REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université d'Akli Mohand Oulhadj - BOUIRA

Faculté des Sciences et des Sciences Appliquées.

Département de Génie Electrique

MEMOIRE

Présenté pour l'obtention du diplôme de Master

Filière : Télécommunications

Spécialité : Système des Télécommunications

Sujet :

Différentes techniques de construction des codes correcteurs d'erreurs de faible densité à court terme

Réalisé par :

- Hanane SIFODIL
- Kenza BENDECHACHE

Dirigé par :

- Dr. S. BENSEGUENI (Maitre de Conférences B)

Année Universitaire : 2017/2018

Remerciements

En préambule, nous souhaitons rendre grâce à Dieu, le clément et le miséricordieux de nous avoir donné la force et la patience de mener à bien ce modeste travail.

Nous tenons évidemment à débuter ces remerciements en témoignant de notre profonde reconnaissance envers Monsieur Skander BENSEGUENI, maître de conférence à l'Université BOUIRA pour nous avoir encadré et dirigé ce travail avec patience, ainsi que pour sa riche contribution et ses précieux conseils.

Nous remercions vivement Monsieur DJEBIRI maître de conférences à l'Université BOUIRA, qui nous a fait l'honneur de présider le jury, ainsi que pour sa contribution à notre formation.

Nous voulons exprimer nos remerciements aux membres de jury,

Monsieur SAIDI enseignant à l'Université BOUIRA

Monsieur ARABI enseignant à l'Université BOUIRA.

Nos remerciements s'adressent aussi, à tous ceux qui nous ont prêté main forte.

DEDICACE

A l'aide de DIEU tout puissant, qui trace le chemin de ma vie, j'ai pu arriver à réaliser ce modeste travail que je dédie à :

A ma chère mère

A mon cher père

Pour l'éducation et le grand amour dont ils m'ont entouré depuis ma naissance.

Et pour leurs patiences et leurs sacrifices.

Qu'ils trouvent dans ce travail

Un Témoignage de mon profond amour et éternelle reconnaissance.

A ma sœur Nassima

A ma sœur Ilhem

A mon frère Koceila

A mon fiancé Mr : Haddouche Ali,

A toute ma famille ainsi ma belle-famille,

A ma chère amie DJahouari Hannane,

A mon ami Boualem Abderaouf,

En lui souhaitant la réussite dans leurs études et dans leur vie.

A tous mes amis

Pour la merveilleuse ambiance qui caractérise notre amitié. Qu'ils soient heureux sur les plans personnel, professionnel et social.

A tous ceux qui m'ont aidé afin de réaliser ce travail.

A tous ceux que j'aime et qui m'aiment.

J'exprime mes sentiments les plus profonds et leur dédie ce modeste travail.

M^elle **Kenza BENDECHACHE**

Dédicace

D'abord Je remercie **ALLAH** pour la faveur de la santé et de L'Islam et le tout puissant de m'avoir donné le courage pour accomplir ce modeste travail que Je dédie :

A toi maman, l'être le plus cher au monde qui n'a jamais cessé de prier pour moi et pour tout ton affection et ton amour que dieu te protège inchallah.

A l'âme de mon cher père Djamel, J'espère que tu habites les paradis.

A mon frère Ali, pour tous ses encouragements et je souhaite toute la réussite dans leur étude. A La Famille SIFODIL et AGGOUNI .

A mon binôme Kenza, A mes chers amis Mouhoub FELLAK, Ismahane, Imene et Karima FODIL PACHA, Ahlem AKSAS, Leila DJEBRI, Louiza CHARAN et Amel OUHMOUCHE pour tous les bons moments qu'on a passés ensemble, je vous souhaite beaucoup de réussite dans votre carrière.

A mon encadreur pour sa patience, ses conseilles, sa disponibilité et pour l'excellence de son accompagnement et la confiance qu'il nous a accordé.

A mes enseignants sans aucune exception, A mes collègues promotion 2017 et à tous ceux que j'aime et qui m'aime.

Hanane SIFODIL

Résumé :

Le développement rapide des réseaux mondiaux et les immenses possibilités offertes par les transactions électroniques en communications continues, posent aujourd'hui de manière cruciale, le problème de la protection de l'information contre les erreurs de transmission d'une part, et d'autre part, il faut que ces données soit non intelligibles sauf à l'auditoire voulu. Afin de pallier ces deux contraintes on utilise le codage de l'information pour combattre les erreurs de transmissions.

La découverte dans les années 90 des Turbo-codes révolutionné la manière d'appréhender un système de communications numériques. Cette avancée notable a permis la redécouverte des codes correcteurs d'erreurs inventés par R. Gallager en 1963, appelés codes Low Density Parity Check (LDPC).

L'intégration des techniques de codage dites avancées, telles que les Turbo-codes et les codes LDPC, se généralise dans les standards de communications. Dans ce contexte, l'objectif de ce mémoire est d'étudier de nouvelles structures de codage de type LDPC réguliers et irréguliers en se basant sur le codage.

Mots clés : Codage de canal, Codes LDPC, turbo codes, graphe de Tanner.

Abstract:

The rapid development of global networks and the tremendous possibilities offered by electronic transactions in continuous communications, pose today crucially, the problem of the protection of the information against the errors of transmission on the one hand, and other On the other hand, this data must be unintelligible except to the intended audience. In order to overcome these two constraints, the coding of information is used to combat transmission errors.

The discovery of Turbo-codes in the early 90s revolutionized the way of understanding a digital communications system. This significant advance allowed the rediscovery of the error-correcting codes invented by R. Gallager in 1963, called Low Density Parity Check (LDPC) codes. The integration of so-called advanced coding techniques, such as Turbo-codes and LDPC codes, is becoming commonplace in communications standards. In this context, the purpose of this thesis is to study new regular and irregular LDPC coding structures based on coding.

Key words : Channel coding, LDPC codes, turbo codes, Tanner graph

ملخص:

التطور السريع للشبكات العالمية والإمكانيات الهائلة التي توفرها المعاملات الالكترونية في مجال الاتصالات الجارية، اليوم تشكل حاسم، مشكلة حماية المعلومات ضد أخطاء الإرسال من جهة، والآخر أولا، يجب أن تكون البيانات غير مفهومة إلا إلى الجمهور المطلوب. للتغلب على هذه القيود باستخدام ترميز المعلومات لمكافحة أخطاء الإرسال.

أحدث ثورة Turbo-codes في التسعينيات ثورة في طريقة فهم نظام الاتصالات الرقمية. وقد سمح هذا تقدما كبيرا في إعادة اكتشاف رموز تصحيح الخطأ اخترعها R. Gallager في عام 1963، رموز يسمى منخفض الكثافة التكافؤ تحقق (LDPC).

دمج ما يسمى تقنيات الترميز المتقدمة، مثل رموز turbo ورموز LDPC، ينتشر في معايير الاتصالات. وفي هذا السياق، فإن الهدف من هذه الورقة هو لاستكشاف الهياكل LDPC الترميز النظامية وغير النظامية نوع جديد يقوم على الترميز

الكلمات المفتاحية: ترميز القناة، الرسم البياني Tanner، شفرات turbo ، شفرات LDPC

Table des matières

Table des matières

Table des matières

Table des matières

Liste des figures

Liste des tableaux

Liste des Figures et des Tableaux

Liste d’abréviation

A

ARQ: Automatic **R**epeat Request

B

BCH : Bose Ray-**C**haudhuri et **H**ocquenghem

BEC: Canal d’**E**ffacement **b**inaire

BSC: Canal **B**inaire **S**ymétrique

C

CC : Codes **C**oncaténés

F

FEC: Forward **E**rror **C**orrection

G

GRS: Generalized **R**eed–**S**olomon

I

IEEE: Institute of **E**lectrical and **E**lectronics **E**ngineers

L

LDPC: low-**d**ensity **p**arity-**c**heck

W

Wi-Fi: Wireless **Fi**delity

Introduction générale

De nos jours, nous vivons dans un monde où les communications jouent un rôle primordial tant par la place qu'elles occupent dans le quotidien de chacun, que par les enjeux économiques et technologiques dont elles font l'objet. Nous avons sans cesse besoin d'augmenter les débits de transmission tout en gardant ou en améliorant la qualité de ceux-ci.

C'est dans la course au débit et à la fiabilité que les codes correcteurs entrent en jeux. La communication avec les sondes spatiales, à l'autre bout du système solaire, pose le problème de la fiabilité du message transmis. Une transmission sur une telle distance est obligatoirement parasitée (notamment à cause de diverses sources de perturbations électromagnétiques). Pourtant, dans ce domaine et dans bien d'autres, il est primordial que les informations collectées par les sondes soient bien reçues. Il y a donc nécessité de «protéger» la transmission : c'est le rôle des codes correcteurs d'erreurs. Ce sont des codes où on rajoute au message à transmettre des informations supplémentaires, qui permettent de reconstituer le message au niveau du récepteur. Un code correcteur d'erreur permet de corriger une ou plusieurs erreurs dans un mot de code en ajoutant aux informations des symboles redondants, autrement dits, des symboles de contrôle.

Les codes LDPC (Low Density Parity-Check) forment une classe de codes en bloc qui se caractérisent par une matrice de contrôle creuse. Ils ont été décrits pour la première fois dans la thèse de Gallager au début des années 60 [1]. Les codes de contrôle de parité (LDPC) étaient à l'origine inventé et étudié par Gallager. L'innovation cruciale a été l'introduction par Gallager des algorithmes de décodage qu'il montré être capable d'atteindre une fraction significative de capacité de canal à faible complexité. Intérêt pour les codes LDPC a été ravivé à la suite de la découverte des codes turbo et les codes LDPC ont été redécouverts indépendamment par les deux MacKay et Neal [2].

Ces dernières années ont apporté de nombreux nouveaux développements dans ce domaine. Premier arrivé plusieurs papiers Luby, Mitzenmacher a introduit de nouveaux outils pour la recherche de messages- passer des décodeurs pour le canal d'effacement binaire (BEC) et le canal binaire symétrique (BSC), et ils ont étendu la définition des codes LDPC pour inclure les codes irréguliers. Les mêmes auteurs ont également présenté des séquences de codes qui, asymptotiquement dans la longueur du bloc.

À bien des égards, les codes LDPC peuvent être considérés comme des concurrents sérieux aux codes turbo. En particulier, les codes LDPC présentent une asymptotisation une

meilleure performance que les codes turbo et ils admettent un large éventail de compromis entre performance et complexité de décodage. Une critique majeure concernant les codes LDPC a été leur apparente complexité d'encodage élevée. Considérant que les codes turbo peut être encodé en temps linéaire, une implémentation simple du codeur pour un code LDPC a la complexité quadratique dans la longueur de bloc. Plusieurs auteurs ont abordé cette question.

Ce mémoire comprend quatre chapitres où on a opté pour une présentation graduelle des concepts par ordre de leur apparition dans le mémoire.

Dans le premier chapitre, il est question des notions de base et de concepts de communications numérique, ainsi que la théorie d'information tel que la modélisation de canaux et surtout du canal binaire symétrique et gaussien, puis on a rappelé quelque définition des codes (binaire et linéaire). Ensuite, on a expliqué les différents types de code correcteur (code en blocs, convolutifs et turbo codes).

Le deuxième chapitre a pour objectif d'étudier les concepts de base sur les codes LDPC en présentant quelques méthodes de décodage et de présentation par les graphes de tanner et matricielles, et la construction des codes LDPC de Gallager.

Dans le troisième chapitre on va appliquer le principe de codage et de décodage des codes correcteurs d'erreurs LDPC par la présentation d'un exemple d'information de taille 4 bits avec la correction des erreurs et sans erreurs. Nous allons à la fin utiliser le principe de codage par la méthode de pivot de gauss, et faire une vérification de l'efficacité de l'algorithme.

Dans le Quatrième chapitre, nous allons appliquer l'algorithme de Gallager régulier pour trouver la matrice de contrôle de parité H et ça représentation par un graphe de Tanner ensuite l'algorithme de réalisation irrégulière est représenté.

Chapitre 1 : Généralités sur les codes correcteurs d'erreurs

1.1. Introduction

La conception des codes est une discipline des mathématiques très utilisées, dont le sujet est la transmission d'informations sur un canal de transmission bruité, en utilisant des objets combinatoires et algorithmiques nommés codes correcteurs d'erreurs. Pour introduire ce sujet, il est nécessaire de détailler les notions de base du codage et de connaitre le chemin qu'empreinte un message depuis l'émetteur jusqu'au destinataire, et de considérer les concepts intéressants qui apparaissent. Il existe trois étapes impliquées dans la chaine de transmission : la source, le récepteur et le support de transmission.

La conception des codes correcteurs d'erreurs a pour but la création de codes capables de détecter et de corriger des erreurs arrivées lors de la transmission d'un message. Elle a pour base théorique la conception de l'information qui a véritablement commencé par l'article de Claude Shannon (1948) [3].

1.2. Schéma de transmission

Pour transmettre l'information numérique du l'émetteur jusqu'au récepteur, on applique un système de transmission numérique. L'information a transporté sur cette chaîne est sous la forme d'une suite binaire, c'est-à-dire des 0 et des 1. Le support physique qui permet de transmettre cette information soit sur un câble, au bien l'espace dans le cas des satellites. La figure 1.1 représente une chaîne de transmission [4].

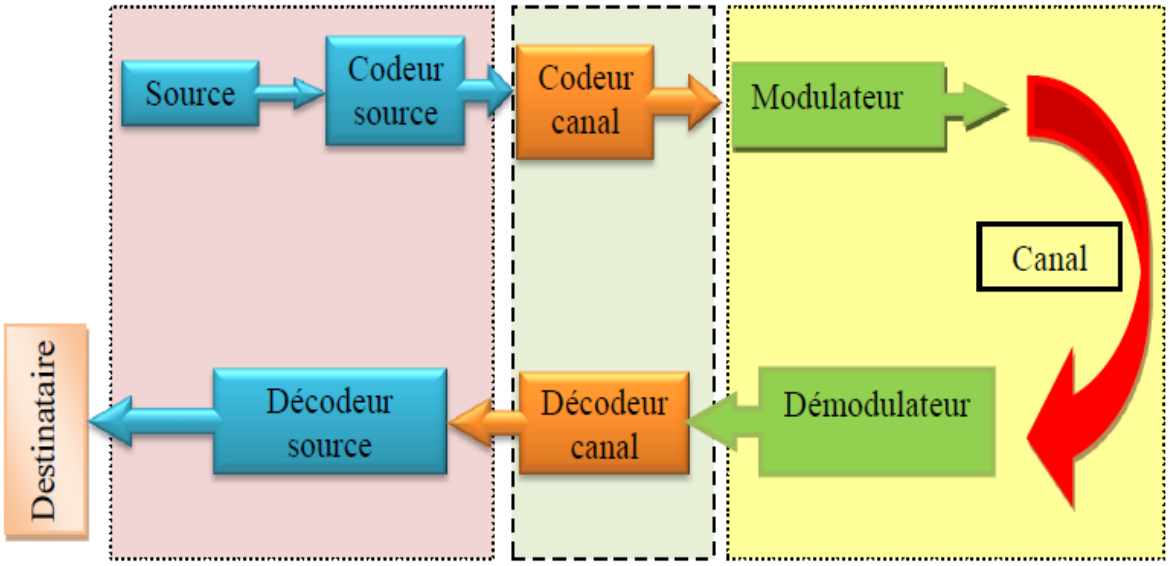

Figure 1. 1: Modélisation d'une chaine de transmission numérique [5].

L'objectif général de chaque bloc du schéma est la fiabilité et la rapidité de la transmission. La première étape, nommé codage de source, comporte à compresser les données et donc augmenter l'efficacité de la transmission. La deuxième étape, notée le codage de canal, ajoute de la redondance au message afin de pouvoir corriger les erreurs qui arriverons au cours de la transmission. La troisième étape, c'est la modulation qui convertit l'information numérique en un signal physique adapté au canal qui va garantir le transport. Le récepteur réalise les opérations en sens inverse à la réception du signal afin de récupérer le message envoyé. Si l'expéditeur souhaite, de plus, protéger son message contre des personnes qui souhaiteraient le lire alors qu'elles ne sont pas destinataires, il va chiffrer son message avec un algorithme de chiffrement. La sécurité de cet algorithme sera garantie par le caractère secret de la clé de chiffrement. Cette opération sera réalisée entre le codage de source et le codage de canal.

- **Codage de source**

Des bits de redondance sont complétés avant que le message ne soit émis par la source. Le codage de source a pour but d'éliminer la redondance de la source. Il faut pour cela répartir l'information sur l'ensemble des symboles x_i. On détermine alors l'entropie d'une source comme étant l'information moyenne émise par la source :

$$H(X) = E\left(\log_2\left(\frac{1}{p_i}\right)\right) = \sum_i p_i \log_2\left({}^{1}/_{p_i}\right) \quad (1.1)$$

Où p_i est la probabilité d'émettre le ième symbole. Pour une source binaire avec p_i =x, on peut alors déduire la fonction d'entropie binaire :

$$H_2(X) = x \log_2\left({}^{1}/_{x}\right) + (1-x)\log_2\left({}^{1}/_{(1-x)}\right) \quad (1.2)$$

Si l'entropie de la source est inférieure à l'entropie maximale possible, alors la source est qualifiée de redondante et la longueur du message émis peut être réduite par un codeur de source.

Concernant le codage Source, le premier théorème de Shannon montre qu'il existe un procédé de codage déchiffrable où la longueur moyenne des mots par symbole de la source est aussi voisine que l'on veut de sa borne inférieure ${}^{H}/_{\log_2(q)}$ où q est la taille de l'alphabet considéré [6].

- **Codage de canal**

Le codage de canal a pour but de protéger le message émis par la source normalisée (sans redondance) contre les bruits et les perturbations introduits par le canal de propagation. Pour garantir un transfert sur un canal bruité, il est nécessaire d'introduire de la redondance dans le message transmis. En réception, le décodeur reçoit le message émis par le codeur perturbé à cause de bruit du canal. L'information mutuelle $I(X,Y)$ en Figure 1.2 entre l'entrée et la sortie du canal peut être exprimée sous la forme

$$I(X;Y) = H(X) - H\left(X/_Y\right) = H(Y) - H(Y/X) \quad (1.3)$$

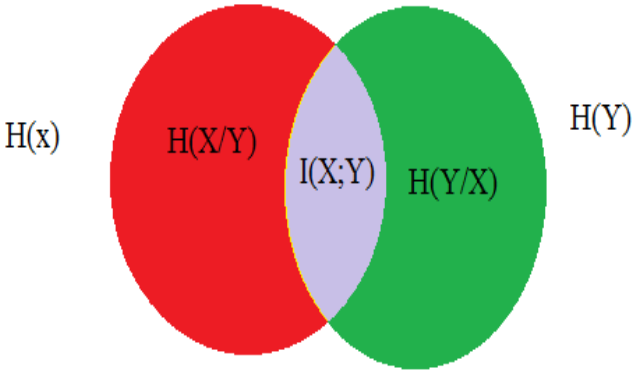

Figure 1. 2: Information mutuelle.

Où $H(X)$est l'entropie de la source et $H(X/Y)$ est le niveau d'incertitude sur la variable X connaissant Y. La démonstration concernant le codage de canal est le résultat le plus important de la théorie de l'information (Figure 1.2). Ce théorème est appelée le deuxième théorème de Shannon [6].

1.3. Théorie de l'information

1.3.1 Modelés de canaux

Un code correcteur est l'ensemble des mots de code $(\in A_n)$ obtenus après le codage. Les systèmes de communication appliquent un canal de transmission, appelé aussi : support de transmission A, afin d'échanger de l'information d'un point à un autre.

Le support peut être de différents types : câble, fibre optique, canal radioélectrique. N'étant pas parfait, il introduit des perturbations, affaiblissement, écho, bruit qui détériorent l'information émise et créent des erreurs dans le message. Afin de fiabiliser le message (empêcher la perte de données due aux perturbations, bruit), les systèmes intègrent un processus de protection du message émis. Le principe global de cette protection est l'ajout de redondance, ça veut dire d'information supplémentaire, la plus optimale possible en termes de coût, de volume et de contraintes qui dépendent largement du canal. D'écrivons dans un premier temps les modèles simplifies de canaux de communication les plus communs [7].

1.3.1.1 Le canal binaire symétrique

C'est un canal binaire caractérise par la probabilité d'erreur *p* qu'au cours de la transmission un bit (0 ou 1) soit modifié en son opposé. Ces modifications se produisent indépendamment sur chacun des bits transmis On représente ce canal par la Figure1.3**.**

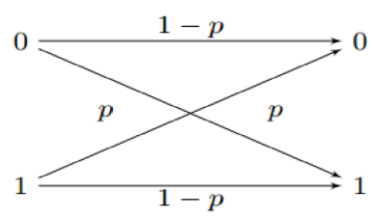

Figure 1. 3: Canal binaire symétrique de probabilité d'erreur $\boldsymbol{\rho}$ [7].

1.3.1.2 Le canal Gaussien

C'est un modèle fréquemment utilisé car il décrit souvent bien la transmission dans un milieu physique. Après modulation, on a un signal en entrée du canal ; en sortie du canal, le signal reçu résulte de l'addition du signal émis et d'un bruit. Pour ce canal, le bruit est modélisé par une variable aléatoire complexe de loi normale $N(0, N_0)$.

Sa projection sur un axe réel est donc une variable aléatoire réelle de loi normale $N\left(0, \frac{N_0}{2}\right)$. Puisque l'on s'intéresse souvent aux distorsions subies dans une direction donnée, on invoque alors la densité mono-latérale de bruit $\frac{N_0}{2}$. On peut montrer que sous certaines modulations et démodulations, pour un alphabet binaire, le canal de transmission composé du modulateur, du canal à bruit blanc gaussien et du démodulateur peut être assimilé à un canal binaire symétrique de probabilité d'erreur [8].

$$\tau = \frac{1}{\sqrt{2\pi}} \int_0^{\infty} \sqrt{\frac{2}{N_0}}\ exp\left(\frac{-t^2}{2N_0}\right) dt \qquad (1.4)$$

1.5. Définition de code binaire

Le code binaire, plus généralement appelé système binaire, est un système de numération utilisant la base 2 avec un nombre exprimé sous forme de série de 0 et de 1. La position des 0 et des 1 indique respectivement l'absence ou la présence d'une puissance de 2 [9].

1.6. Définition d'un code linéaire

Un mot de code d'un code en blocs linéaire $C(n,k)$ (avec $k < n$) construit sur un corps F se compose de :

- k symboles composés de la séquence d'information à transmettre répartis dans l'ensemble du message.
- $n-k$ symboles calculés à partir d'une combinaison linéaire d'une partie prédéterminée des symboles d'information et eux aussi répartis dans le message. Il s'agit des symboles de parité ou de redondance.

$C(n,k)$ est un sous-espace vectoriel de dimension k de l'espace engendré par $GF(2^n)$, n correspond à la longueur du code, k à sa dimension et k/n au rendement du code [10].

Exemple :

$$\begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Nous remarquons que, la multiplication de message i par la matrice génératrice nous a donné un mot de code (information + redondance = c) mais, il est lisible que l'information a été modifiée, donc ce n'est pas un vrai codage. Pour régler ce problème, il suffit de transférer la matrice génératrice à une autre matrice qui se commence par la matrice identité permettant de conserver le même message, et pour cela il suffit d'appliqué la méthode d'élimination de Gauss par exemple, comme le montre la figure suivante :

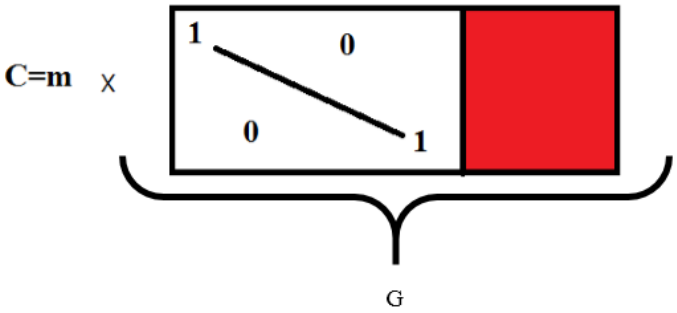

Figure 1. 4: G, matrice génératrice sous forme systématique [10].

1.7. Généralités sur le décodage

Plusieurs méthodes de décodage existent, le décodage au maximum de vraisemblance est la méthode la plus souhaitable mais n'est pas toujours réalisable en pratique, sauf pour les codes convolutifs. Elle consiste à partir d'un modèle probabiliste du canal a calculé le mot le plus probable compatible avec le mot reçu. Il est facile de montrer que le décodage au maximum de vraisemblance consiste à retrouver le mot de code le plus proche (au sens de Hamming) du mot reçu.

Le décodage en revanche va utiliser la structure algébrique du code. La connaissance de la seule matrice génératrice ne suffit en général pas à décoder de manière efficace, et la structure algébrique n'est pas nécessairement facile à obtenir à partir de la matrice génératrice [11].

1.8. Différents codes correcteurs

1.8.1. Codes en Bloc

Le message à transmettre est découpé en blocs de k bits, qui sont alors traités séparément par le codeur [12]. On appelle code un ensemble C de $2k$ n-uplets de bits, avec $n > k$, dont les éléments sont appelés mots. Les blocs à transmettre sont traduits en des éléments du code, chacun des $2k$ messages différents possibles correspondant de manière unique à un des mots du code. On appelle distance de Hamming entre deux mots du code le nombre de composantes par lesquelles ils différents. La distance minimale d'un code C, notée d, est alors la distance de Hamming minimale entre deux mots quelconques. Un code est alors défini par les trois paramètres$[n, k, d]$, où n est la taille du code, k sa dimension et d sa distance minimale [13].

Tableau 1. 1: Exemple qui peut considérer le code suivant, de paramètres 5,2,3.

Message	Mot correspondant
(0,0)	(0,0,1,1,0)
(0,1)	(0,1,0,1,1)
(1,0)	(1,0,1,0,1)
(1,1)	(1,1,0,0,0)

On montre qu'un code est capable de corriger $e = E\left({}^{(d-1)}/_{2}\right)$ erreurs 1. En effet, si on sait qu'un mot x a été transmis avec e erreurs ou moins, et si on reçoit le n-uplet x' alors x est le mot a du code tel que $d(x', a)$ soit minimal : le décodage s'effectue donc en calculant les distances entre x' et tous les mots du code. Le rapport d/n renseigne donc sur la fiabilité du code. Le rapport $R = {}^{k}/_{n}$ est appelé taux du code. Plus il est proche de 0 et plus la redondance introduite, et donc le temps de transmission, sont importants. On voit qu'il est impossible d'avoir en même temps une fiabilité et un taux élevés. Toute la difficulté de la construction des codes est de trouver le bon compromis [14].

1.8.1.1. Le code à répétition

Le code à répétition consiste à répéter n fois le bit d'information. C'est un code [n; 1; n], de dimension 1, de longueur n et de distance n. Pour n impair, ce code est parfait et optimal lorsque l'on protège un seul bit de donnée. En théorie de l'information pour un modèle de bruit donne (canaux sans mémoire) et le rendement constant on peut avoir une probabilité d'erreur qui tend vers 0. Pour un code à répétition le rendement décroît pour une probabilité d'erreur qui tend vers 0. Ce code ne tire malheureusement pas parti du faite qu'il est plus efficace de protéger simultanément plusieurs bits d'information. Ce code est surtout utilisé en télécommunication pour reconnaître une modulation de signal [15].

1.8.1.2. Codes de Hamming

Le code de Hamming se définit simplement au moyen d'une matrice de parité H.de dimension m × n et de longueur $n = 2^k - 1$et $m = 2^k - 1 - k$ et $n - k = m$; Par exemple, pour $k = 3$ une des matrices de parité s'écrié [15].

$$H = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Les colonnes de la matrice de parité étant toutes distinctes, la distance minimale du code est 3. Plus généralement, les codes de Hamming sont des codes $[2^k - 1, 2^k - k - 1.3]$.

Exemple :

- Si k=2 : alors $H = \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$ donc $G = (1 \quad 1 \quad 1)$.
  On retrouve le code (3,1) par répétition : c'est le plus petit code de Hamming.
- Si k=3 : on construit $H = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ par exemple, donc
  $G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$ c'est un code de hamming de taille (7,4).

1.8.1.3. Codes cycliques :

Les codes cycliques sont une sous famille des codes linéaires pour lesquels par définition, toute permutation circulaire d'un mot de code est aussi un mot de code. Comme nous allons le voir ci-dessous, ceci permet une simplification matérielle au niveau du codeur, et du décodeur, en évitant les multiplications matricielles (3.3) et (3.4), compliquées à mettre en œuvre lorsque, comme pour la majorité des applications, la taille n du code devient grande [16].

Définition :

On appelle représentation polynomiale du vecteur $c = (c_0 \ldots . c_{n-1})$, le polynôme ayant le même coefficient que c

$$C(X) = C_0 + C_1 X + \cdots + C_{n-1} X^{n-1} \qquad (1.5)$$

Cette représentation polynomiale de degré $n - 1$ peut etre vue comme le résultat d'une opération modulo un polynôme F de degré n et on choisit $F(X) = X^n - 1$. Nous allons caractériser$C[X]$, l'ensemble des représentations polynomiales des mots de code.

1.8.1.4. Codes alternants

Les codes alternants ont été introduits par Helgert. Des résultats importants sur ces codes. De nombreux codes appartiennent à cette grande famille de codes linéaires [17]. Nous étudierons les codes de Goppa classiques qui forment une sous-famille parmi ces codes. On

notera que les codes de Goppa ont été découverts avant les codes alternants. Nous travaillerons aussi avec les codes BCH qui sont des codes cycliques alternants. Pour autant, tous les codes alternants ne sont pas cycliques et réciproquement, tous les codes cycliques ne sont pas alternants. Les codes alternants sont dérivés des codes de Reed-Solomon généralisés (GRS). Les codes de Reed-Solomon ont été découverts en 1960. Il s'agit de codes d'évaluation.

Cela signifie que les mots d'un tel code peuvent être obtenus en évaluant des polynômes en un nombre fixé de points.

(Code de Reed-Solomon généralisé). Le code GRSk (α, ν) de longueur n sur F_{q^m} est :

$$\{(v_1 f(\alpha_1), v_2 f(\alpha_2), \ldots., v_n f(\alpha_n)) \in F_{q^m}^n : \forall f \in F_{q^m}[x]_k\} \quad (1.6)$$

1.8.1.5. Codes Concaténés (CC)

Les codes concatènes ont constitué, et constituent toujours, l'une des constructions les plus utilisées pour obtenir une protection contre des niveaux de bruit importants. Ils risquent d'être supplantes dans les années µa venir par les turbo-codes et les codes LDPC.

Dans un schéma concatène, l'information est codée deux fois séquentiellement. Tels qu'ils ont été décrits initialement par Forney, ils utilisent deux codes en bloc. Le premier, le code externe, est défini sur un alphabet de grande taille q et le second, le code interne, binaire en général, codera chaque symbole q-aire an de fournir une protection supplémentaire. Par la suite, le code interne à été remplace par un code convolutif. Ce schéma concatène avec un code convolutif interne et un code de Reed-Solomon externe a été standardise pour les communications spatiales.

Si l'on se place à la sortie du canal, tout se passe comme si la séquence codée provenait du code interne seul. Ce codeur permet une forte résistance au bruit. L'ajout de redondance sur un code en possédant déjà avec le premier codeur ore la possibilité de retrouver l'information là ou un seul codeur n'aurait pas suffit pour la retrouver.

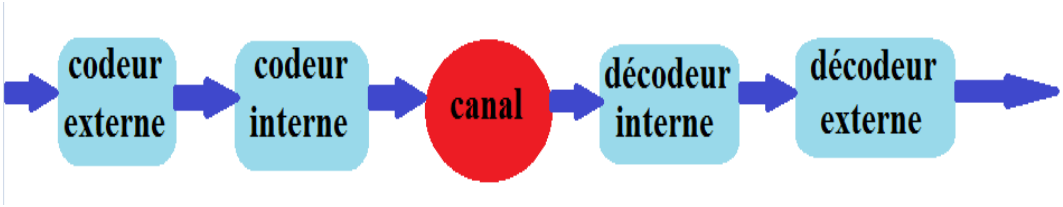

Figure 1. 5: Schéma de codage concaténé [18].

Comme le montre le schéma de codage d'un code concaténé (Figure1.5) le décodage s'effectue dans l'ordre inverse du codage. Chacun des codes utilisera un algorithme ad hoc. La construction est efficace car les codes sont complémentaires.
Schématiquement, le code interne va corriger les erreurs isolées et le code externe les rafales. De plus, dans les décodeurs les plus élabores, le résultat du décodage interne pourra être assorti d'une information de fiabilité qui améliore (Marginalement) les performances du décodeur externe (plus le décodeur interne a corrige d'erreurs, moins l'information fournie est fiable) [18].

1.8.1.6. Codes Produits

Les codes produits, inventés par P. Elias en 1954 [19], sont construits par la concaténation série de deux ou plusieurs codes en blocs linéaires à faible pouvoir de correction. En général, les codes utilisés sont les codes BCH et Hamming. Le produit de deux codes linéaires en bloc sur un même alphabet q-aire est défini comme l'ensemble des matrices génératrices dont toutes les lignes sont dans un code et toutes les colonnes dans un autre code.

1.8.1.7. Code LDPC

Les codes LDPC ("low-density parity-check") ont été introduits par Gallager en tant qu'une classe de codes avec une probabilité d'erreur évanescente lorsque la taille du bloc est infinie et une complexité de décodage raisonnable. La construction de code initialement proposée par Gallager est l'ensemble des codes LDPC réguliers de longueur n et défini par une matrice de parité H de dimension (m × n), ou m représente le nombre de bits de parité. Pour les codes LDPC réguliers, la matrice H contient dc (resp. dv) uns dans chaque ligne (resp. colonne). Typiquement, dc et dv sont des entiers petits de telle sorte que H soit clairsemé. Cet ensemble est généralement appelé l'ensemble des (n, dv, dc) codes LDPC réguliers. Les positions des uns sont choisies au hasard et la séquence binaire c = (c1, . . . , cn) de longueur n est un mot de code si et seulement si $HC^T = 0$. Un code LDPC peut aussi être représenté par son graphe bipartite associe ou chaque bit dans le mot de code est représenté par un nœud de variable ou "à gauche" et chaque équation de parité est représenté par un nœud de parité ou "`à droite" ; et une connexion relie un nœud de variable à un nœud de parité si et seulement si le nœud de parité intervient dans l'équation de parité correspondant au nœud de parité. Le nombre de connexions reliées à un nœud est appelé le degré du nœud [18].

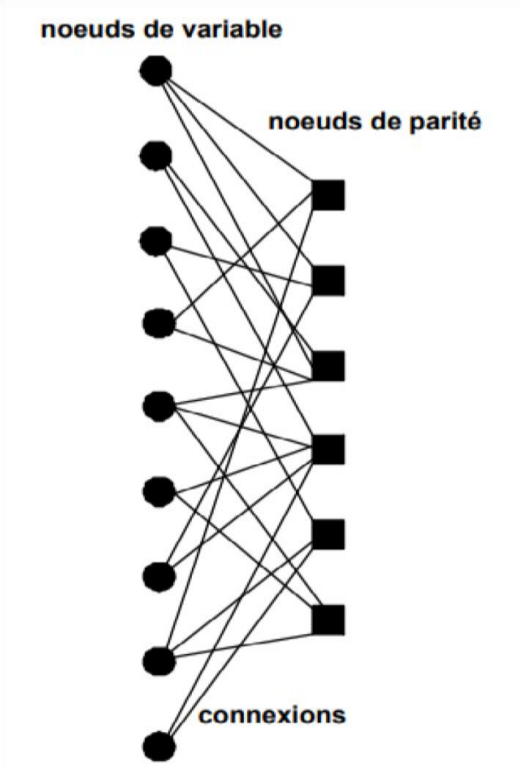

Figure 1. 6: Graphe bipartite d'un code LDPC irrégulier.

1.8.2 Codes convolutifs et Turbo-codes

1.8.2.1 Les codes convolutifs

Introduits en 1955 par Elias, forment une classe efficace de codes correcteurs d'erreurs. Ce sont les codes les plus utilisés dans le système de télécommunications fixes et mobiles. Théoriquement, ils ont les mêmes caractéristiques que les codes en blocs sauf pour la valeur de leur dimension et leur longueur. Les codes convolutifs s'appliquent sur une séquence innie de symboles d'information et génèrent une séquence innie de symboles codés [20].

Nous nous intéressons uniquement au cas binaire, un code convolutif C de longueur n et de dimension k possède une matrice génératrice de k lignes et n colonnes. Cette matrice génératrice permet de définir le code convolutif mais ne permet pas l'encodage.

Le codage peut s'effectuer de plusieurs façons, la plus courante étant celle utilisant des registres a décalage, mais il peut également être réalise en effectuant le produit de l'information par une matrice génératrice binaire infinie ou en utilisant un treillis de codage (autrement dit un graphe).

Un codeur convolutif binaire de longueur n, de dimension k, note (n, k) et d'ordre m est défini par $m + 1$ matrices binaires $G_0, \ldots, G_m$ de taille $k \times n$. Le code convolutif C est l'ensemble

des mots obtenus en effectuant le produit de l'information ***I*** par la matrice binaire infinie G [21] :

$$\mathcal{G} = \begin{pmatrix} \mathcal{G}_0 & \mathcal{G}_1 & \cdots & \mathcal{G}_m & & & \\ & \mathcal{G}_0 & \mathcal{G}_1 & \cdots & \mathcal{G}_m & & \\ & & \ddots & \ddots & \ddots & \ddots & \\ & & & \mathcal{G}_0 & \mathcal{G}_1 & \cdots & \mathcal{G}_m \\ & & & & \ddots & \ddots & \ddots \end{pmatrix}$$

Autrement dit, le mot de code **c** = (c_0 c1 . . .) est défini par **c** = **IG** = (I0 I1 . . .)**G.**

$$\begin{aligned} c_0 &= \mathrm{I}_0\mathcal{G}_0 \\ c_1 &= \mathrm{I}_0\mathcal{G}_1 + \mathrm{I}_1\mathcal{G}_0 \\ c_2 &= \mathrm{I}_0\mathcal{G}_2 + \mathrm{I}_1\mathcal{G}_1 + \mathrm{I}_2\mathcal{G}_0 \\ &\vdots \\ c_{i+m} &= \sum_{j=0}^{i} \mathrm{I}_{i+j}\mathcal{G}_{m-j} \end{aligned}$$

C_i (Le i^{éme} bloc de *n* bits du mot de code) dépend donc des *m* blocs de *k* bits [21].

Exemple :

Pour le codeur convolutif 2-mémoires 1/2-taux, [22] les générateurs sont g_0 = (1, 1, 1) et g_1 = (1, 0, 1).

$$G = \left(\begin{array}{cc|cc|cc|cc|cc|cc} 1 & 1 & 1 & 0 & 1 & 1 & & & & & & \\ & & 1 & 1 & 1 & 0 & 1 & 1 & & & & \\ & & & & 1 & 1 & 1 & 0 & 1 & 1 & & \\ & & & & & & 1 & 1 & 1 & 0 & 1 & 1 \\ & & & & & & & \ldots & & & & \end{array}\right)$$

Si I = (110100) alors v = I.G = (11, 01, 01, 00, 10, 11).

1.8.2.2. Turbo-code

Le principe du Turbo-code fut introduit par C. Berrou, A Glavieux et P. Thitimajshima, en 1993 [23]. Pour la première fois, un code correcteur d'erreurs fonctionnant à moins de 0.5 dB de la limite de Shannon fut démontré. Cette rupture technologique dans le domaine du codage de canal a tout d'abord surpris la communauté scientifique, mais les résultats ont été très rapidement confirmés. Un Turbo-code est caractérisé par ces codes constituants et la

fonction d'entrelacement. Tout d'abord, dans le cas des Turbo-codes parallèles, au moins l'un des codes Constituants doit être un code convolutif récursif. En pratique les codes constituants sont choisis identiques. La fonction d'entrelacement permet d'introduire une fonction d'aléa entre les décodeurs. Ainsi, plus l'effet de brassage sera important, plus les informations extrinsèques seront décarrelées, ce qui améliorera la qualité du décodage. Cette fonction a aussi un rôle important sur la propriété de distance minimale du code [24].

Les Turbo-codes ainsi présentés ont ouvert de nombreuses voies de recherche dans le domaine du codage de canal. On peut citer par exemple les Turbo-codes en bloc [25] ou les Turbo-codes séries [26]... Ce bouleversement dans la façon de concevoir un système de décodage a également permis de redécouvrir les travaux de Gallager sur les codes LDPC [27].

1.9. Conclusion :

Dans ce chapitre nous avons présenté le schéma général d'une transmission sans fil, puis rappelé quelques définitions de quelques types de codes comme le code binaire et le code linéaire avec des généralités sur leurs techniques de décodage, à la fin, on a expliqué comment on peut construire différents types de codes correcteurs, comme les codes en blocs, les codes convolutifs et les turbo-codes.

Chapitre 2 : Principe de code correcteur LDPC

2.1. Introduction

Dans les transmissions sans fils de nos jours, les codes correcteurs d'erreurs LDPC (low-density parity-check) sont considérés parmi les plus utilisés. Ils ont été inventés par Gallager mais n'ont pas été utilisés à l'époque, ils ont été oubliés pendant des années, avant d'être redécouverts en 1996 par MacKay. Par leur nom l'indique, ils sont caractérises par une matrice de parité H très creuse. Un code est dit LDPC, s'il admet une matrice de parité H qui possède très peu de coefficients non nuls.

Dans ce chapitre, nous rappelons quelques notions de base concernant les codes LDPC et leur représentation telle que leur codage et leur décodage.

2.2. Historique

Les codes LDPC font leur première apparition en 1962 par Gallager [27], [28], mais il a proposé juste une méthode générale pour construire des codes LDPC pseudo aléatoires ; les bons codes LDPC sont générés par ordinateur et leur décodage est très complexe dû au manque de structure. Ces codes ont été ignorés jusqu'à 1981 quand Tanner introduit les graphes pour décrire ces codes, et étend les opérations de vérification de parité vers des fonctions plus générales [29], Sa théorie a été aussi ignorée pour les prochaines 14 années Jusqu'au jour où quelques chercheurs en codage ont commencé à étudier les codes en graphes et le décodage itératif. Deux chercheurs, en 1995, McKay et Neal, ont fait une nouvelle classe de codes de blocs pour posséder plusieurs caractéristiques des nouveaux codes turbo[2].Ces codes de blocs étaient une redécouverte des codes LDPC développés par Gallager. Mackay présente des constructions de codes LDPC.

Il existe certain des nombres de chercheurs nouveaux pour des codes LDPC, y compris Luby, Mitzenmacher, Shokrollahi, Spielman, Richardson et Urbanke. Ils ont produit de nouveaux codes LDPC irréguliers en 2001 [30], qui surpassent facilement les meilleurs codes turbo, Ainsi que d'offrir certains avantages pratiques et une configuration sans doute plus propre pour les résultats théoriques. Aujourd'hui, il existe des techniques de conception pour les constructions des codes LDPC qui s'approchent de la capacité de Shannon à l'intérieur des centièmes de décibels [3],[31], les codes LDPC sont rapidement développés avec le temps et ont adopté aux plusieurs applications comme la radiodiffusion numérique par satellite et les normes de communication optique de longue distance et sont très susceptibles d'être adoptés dans la norme de réseau local sans fil IEEE [27].

2.3. Problématique des codes correcteurs d'erreurs

2.3.1 Utilisation :

Les codes correcteurs d'erreurs sont utilisés dans le système de transmission telle que les satellites, la téléphonie, les disques lasers, les téléviseurs haute définition, le traitement d'image et de la parole, cryptographie (signature, carte à puce), compression de données, etc.…

2.3.2 Origine des codes correcteurs d'erreurs :

C'est la théorie de l'information initiée par C. Shannon dans les années cinquante qui a était à l'origine de la création des codes correcteurs d'erreurs.

2.3.3 Problématique :

La problématique des codes correcteurs d'erreur est la suivante : un émetteur A envoie un message X_m à B (le récepteur), pendant la transmission de ce message il introduit des perturbations, affaiblissement, écho, bruit qui détériorent l'information émise et créent des erreurs dans le message. Il s'agit de trouver comment faire pour que B,

1- d'une part détecte s'il y a des erreurs,
2- d'autre part, si elles ne sont pas trop nombreuses, il faut les corriger.

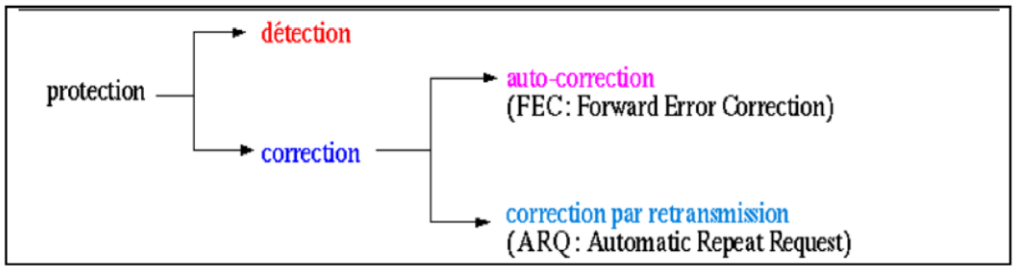

Figure 2. 1: Stratégies de protection contre les erreurs de transmission [32].

2.4. Concepts de base :

Le but principal de la communication est de transférer l'information de la source vers un ou plusieurs utilisateurs à travers un canal de transmission avec la plus grande fiabilité possible. Les communications numériques ne sont pas parfaites, quel que soit le canal utilisé, des erreurs peuvent se produire. Un codage à l'émission est introduit pour réduire ces erreurs.

2.4.1. Codes en bloc :

2.4.1.1. Définitions :

Le codage en bloc consiste à associe à un bloc de données m de k symboles issue de la source d'information à un bloc C qui est le mot de code de n symboles avec $n > k$ et $(n - k)$ représente les bits de parité.

Figure 2. 2: Schéma simplifié d'un codeur en bloc.

2.4.1.2. Matrice génératrice

Soit $c(n,k)$ un code en bloc linéaire et que $(g_1, g_2, \ldots\ldots g_k)$ soient des vecteurs linéairement indépendants. Chaque mot de code est une combinaison linéaire de ceux-ci :

$$c = m_1 . g_1 + m_2 . g_2 + \cdots + m_k . g_k \tag{2.1}$$

Sauf indication contraire, toutes les opérations vectorielles et matricielles sont modulo 2. Ces vecteurs linéairement indépendants peuvent être disposés dans une matrice appelée matrice génératrice G :

$$G = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_k \end{bmatrix} = \begin{bmatrix} g_{1,1} & g_{1,2} & \cdots & g_{1,n} \\ g_{2,1} & g_{2,2} & \cdots & g_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k,1} & g_{k,2} & \cdots & g_{k,n} \end{bmatrix} \tag{2.2}$$

Pour un vecteur de message donné $m = (m_1, m_2, \ldots . m_k)$, le mot de code correspondant est obtenu par multiplication matricielle :

$$c = m . G = (m_1, m_2, \ldots . . m_k) . \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_k \end{bmatrix} = m_1 . g_1 + m_2 . g_2 + \cdots + m_k . g_k \tag{2.3}$$

2.4.1.3. Matrice de contrôle de parité :

A chaque matrice génératrice de dimension$(k \times n)$, on associé une matrice H de dimension $((n-k) \times n)$ tels que les lignes G soient orthogonale à celle de matrice H.

La matrice H peut s'écrire :

$$H = [P^T \quad I_{n-k}] \tag{2.4}$$

C'est le double espace du code c, c'est-à-dire $G.H^T = 0$.

$$\mathrm{H} = \begin{bmatrix} \mathrm{h}_1 \\ \mathrm{h}_2 \\ \vdots \\ \mathrm{h}_{\mathrm{n-k}} \end{bmatrix} = \begin{bmatrix} \mathrm{h}_{1,1} & \mathrm{h}_{1,2} & \dots & \mathrm{h}_{1,\mathrm{n}} \\ \mathrm{h}_{2,1} & \mathrm{h}_{2,2} & \dots & \mathrm{h}_{2,\mathrm{n}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{h}_{\mathrm{n-k},1} & \mathrm{h}_{\mathrm{n-k},2} & \cdots & \mathrm{h}_{\mathrm{n-k,n}} \end{bmatrix} \tag{2.5}$$

On peut également vérifier que les équations de contrôle de parité peuvent être obtenues à partir de la matrice de contrôle de parité H, c'est-à-dire $G.H^T = 0$. Par conséquent, cette matrice spécifie également complètement un code de bloc donné.

2.4.1.4. Codes en bloc en forme systématique :

La structure du mot de code dans une forme systématique est représentée sur la Figure 2.3. Sous cette forme, un mot de code se compose de k bits d'information suivi de $(n-k)$ bits de parité.

k:bits d'information	(n-k):bits de parité

Figure 2. 3: Forme systématique d'un mot de code d'un code en bloc.

Ainsi, un code systématique en bloc linéaire (n, k) peut être défini par la matrice génératrice suivante :

$$G = \left[I_{k\times k} \quad p_{k\times(n-k)} \right] \tag{2.6}$$

Ainsi, un code de bloc linéaire systématique c (n, k) peut être spécifié par la matrice de générateur suivante :

$$G = \left[\begin{array}{cccc|ccc} 1 & 0 & \dots & 0 & p_{1,k+1} & p_{1,k+2} & \cdots & p_{1,n} \\ 0 & 1 & \dots & 0 & p_{2,k+1} & p_{2,k+2} & \cdots & p_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & p_{k,k+1} & p_{k,k+2} & \cdots & p_{k,n} \end{array}\right] \quad (2.7)$$

Identity matrix **Parity-check matrix**

$\boldsymbol{I_{k\times k}}$ $\boldsymbol{P_{k\times(n-k)}}$

Dans une notation compacte, est $G = [I_{k\times k}\ P_{k\times(n-k)}]$ .La matrice de contrôle de parité correspondante est donnée par $H = [P^T_{(n-k)\times k}\ I_{(n-k)\times(n-k)}]$. Cela veut dire que les bits d'information et les bits de redondances ne sont pas mélangés.

2.4.1.5. Décodage des codes en bloc linéaire :

Nous pouvons observer sur la figure 2.4 qu'à la suite de sa transmission à travers un canal bruité, un mot de code peut être reçu avec des erreurs. Le vecteur reçu peut donc être différent du mot de code émis correspondant, et il sera noté comme :

$Y = (Y_1, Y_2, Y_3 \dots\dots\dots\dots, Y_n)$

Un évènement d'erreur peut être modélisé comme un vecteur d'erreur ou un motif d'erreur :

$e = (e_1, e_2, \dots\dots\dots\dots, e_n)$

Où : $e = Y + c$

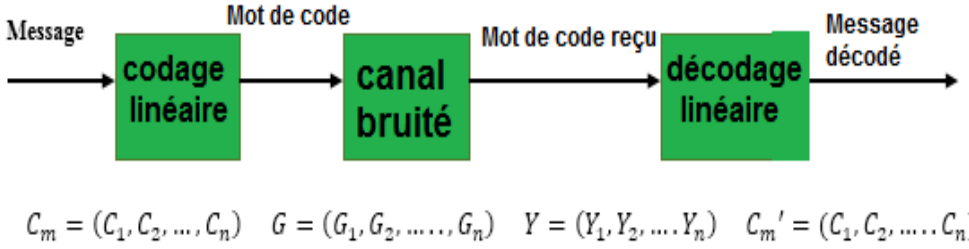

Figure 2.1: Diagramme d'un codeur en bloc.

Pour détecter les erreurs, on utilise le fait que tout mot de code valide doit obéir à la condition.

$$G \times H^T = 0$$

Un mécanisme de détection d'erreur est basé sur l'expression ci-dessus, qui adopte la forme suivante :

$$S = Y \times H^T$$

Où : $S = (S_1, S_2, \dots\dots, S_n)$ est appelé le vecteur de syndrome. L'opération de détection est effectuée sur le vecteur reçu :

- Si S (syndrome) est le vecteur nul, le vecteur reçu est un mot de code valide.

Sinon, il y a des erreurs dans le vecteur reçu. Il y a des méthodes qui est vérifié pour trouver le motif d'erreur correspondant e_m pour $m = 1,2,3, \dots\dots, n$, et le message décodé est obtenu par :

$$\widehat{C_m} = Y + e_m$$

2.5. Définition de code LDPC :

Le code LDPC $\mathrm{C(n,k)}$ est un code correcteur d'erreur de catégorie code block linéaire. Il est détermine par la matrice de parité [33]. Cette matrice est globalement binaire et de plus les lignes et les colonnes doivent être limitées pour que l'ensemble des matrices de parité reste des matrices de faible densité. Comme tous les codes de block linéaire, les mots de codes sont les vecteurs d'un sous espace de dimension k plongé dans un trajet vectoriel de dimension n. L'élément n se dénomme la longueur du code. Le rendement qui mesure la quantité d'information possédée dans un bit d'un mot de code définit par $\mathrm{R = k/n}$.

Premièrement, Gallager avait déterminé les codes LDPC par des contraintes plus fortes. En effet, il examine dans [27] que la matrice de parité doit contenir précisément les valeurs non nulles par ligne et les valeurs non nulles par colonne. Par conséquent, chaque élément du mot code associe à des équations de contrôle de parité et chacune de ces équations est composée de bits. Cet ensemble de codes LDPC est paramétré, et dont le rendement vérifie la matrice H peut être aussi représenté par un graphe bipartie connu par le graphe de Tanner [1].

Il y a deux caractéristiques évidentes pour les codes LDPC :

- **Contrôle de parité** : les codes LDPC sont représentés par une matrice de contrôle de parité H, où H est une matrice binaire.
- **Faible densité** : H est une matrice clairsemée (c'est-à-dire que le nombre de '1' est beaucoup plus faible que le nombre de '0'). C'est la faible densité de H qui garantit le faible taux de complexité [34].

Exemple : On considère la matrice de contrôle de parité suivante d'un code de rendement 1/2 et produisant 4 bits de redondance [35] :

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

2.5.1. Les avantages :

Les codes LDPC sont en grande compétition avec les codes turbo dans les systèmes de communication numériques qui demandent une fiabilité élevée. Aussi, les codes LDPC ont quelques avantages par rapport aux codes turbo :

- ils ne nécessitent pas d'entrelacer pour réaliser une bonne performance d'erreur.
- ils ont une meilleure performance par trame.
- leur plancher d'erreur se produit à un niveau de BER de beaucoup inférieur.
- leur décodage n'est pas basé sur un treillis et peut être réalisé par un processus parallèle.

2.6. Représentation du code LDPC

Le code LDPC peut être indiqué sous deux formes : la forme matricielle et la représentation graphique dit aussi le graphe de Tanner.

2.6.1 La représentation matricielle :

Le code LDPC peut être représenté par la matrice de parité $H(n, m)$ caractérisée par :
m équations de contrôle de parité qui doivent être satisfaites par les bits codés (nœuds de contrôle).

n : est le nombre de bits du mot-code (nœuds des variables ou de bit).

Les nœuds de variables (variable nodes) ou les bits nœuds w_c et les nœuds de contrôle (check nodes) w_r représentent le nombre les éléments non nul dans chaque ligne et colonne respectivement.

On peut noter, que pour une matrice de parité H de faible densité il faut que $w_c \ll n$ et $w_r \ll m$. Exemple d'une matrice de parité $H = (8,4)$:

$$H = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Dans cette matrice : $w_r = 4$ pour chaque ligne et $w_c = 2$ pour chaque colonne.

2.6.2 La représentation graphique (graphe de Tanner) :

On peut représenter la matrice de parité H sous forme d'un graphe bipartie dit le graphe de Tanner de la manière suivante [29] :

Les nœuds sont de deux types : les nœuds de variables et les nœuds de parités. Les nœuds de variables représentent les symboles du mot de code (code Word) et les nœuds de parités (check nodes) correspondent aux équations de parité. Un nœud de variable v_i est relié à un nœud de parité par c_j une branche, si et seulement si, l'élément h_{ij} de la matrice de contrôle de parité H est non nul($h_{ij} = 1$).

Par convention, les nœuds de variables seront représentés par des cercles et les nœuds de parité par des carrés. La représentation par un graphe d'un code LDPC nous permet aussi d'introduire l'idée de cycle. Un cycle existe dans un graphe dès lors qu'il y a un chemin pour quitter un nœud et revenir sans passer par les mêmes branches. Le nombre de branches traversées détermine la longueur du cycle. Un graphe sans cycle est appelé un arbre. Le graphe de Tanner ci-dessous représente la matrice précédente :

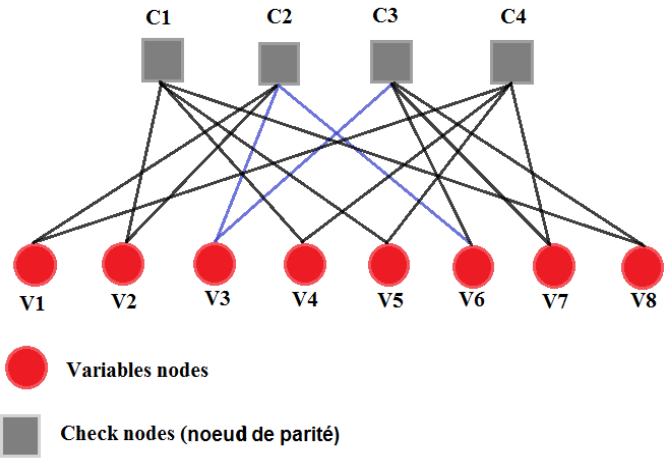

Figure 2. 5 : Le graphe de Tanner.

Un exemple de court cycle dans le graphe de Tanner de la Figure 2.5 :

$C_2 \rightarrow V_1 \rightarrow C_1 \rightarrow V_5 \rightarrow C_2$

Il doit éviter ce type de cycle puisqu'ils ont une mauvaise performance de décodage, par l'augmentation de la largeur de la matrice de contrôle de parité.

2.7. Les codes LDPC réguliers

En utilisant la représentation de Tanner, on détermine les nœuds de variable, les nœuds qui représentent les bits (information et redondance) du mot de code. On définit également les nœuds de parité, ceux représentant la contrainte placée sur les nœuds de variable.

Auxquels il est connecté, les originels codes LDPC proposés par Gallager dans ont une structure régulière. Les nœuds de variable et les nœuds de parité ont des degrés de connexion d_v (Respectivement) constants .Toutes les colonnes ont alors le même nombre de positions non nulles [36]. Cette condition est valable aussi pour les lignes. Le nombre total de positions non nulles dans la matrice est égal au nombre d'arêtes du graphe. On en dérive :

$$N.d_v = (N-K).d_c \Longrightarrow \frac{K}{N} = R = 1 - \frac{d_v}{d_c} \quad (2.8)$$

Avec le même couple $(\mathrm{d_v}, \mathrm{d_c})$, plusieurs codes réguliers peuvent être définis selon le choix des positions non nulles dans la matrice H.

Les conditions à remplir dans la construction de la matrice de contrôle de parité H d'un code LDPC régulier binaire sont :

- La matrice de contrôle de parité H correspondante doit avoir un poids de colonne fixe w_c.
- La matrice de contrôle de parité correspondante H doit avoir un poids de ligne fixe w_r.
- Le nombre de "l" entre deux colonnes n'est pas supérieur à 1.
- w_c et w_r doivent tous deux être de petits nombres comparés à la longueur de code n et au nombre de lignes de H.

Normalement, le taux de code des codes LDPC est $R = 1 - (\frac{w_c}{w_r})$.

Exemple : Code régulier :

$$H = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

$d_v = 2, d_c = 4$.

2.8. Les codes LDPC irréguliers :

Au lieu d'avoir des degrés de connexion fixes, les nœuds du graphe d'un code LDPC peuvent avoir des degrés de connexion différents [37], d'où l'appellation de « codes irréguliers ».

Luby et *al*. , donnent une extension de l'étude de Gallager sur des graphes irréguliers. Ils présentent que les performances des codes irréguliers sont meilleures et donnent une première approche de construction de codes irréguliers [38]. Cette approche a été développée plus tard pour prendre des performances proches de la limite de capacité de Shannon. La structure du code est définie à l'aide des deux polynômes $\boldsymbol{\lambda}(x)$ et $\boldsymbol{\rho}(x)$:

$$\boldsymbol{\lambda(x) = \sum di=v2\ \lambda i.\ xi-1} \qquad \boldsymbol{0 \le \lambda i \le 1} \qquad \boldsymbol{\sum di=v2\ \lambda i = 1} \tag{2.9}$$

$$\boldsymbol{\rho(x) = \sum di=c2\ \rho i.\ xi-1} \qquad \boldsymbol{0 \le \rho i \le 1} \qquad \boldsymbol{\sum id=c2\ \rho i = 1} \tag{2.10}$$

$\boldsymbol{\lambda_i}$ *et* $\boldsymbol{p_i}$ Sont les proportions des branches du graphe connectées à des nœuds de variable (Respectivement de parité) dont le degré de connexion est égal à **i**.

Soit t le nombre total d'arêtes dans le graphe, on note par vi (resp. ci) le nombre de nœuds de variable (resp. de parité) de degré i. Les égalités suivantes lient alors les paramètres du code à sa structure :

$$v_i = \frac{t - \lambda_i}{i} \qquad N = l.\sum_{i=2}^{d_c} \frac{\lambda_i}{i} = l.\int_1^0 \lambda(x)dx \tag{2.11}$$

$$C_i = \frac{t - p_i}{i} \qquad M = l.\sum_{i=2}^{d_c} \frac{p_i}{i} = l.\int_1^0 p(x)dx \tag{2.12}$$

$$R = 1 - \frac{\sum_{i=1}^{d_c} \frac{p_i}{i}}{\sum_{i=1}^{d_c} \frac{\lambda_i}{i}} \tag{2.13}$$

La figure (2.6) montre deux graphes de codes LDPC régulier et irrégulier :

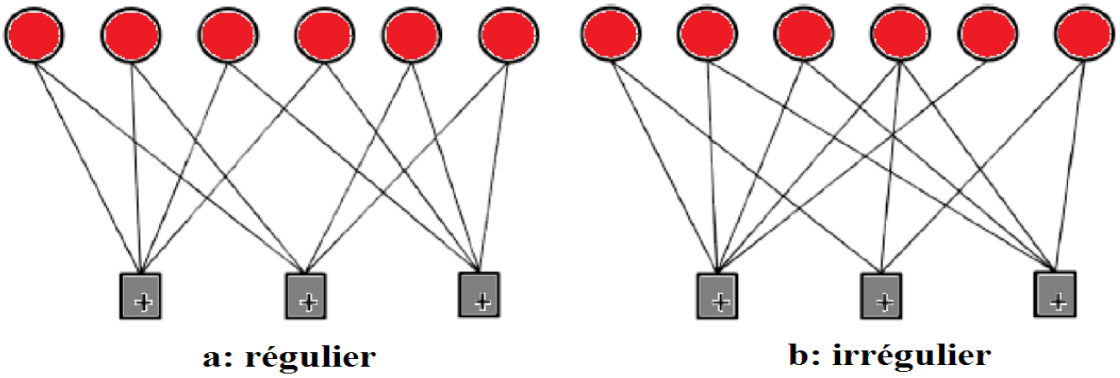

Figure 2. 6 : Graphe de Tanner de codes LDPC régulier et irrégulier

2.9. Construction des codes LDPC :

La construction d'un code LDPC binaires comporte à attribuer un petit nombre de valeurs dans une matrice de zéros à 1 de sorte que les lignes et les colonnes ont la distribution du degré requis.

La construction est basée sur différents critères de conception à mettre en œuvre pour un encodage et un décodage efficace, afin d'obtenir une capacité voisine de la capacité théorique. Plusieurs méthodes pour construire de bons codes LDPC peuvent être résumées en deux classes principales : constructions aléatoires et structurelles.

Pour des codes LDPC de grande taille une construction aléatoire est privilégier [38], [39] ainsi que pour des codes de petite ou de moyenne taille une construction structurée est adoptée. Cette dernière classe principalement deux méthodes existe : La première méthode est basée sur des géométries finies, tandis que la deuxième catégorie est basée sur des matrices de permutation circulantes. Dans ce travail c'est la deuxième méthode qui nous intéresse le plus.

2.9.1. Construction de Gallager

Les lignes de contrôle de parité des matrices de Gallager sont divisées en ensembles w_c avec $\frac{M}{w_r}$ lignes dans chaque série. Le premier ensemble de lignes contient w_r nombre de '1' consécutifs ordonnés de gauche à droite à travers les colonnes. (C'est-à-dire que pour $i \leq \frac{M}{w_r}$, la $i^{éme}$ ligne n'a pas d'entrée nulle de la $\left((i-1)+1\right)^{éme}$ jusqu'à la ${iw_r}^{éme}$colonne). Tout autre ensemble de lignes est une permutation de colonne adoptée au hasard de cet ensemble première. Par conséquent, toutes les colonnes de *H* comportent une '1' entrée une fois dans chacun des ensembles w_c .

Exemple : Une matrice de control de parité régulière (Gallager) M=12 $(w_c = 3, w_r = 4)$ est :

$$H = \left[\begin{array}{cccccccccccc} 1&1&1&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&1&1&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&1&1&1 \\ \hline 1&0&1&0&0&1&0&0&0&1&0&0 \\ 0&1&0&0&0&0&1&1&0&0&0&1 \\ 0&0&0&1&1&0&0&0&1&0&1&0 \\ \hline 1&0&0&1&0&0&1&0&0&1&0&0 \\ 0&1&0&0&0&1&0&1&0&0&1&0 \\ 0&0&1&0&1&0&0&0&1&0&0&1 \end{array}\right]$$

2.10. Conclusion

Chapitre 2 : Principe de code correcteur LDPC

Dans ce chapitre, nous avons présenté l'historique des codes LDPC et la problématique incitant la création des codes correcteurs d'erreurs puis on a rappelé quelques définitions et quelques avantages des LDPC par rapport aux turbo-codes. Ensuite, le détail de la méthode de codes LDPC ainsi que leurs deux types de représentations matricielle et graphique, ont été présentés. On a expliqué, aussi, comment on peut construire différents types de codes LDPC, comme les codes LDPC réguliers et les codes LDPC irréguliers.

Après, on a fait une brève description des méthodes de construction de codes LDPC telle que les méthodes de Gallager.

3.1. Introduction :

Dans les codes correcteurs d'erreurs, il existe plusieurs manières pour générer les matrices génératrices qui sont à l'origine de construction des codes LDPC réguliers, notre construction est basée sur une méthode systématique simple des codes LDPC en utilisant des vecteurs linéairement indépendants. Cependant, avant d'appliquer ce type de matrices systématique, nous allons appliquées la méthode de pivot de gauss pour donner une matrice génératrice systématique à partir d'une matrice génératrice G quelconque. Cette méthode est utilisée pour la première fois dans ce mémoire dans le cas binaire pour faciliter le décodage des codes LDPC.

Cette construction du code LDPC sera utilisée à la fin de ce chapitre afin de mettre en évidence notre algorithme de codage et décodage de données en générant et corrigeant des erreurs et en récupérant les données originales.

3.2. Principe de codage LDPC :

Pour encoder un message X_m on le multiplie par une matrice génératrice G, on obtient un mot de code C_m. Ce principe de LDPC on peut le représenté on deux forme, matricielle et graphe de Tanner.

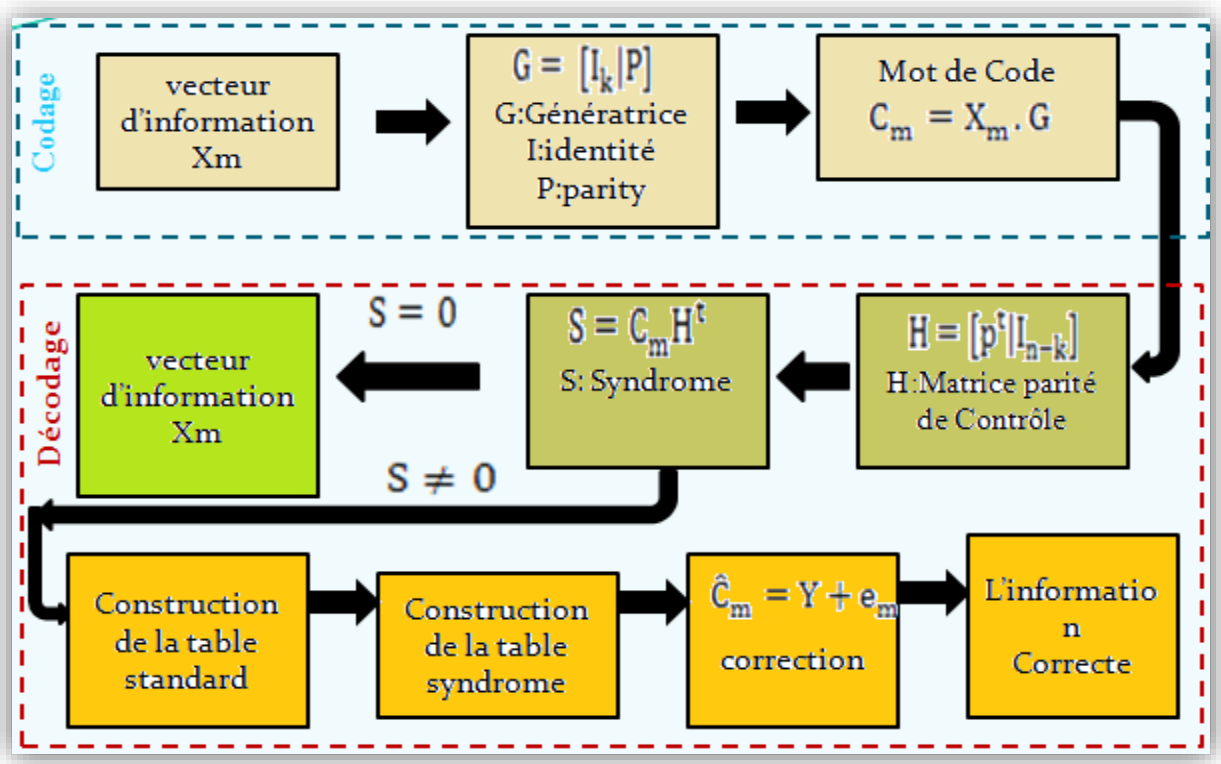

Figure 3. 1: Etapes d'un algorithme de codage et décodage LDPC.

3.3. Codage d'information

3.3.1. Vecteurs d'information et de code à 4 bits

Nous allons coder les bits d'information représentés par le vecteur X_m en un mot de code C_m. Ce vecteur d'information est donné par tous les cas possibles sur quatre bits :

$$Xm = \begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{matrix}$$

3.3.2. Matrice génératrice :

Le mot de code C_m associe à chaque bloc d'information une matrice G de taille $n \times k$ tels que :

$$C_m = X_m \times G \tag{3.1}$$

Les n éléments binaires formant un mot de code se répartit de la manière suivante :

k Bits d'information

$n - k$ Bits de contrôle

Nous allons utiliser la matrice identité de cette information de (4x4) qui est une matrice diagonale (I) ; on a dans cet exemple que les bits de la matrice de parité de 3 bits donnent la matrice génératrice :

$$\mathrm{G} = \mathrm{Xm}.\,\mathrm{Cm}^{-1} \tag{3.2}$$

La matrice G peut s'écrire aussi sous la forme réduite :

$$\mathrm{G} = [\mathrm{I} \quad \mathrm{P}] \tag{3.3}$$

Ou I : La matrice identité de dimension $k \times k$

P : La matrice de parité de dimension $k \times (n - k)$

On donne la matrice de parité et la matrice identité comme suit :

$$p_1 = I_1 + I_2 + I_3$$
$$p_2 = I_2 + I_3 + I_4$$
$$p_3 = I_1 + I_2 + I_4$$

$$I=\begin{matrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{matrix} \qquad p=\begin{matrix}1&0&1\\1&1&1\\1&1&0\\0&1&1\end{matrix}$$

Le résultat obtenu à partir de P et I est la matrice génératrice :

$$G = \begin{bmatrix}1&0&0&0&1&0&1\\0&1&0&0&1&1&1\\0&0&1&0&1&1&0\\0&0&0&1&0&1&1\end{bmatrix}$$

Le résultat obtenu de mot de code C_m après la multiplication de message d'information X_m et la matrice génératrice G.

$$C_m = \begin{bmatrix}0&0&0&0&0&0&0\\0&0&0&1&0&1&1\\0&0&1&0&1&1&0\\0&0&1&1&1&1&1\\0&1&0&0&1&1&1\\0&1&0&1&1&1&1\\0&1&1&0&1&1&1\\0&1&1&1&1&0&1\\1&0&0&0&1&0&1\\1&0&0&1&1&1&1\\1&0&1&0&1&1&1\\1&0&1&1&1&1&1\\1&1&0&0&1&1&1\\1&1&0&1&1&1&0\\1&1&1&0&0&1&1\\1&1&1&1&0&0&0\end{bmatrix}$$

3.4. Principe de décodage LDPC :

Par rapport aux autres types de codes, le décodage des codes LDPC ne pose pas autant de difficultés pour les chercheurs que leur construction. Le travail le plus difficile est de trouver les meilleures méthodes pour construire des codes LDPC efficaces.

La matrice de contrôle de parité

$$H = [p^t | I_{n-k}] \qquad (3.4)$$

Elle possède les propriétés suivantes :

$$C_m H^t = 0$$

$$GH^t = 0$$

3.5. Décodage d'information à 4 bits

3.5.1. Matrice de contrôle à 4 bits :

La matrice de contrôle de parité il constitué la matrice transposé de parité et la matrice identité de dimension $n - K = 7 - 4 = 3$

On donne la matrice transposé de parité et la matrice identité comme suit :

$$p^t = \begin{matrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{matrix} \qquad I_{n-k} = \begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$$

On a trouvé la matrice comme suit : $H = [p^t | I_{n-k}]$

$$H = \begin{matrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \end{matrix}$$

3.5.2. Le syndrome :

Pour un code *C*, le syndrome d'un mot $C_m \in F_2^n$ (vecteur reçu après transmission) est :

$$S = C_m * H^t \qquad (3.5)$$

Avec *H* la matrice de vérification de parité (matrice de contrôle) du code *C*.

D'après la figure 3.1 on détermine que :

- Si Le résultat obtenue après la multiplication de la transposé da la matrice de contrôle et le code C_m est égal à 0 donc le code C_m a été sans une erreur, car il donne une valeur de syndrome nulle :

$$S = 0\ 0\ 0$$

- Si le syndrome est différent de 0 c'est-à-dire que le code C_m a été avec une erreur donc on va faire la construction et la correction.

3.6. Méthode de correction des erreurs

Nous allons retrouvé une erreur dans le mot de code C_m donc on va faire une construction de l'erreur on utilisant :

$$S = Y \times H' \neq 0 \qquad (3.6)$$

Donc on va suivi les étapes suivants :

3.6.1. Construction de la table standard automatique et les vecteurs d'erreurs

Un tableau standard pour un code linéaire $[n, k]$est une table listant tous les vecteurs y pouvant être reçu (après transmission d'un mot de code). Le tableau est construit pour pouvoir lire pour chaque vecteur y le plus proche mot de code C_m.

Soit C_i , un des 2^k mots de code du code$[n, k]$.

Soit e_i , un des $2^{n-k} - 1$ vecteurs erreur.

Tableau 3. 1: La table standard et les vecteurs d'erreurs.

Erreur	**Mot de code**			
	$\mathbf{C_0 = 0}$	$\mathbf{C_1}$	…	$\mathbf{C_{2^k-1}}$
$\mathbf{e_0 = 0}$	$\mathbf{C_0 + e_0 = 0}$	$\mathbf{C_1 + e_0 = C_1}$	…	$\mathbf{C_{2^k-1} + e_0 = C_{2^k-1}}$
$\mathbf{e_1}$	$\mathbf{C_0 + e_1 = e_1}$	$\mathbf{C_1 + e_1}$	…	$\mathbf{C_{2^k-1} + e_1}$
…	…	…	…	…
$\mathbf{e_{2^{n-k}-1}}$	$\mathbf{C_0 + e_{2^{n-k}-1} = e_1}$	$\mathbf{C_1 + e_{2^{n-k}-1}}$	…	$\mathbf{C_{2^k-1} + e_{2^{n-k}-1}}$

3.6.1.1. Construction de la table standard de 8 bits :

Nous allons utiliser la ligne initiale qu'il possède le 2^4 mot de codes C_m possible en commençant par le mot dont tous les éléments sont des 0. Cette ligne représente donc les mots transmis sans erreur $e_i = 0$.

Et la première colonne, linéaire à 2^{8-4} élément 4 étant la dimension de S contient des vecteurs d'erreur $\{e_i\}$ choisi parmi les 2^8.

Pour la vérification des erreurs, nous calculons automatiquement la table syndromes.

3.6.1.2. Correction d'erreurs :

D'après l'équation :

$$\hat{C}_m = Y + e_m \tag{3.7}$$

et l'équation de syndrome :

$$S = Y.H^t \quad (3.8)$$

On a trouvé le tableau 3.2 :

Tableau 3. 2: La table standard de 8 bits.

L'erreur	00000000	00011111	00100110	00111101	01001111	01011101	01100100
Mot de code	00000001	00011110	00100111	00111100	01001110	01011100	01100101
	00000010	00011101	00100100	00111111	01001101	01011111	01100110
	00000100	00011011	00100010	00111001	01001011	01011001	01100000
	00001000	00010111	00101110	00110101	01000111	01010101	01101100
	00010000	00001111	00110110	00101101	01011111	01001101	01110100
	00100000	00111111	00000110	00011101	01101111	01111101	01000100
	01000000	01011111	01100110	01111101	00001111	00011101	00100100
	10000000	10011111	10100110	10111101	11001111	11011101	11100100

01111001	10000010	10011101	10100110	10111011	11001000	11010010
01111000	10000011	10011100	10100111	10111010	11001001	11010011
01111011	10000000	10011111	10100100	10111001	11001010	11010000
01111101	10000110	10011001	10100010	10111111	11001100	11010110
01110001	10001010	10010101	10101110	10110011	11000000	11011010
01101001	10010010	10001101	10110110	10101011	11011000	11000010
01011001	10100010	10111101	10000110	10011011	11101000	11110010
00111001	11000010	11011101	11100110	11111011	10001000	10010010
11111001	00000010	00011101	00100110	00111011	01001000	01010010

11101100	11111111
11101101	11111110
11101110	11111101
11101000	11111011
11100100	11110111
11111100	11101111
11001100	11011111
10101100	10111111
01101100	01111111

3.6.1.3. Table de syndrome de 8 bits :

Pour réaliser la table standard de 8 bits, on utilise les étapes suivantes :

- remplir la ligne listant l'ensemble des mots d'information et calculer la ligne des mots de code associes (via la matrice G).
- remplir la première ligne correspondant à une erreur nulle (recopie de la ligne précédente).
- génération des lignes :
  - ❖ déterminer un vecteur d'erreur de poids le plus petit $e_i \in F_2^n$ et n'appartenant pas aux lignes précédentes.
  - ❖ en déduire le syndrome associe $S_i = e_i H'$
  - ❖ remplir la ligne pour ce vecteur d'erreur (première ligne + erreur e_i)
- répéter jusqu'à avoir rempli 2^{n-k} lignes.

Tableau 3. 3: La table de syndrome et les vecteurs d'erreurs.

Syndrome s_i	**erreur**	**Mot de code**			
		$c_0 = 0$	c_1	…	c_{2^k-1}
0	$e_0 = 0$	$\mathbf{C_0 + e_0 = 0}$	$\mathbf{C_1 + e_0 = C_1}$	…	$\mathbf{C_{2^k-1} + e_0 = C_{2^k-1}}$
s_1	e_1	$\mathbf{C_0 + e_1 = e_1}$	$\mathbf{C_1 + e_1}$	…	$\mathbf{C_{2^k-1} + e_1}$
…	…	…	…	…	…
$s_{2^{n-k}-1}$	$e_{2^{n-k}-1}$	$\mathbf{C_0 + e_{2^{n-k}-1} = e_1}$	$\mathbf{C_1 + e_{2^{n-k}-1}}$	…	$\mathbf{C_{2^k-1} + e_{2^{n-k}-1}}$

Nous construisons automatiquement la table des syndromes en examinant pour chaque vecteur d'erreur l'existence ou la non-existence du syndrome correspondant dans la table. Ensuite, si ce syndrome existe, on le change par le syndrome suivant et son vecteur d'erreur correspondant.

Tableau 3. 4: La table de syndrome de 8 bits.

Syndrome	**Vecteur d'erreur**
000	00000000
001	00000001
010	00000010
100	00000100
011	00001000
110	00100000
111	01000000
101	10000000

3.7. Principe de codage par la méthode de pivot de gauss :

3.7.1. Méthode de pivot de Gauss :

Cette méthode permet d'associer à tout système linéaire un système équivalent plus facile. Elle consiste à sélectionner une équation qu'on va garder intacte, et dans laquelle on va rendre une inconnue facile (en l'éliminant des autres équations). Dans cette démarche, ce qu'on appelle le pivot, c'est la paire (équation, inconnue).

Pour résoudre un système, on applique une première fois la méthode au système donné, puis a une deuxième fois au système dérivé du système facile obtenu, et ainsi de suite, jusqu'à obtenir une équation impossible ou un système à une ou deux équations, qu'on sait résoudre.

3.7.2. Codage conventionnel basé sur l'élimination de Gauss-Jordan

L'algorithme de codage classique est basé sur l'élimination de Gauss-Jordan et la réorganisation des colonnes pour calculer le mot de code.

Semblable à la méthode générale de codage des codes en blocs linéaires, Neal a proposé un schéma simple .Pour un mot de code C donné et une matrice de contrôle de parité H irrégulière de taille $(m \times n)$, on partitionne le mot de code C en bits de message m, et des bits de contrôle de parité p.

$$c = [m \quad p] \tag{3.9}$$

Après l'élimination de Gauss-Jordan, la matrice de contrôle de parité H est convertie en forme systématique et ensuite divisé en une matrice A de taille $m \times (n - m)$sur la gauche et une matrice B de taille $m \times m$ sur la droite.

$$H = [A \quad B] \tag{3.10}$$

De la condition $C \times H^T = 0$, nous avons :

$$A.m^T + B.p^T = 0 \tag{3.11}$$

Par conséquent,

$$p^T = B^{-1}\, A.m^T \tag{3.12}$$

Donc, cette dernière équation peut être utilisée pour calculer les bits de contrôle sous la condition que B soit non singulière.

D'une manière générale, la matrice de contrôle de parité H ne sera pas une matrice sparce après le prétraitement.

Ainsi, la complexité des procédés classiques pour le codage de ce code LDPC est enlevée.

3.8. Algorithme de pivot de Gauss :

L'algorithme de pivot de Gauss, peut être résumé dans les étapes suivantes :

1. Résoudre le système d'équations linéaires.

2. Utiliser la méthode d'élimination de Gauss :

- m : numbre de ligne
- n : numbre de colonne
- pivotant
- Convertir les éléments sous la diagonale principale en zéros
- Retour de sous-estimation
- afficher le résultat

3. Utiliser la méthode de Gauss Jordan :

- m : numbre de ligne
- n : numbre de colonne
- pivotant
- Convertir les éléments sous la diagonale principale en zéros
- Convertir des éléments au-dessus de la diagonale majeure en zéros
- Convertir les éléments sur la diagonale majeure en un
- afficher le résultat

4. Cette méthode de pivot de Gauss va donner une matrice résultante qu'on la divisera en deux matrices : une matrice B diagonale et l'autre matrice A de parité.

5. La matrice H systématique est H=$[A \quad B]$ (3.13)

6. La matrice G=$[B \quad A^t]$ (3.14)

7. On vérifie à la fin que notre algorithme donne une matrice juste par le calcul de $bx = H * G$ qui représente la matrice des syndromes qui doit être nulle, si cette matrice est non nulle donc le résultat est faux.

3.8.1. Résultats d'algorithme pour rendre une matrice génératrice systématique :

Nous allons utilisé la matrice a :

$$a = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

On a appliqué dans la matrice de contrôle de parité H(12.6) dont le graphe de Tanner est le suivant :

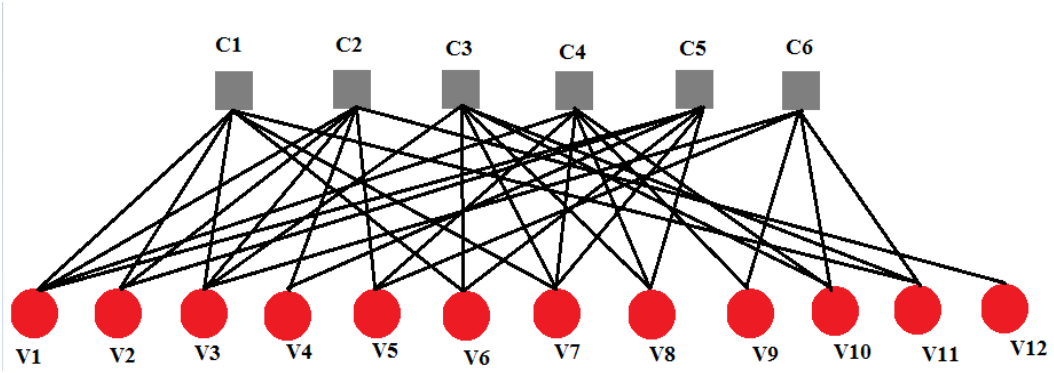

Figure 3.2 : Présentation de graphe de Tanner.

Les étapes suivantes représentent les résultats de la méthode d'élimination de Gauss puis de l'utilisation de la méthode Gauss-jordan :

:

$$a=\begin{bmatrix}1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0\\ 0 & -1 & -1 & 0 & 1 & -1 & 0 & 1 & 1 & 1 & -1 & 0\\ 0 & 0 & 0 & 1 & 1 & -1 & 1 & 0 & 0 & 0 & -1 & 1\\ 0 & -1 & 0 & 0 & 1 & 0 & 1 & 2 & 1 & 2 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & -1 & 2 & -3 & 0 & -2 & 0 & 0\end{bmatrix}$$

puis on aura :

$$a=\begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 2 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & -1 & 1\\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 2 & 1 & 1 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 0 & 1 & 0 & 0\end{bmatrix}$$

La matrice résultante systématique est donnée par la réalisation suivante :

$$a=\begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1\\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0\end{bmatrix}$$

Cette matrice est le résultat de conversion en binaire d'une matrice réelle, ce qui donne la partition en deux matrices suivante :

$$B=\begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 0 & 1\end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

La matrice de parité H systématique est donc, donnée par :

$$H = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

La matrice génératrice G sous forme systématique (par la méthode de pivot de Gauss) est donnée par :

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

La vérification l'efficacité de notre méthode donne une matrice nulle :

$$bx = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

3.8.2. Résultats de notre algorithme :

Nous allons utilisé une autre matrice pour vérifier l'efficacité de notre algorithme :

$$a = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Cette matrice de contrôle de parité H(12.6) est représenté par le graphe de Tanner suivant :

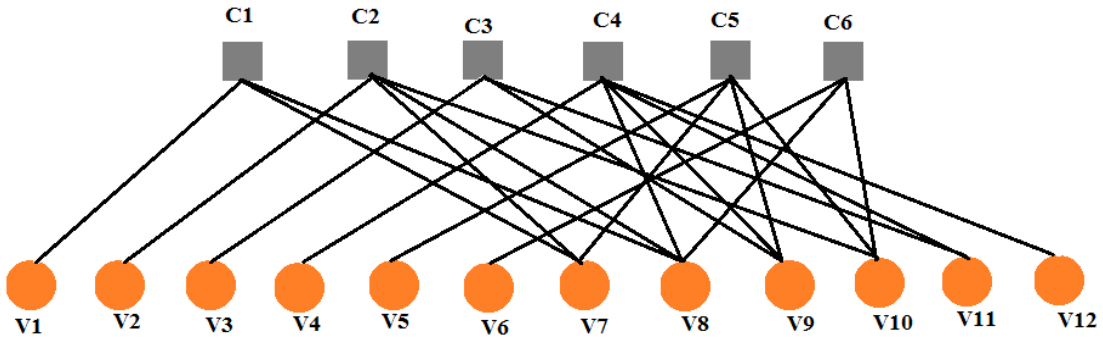

Figure 3.3 : Présentation de graphe de Tanner.

Les résultats de méthode d'élimination Gauss donne la matrice suivante :

$$a = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Alors que les résultats de la méthode de gauss jordan sont représentés par la matrice suivante :

$$a = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

La matrice de parité H systématique :

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

La matrice G sous forme systématique (par le pivot de Gauss) :

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

3.9. Conclusion

Dans ce chapitre on a appliqué le principe de codage et de décodage des codes correcteurs d'erreurs LDPC par la présentation d'un exemple d'information de taille de quatre bits avec la correction des erreurs dans l'étape de décodage. Nous avons ensuite, utilisé le principe de codage par la méthode de pivot de Gauss, qui facilite le décodage de l'information codée par la simplification de la matrice génératrice en une matrice systématique ce qui rend l'extraction de l'information rapide. Une vérification de ces résultats était réalisée afin de vérifier l'efficacité de l'algorithme.

4.1. Introduction :

Dans les codes LDPC il existe plusieurs manières de construction des matrices génératrices des codes, et parmi ces méthodes, on peut citer la méthode de Gallager. Dans ce chapitre nous allons appliquer deux méthodes de construction de codes, la première construction de la matrice de parité H est régulière alors que dans la deuxième nous appliquerons une construction d'une matrice irrégulière. Ces deux constructions vont être utilisées pour vérifier notre algorithme de codage et décodage LDPC pour des matrices de contrôle de parité plus complexes.

4.2. Construction de Gallager (régulière) :

Ce type de construction est caractérisé par les propriétés suivantes :

- Les codes de contrôle de parité de faible densité sont des codes spécifiés par une matrice de contrôle de parité H contenant principalement des 0 et seulement un petit nombre de 1.
- Un code LDPC régulier (n, wr, wc) est un code de longueur de bloc n avec une matrice de contrôle de parité $m * n$ où chaque colonne contient un petit nombre fixe, $wc >= 3$ de 1 et chaque ligne contient un petit nombre fixe $wr >= wc$, de 1 [40].
- En d'autres termes :
  - ✓ Chaque contrainte de contrôle de parité implique des bits de code w_r , et chaque bit de code est impliqué dans des contraintes w_c.
  - ✓ Faible densité implique que $wc \ll m \; et \; wr \ll n$.
  - ✓ Nombre de ceux dans la matrice de contrôle de parité $H = w_c * n = w_r * m$.
- Soit n la longueur de bloc transmise d'une séquence d'information de longueur k. m est le nombre des équations de contrôle de parité
- Construisez une matrice $m * n$ avec wc 1s par colonne et wr 1s par ligne (un code(n, wr, wc)).
- Diviser une matrice $m * n$ en sous-matrices $wc\ m/\ wc * n$, chacune contenant un seul 1 dans chaque colonne.

4.2.1. Principe de l'algorithme :

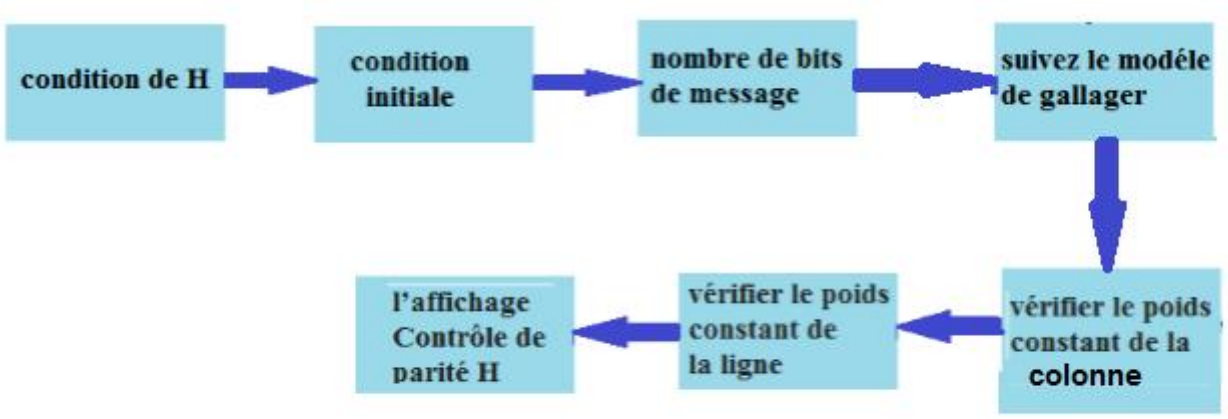

Figure 4.1 : Algorithme de construction régulière de Gallager.

4.2.2. Etapes de l'algorithme :

L'algorithme de construction est basé sur l'utilisation des étapes suivantes :

1. Conditions sur la matrice H :

- **a.** (n / nombre de ligne) > nombre de colonne
- **b.** nombre de colonne >= 3
- **c.** nombre de ligne > nombre de colonne

2. Conditions initiales, nous choisissons les conditions initiales suivantes :

- **a.** n=12
- **b.** nombre de ligne=4
- **c.** nombre de colonne=3

3. Nombre de bit de message :

k = (n / nombre de ligne) * nombre de colonne

4. Suivez le modèle de Gallager

5. Vérifier le poids constant de la colonne

6. Vérifier le poids constant de la ligne.

7. Affichage de la matrice de contrôle de parité H

4.2.3. Résultats de l'algorithme :

On a appliqué notre algorithme de construction de la matrice de contrôle de parité, et il a donné les résultats suivants :

La matrice de contrôle de parité est régulière et ayant un nombre de quatre uns par ligne, et elle a la structure ci-dessous :

$$H=\begin{bmatrix}1&1&1&1&0&0&0&0&0&0&0&0\\0&0&0&0&1&1&1&1&0&0&0&0\\0&0&0&0&0&0&0&0&1&1&1&1\\1&1&1&0&0&0&0&1&0&0&0&0\\0&0&0&0&1&1&1&0&0&0&0&1\\0&0&0&1&0&0&0&0&1&1&1&0\\0&1&0&0&0&0&0&0&1&0&1&1\\1&0&1&1&0&1&0&0&0&0&0&0\\0&0&0&0&1&0&1&1&0&1&0&0\end{bmatrix}$$

Le graphe de Tanner correspondant à la matrice de contrôle de parité H(12.9) est donnée dans la figure 4.2 ci-dessous:

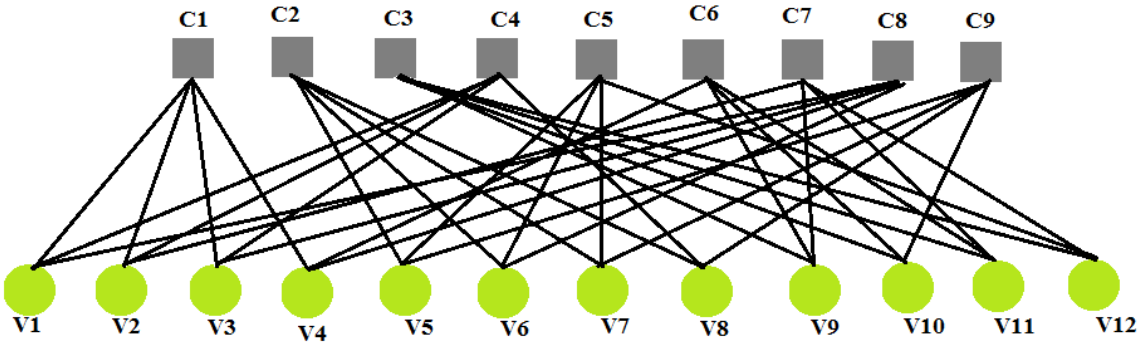

Figure 4.2 : Présentation de graphe de Tanner.

4.3. Construction basée sur une triangulation inférieure approximative :

La complexité des algorithmes de codage conventionnels est essentiellement proportionnelle au carré de la longueur du code et devient un problème important lorsqu'on traite de longueurs de code longues. Pour résoudre ce problème, Richardson et Urbanke ont proposé un algorithme de codage efficace pour les codes LDPC [41]. Nous donnerons une description détaillée de cet algorithme de codage dans ce qui suit. L'idée est de faire une transformation de la matrice de contrôle de parité en utilisant uniquement des permutations de lignes et de colonnes afin de garder H clairsemé. Toute matrice clairesemée arbitraire peut être convertie en la matrice de contrôle de parité souhaitée H avec une forme triangulaire inférieure approximative, comme le montre la Figure 4.3 :

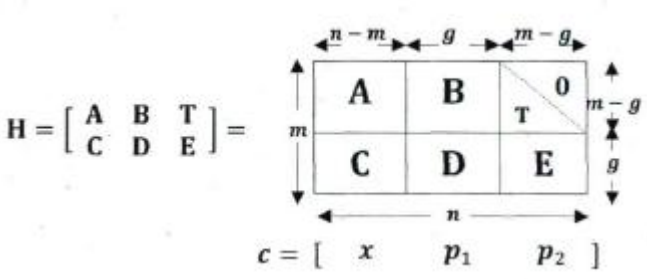

Figure 4.3 : Matrice de contrôle de parité H sous forme triangulaire inférieure approximative.

4.3.1. Algorithme de codage de Richairdson-Urbanke

Pour ce type de construction, on suit les étapes suivantes :

1. Effectuer la permutation des lignes et des colonnes pour amener H en une forme triangulaire inférieure approximative

$$H=\begin{bmatrix} A & B & T \\ C & D & E \end{bmatrix} \tag{4.1}$$

Où A est $(m-g)\times(n-m)$, B est $(m-g)*g, T\ est$ un $(m-g)\times(m-g)$ une matrice triangulaire inférieure, C est $g*(n-m)$, D est $g*g$ et finalement E est $g*(n-m)$. Les lignes g de H sont appelées l'intervalle de la représentation approximative, et plus g est petit, plus la complexité d'encodage des codes LDPC est faible.

2. Une fois le format triangulaire supérieur de T obtenu, on utilise l'élimination de Gauss pour effacer E qui équivaut à la pré-multiplication suivante :

$$\begin{bmatrix} I & 0 \\ -ET^{-1} & I \end{bmatrix}\begin{bmatrix} A & B & T \\ C & D & E \end{bmatrix}=\begin{bmatrix} A & B & T \\ -ET^{-1}A+C & -ET^{-1}B+D & 0 \end{bmatrix}=\begin{bmatrix} A & B & T \\ \hat{C} & \widehat{D} & 0 \end{bmatrix} \tag{4.2}$$

Où nous dénotons :

$$\hat{C}=-ET^{-1}A+C \tag{4.3}$$

$$\widehat{D}=-ET^{-1}B+D \tag{4.4}$$

3. Codage :

Considérons le mot de code c consistant en une partie systématique x et deux parties de parité $p1$ et $p2$, respectivement de longueur g et $(m-g)$. Comme le mot de code $c=[x\ p1\ p2]$ doit satisfaire l'équation de contrôle de parité $x^T=0^T$, nous avons

$$Ax^T+Bp_1^T+Tp_2^T=0 \tag{4.5}$$

$$\hat{C}x^T+\widehat{D}p_1^T+0p_2^T=\hat{C}x^T+\widehat{D}p_1^T=0 \tag{4.6}$$

Supposons que $\widehat{D}$ soit inversible, p_1 peut être trouvé à partir de (4.5) :

$$p_1^T=\widehat{-D^{-1}}\,\hat{C}x^T=\widehat{-D^{-1}}(-ET^{-1}\mathrm{A}+\mathrm{C})x^T \tag{4.7}$$

Où la faible densité de A, B et T peut être utilisée pour maintenir la complexité de cette opération à un niveau bas ; Puisque T est triangulaire supérieur, p_2 peut être trouvé en utilisant la substitution arrière.

$$p_2^T=-T^{-1}(Ax^T+\mathrm{B}p_1^T) \tag{4.8}$$

Ainsi, une fois que $g*(n-m)$ la matrice $\widehat{D^{-1}}\hat{C}x^T$ à été pré-calculée, la détermination de px peut être réalisée avec la complexité $O(g^2)$ simplement en effectuant une multiplication avec cette matrice, comme indiqué dans le tableau 4.1. La complexité correspondante de p_2 est O(n) comme indiqué dans le tableau 4.2 :

Tableau 4.1 : calcule efficace pour $p_1^T = \widehat{-D^{-1}}(-ET^{-1}\text{A+C})x^T$

Opération	Commentaire	Complexité
Ax^T	Multiplication pour matrice clairsemée	$O(n)$
$T^{-1}Ax^T$	$T^{-1}Ax^T = y^T \leftrightarrow Ax^T = Ty^T$	$O(n)$
$-ET^{-1}Ax^T$	Multiplication pour matrice clairsemée	$O(n)$
Cx^T	Multiplication pour matrice clairsemée	$O(n)$
$-ET^{-1}Ax^T+Cx^T$	addition	$O(n)$
$\widehat{-D}^{-1}(-ET^{-1}Ax^T + Cx^T)$	Multiplication pour dimension de matrice $g \times g$	$O(g^2)$

Tableau 4.2 : calcule efficace pour $p_2^T = -T^{-1}(Ax^T+\text{B}p_1^T)$

Opération	commentaire	complexité
Ax^T	multiplication par matrice clairsemée	$O(n)$
Bp_1^T	$T^{-1}Ax^T = y^T \leftrightarrow Ax^T = Ty^T$	$O(n)$
$Ax^T + Bp_1^T$	multiplication par matrice clairsemée	$O(n)$
$-T^{-1}(Ax^T + Bp_1^T)$	$-T^{-1}(Ax^T + Bp_1^T) = y^T$ $\leftrightarrow -(Ax^T + Bp_1^T) = Ty^T$	$O(n)$

Cette méthode est la plus utilisée pour coder les codes LDPC et L'avantage de ces codes est leur construction qui est faite de manière systématique qui diminue la complexité de l'encodage et réduit les besoins en mémoire. En revanche, sur la base de la méthode de Richardson-Urbanke, la matrice de contrôle de parité H est divisée en six sous-matrices illustrées à la figure 4.3.

4.3.2. Application de l'algorithme :

4.3.2.1. Premiers résultat :

Nous avons appliqué notre algorithme pour construire la matrice de contrôle de parité H, avec un code régulier de longueur 12, et nous obtenons la matrice H suivante :

$$H=\begin{bmatrix}1&1&1&0&0&1&1&0&0&0&1&0\\1&1&1&1&1&0&0&0&0&0&0&1\\0&0&0&0&0&1&1&1&0&1&1&1\\1&0&0&1&0&0&0&1&1&1&0&1\\0&1&0&1&1&0&1&1&1&0&0&0\\0&0&1&0&1&1&0&0&1&1&1&0\end{bmatrix}$$

Le graphe de Tanner qui correspond à la matrice de contrôle de parité H(12.6) précédente est le suivant :

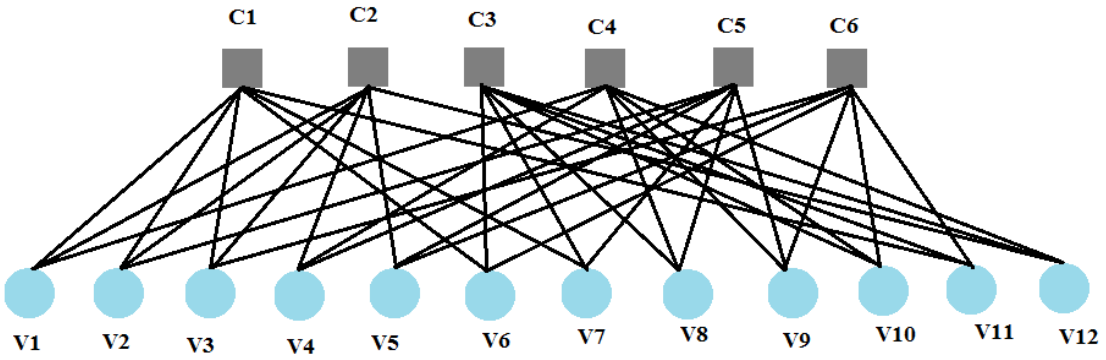

Figure 4.4. Présentation de graphe de Tanner.

Nous avons appliqué aussi, l'algorithme de construction de Richardson-Urbanke pour avoir en résultats la matrice de contrôle de parité (25,50) suivante :

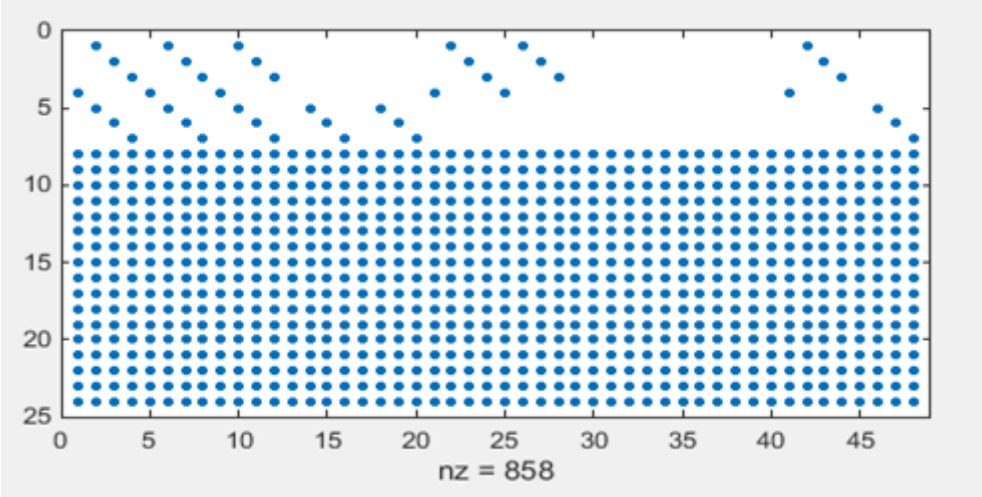

Figure 4.5. Construction de Richardson-Urbanke de H(25,50).

Dans cette figure 4.5, une représentation d'une matrice de contrôle de parité d'un code LDPC en respectant les principes de l'algorithme de Richardson-Urbanke. On a utilisé pour cela $z = 4$. Ce qui donne le résultat $n * z$=858.

Dans cette figure la matrice est de taille *25 x 50*, c'est-à-dire *25* lignes et *50* colonnes. Les valeurs non nulles sont représentées par des points en bleu.

4.3.2.2. Deuxième résultat :

On a appliqué la matrice H sous forme de cette figure :

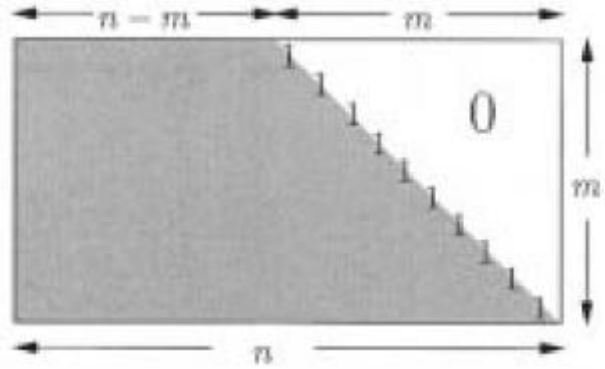

Figure 4.6 : Une matrice de contrôle de parité équivalente sous forme triangulaire inférieure.

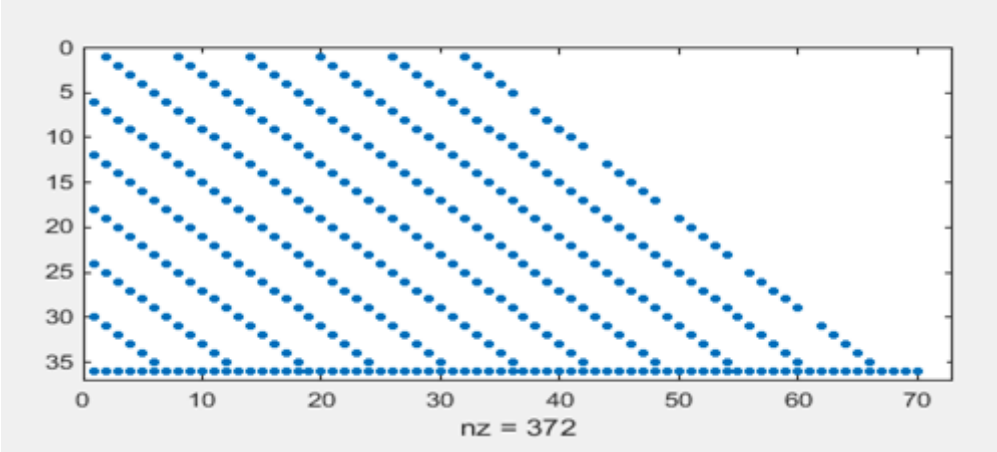

Figure 4.7 : Une matrice H sous forme triangulaire inférieure.

Dans cette figure on a appliqué une représentation d'une matrice de contrôle de parité d'un code LDPC sous forme presque triangulaire de la matrice de parité en respectant ces principes. On a met $z = 4$. Le résultat de $n * z = 372$

Dans cette figure la matrice est de taille *35 x 70*, c'est-à-dire *35* lignes et *70* colonnes. . Les valeurs non nulles sont représentées par des points en bleu.

4.3.2.3. Troisième résultat :

On a appliqué la matrice H sous forme de cette figure :

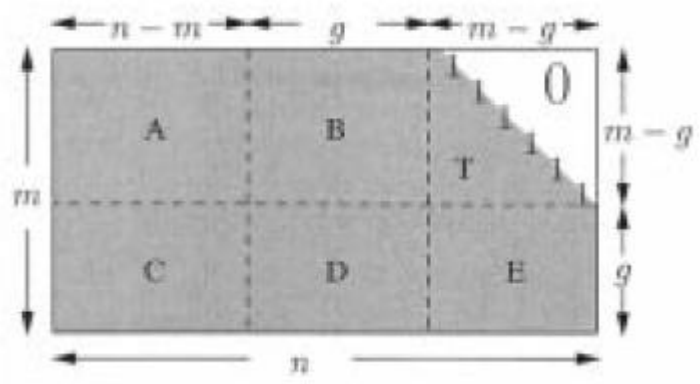

Figure 4.8. La matrice de contrôle de parité en forme triangulaire inférieure approximative.

On peut noter que cette transformation a été réalisée simplement par permutations. En particulier, on a réalisé la matrice de contrôle de parité suivante :

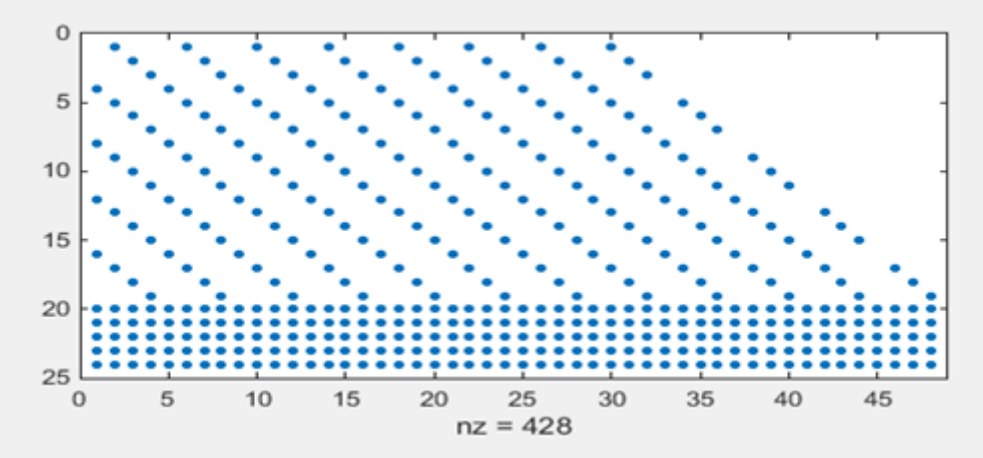

Figure 4.9 : Une matrice H sous forme triangulaire inférieure modifié par permutation.

Dans cette figure on a une représentation d'une matrice de contrôle de parité d'un code LDPC sous forme presque triangulaire de la matrice de parité en forme triangulaire inférieure approximative en respectant que $z = 4$. Ce qui donne le résultat $n * z = 428$.

Dans cette figure, la matrice est de taille *25 x 50*, c'est-à-dire *25* lignes et *50* colonnes. Les valeurs non nulles sont représentées par des points en bleu.

4.4. Conclusion :

Dans ce chapitre, nous avons appliqué plusieurs algorithmes de construction de la matrice de contrôle de parité H, les uns sont régulières et les autres irrégulières. Nous avons obtenu des matrices de contrôle de parité du code LDPC respectant les propriétés de ce type de codage (un petit nombre de uns, et des vecteurs de la matrice de contrôle de parité linéairement indépendantes).

Les résultats ont été affiché sous forme d'image représentant la densité de la matrice (petit nombre de points bleus dans les figures 4.5, 4.7 et 4.9) et une représentation par un graphe de Tanner.

Conclusion générale et perspectives

Les codes LDPC deviennent maintenant un sujet d'actualité dans le domaine des télécommunications sans fil, pour cela, et au terme de ce travail il nous est possible de dégager quelques conclusions sur les travaux menés dans ce mémoire.

La théorie des codes correcteurs d'erreurs est présentée dans l'ensemble des transmissions numériques de nos jours. Il est indispensable d'avoir des algorithmes de décodages efficaces, et les recherches actuellement vont vers une accélération et une amélioration des algorithmes existants. Nous avons présenté ici des notions importantes de la théorie des codes correcteurs d'erreurs et précisément des codes LDPC qui sont des codes correcteurs basés sur le décodage de l'information en blocs linéaires.

La plus importante caractérisation est que l'application des codes LDPC apporte une réelle amélioration des performances en matière de taux d'erreur binaire. Cependant, cela n'est pas sans inconvénients, puisque l'amélioration se fait au détriment d'une augmentation de la complexité des architectures des émetteurs et des récepteurs.

L'objectif du travail présenté dans le troisième chapitre est de représenter des codes de LDPC qui permettent de corriger des erreurs effacements grâce à des matrices de contrôle de parité très simple facilitant le décodage de l'information.

L'objectif visé dans le quatrième chapitre, est la construction de matrice de contrôle de parité plus complexe, La complexité du codeur/décodeur dépend du type de codage LDPC. Ce chapitre traite l'aspect mathématique du codage et du décodage pour des codes réguliers et irréguliers où nous avons présenté, dans ce cadre l'algorithme ainsi que les paramètres d'implémentation sous Matlab. Les résultats des simulations menées sont présentés et montre l'efficacité de la méthode.

❖ ***Perspectives***

La performance des codes correcteurs d'erreurs repose sur les algorithmes de codage et décodage établis, ce qui implique qu'il y a toujours des recherches et des améliorations dans ce domaine. Les futures améliorations pour notre projet sont :

- Rechercher d'autres algorithmes plus performants de codage/décodage, et tester diverses architectures.
- Utilisation des codes LDPC non binaire, qui présentent une meilleure performance que les codes LDPC binaire.
- L'association de différentes techniques au système étudié dans ce mémoire.
- L'association des codes LDPC à d'autres codes d'ordres plus élevés pour améliorer les performances.

Références bibliographiques

[1] R. Gallager, *Low Density Parity Check Codes*, IRE Trans. Inform. Theory, vol. IT-8, pp. 21-29, 1962.

[2] D. MacKay and R. M. Neal, *Near Shannon limit performance of low density parity-check codes*, Electronic Letter, August 1996.

[3] C. E. Shannon. *A mathematical theory of communication*. In from The Bell System Technical Journal, volume 27, July, October 1948.

[4] M.Candau, *Codes correcteurs d'erreurs convolutifs non-commutatifs*, THÈSE doctorat,

Sciences et Technologies de l'Information et de la Communication - Spécialité Communications Numériques École Doctorale, UNIVERSITÉ DE BRETAGNE OCCIDENTALE sous le sceau de l'Université européenne de Bretagne, 9 décembre 2014.

[5] H. JABER, *CONCEPTION ARCHITECTURALE HAUT DÉBI ET SÛRE DE FONCTIONNEMENT POUR LES CODES CORRECTEURS D'ERREURS*, THÈSE DE DOCTORAT, École Doctorale IAEM – Lorraine, Département de Formation Doctorale Électronique – Électrotechnique, l'Université Paul Verlaine – Metz.

[6] Ch. A. Berthelemot, *Performances des codes correcteurs d'erreur LDPC appliqués au lien Fronthaul optique haut-débit pour l'architecture C-RAN du réseau 5G : conception et implantation sur FPGA*, Thèse de Doctorat, École Doctorale Sciences et Ingénierie pour l'Information, Mathématiques. Université de Limoges, 18 décembre 2017.

[7] M. Cote, *Reconnaissance de codes correcteurs d'erreurs*, Thèse de Doctorat, Docteur de l´Ecole Polytechnique en Informatique.

[8] M.CLUZEAU, *Reconnaissance d'un schéma de codage*, Thèse de Doctorat, L'ÉCOLE POLYTECHNIQUE DOCTEUR EN SCIENCES, Spécialité Informatique, 28 novembre 2006.

[9] A. VALOMBOIS, *Décodage détection et reconnaissance des codes linéaires binaires*, Thèse de Doctorat, université de limoge en mathématique et informatique ,20 octobre 2000.

[10] F.J. M. Williams et N.J.A. Soane. *The Theory of error correcting codes*, North Holland publishing company, 1978.

[11] G. Bernault, Exploration *architecturale pour le décodage de Codes Polaires*, Thèse de Doctorat, l'École doctorale Sciences Physiques et de l'Ingénieur Spécialité Électronique, université de Bordeaux.

[12] G. LACHAUD, S. VLADUT, *Les codes correcteurs d'erreurs*, La Recherche, col. 26, n°278, pp. 778-782, juillet-août 1995.

[13] R. E. Blahut. *Algebraic Codes for data transmission*. MIT press Cambridge university press, 2003.

[14] Shu Lin est Daniel J. Costello. *Error Control Coding Fundamentals and application*. Pearson éducation Inc., 2004.

[15] N. Sendrier, *Reconnaissance de codes correcteurs d'erreurs*, Thèse de Doctorat, l´Ecole Polytechnique ,22 mars 2010.

[16] Ch. VANSTRACEELE, *Turbo Codes et estimation paramétrique pour les communications ‡ haut débit*, THESE DE DOCTORAT, L'école NORMALE SUPERIEURE DE CACHAN, Domaine : Electrotechnique-Electronique-Automatique (Spécialité Traitement du Signal), 26 janvier 2005

[17] V. HERBERT, *DIPLÔME DE DOCTORAT ÈS SCIENCES dans la spécialité INFORMATIQUE DES CODES CORRECTEURS*, Thèse de Doctorat, Université Paris 6, le 05 décembre 2011.

[18] F. LEHMANN, Les *Systèmes de Décodage Itératif et leurs Applications aux Modems Filaires et Non-filaires*, THESE de doctorat, de l'Ecole Doctorale Electronique, Electrotechnique, Automatique, télécommunications, Signal présenté, Spécialité : Signal, Image, Parole, Télécom ,10 décembre 2002.

[19] P. Elias, Error *Free Coding*, IRE Transaction on Inf. Theory, vol. IT-4, pp. 29–37, September 1954.

[20] H. Liu, *Contributions à la maîtrise de la consommation dans des turbo-décodeurs*, Thèse de doctorat, l'Université de Bretagne-Sud ,1 juillet 2009.

[21] A. Texier, Reconnaissance *de codes correcteurs*, thèse de doctorat de l'université PIERRE et MARIE CURIE.,14 octobre 2015

[22] M. Chaumont, *Codes Correcteurs d'Erreurs Les codes convolutifs binaires*, Novembre 12, 2008.

[23] C. Berrou, A. Glavieux, et P. Thitima-jshima, *Near Shannon limit error- correcting coding and decoding: turbo-codes*, IEEE International Conference on Communications, vol. 2, may 1993.

[24] C. Berrou, Y. Souter, C. Rouillard, S. Kerouedan, and M. Jezequel, *Designing good permutations for turbo codes: towards a single model*, IEEE Internatio- nal Conference on Communications, June 2004.

[25] R. Pyndiah, A. Glavieux, A. Picart, et S. Jacq, *Near-optimum decoding of products codes*, IEEE Global Conference on Communications, Nov 1994.

[26] S. Benedetto, D. Divsalar, G. Montorsi, et F. Pollara, *Serial concatenation of interleaved codes: Performance analysis, design, and iterative decoding*, IEEE Transactions on Information Theory, May 1998.

[27] R. G. Gallager, *Low-density parity-check codes*, Ph.D. dissertation, 1963.

[28] M. Bossert. *Channel Coding for telecommunications*. John Wiley & sons, 1999.

[29] R. Tanner, *A recursive approach to low complexity codes*, IEEE Transactions on Information Theory, vol. 27, sept 1981.

[30] M. Luby, M. Mitzenmacher, A. Shokrollahi, et D. Spielman, *Analysis of low density codes and improved designs using irregular graphs*, Proceeding of 30th ACM Symp. On Theory of Computing, 1998.

[31] M. Karkooti, *Semi-Parallel Architectures for Real-Time LDPC Coding*, Houston, TX: Rice University, 2004.

[32] : article de F. Jambon, La *prévention d'erreur par la technique des Fonctions de Contrainte*, Onzième conférence sur l'Interaction Homme-Machine (IHM'99), Montpellier, France, vol (40), 22-26 novembre 1999. p. 102-109.

[33] C. Berrou, *Codes et turbocodes* ; 1e édition, Springer - Verlag, France, 2007

[34] R. Yang, *LDPC-coded Modulation for Transmission over AWGN and Flat Rayleigh Fading Channels*.

Références Bibliographiques

[35] J-Baptiste Doré *Optimisation conjointe de codes LDPC et de leurs architectures de décodage et mise en oeuvre sur FPGA*. Traitement du signal et de l'image. INSA de Rennes, 2007.

[36] V. Anjitha et G.k. Sadanandan, *Low-Density Parity-Check (LDPC) decoder using low complexity min sum algorithm, ECE department Touch institute of science and technology, vol* 4, pp 529 ,2016

[37] S. Lin et D. Costello, *Error control coding*. Prentice Hall, 2004.

[38] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, et D. A. Spielman, *Improved low density parity check codes using irregular graphs*, IEEE Transactions on Information Theory, Vol. 47, pp. 585-598, Feb. 2001.

[39] T. Richardson, M. Shokrollahi et R. Urbanke, *Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes*, IEEE Trans. Inform. Theory, vol. 47(2), pp. 619-637, 2001.

[40] S. Myung, K. Yang and J. Kim, "Quasi-cyclic LDPC codes for fast encoding", IEEE Transactions on Information Theory, Vol. 51, no.8, pp. 2894- 2900, Aug. 2004.

[41] T. J. Richardson and R. L. Urbanke, "Efficient encoding of low-density parity-check codes", IEEE Transactions on Information Theory, Vol. 47, no. 2, pp. 638-656, Feb. 2001.