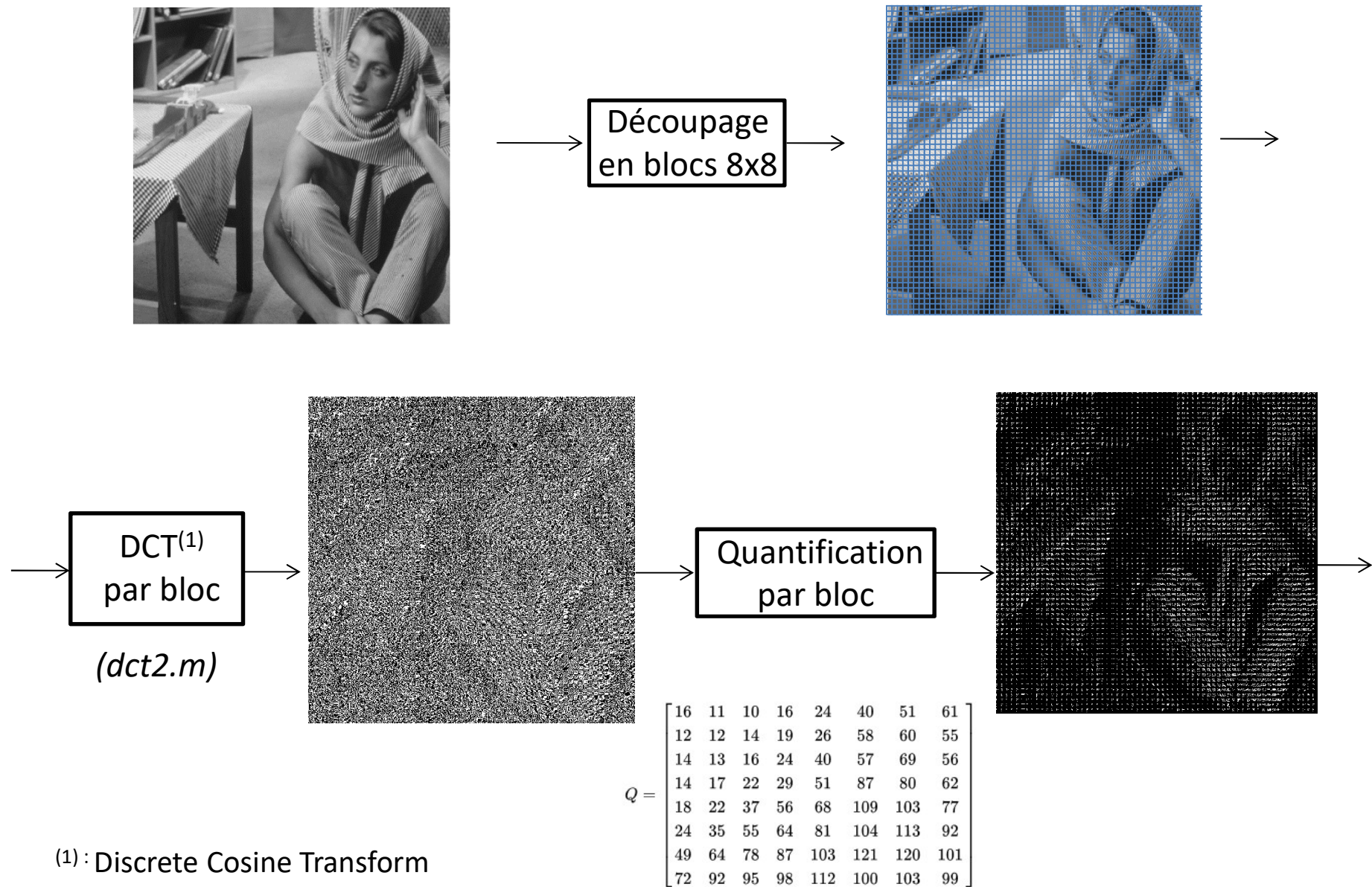


DSP du signal associé après embrouillage



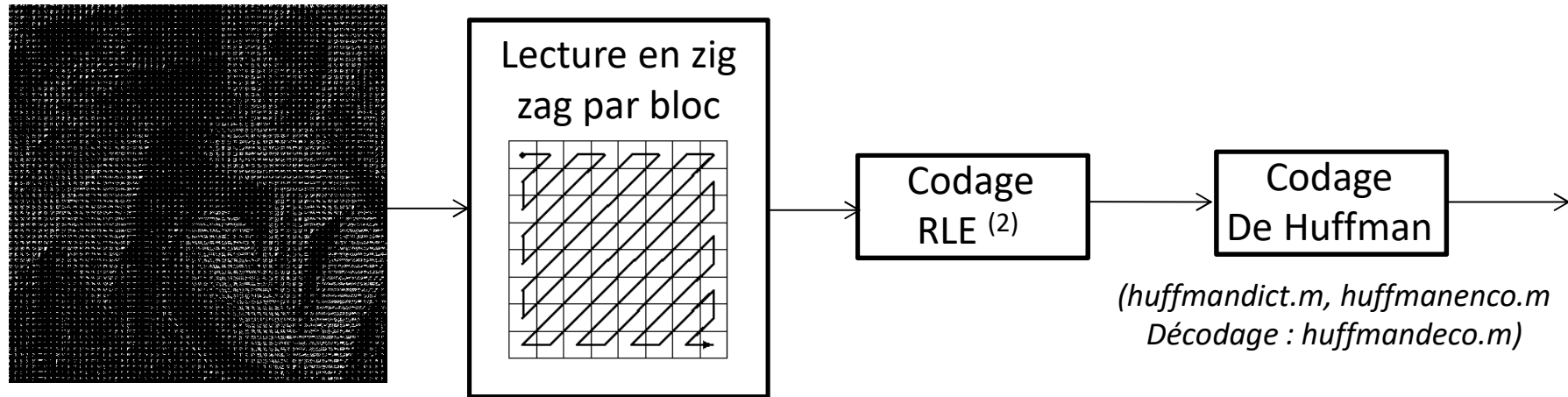
Transmission via la chaine DVB-S d'une image compressée

Compression d'image au format jpeg



Transmission via la chaine DVB-S d'une image compressée

Compression d'image au format jpeg



Exemple sur un bloc :

En sortie du quantifieur :

33	51	-2	0	0	54	33	-1
1	0	1	0	0	23	14	0
0	0	0	0	0	0	0	0
0	0	11	10	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Après lecture en zigzag :

```
[33 51 1 00-2010000000540011000001002333-114000000000000000
00000000000000000000]
```

Après RLE :

```
[(1)33 (1)51 (1) 1 (2)0 (1)-2 (1)0 (1)1 (7)0 (1)54 (2)0 (1)11 (5)0 (1)10 (1)0 (1)23 (1)33 (1)-1 (1)14 (34)0]
```

Après RLE + codage de Huffman (voir diapos suivantes) :

```
[101 0110 100 11 100 11 0101 11 0011 11 0010 11 0101 101 0001 0100 11]
```

Nombre d'apparitions : [1 1 1 2 1 1 1 7 1 2 1 5 1 1 1 1 1 34]

(2) : Run Length Encoding

Transmission via la chaine DVB-S d'une image compressée

Passages image -> flux binaire, flux binaire -> image

→ Lecture d'une image et conversion en flux binaire :

```
%Lecture de l'image (noir et blanc)
image = imread('barbara.png');
%Affichage de l'image
figure;
imshow(image);
% transformation de l'image en un flux binaire
y=de2bi(image);
flux_binaire_image_png=reshape(y.',1,size(y,1)*size(y,2));
```

→ Conversion du flux binaire reçu en image et affichage :

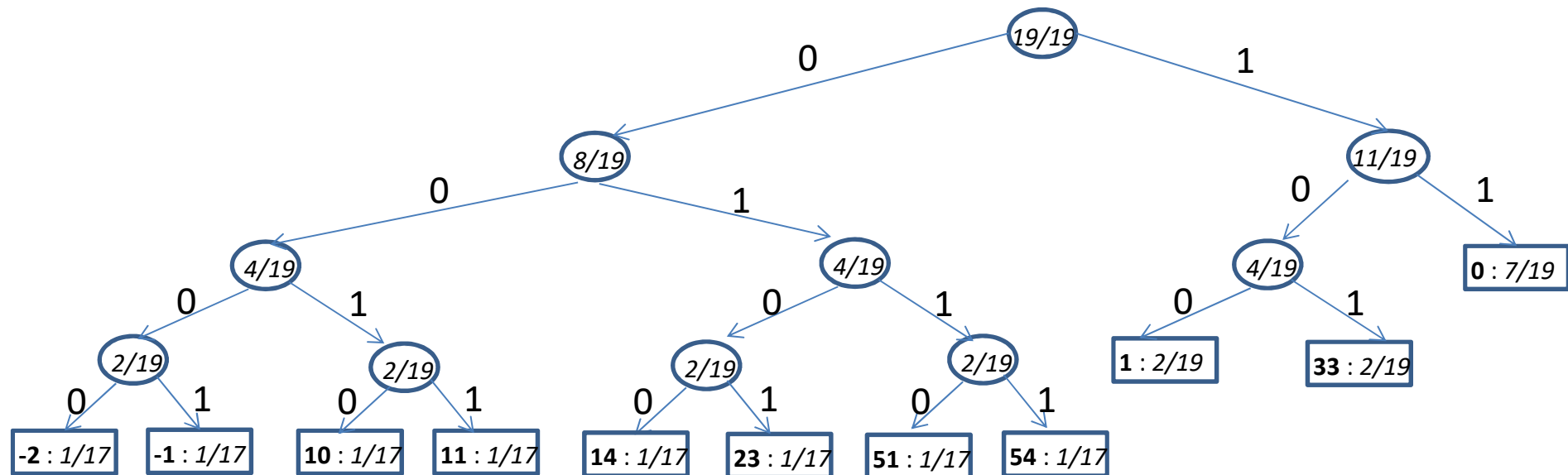
```
%On repasse des bits reçus à l'image reçue
bits=reshape(bits_recus,8,length(bits_recus)/8);
image_recue_png=reshape(bi2de(bits.').',size(image,1),size(image,2));
%Affichage
figure
imagesc(image_recue_png)
colormap('gray')
%Remarque : selon les traitements précédents il peut être nécessaire d'utiliser
uint8 sur les données binaires avant affichage de l'image.
```

Transmission via la chaine DVB-S d'une image compressée

Compression d'image au format jpeg – codage de Huffman

→ Exemple de codage de Huffman sur un bloc [33 51 1 (2)0 -2 0 1 (7)0 54 (2)0 11 (5)0 10 0 23 33 -1 14 (34)0]

-2 : 1/19 -1 : 1/17 10 : 1/19 11 : 1/19 14 : 1/19 23 : 1/19 51 : 1/19 54 : 1/19 1 : 2/19 33 : 2/19 0 : 7/19



Nombres à coder :	0	1	33	-2	-1	10	11	14	23	51	54
Probabilités d'apparition :	7/19	2/19	2/19	1/19	1/19	1/19	1/19	1/19	1/19	1/19	1/19
Codage :	11	100	101	0000	0001	0010	0011	0100	0101	0110	0101

Transmission via la chaine DVB-S d'une image compressée

Compression d'image au format jpeg – codage de Huffman

Bloc : [33 51 1 (2)0 -2 0 1 (7)0 54 (2)0 11 (5)0 10 0 23 33 -1 14 (34)0]

→ Codage de Huffman :

Nombre :	0	1	33	-2	-1	10	11	14	23	51	54
Probabilité d'apparition :	7/19	2/19	2/19	1/19	1/19	1/19	1/19	1/19	1/19	1/19	1/19
Codage :	1	000	001	01000	01001	01010	01011	01100	01101	01110	01111

[101 0110 100 11 100 11 0101 11 0011 11 0010 11 0101 101 0001 0100 11]

→ Codage « naturel »

Nombre :	0	1	33	-2	-1	10	11	14	23	51	54
Probabilité d'apparition :	7/19	2/19	2/19	1/19	1/19	1/19	1/19	1/19	1/19	1/19	1/19
Codage :	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010

[0010 1001 0001 0000 0011 0000 0001 0000 1010 0000 0110 0000 0101 0000 1000 0010 0100 0111 0000]

Transmission via la chaine DVB-S d'une image compressée

Codage/Décodage de Huffman en utilisant les fonctions Matlab

→ Codage de Huffman d'un vecteur de valeurs X

```
%Mise en place du dictionnaire pour le codage de Huffman
%valeurs = vecteur des valeurs possibles dans X
%proba = probabilité d'apparition de ces différentes valeurs
DICT = huffmandict(valeurs, probas);

%Lecture et affichage du dictionnaire
temp = DICT;
for i = 1:length(temp)
    temp{i,2} = num2str(temp{i,2});
end
Temp

%Codage de Huffman du vecteur de valeurs X
Y = huffmanenco(X, DICT);
```

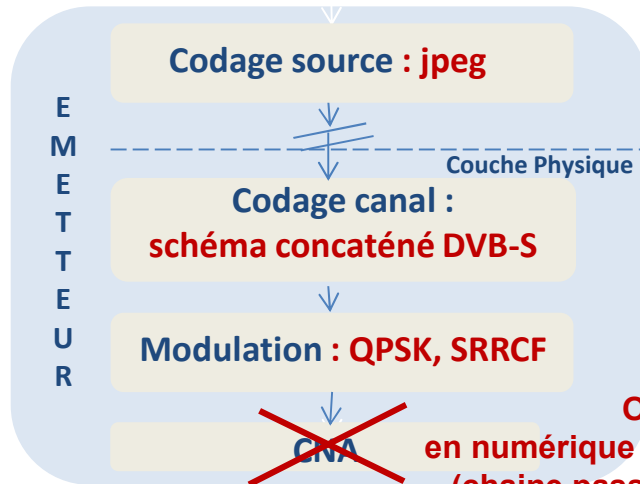
→ Décodage de Huffman d'un vecteur de valeurs Y

```
%Décodage de Huffman du vecteur Y
X= huffmandeco(Y, DICT);
```

Chaine de communication numérique à implanter

Information binaire à transmettre : 0 1 1 0 0 1 0 ...

Débit binaire R_b

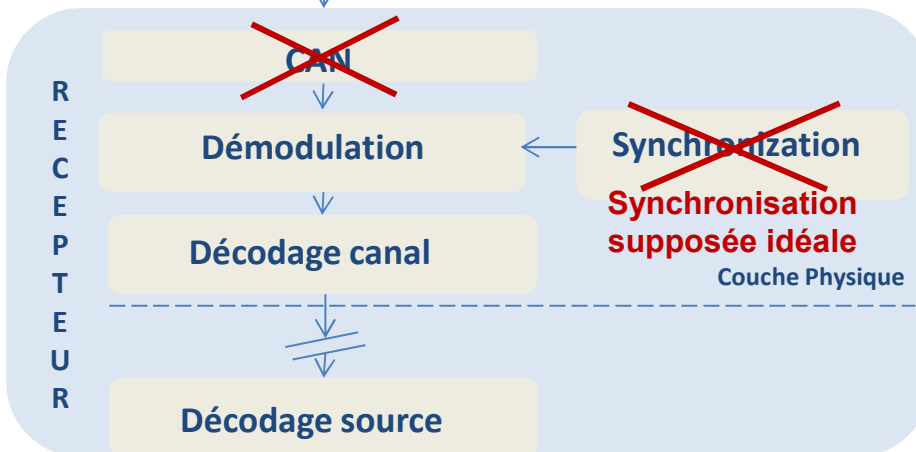


Efficacité spectrale :
Bande B nécessaire pour passer le débit R_b souhaité

Bande de transmission B nécessaire



SNR nécessaire par bit



Efficacité en puissance :
SNR par bit nécessaire à l'entrée du récepteur pour atteindre le TEB souhaité

Taux d'erreur binaire₃₀ (TEB)

Information binaire reçue : 0 1 0 1 0 1 1 ...