



HAL
open science

De l'usage des codes fontaines dans les réseaux de capteurs multisauts

Anya Apavatjrut

► **To cite this version:**

Anya Apavatjrut. De l'usage des codes fontaines dans les réseaux de capteurs multisauts. Autre. INSA de Lyon, 2011. Français. NNT : 2011ISAL0064 . tel-00708449

HAL Id: tel-00708449

<https://theses.hal.science/tel-00708449>

Submitted on 15 Jun 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

DE L'USAGE DES CODES FONTAINES DANS LES RÉSEAUX DE
CAPTEURS MULTISAUTS

présentée devant L'INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE LYON

pour obtenir
LE GRADE DE DOCTEUR

Ecole doctorale : INFORMATIQUE ET MATHÉMATIQUES

soumise le 26 mai 2011

Par
Anyà APAVATJRUT

Soutenue le 12 juillet 2011 devant la commission d'examen

JURY

DIOURIS Jean-François	<i>Professeur</i> Polytech'Nantes	Rapporteur
JULIEN-VERGONJANNE Anne	<i>Professeur</i> Université de Limoges	Rapporteur
HELARD Jean-François	<i>Professeur</i> INSA de Rennes	Examineur
GUERIN-LASSOUS Isabelle	<i>Professeur</i> Université de Lyon I	Examineur
GORCE Jean-Marie	<i>Professeur</i> INSA de Lyon	Directeur
GOURSAUD Claire	<i>Maître de conférence</i> INSA de Lyon	Co-Directeur
JAFFRES-RUNSER Katia	<i>Docteur</i> INSA de Lyon	Co-Encadrant

Les travaux présentés dans ce mémoire ont été réalisés au Centre d'Innovation en Télécommunications et Intégration de Services (CITI), INSA de Lyon, INRIA, Rhône-Alpes, sous la direction de Jean-Marie GORCE et Claire GOURSAUD, et financés par le gouvernement Thaïlandais

*À la mémoire de mon grand-père
À ma grand-mère et toute la famille APAVATJRUT*

Remerciements

C'est un plaisir de rédiger cette page de remerciement. Tout d'abord, je voudrais exprimer mes remerciements à Professeur Jean-Marie GORCE, mon directeur de thèse, qui m'a confié ce sujet de thèse et m'a orienté vers le monde de recherche pour son soutien tout le long de ce travail ainsi que pendant mon stage de mastère et Projet de Fin d'Etude(PFE). Je tiens à remercier Claire Goursaud pour son encadrement, son soutien et sa confiance par rapport aux différentes problématiques traitées durant mon doctorat.

La rédaction de ma thèse n'aurait pas été possible sans l'aide de Katia Jaffrès-Runser et Cédric Louradoux qui ont consacré leur temps en me prodiguant leurs conseils scientifiques. Je tiens à les remercier vivement pour leur patience, leur enthousiasme, et pour leurs corrections qui m'ont permis d'améliorer ce manuscrit.

J'ai eu plaisir de collaborer avec le laboratoire WNET à *Stevens Institute of Technologie* à New Jersey, Etats-Unis où j'ai passé six mois agréables. Je voudrais remercier Christina Comaniciu qui m'a accueillie pour son soutien et ses précieux conseils pendant mon séjour à Stevens.

J'aimerais remercier les rapporteurs de cette thèse, Professeur Jean-François DIOURIS, Professeur Anne JULIEN-VERGONJANNE, les examinateurs, Professeur Jean-François HELARD et Professeur Isabelle GUERIN-LASSOUS qui ont accepté de juger ce travail et m'ont donné des remarques constructives.

Une pensée à tous les membres du laboratoire CITI avec qui j'ai partagé une salle, un repas pendant ces années : Amira, Anis, Cédric, Hajer, Herve, Ibrahim, Ioan, Karel, Leila, Loïc, Ochir, Paul, Stéphane, Virgile, etc. Hors du travail, une pensée à mes amis Thaïlandais à Lyon.

Toute la famille APAVATJRUT ne se lasse pas de me prodiguer leur soutien et leur encouragement. Je voudrais exprimer ma gratitude à mes grand-parents, mes parents, mon oncle, mes tantes et toute ma famille notamment à mes deux tantes Pimchai et Supaporn pour leur affection, leur patience, leur compréhension depuis mon enfance.

Résumé

Important sujet de recherche dans les télécommunications contemporaines, les réseaux de capteurs sont des réseaux sans fil constitués de plusieurs nœuds pouvant communiquer entre eux. Chaque capteur est autonome et possède une durée de vie limitée, liée à la taille de sa batterie. Dans ce contexte, l'énergie est une ressource critique qui peut être économisée en minimisant le nombre de paquets émis. De part la nature du médium radio, les données transmises subissent des pertes induites par le bruit, l'évanouissement ou encore l'affaiblissement sur le canal. La fiabilisation de données dans ce contexte n'est pas simple et devient d'autant plus problématique lorsque la taille de réseau augmente.

C'est dans ce contexte que s'inscrit cette thèse qui vise à fournir une technique de fiabilisation des transmissions dans un réseau de capteur. Pour cela, nous proposons de mettre en œuvre un mécanisme de transmission qui exploite le *code fontaine*. Ce code est sans rendement et les symboles de redondance sont générés à la volée. Il permet de fiabiliser la transmission avec l'utilisation d'un canal de retour limité. Le code fontaine permet d'alléger le mécanisme de contrôle des transmissions tout en assurant un lien complètement fiable, ce qui permet de réduire la latence et la consommation énergétique d'une transmission.

Afin d'optimiser la performance globale du réseau, nous étudions également dans cette thèse le cas où les nœuds sont autorisés à coopérer pour le relayage multi-sauts de paquets destinés à des nœuds distants. Nous montrons dans cette thèse que la technique de codage réseau permet d'introduire de la diversité d'information et ainsi d'améliorer la performance globale de transmissions multi-sauts mono-chemin. Ce résultat a été étendu à la transmission coopérative pour laquelle nous avons à la fois pu exploiter la diversité d'information et la diversité spatiale.

De façon plus générale, cette thèse tend à exploiter l'avantage offert par le code fontaine et le codage réseau pour réduire le délai de transmission et la consommation énergétique dans les réseaux de capteurs sans fil, point critique pour le fonctionnement d'un tel réseau.

Mots clefs : réseaux de capteurs, codage réseau, codes fontaine, code LT

Abstract

This thesis is dedicated to the deployment of fountain codes and network coding in a wireless sensor network (WSN). A WSN is composed of sensor nodes with restricted capacities : memory, energy and computational power. The nodes are usually randomly scattered across the monitored area and the environment may vary. In the presence of fading, outage and node failures, fountain codes are a promising solution to guaranty reliability and improve transmission robustness.

The benefits of fountain codes are explored based on an event-driven WSN simulator considering realistic implementation based on standard IEEE802.15.4. Fountain codes are rateless and capable of adapting their rate to the channel on the fly using a limited feedback channel. In this thesis, we highlight the benefits brought by fountain code in terms of energy consumption and transmission delay.

In addition to the traditional transmission with fountain code, we propose in this thesis to study the network coding transmission scheme where nodes are allowed to process the information before forwarding it to their neighbors. By this means, we can say that packet diversity is exploited as each individual packet is unique and contains different representations of binary data. Redundancy is thus optimized since repetitions are avoided and replaced with diversified information. This can further lead to an overall improved performance in cooperative communication where nodes are allowed to assist in relaying packets from the source the destination.

We highlight in this thesis the benefits of fountain code combined to network coding and show that it leads to a reduction in transmission delay and energy consumption. The latter is vital to the life duration of any wireless sensor network.

Key words : Wireless Sensor Network, network coding, LT code, Fountain code

Table des matières

Remerciements	v
Résumé	vii
Abstract	ix
1 Introduction	1
2 Codes correcteurs pour le canal à effacements	3
2.1 Canal de communication	4
2.1.1 Notions de la théorie de l'information	4
2.1.2 Capacité du système de transmission	5
2.2 Code correcteur d'erreurs	6
2.2.1 Principes des codes correcteurs d'erreurs	6
2.2.2 Codes d'effacement	8
2.3 Codes d'effacement à rendement fixe	9
2.3.1 Code MDS	9
2.3.2 Code LDPC	10
2.4 Codes Fontaine : Code d'effacement sans rendement	11
2.4.1 Codes fontaine aléatoires	13
2.4.2 Code LT	15
2.4.3 Code Raptor	20
2.5 Variantes des codes fontaine	20
2.6 Domaine d'application des codes fontaine	21
2.7 conclusion	23
3 Codes Fontaines dans les Réseaux de Capteurs sans fil	25
3.1 Réseau de capteurs	27
3.1.1 Architecture d'un capteur	27
3.1.2 Types et topologies de réseau	27
3.1.3 Problématique des réseaux de capteurs	29
3.2 Codes fontaine pour les réseaux de capteurs sans fil	31
3.2.1 Etat de l'art	31
3.2.2 Motivation de notre étude	31
3.3 Modèles et Outils	33

3.3.1	Solutions Protocolaires	33
3.3.2	Implémentation du codage dans la pile protocolaire	35
3.3.3	Modélisation du lien radio	36
3.3.4	Paramètre de simulations	37
3.3.5	Critères d'évaluation de performance	39
3.4	Performance d'un code à faible dimension	41
3.4.1	Topologie Point-à-point	41
3.4.2	Topologie en étoile	47
3.4.3	Réseau linéaire	47
3.5	Acquittement et débordement	50
3.5.1	Modélisation	51
3.5.2	Surcharge d'acquittement	53
3.5.3	Résultats de simulation	53
3.6	Conclusion	56
4	Diversité d'information par le codage réseau avec code fontaine	59
4.1	Enjeux et problématique : code fontaine et codage réseau	61
4.1.1	Problématique	61
4.1.2	Etat de l'art	62
4.2	Étude de la distribution de degré obtenue après des combinaisons linéaires XOR	63
4.2.1	Une combinaison linéaire XOR	64
4.2.2	Expression analytique de la probabilité du degré du flux sortant	64
4.3	Comment XORer les paquets : approche théorique	71
4.3.1	Modélisation théorique	71
4.3.2	Pertinence du modèle théorique	74
4.4	Comment XORer les paquets : approche heuristique	77
4.4.1	Modèle algorithmique	77
4.4.2	Validation de l'algorithme de codage réseau	78
4.4.3	Optimisation d'algorithme	80
4.5	Codage réseau pour un système linéaire multisauts	82
4.5.1	Contexte d'étude	82
4.5.2	Scénarios d'études	82
4.5.3	Simulations et Résultats	84
4.6	Conclusion	85
5	Code fontaine pour la coopération	89
5.1	Communication coopérative	90
5.2	Code fontaine pour la coopération	91
5.2.1	Etat de l'art	91
5.2.2	Motivation	92
5.3	Codage réseau pour le canal à relai	92
5.3.1	Caractéristique d'un canal à relai	92
5.3.2	Modélisation de la transmission	93
5.3.3	Etude théorique	94

5.3.4	Approximation pour des régimes quasi-stationaires	99
5.3.5	Comparaison du modèle théorique aux simulations	101
5.3.6	Optimisation du débit de relaying	103
5.3.7	Etude par simulation	105
5.4	Code fontaine distribué pour un canal à relai	107
5.4.1	Contexte d'étude	107
5.4.2	Simulations et résultats	107
5.5	Code fontaine et codage réseau pour la diffusion dans un réseau de capteurs	109
5.5.1	Diffusion par gradient	109
5.5.2	Codage fontaine et codage réseau pour la diffusion par gradient	110
5.5.3	Simulations	111
5.5.4	Résultats	112
5.6	Conclusion	117
6	Conclusion et perspectives	119
6.1	Conclusion	119
6.2	Perspectives	121
	Bibliographie	123
	Liste de publications	135
	Liste d'acronymes	137
	Notations	139

Liste des tableaux

2.1	L'addition et la multiplication dans le corps \mathbb{F}_2	6
2.2	Coût d'encodage et décodage pour les différent types de codes	20
3.1	Paramètres de simulation	39
5.1	Les différents protocoles TDMA	93
5.2	Les séquences de transmissions lorsqu'on suppose que le lien S – R est parfait avec $\gamma_{SR} = 1$	94
5.3	Les séquences de transmission du relayage de paquets XORés entre deux paquets consécutivement envoyés par la source pour $\gamma_{S-R} \neq 1$	98
5.4	Une séquence de relayage des paquets XORés entre deux paquets consécutivement reçus par le relai pour $\gamma_{S-R} \neq 1$	99

Table des figures

2.1	Système de communication	4
2.2	Canal binaire symétrique	5
2.3	Canal binaire à effacement	5
2.4	Représentation du code en bloc	7
2.5	Réprésentations du code LDPC : graphe biparti et graphe de Tanner	11
2.6	Code Tornado	12
2.7	Analogie du code fontaine avec le remplissage d'un verre par une fontaine d'eau	12
2.8	Distribution binomiale	14
2.9	Représentation du code LT sur le graphe biparti	15
2.10	Distribution de Soliton idéale	17
2.11	Distribution de Soliton robuste	17
2.12	L'encodage LT	18
3.1	Un capteur et son architecture [5]	27
3.2	Réseau personnel	28
3.3	Réseau corporel	28
3.4	Environnement monitoré par des capteurs à grand échelle	28
3.5	La fiabilité de transmissions dans WSN [170]	30
3.6	Pile protocolaire du capteur	33
3.7	Implémentation de la couche de codage sur la pile protocolaire du capteur	36
3.8	La probabilité du lien en présence du bruit AWGN et évanouissement de Rayleigh par bloc	38
3.9	Mécanisme d'acquiescement : ARQs vs. Codes fontaine	43
3.10	La consommation d'énergie pour la transmission point-à-point, $K = 10$	44
3.11	La consommation d'énergie pour la transmission point-à-point, $K = 100$	45
3.12	La consommation d'énergie en fonction de la puissance d'émission, $K = 100$	46
3.13	Topologie en étoile	47
3.14	La consommation d'énergie pour la topologie en étoile en fonction du nombre de sources	47
3.15	Un réseau linéaire multisaut	48
3.16	La consommation d'énergie pour la transmission multisauts	49
3.17	Le délai de transmission pour la transmission multi-sauts	50
3.18	Une séquence de transmissions pour $n_h = 4$ et une progression $h = 1$.	52
3.19	Taille du débordement pour les stratégies CP et CPR-XA (simulation et modèle).	54

3.20	Ecart type du débordement pour les stratégies CP et CPR-XA	54
3.21	Le taux d'inondation du trafic mesuré par divers scénarios de transmissions (a)CB, (b)CP, (c)CH et (d)CPR-XA pour $n_h = 5, 10, 15, 30$ et $K = 5$	55
3.22	Le taux du trafic mesuré par divers scénarios de transmission pour $n_h = 10$ et $K = 5$	56
4.1	Trames de données	64
4.2	Probabilité qu'une combinaison XOR entre deux paquets conduise à la réduction du degré du paquet résultant	65
4.3	Évolution de degré après XORs pour le code RLF	67
4.4	Évolution de degré après XORs pour le code LT	67
4.5	Espérance du degré après e étape de XORs successives	68
4.6	Évolution de degré après e étapes de XORs successives pour le code RL	69
4.7	Évolution de degré après e étapes de XORs successives pour le code LT	70
4.8	Distribution de p_{choice} théorique	73
4.9	Réseau linéaire à 3 sauts	74
4.10	Distribution des degrés résultant pour l'approche théorique	75
4.11	Probabilité de répartition des fragments pour l'approche théorique	75
4.12	Nombre de paquets décodés en fonction du nombre de paquets reçus pour le cas sans XOR (courbes en bleu) en comparaison avec le cas avec XOR réalisé par l'approche théorique (courbes en rouges)	76
4.13	Nombre de fois le message est décodé avec succès en fonction du nombre de paquets reçus au décodeur	76
4.14	Distribution des degrés pour l'approche heuristique	78
4.15	Probabilité de répartition de fragments pour l'approche heuristique	78
4.16	Nombre de paquets décodés en fonction du nombre de paquets reçus pour le cas sans XOR (courbes en bleu) en comparaison avec le cas avec XOR réalisé par l'approche heuristique (courbes en verts)	79
4.17	Nombre de fois le message est décodé avec succès en fonction du nombre de paquets reçus au décodeur	79
4.18	Distribution des degrés pour l'approche heuristique avec optimisation	80
4.19	Probabilité de repartition de fragments pour l'approche heuristique avec optimi- sation	80
4.20	Évolution du nombre moyen de paquets requis pour récupérer la totalité des fragments d'information en fonction de la probabilité de XOR.	81
4.21	Réseau linéaire à n_h sauts	82
4.22	Performance du système pour le décodage par vraisemblance maximale (ML)	85
4.23	Performance du système pour le décodage par propagation de croyance (BP)	86
4.24	Distribution des degrés après 10 sauts pour divers stratégies de transmission : (a)CPR, (b)CPR-XT, (c)CPR-XH et (d)CPR-XH optimisé avec $\xi = 0.2$	87
5.1	Canal à relai	93
5.2	Les scénarios des cas possibles qui peuvent se produire pendant la transmission d'un paquet p_i	95

5.3	Résultats théoriques et par simulation pour les probabilités de succès avec relayage de XOR entre deux paquets consécutivement envoyés par la source (a) et consécutivement reçus par le relai (b) pour $\gamma_{SR} = 0.5, E = 10$	102
5.4	Probabilité de succès en fonction de γ_{RD} pour les algorithmes 6 et 7 (avec $\gamma_{SD} = 0.5$ et $\gamma_{SD} = 0.1$) : comparaison du modèle quasi-stationnaire et des simulations pour $E \in \{10, 100, 1000\}$	103
5.5	Nombre total de transmissions du système en fonction du taux de transmission pour le cas où on applique les opérations XOR entre deux paquets consécutivement envoyés par la source	104
5.6	Nombre de transmissions minimal et taux de transmission optimal en fonction de γ_{SD} ($\gamma_{SR} = \gamma_{RD} = 0.5$). Les trois courbes représentent le cas sans codage réseau (Sans XOR), le cas où on XOR les deux paquets émis par la source (Algo 6) et le cas où on XOR les deux paquets reçus par le relai (Algo 7).	105
5.7	Nombre de transmission minimal et taux de transmission optimal en fonction de γ_{SD} ($\gamma_{SR} = \gamma_{RD} = 1$). Les trois courbes représentent le cas sans codage réseau (Sans XOR), le cas où on XOR les deux paquets émis par la source (Algo 6) et le cas où on XOR les deux paquets reçus par le relai (Algo 7).	106
5.8	Nombre total de transmissions N_{tx} pour différents scénarios de relayage d'un code fontaine. ($\gamma_{SR} = \gamma_{RD} = 0.5$)	108
5.9	Nombre total de transmissions N_{tx} pour différents scénarios de relayage d'un code fontaine. ($\gamma_{SR} = \gamma_{RD} = 1.0$)	108
5.10	Positions et coûts associés aux capteurs par un algorithme à diffusion par gradient	110
5.11	Nombre de transmissions par capteur pour une communication entre S et D . . .	110
5.12	Taux de succès de transmission	113
5.13	La consommation énergétique du réseau	114
5.14	Le délai de transmission	115
5.15	Redondance en fonction de la probabilité de XOR et du taux de relayage	117

1

Introduction

Important sujet de recherche dans le domaine contemporain des télécommunications sans fil, les réseaux de capteurs sont constitués de plusieurs terminaux, appelés nœuds, pouvant communiquer entre eux. Chaque capteur est autonome et possède une durée de vie limitée, liée à la taille de sa batterie. Dans ce contexte, l'énergie est une ressource critique qui peut être économisée en minimisant le nombre de transmissions à effectuer dans le réseau. Cependant, cette utilisation parcimonieuse de l'énergie ne doit pas se faire au détriment de la fiabilité des communications, tout particulièrement dans le cas d'applications critiques de surveillance ou de contrôle. Pourtant, l'information transmise par paquets successifs est soumise à des erreurs de transmission sur le canal radio liées au canal lui-même (affaiblissement, évanouissements) ou à la coexistence de transmissions multiples (interférences, collisions). La fiabilisation des transmissions se fait classiquement par l'utilisation de mécanismes d'acquittements et de retransmissions, mais ces mécanismes coûtent cher en énergie lorsque la taille du réseau est grande et en particulier lorsque de nombreuses communications multi-sauts sont effectuées simultanément.

C'est dans ce contexte que s'inscrit cette thèse qui vise à exploiter les codes fontaine et le codage réseau pour fiabiliser les transmissions dans les réseaux de capteurs tout en limitant le surcoût énergétique. L'intérêt du code fontaine est qu'il s'agit d'un code sans rendement prédéfini, c'est-à-dire que les symboles de redondance sont générés à la volée jusqu'à ce que toute l'information soit reçue au(x) destinataire(s). Ainsi, le code fontaine permet potentiellement de fiabiliser la transmission avec une utilisation limitée du canal de retour. C'est ce principe que nous étudions dans cette thèse, où nous analysons différents scénarios permettant un compromis entre énergie, latence et fiabilité dans les communications multi-sauts. Nous montrons que, malgré le surplus de transmission lié à la redondance du code, le code fontaine peut permettre de réduire la latence ainsi que la consommation d'énergie.

Dans le chapitre 2, nous introduisons les notions fondamentales de théorie des codes, et

présentons les propriétés spécifiques des codes fontaine ainsi que leurs principales applications pour faciliter la compréhension du reste de la thèse.

Nous considérons ensuite dans le chapitre 3 la problématique de l'utilisation des codes fontaine dans les réseaux de capteurs sans fil pour une transmission sur une liaison directe. Dans ce cas, le bénéfice principal des codes fontaine est de réduire l'utilisation du canal de retour. Nous montrons que cette technique s'avère efficace même pour une taille de paquet relativement petite.

Puis, afin d'optimiser les performances globales du réseau, nous étudions le cas où les nœuds ont la possibilité de coopérer en servant de relais (simple réémission) pour les nœuds plus lointains. Nous associons au code fontaine un principe de réencodage de type codage en réseau, et nous montrons comment cela permet d'apporter de la diversité sur l'information transmise de proche en proche et ainsi permet d'améliorer la performance globale de transmission. Nous considérons ainsi le cas où les nœuds ont la possibilité de réaliser des opérations XOR sur les paquets reçus avant de les réémettre. Cette opération modifie une propriété fondamentale des codes fontaines : la distribution des degrés. Ainsi, dans le chapitre 4, nous évaluons la distribution des degrés des paquets obtenus après XOR, et proposons des algorithmes de sélection des paquets à XORer permettant de conserver la distribution souhaitée.

Enfin, dans le chapitre 5, nous étudions l'impact des combinaisons XOR sur la quantité d'information reçue au destinataire. Ce gain peut être étendu pour optimiser la diversité spatiale dans un schéma de transmission coopérative. Nous évaluons théoriquement cet impact pour un lien multi-saut. Enfin, nous l'étendons par simulation à un réseau de plus grande échelle, dans lequel le routage se fait par gradient.

Les résultats présentés dans ce manuscrit sont le fruit de mes trois ans et demi de thèse, de novembre 2007 à juin 2011, à l'Institut National des Sciences Appliquées (INSA) de Lyon au Centre d'Innovation en Télécommunications et Intégration de service (CITI). Ces travaux de recherche ont fait partie du projet *Body Area Networks and Technologies* (BANET) et font l'objet de plusieurs publications dans des journaux [11, 15], des conférences internationales [16, 12, 112] et des conférences nationales [10, 21, 13]. Les travaux les plus récents, présentés dans le chapitre 5 sont en cours de soumission.

2

Codes correcteurs pour le canal à effacements

Le point de départ de la théorie de l'information a été introduit par Claude E. Shannon dans son article [142] de 1948. Shannon a défini l'information comme une grandeur observable et mesurable. Il a démontré que pour n'importe quel canal, il est possible de transmettre l'information avec un taux d'erreur arbitrairement proche de zéro, tant que le débit du code est inférieur à un seuil appelé *capacité de canal*. A part des travaux de Shannon qui focalisent principalement sur la théorie d'information pour une communication point-à-point, El Gamal et al. propose quelques notions de la théorie d'information pour un système de transmission formé en réseau avec les caractéristiques de la communication plus réalistes. En effet, dans un réseau constitué de plusieurs nœuds communicants, il est important de considérer les effets provenant des transmissions multiples comme la variance de canal, la redondance, la corrélation de canal, les interférences, etc. La borne de capacité pour plusieurs modèles de communications a été proposés par exemple, un système à multi-sources, multi-destinataires, système à relai, etc [52, 42]. Malheureusement, la borne de capacité théorique d'un système de communication est en pratique difficile à obtenir. Ces preuves ne donnent pas les moyens pratiques d'atteindre cette limite. Depuis les années 1960, de nombreux travaux de recherche ont tenté d'approcher cette borne de capacité.

Une des solutions intéressantes qui permettent d'approcher cette borne est d'utiliser les codes correcteurs d'erreurs. Pour rendre la transmission résiliente aux erreurs, les codes correcteurs d'erreurs introduisent de la redondance dans les messages. Les erreurs peuvent provenir du bruit, des interférences ou des évanouissements du canal. Les codes correcteurs d'erreurs sont principalement employés dans le domaine des télécommunications mais ils sont également utilisés pour le stockage de données enregistrées pour lutter contre la détérioration du support. Un code correcteur d'erreurs peut être appliqué au niveau de chaque bit de données afin de détecter, voir de corriger les bits erronés. Il peut être également appliqué au niveau des paquets

pour corriger l'effacement de certains d'entre eux et protéger l'ensemble de données transmises.

Dans ce chapitre, nous aborderons les notions fondamentales liées aux codes correcteurs d'erreurs plus particulièrement aux techniques liées au canal à effacements. Nous focaliserons sur les codes fontaines, leurs principes et leur applications. Nous discuterons des principales catégories du code fontaine et nous comparerons les avantages et les inconvénients de ces codes ce qui nous permettra de justifier le choix des codes qui sont les plus adaptés aux réseaux de capteurs.

2.1 Canal de communication

2.1.1 Notions de la théorie de l'information

Un système de communication se représente communément par la Figure 2.1. Nous définissons par \mathcal{M} l'information numérique formée d'une suite de symboles que l'on appelle *message*. Ces symboles prennent leurs valeurs dans l'alphabet \mathcal{A} à q éléments. Nous parlons de message binaire lorsque les symboles d'information appartiennent à l'alphabet $\mathcal{A} = \{0, 1\}$ ($q = 2$). Notre but est de communiquer ce message \mathcal{M} sur le canal de transmission. N'étant pas parfait, le canal est perturbé par du bruit, de l'affaiblissement, etc. Cette perturbation entraîne la perte et/ou introduit des erreurs dans les messages transmis.

A l'entrée du canal se trouve un *encodeur* qui encode le message avant de l'envoyer sur le canal. L'encodeur a pour but de tenter de protéger les données contre les erreurs en rajoutant de la redondance comme protection (codage canal). Il convertit le message \mathcal{M} en un *mot de code* X . A la sortie du canal, ce mot code est modifié par le bruit et il devient Y . Un *décodeur* cherche ensuite à reconstruire $\hat{\mathcal{M}}$ l'information la plus proche possible du message \mathcal{M} à partir de Y .

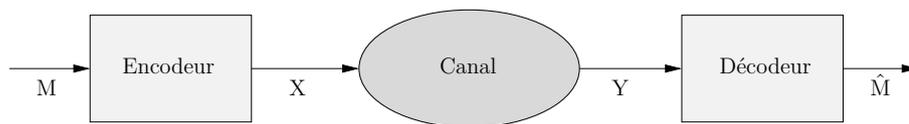


FIGURE 2.1 – Système de communication

Nous introduisons dans cette partie les différentes notions de codage et de la théorie de l'information. Ces définitions peuvent être retrouvées en détail dans [35, 104].

Definition 2.1.1.1 (Canal de communication). *On peut définir un canal de transmission par sa loi de transition qui est la probabilité d'obtenir Y à la sortie du canal sachant que l'entrée était X . Cette probabilité conditionnelle est notée $P(Y|X)$.*

Definition 2.1.1.2 (Entropie). *La fonction entropie $H(X)$ permet de quantifier l'incertitude de la variable X à valeurs dans $\{x_i | i = 0, \dots, N\}$. Supposons que la probabilité de masse d'une variable aléatoire x_i s'écrit $p(x_i)$. L'entropie $H(X)$ peut être exprimé comme :*

$$H(X) = - \sum_{i=1}^N p(x_i) \log_2 p(x_i) \quad (2.1)$$

Definition 2.1.1.3 (Information mutuelle). *L'information mutuelle $I(X, Y)$ de deux variables aléatoires X et Y mesure la dépendance statistique entre ces deux variables. Elle s'écrit :*

$$I(X, Y) = H(X) - H(X|Y) \quad (2.2)$$

Dans un canal parfait sans erreur, $I(X, Y) = H(X)$. Dans le pire cas pour un canal très dégradé $I(X, Y) \approx 0$.

Definition 2.1.1.4 (Capacité de canal). *La capacité de canal C se définit comme la quantité d'information maximale que l'on peut transmettre sur le canal.*

$$C = \max_X(I(X, Y)) \quad (2.3)$$

2.1.2 Capacité du système de transmission

Plusieurs modèles de canaux ont été définis. Pour chaque modèle, il est possible de déterminer la capacité du système.

Canal binaire symétrique

Nous considérons dans la Figure 2.2 un canal de transmission binaire symétrique ou *Binary Symmetric Channel* (BSC). Ce canal transmet des symboles binaires $\mathcal{A} = \{0, 1\}$. La probabilité qu'un symbole ne soit pas modifié par le canal est de $1 - p_e$. Elle est égale à p_e dans le cas contraire. La loi de transition est symétrique et les probabilités d'erreurs sur le bit 0 et 1 sont identiques. La capacité du canal est égale à $1 - H(p_e)$ avec $H(p_e)$ l'entropie du canal BSC en fonction de p_e .

Canal binaire à effacement

Introduit par Elias en 1995 [43], le canal binaire à effacement ou *Binary Erasure Channel* (BEC) représenté à la Figure 2.3 prend en entrée un symbole binaire et le transforme en un symbole qui appartient à $\{0, 1, e\}$ où e représente l'effacement qui se produit avec la probabilité p_e . Chaque bit transmis sur le canal est soumis à la probabilité $1 - p_e$ d'être transmis avec succès ou avec la probabilité p_e d'être effacé. La capacité du canal binaire à effacement est égale à $1 - p_e$.

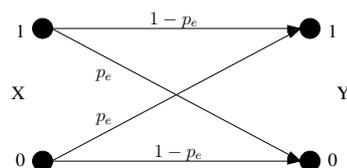


FIGURE 2.2 – Canal binaire symétrique

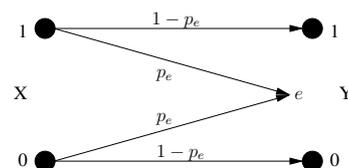


FIGURE 2.3 – Canal binaire à effacement

Canal à effacement de paquets

Dans la majorité des applications, au lieu de considérer la transmission individuelle de chaque bit d'information, on considère la transmission d'un ensemble de bits formant un paquet. Nous faisons appel ainsi au canal à effacement de paquets pour modéliser la probabilité de réception de chaque paquet sur le lien. En plus du codage canal, un code détecteur d'erreur est appliqué sur les paquets. Le paquet est considéré comme transmis avec succès si aucune erreur n'est détectée. On utilise, par exemple, pour la détection d'erreur le code de parité, bit de contrôle et/ou la redondance cyclique *Cyclic Redundancy Check* (CRC), etc. Le paquet est considéré soit transmis avec succès ou sinon totalement effacé avec la probabilité d'effacement de paquet p_e égale à la probabilité d'erreur par paquet *Packet Error Rate* (PER). La capacité de canal à effacement par paquet est égale à $1 - p_e$.

2.2 Code correcteur d'erreurs

L'objectif principal du code correcteur d'erreurs est de fiabiliser la transmission grâce à la redondance introduite par le codeur. Pour la bonne compréhension de la suite de ce manuscrit, nous proposons d'aborder dans cette partie les notions fondamentales relatives aux codes. Puis, nous décrirons les différentes familles de codes que nous utiliserons dans la suite de notre étude. L'introduction détaillée du code correcteur d'erreurs peut être retrouvée dans [104].

2.2.1 Principes des codes correcteurs d'erreurs

Tout d'abord, nous allons définir l'alphabet \mathcal{A} sur lequel le codage va s'appliquer.

Definition 2.2.1.1 (Corps finis). *Un corps fini appelé également corps de Galois est un corps commutatif (dans lequel l'addition, la multiplication, la soustraction et la division dans ce corps sont finis). A un isomorphisme près, un corps fini est entièrement déterminé par son cardinal qui est toujours la puissance d'un nombre premier. Pour tout nombre premier $p \in \mathbb{N}$ pour tout entier relatif $N \in \mathbb{N}$, l'ensemble des entiers relatifs modulo p est un corps qui contient $q = p^N$ éléments notés $GF(q)$ ou \mathbb{F}_q^N .*

Soit \mathcal{A} un ensemble d'alphabets muni d'une structure de corps, nous pouvons écrire \mathcal{A}_q^N un corps fini à q éléments de taille N . Nous prenons exemple un corps binaire \mathbb{F}_2^N , $\mathcal{A} = \{0, 1\}$. L'addition modulo 2 dans ce corps se réalise par l'opération ou exclusif, la multiplication dans ce corps se réalise par l'opération et selon la Table 2.1.

+	0	1
0	0	1
1	1	0

X	0	1
0	0	0
1	0	1

TABLE 2.1 – L'addition et la multiplication dans le corps \mathbb{F}_2

Dans ces travaux, nous travaillons principalement dans \mathbb{F}_2 .

Definition 2.2.1.2 (Code en bloc). *Le code en bloc est un code correcteur d'erreurs qui est un sous espace vectoriel de dimension K de l'espace vectoriel \mathbb{F}_q^N . Ce code convertit les séquences*

initiales de messages \mathcal{M} de taille K , $\mathcal{M} = \{m_1, m_2, \dots, m_k\} \in \mathbb{F}_q^K$ à ces séquences de N symboles encodés $\mathcal{X} = \{x_1, x_2, \dots, x_n\} \in \mathbb{F}_q^N$ appelés mots codes. On désigne par $\mathcal{C}(N, K)$ un code de longueur N et de dimension K ($K < N$).

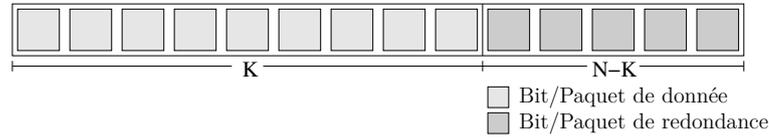


FIGURE 2.4 – Représentation du code en bloc

La Figure 2.4 représente un code en bloc où chaque carré présente un symbole d'information. Ces symboles peuvent être des bits (dans le cas du code correcteur d'erreur par bit) ou des paquets (pour le cas du code à effacement) selon les cas. Cette Figure est un exemple particulier du code en bloc linéaire symétrique dans lequel les K premiers carrés correspondent à K symboles initiaux à transmettre et les derniers $N - K$ carrés correspondent aux $N - K$ symboles de redondance introduit par le code. Dans un cas général, ces symboles de redondance résultent de l'addition linéaire de K symboles initiaux dans le corps \mathbb{F}_q selon les différentes techniques spécifiées par le code.

Definition 2.2.1.3 (Rendement). *Le rendement d'un code est le rapport K/N . Cette quantité est également appelée taux du code ou débit de codage. Nous emploierons dans ce manuscrit le terme rendement.*

Definition 2.2.1.4 (Code linéaire). *On appelle code linéaire, un code bloc $\mathcal{C}(N, K)$ qui est linéaire dans le corps \mathbb{F}_q^N .*

Definition 2.2.1.5 (Matrice génératrice). *Une matrice génératrice \mathbf{G} du code $\mathcal{C}(N, K)$ est une matrice de l'application de \mathbb{F}_q^K dans \mathbb{F}_q^N de dimension $K \times N$. Les lignes de la matrice \mathbf{G} forment une base de l'espace vectoriel du code.*

Soit M un message vu comme un vecteur de dimension K dans \mathbb{F}_q^K . Soit \mathbf{G} une matrice génératrice d'un code $\mathcal{C}(N, K)$. L'opération de codage est le produit :

$$X = \mathbf{G} \cdot M. \quad (2.4)$$

Definition 2.2.1.6 (Matrice de parité). *La matrice de parité H du code $\mathcal{C}(N, K)$ est une matrice de taille $(N - K) \times K$ dont chaque ligne forme un sous-espace vectoriel orthogonal à l'espace des mots de codes. Pour tout mot de code X , on a donc $\mathbf{H} \cdot X = 0$.*

Definition 2.2.1.7 (Code linéaire aléatoire). *Le code linéaire aléatoire est un code linéaire $\mathcal{C}(N, K)$ sélectionné aléatoirement et uniformément parmi les symboles à l'entrée du codeur. Cet ensemble de code peut être défini par le choix de la matrice génératrice ou par le choix de la matrice de parité.*

Definition 2.2.1.8 (Code systématique). *On parle d'un code systématique quand les mots de code contiennent explicitement les symboles de la source plus les bits de redondance. La matrice génératrice du code systématique \mathbf{G}_{sys} contient la matrice identité :*

$$\mathbf{G}_{sys} = [\mathbf{I}_K | \mathbf{R}]. \quad (2.5)$$

A partir de Équation(2.5), la matrice de parité \mathbf{H}_{sys} peut être construit comme :

$$\mathbf{H}_{sys} = [-\mathbf{R}^T | \mathbf{I}_{N-K}]. \quad (2.6)$$

Definition 2.2.1.9 (Poids de Hamming). *Le poids de Hamming d'un mot de code $X \in \mathbb{F}_q^N$ présenté par $(W_H(X))$ est défini par le nombre de ses coordonnées non nulles.*

Definition 2.2.1.10 (Distance de Hamming). *La distance de Hamming $d_H(X, X')$ entre deux mots de code X et X' de longueur N ($X, X' \in \mathbb{F}_q^N$) est le nombre de positions pour lesquelles les symboles correspondants sont différents.*

$$d_H(X, X') = |\{i | 1 \leq i \leq N, x_i \neq x'_i\}| \quad (2.7)$$

Definition 2.2.1.11 (Distance minimal d'un code). *La distance minimale du code est la plus petite distance entre deux mots de code quelconques. Pour le code linéaire, c'est aussi égale au nombre minimal de symboles non nul de tous les mots de code.*

Theorem 1 (Borne de Singleton). *La distance minimale d_{min} d'un code linéaire $\mathcal{C}(N, K)$ vérifie la propriété suivante :*

$$d_{min} \leq 1 + N - K. \quad (2.8)$$

Theorem 2 (Capacité de détection et correction). *La capacité de détection et correction d'un code est définie par le nombre de symboles erronées qu'il est capable de détecter ou corriger. Pour qu'un code en bloc de longueur N ait une capacité de détection et correction d'erreur d'ordre e , il faut que sa distance minimal d_{min} soit supérieur à $1 + e$ et $1 + 2e$ respectivement.*

2.2.2 Codes d'effacement

Selon le second théorème de Shannon [142], on apprend que pour un canal discret sans mémoire de capacité C et pour un rendement inférieur à C , il existe des familles de codes en bloc $\mathcal{C}(N, K)$ tels que la probabilité d'erreur du code tend vers 0 lorsque N tend vers l'infini. On cherche des codes linéaires en bloc ayant un bon rendement et corrigeant un maximum d'erreur. Elias a montré dans [43] qu'il existe des codes linéaires qui vérifient ce théorème.

Nous avons défini dans la section précédente la borne de Singleton. Très peu de codes linéaires binaires ($\mathcal{A} = \mathbb{F}_2^N$) atteignent cette borne avec égalité (on trouve certains codes de Hamming et de Golay). On trouve des codes qui atteignent la borne de Singleton pour des alphabets plus grands $q > 2$. Par la suite, notre étude porte sur le canal à effacement de paquets.

L'utilisation de codage de correction préventive des erreurs *Forward Error Correction* (FEC) à l'aide de code d'effacement a été ainsi introduit pour fiabiliser la transmission multiple des données [53, 74, 7].

Afin d'uniformiser les notations entre le monde du réseau protocolaire et celui des codes, nous adopterons le langage suivant :

- Nous aurons toujours un message \mathcal{M} à transmettre.

- Les symboles constituant le message sont désignés par le terme *fragment* $F = \{f_1, f_2, \dots, f_K\} \in \mathbb{F}_2^K$
- Les symboles du mot de code sont les *paquets* $P = \{p_1, p_2, \dots, p_N\} \in \mathbb{F}_2^N$ transmis

Le code d'effacement a pour objectif de protéger l'ensemble de données divisées en fragments. Lorsque les paquets ne sont pas de bonne taille, la fragmentation et l'ajout de bits de bourrage (bits de zéros), nécessaires pour avoir la taille de données requise. L'encodage et le décodage dans ce cas appliquent l'opération *ou exclusif* souvent appelé l'opération XOR sur les paquets à transmettre. Nous focalisons dans cette section sur les différents caractéristiques des codes d'effacement.

2.3 Codes d'effacement à rendement fixe

Nous considérons dans un premier temps les codes d'effacement ayant un rendement fixe : le nombre N de paquets transmis par la source est fixe. Nous présentons brièvement les principales familles de code d'effacement à rendement fixe tel que

- les codes MDS,
- les codes LDPC.

2.3.1 Code MDS

Les codes *Maximum Distance Separable* (MDS) ont comme principale caractéristique de vérifier la borne de Singleton avec égalité. Ceci peut se traduire de la façon suivante : Si on reçoit au moins K paquets parmi les N générés par la source, alors on sait retrouver le message d'origine. Ces codes ont un rendement optimal. On parle du code parfait.

Les codes MDS les plus connus sont les codes Reed-Solomon (RS) [128] utilisés dans de nombreuses applications comme l'ADSL, la technologie RAID ou les CDs/DVDs. Pour le canal BSC, l'algorithme de Berlekamp-Massey [19, 107] permet de créer un code RS avec une complexité de décodage de $\mathcal{O}(n(\log(n))^2)$. Pour le canal à effacement, des méthodes optimales ayant des bonnes complexités existent [38]. Il existe aussi des algorithmes de décodage permettant de décoder au-delà de la capacité du code : c'est le *list decoding* de Sudan [149].

Pour un code MDS $\mathcal{C}(N, K)$ toutes les K colonnes de la matrice génératrice \mathbf{G} ainsi que toutes les $N - K$ colonnes de la matrice de parité \mathbf{H} doivent être linéairement indépendants [104]. Nous donnons deux exemples du code MDS dans cette partie qui sont deux variantes de code RS. Ces codes permettent d'avoir une bonne complexité de codage en utilisant des matrices de Cauchy et des matrices de Vandermonde.

Code Reed Solomon par la matrice de Cauchy Blömer et al. ont proposé d'utiliser la matrice de Cauchy pour implémenter des codes RS afin de diminuer la complexité de décodage sur le canal à effacement. Les sous-matrices de Cauchy sont inversibles et permettent d'accélérer le processus de décodage. La matrice de Cauchy \mathbf{G} de taille $K \times N$ possède des coefficients g_{ij} tels que :

$$g_{ij} = \frac{1}{\alpha_i - \beta_j}, \alpha_i - \beta_j \neq 0, 1 \leq i \leq K, 1 \leq j \leq N. \quad (2.9)$$

Code Reed Solomon par la matrice de Vandermonde Un code RS peut être également implémenté à l'aide de matrices de Vandermonde [120, 129]. Ce code est aujourd'hui standardisé [82] et sert de référence pour l'évaluation des codes à effacement. La matrice de Vandermonde \mathbf{G} peut être présentée par ses coefficients :

$$g_{ij} = \alpha_i^{j-1}, 1 \leq i \leq K, 1 \leq j \leq N. \quad (2.10)$$

L'intérêt de cette matrice est que toutes les sous-matrices carrées de cette matrice sont inversibles si et seulement si α_i sont deux à deux distincts, comme pour le cas de la matrice de Cauchy. Cependant, selon [129] le code construit à partir de la matrice de Vandermonde pose une complexité de calcul plus faible que celle de Cauchy.

Généralement, la complexité de l'algorithme de codage RS est $\mathcal{O}(NK)$ sur l'ensemble des paquets [17]. Le décodage peut être réalisé par multiplication matricielle. Le plus courant est la résolution du système d'équations par la technique du pivot de Gauss dont la complexité est du $\mathcal{O}(K^3)$.

L'inconvénient principal des codes RS est la complexité importante de décodage. Une technique proposée dans [105] permet de réduire la complexité jusqu'à obtenir $\mathcal{O}(N(\log N)^2 \log \log N)$. Cependant, l'algorithme qui permet d'obtenir cette complexité est asymptotiquement inefficace. D'autres algorithmes de décodage proposés ayant une complexité quadratique et log-quadratique en temps apparaissent plus pratique à réaliser [85, 38].

2.3.2 Code LDPC

Les codes MDS sont optimaux en terme d'erreur, cependant, ils entraînent une complexité de décodage élevée. Nous nous intéressons donc à chercher une technique de décodage plus simple. Gallager a proposé dans [51] un code qui permet d'avoir une complexité de décodage moins élevée grâce à la technique de décodage par propagation de croyance ou *Belief Propagation* (BP). Les codes de Gallager sont appelés codes à faible densité ou *Low Density Parity Check* (LDPC).

Pour faciliter le décodage BP, la matrice de parité de ce code doit avoir la particularité d'être à faible densité (matrice creuse). Cet algorithme de décodage BP est asymptotiquement optimal et permet d'assimiler le problème à un système d'équation. On utilise un algorithme itératif qui permet d'obtenir une complexité de codage et de décodage linéaires en temps. Si la matrice est creuse, il a été montré dans [94, 97, 99] qu'un code LDPC permet d'atteindre la borne en capacité pour le canal à effacement.

La modélisation du code LDPC est simple et rapide. Il peut être systématique ou non. L'encodage nécessite un bon choix du graphe bipartite (graphe de Tanner [151]). On utilise en pratique la matrice de parité ou le graphe bipartite pour réaliser le codage.

Soit F un vecteur qui présente l'ensemble des fragments à encoder. La matrice de parité \mathbf{H} satisfait :

$$\mathbf{H} \cdot F = 0 \quad (2.11)$$

Prenant en exemple le cas suivant :

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

notons ici que la matrice \mathbf{H} présentée ci-dessus est juste un exemple simple qui n'est pas creuse selon la définition du code LDPC. F représente un vecteur qui contient K fragments initiaux à encoder. La Figure 2.5 représente pour cette matrice la relation entre les paquets de redondance en fonction des fragments initiaux. Ici, le codeur envoie les séquences de paquets $P = \{f_1, f_2, f_3, r_1, r_2\}$. La deuxième figure à droite représente ces relations sur le graphe de Tanner duquel nous pouvons déduire les relations entre les fragments et les redondances transmises qui permettent de décoder.

Il existe plusieurs catégories du code LDPC : régulier, irrégulier, systématique, en cascade (LDPC staircase), etc [102]. Le code LDPC en cascade est standardisé à l'IETF au sein du RFC5170 [131].

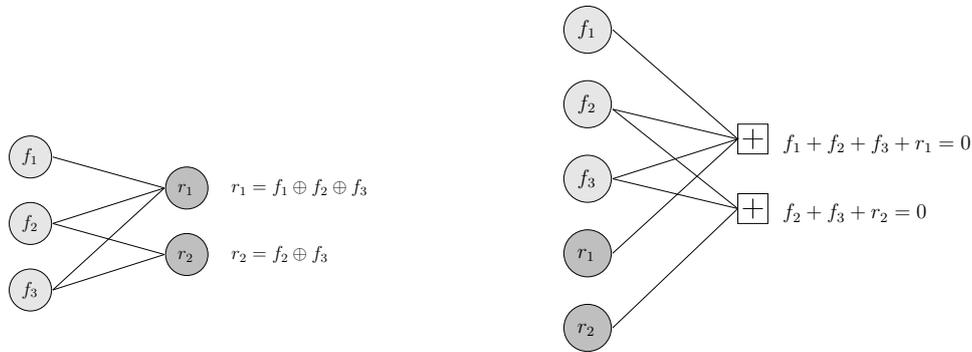


FIGURE 2.5 – Représentations du code LDPC : graphe biparti et graphe de Tanner

Il existe également le code *Tornado* qui est formé à partir de la concaténation du code LDPC en cascade avec le code RS en dernière étape de décodage. En effet, dans une première étape, les fragments sont encodés avec le code LDPC. Ensuite, les paquets sont recombinaés successivement par le code RS selon la Figure 2.6. Le code LDPC est intéressant en terme du temps de calcul mais n'est pas optimal en terme de correction d'erreurs. C'est la raison pour laquelle le code RS est conjointement introduit pour permettre de récupérer l'information plus efficacement [24, 98].

2.4 Codes Fontaine : Code d'effacement sans rendement

Lorsque la réalisation de canal est connue a priori, nous pouvons trouver un code qui possède le rendement fixe approprié pour contourner l'effacement du canal. Quand le canal n'est pas connu a priori, on doit utiliser d'autres techniques. Les codes correcteurs d'erreurs permettant de travailler dans ce type d'hypothèse sont dits *sans rendement* (*rateless*). Ces codes permettent de s'adapter aux changements de l'état du canal. Les codes fontaine font partie des codes sans rendement : à partir d'un nombre fini de fragments d'information, on génère des paquets jusqu'à ce que le destinataire réussisse à décoder. Ce débit variable implique une adaptation en temps

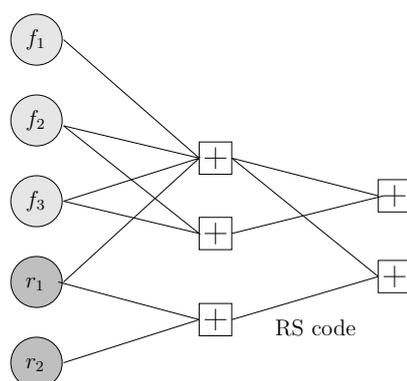


FIGURE 2.6 – Code Tornado

réel de la transmission depuis la source vers le destinataire. Les travaux de Verdu et al. [165] présentent de manière théorique les différents problèmes de codage et discute des problèmes liés aux codes sans rendement.

Le principe de transmission des codes fontaine est analogue au cas d'une fontaine d'eau : pour remplir un verre à la fontaine (Figure 2.7) on s'intéresse seulement à la quantité d'eau nécessaire à remplir le verre sans considérer l'ordonnancement des gouttes d'eau ni celles qui tombent à côté du verre [24, 23]. Cette idée amène à la réalisation des codes ayant cette caractéristique. Notons que chaque paquet p_i est une combinaison linéaire dans \mathbb{F}_2 des K fragments pris dans $F = \{f_1, f_2, \dots, f_K\}$ et que le décodage est accompli après la réception de $N (N > K)$ paquets à la destinataire.

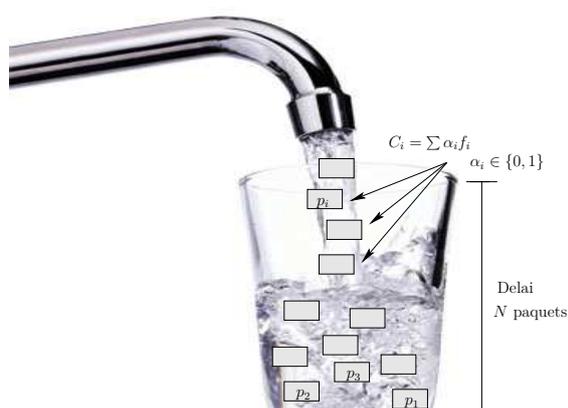


FIGURE 2.7 – Analogie du code fontaine avec le remplissage d'un verre par une fontaine d'eau

Selon Maymounkov [108], pour tout message constitué de K fragments à l'entrée de l'encodeur et pour $\epsilon \in \mathbb{R}^+$, il existe un code sans rendement qui permet de récupérer ces K fragments à partir de n'importe quel $N = (1 + \epsilon)K$ paquets reçus avec une probabilité proportionnelle à $K \log(1/\epsilon)$. Les codes fontaine sont les premiers codes qui vérifient cette propriété et qui

sont dits *universels*, c'est-à-dire, leurs performances sont bonnes pour tous les types de canaux de transmission. Bien qu'ils aient été initialement conçus pour les canaux à effacement, ces codes ont été montrés efficaces pour le canal discret sans mémoire, le canal AWGN, le canal de Rayleigh [41, 45, 118, 30].

Contrairement aux codes à rendement fixe, les codes sans rendement adaptent leur débit de transmission selon la capacité du système de transmission. Du fait de leur caractéristique aléatoire, il est important de fournir au décodeur les caractéristiques du code qui permettent de retrouver le message initial. Il existe plusieurs mécanismes pour communiquer ces informations selon les différentes applications. Selon [95], le degré et la liste de voisins peuvent être indiqués pour chaque paquet envoyé. En outre, ce degré et l'ensemble de voisins peut être calculé par le temps de réception ou la position du paquet encodé relativement aux positions des autres paquets reçus. Un transfert de la clé peut être aussi associé à l'envoi du paquet. Ce dernier nécessite la connaissance de la clé soit au niveau d'encodeur et de décodeur pour pouvoir retrouver les informations nécessaires au codage. Dans ce travail nous avons choisi d'ajouter l'entête de K bits à chaque paquet envoyé. Ces K bits indiquent la présence des symboles contenu dans le paquet par ses combinaisons linéaires XOR. L'indice de i -ème symbole présenté dans le paquet est marqué comme le bit 1, sinon il est présenté par le bit 0. Chaque paquet p_i envoyé est composé donc de la partie en-tête e_i et la partie message m_i . Nous avons $p_i = \{e_i || m_i\}$ avec e_i décrivant les combinaisons linéaires dans \mathbb{F}_2^K . La partie de données contient les combinaisons de fragments comme indiqué dans l'entête du paquet $m_i = \bigoplus_{i=1}^K e_i f_i$. Par la suite, nous décrirons principalement les opérations de codage effectuées sur l'entête de paquet.

Definition 2.4.0.1 (degré de paquet). *Nous notons le degré d de paquet p , le poids de Hamming $W_H(e)$ tel que : $d = W_H(e) = \sum_{i=1}^K e_i$*

Definition 2.4.0.2 (distribution de degré). *La distribution de degré définit la probabilité où un paquet de degré d est émis par l'encodeur, autrement dit, la probabilité où d fragments sont choisis pour être combinés dans des paquets transmis.*

Nous trouvons les principales catégories de codes fontaine connus jusqu'à présent :

- les codes aléatoires,
- les codes LT,
- les codes Raptors.

Il existe d'autres familles de codes fontaines qui permettent de s'adapter à certaines contraintes de l'environnement : online code [108], growth code [71], code pour les applications à temps réel [159], etc. Par la suite, nous étudierons les trois grands familles de code fontaine citées précédemment.

2.4.1 Codes fontaine aléatoires

La première famille de code fontaine est appelée aléatoire linéaire ou *Random Linear Fountain*(RLF). Il s'agit de codes à rendement optimal par rapport aux autres familles des codes fontaine. Chaque paquet encodé RLF est généré à partir de la combinaison linéaire XOR de fragments uniformément choisis avec une probabilité de 1/2 et qui sont combinés à l'aide de l'opération binaire XOR. Le but du code est d'envoyer les paquets encodés pour que le récepteur puisse décoder l'information à partir d'environ K paquets encodés linéairement indépendants.

Encodage du code RLF

La matrice génératrice \mathbf{G} du code RLF est construite de telle façon que chaque élément de la matrice est choisi dans $\mathcal{A} = \{0, 1\}$ avec une probabilité équivalente égale à 0.5. Nous prenons par exemple la matrice $\mathbf{G}_{K \times N}$:

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Les paquets encodés prennent la forme :

$$P = \mathbf{G} \cdot F \quad (2.12)$$

Distribution Binomiale

Le fait que chaque élément de la matrice génératrice \mathbf{G} soit choisi aléatoirement et uniformément dans $\mathcal{A} = \{0, 1\}$ est équivalent à dire que les fragments initiaux dans $F = \{f_1, f_2, \dots, f_N\} \in \mathbb{F}_2^N$ subissent la même probabilité φ d'être choisis avec $\varphi = 0.5$.

La distribution de degré des paquets après l'encodage est défini par rapport aux probabilités $\rho(d)$ de choisir exactement d fragments à combiner dans un paquet donné. Ceci prend la forme de la distribution binomiale (DB) qui est égale à :

$$\rho(d) = C_K^d \cdot \varphi^d (1 - \varphi)^{K-d}. \quad (2.13)$$

La Figure 2.8 illustre la distribution binomiale qui est la probabilité de construction de paquet du degré d à partir du code RLF.

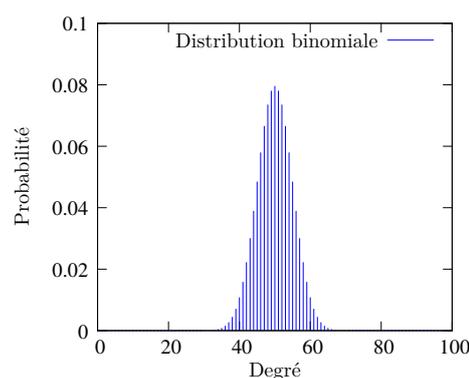


FIGURE 2.8 – Distribution binomiale

Décodage du code RLF

Le décodage du code RLF suit la technique de décodage par vraisemblance maximale ou *Maximum Likelihood* (ML) avec l'inversion matricielle. Pour le code linéaire aléatoire, ceci

revient à résoudre un système d'équations par la technique du *pivot de Gauss*. La résolution linéaire est réalisée par :

$$F = \mathbf{G}^{-1} \mathbf{G} \cdot F = \mathbf{G}^{-1} \cdot P \quad (2.14)$$

Selon [103], le nombre de paquets nécessaires pour récupérer K symboles d'information avec la probabilité $1 - p_e$ de succès est égale à environ $K + \log_2(1/p_e)$. Ce code permet asymptotiquement d'atteindre la capacité de canal, la meilleure efficacité pour un temps de décodage le plus court et le nombre de transmissions le plus petit. Malheureusement, il a une complexité de décodage élevée de l'ordre de $\mathcal{O}(K^3)$ dans le temps.

2.4.2 Code LT

Luby a proposé dans [98] une famille de codes dont le décodage est moins complexe que celui du code RLF. Ces travaux ont amené la naissance des codes *Luby Transform* (LT) en 2002 [95] et l'utilisation du décodage par propagation de croyance *Belief Propagation* (BP). Le code LT est la première classe de code fontaine universel (la performance est asymptotiquement optimale sur tous les canaux de transmissions). La Figure 2.9 présente le graphe biparti du code LT. Contrairement aux exemples du graphe biparti du code LDPC où les redondances sont présentées sur la droite du graphe biparti, ces éléments pour la représentation du code LT résultent des combinaisons linéaires XOR entre les fragments initiaux sous une forme non systématique. Le décodage BP est basé sur la résolution itérative du graphe biparti et permet de faire un compromis entre la complexité et l'efficacité de décodage : l'algorithme est plus simple, mais il permet de décoder moins rapidement. Pour obtenir la performance de décodage optimale avec l'algorithme BP, les paquets doivent être encodés d'une manière spécifique. En effet, les fragments qui sont combinés dans un paquet doivent être choisis conformément à une certaine distribution statistique. Luby a proposé la loi de Soliton robuste ou *Robust Soliton Distribution* (RSD) comme une solution optimale en terme de capacité.

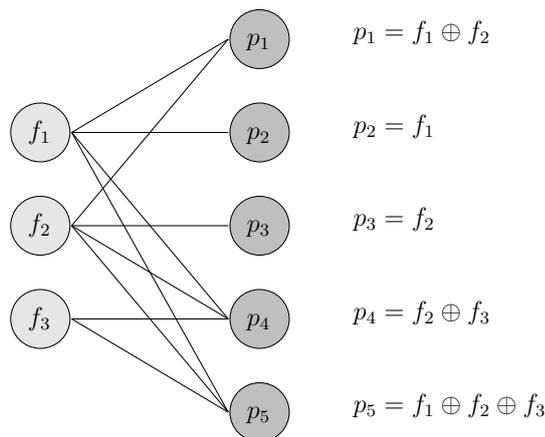


FIGURE 2.9 – Représentation du code LT sur le graphe biparti

Le code LT est très attirant par sa faible complexité de codage. L'encodage du code LT a

une complexité en $\mathcal{O}(\log(K))$ opérations par paquet envoyé. La reconstruction générale de K fragments peut être effectuée à partir de n'importe quelle combinaison de $K + \mathcal{O}(\sqrt{K} \log^2(K))$ paquets en utilisant un décodeur itératif qui a un coût de l'ordre de $\mathcal{O}(K \log(K))$ [17]. Le nombre de symboles nécessaires au décodage tend vers K lorsque la dimension K du code tend vers l'infini. Autrement dit le code LT est asymptotiquement parfait.

Encodage du code LT

L'encodage des codes LT est réalisé selon l'Algorithme 1 qui a comme paramètre important la distribution RSD. Cette distribution est définie par rapport aux probabilités de choisir exactement d fragments à combiner dans un paquet donné. La combinaison des fragments se fait dans \mathbb{F}_2^K par l'addition modulo 2. Cet algorithme s'arrête lorsque le décodeur reçoit suffisamment de paquets pour décoder ou bien lorsqu'un temps d'attente maximal (*timeout*) est atteint.

Algorithme 1 : Encodage du code LT

```

tant que on n'a pas reçu l'indication d'arrêter faire
  choisir  $d$  à l'aide de la distribution RSD
  choisir aléatoirement et uniformément  $d$  fragments parmi  $K$  tel que l'ensemble de
  fragments choisit  $F$  égale à  $\{f_1, f_2, \dots, f_d\}$ 
  faire en tête
  envoyer  $p_j = f_1 \oplus f_2 \oplus \dots \oplus f_d$ 
  incrémenter  $N_{tx}$ 
fin tant que

```

Distribution de Soliton Robuste

La distribution de Soliton robuste (RSD) a été proposée pour que le décodage BP se poursuive de manière efficace. Il est nécessaire que le degré du paquet soit assez grand pour assurer que chaque fragment f_i soit contenu dans au moins un paquet p_i . De plus, le degré des paquets doit rester faible de manière à permettre le déroulement du processus de décodage à chaque itération. Théoriquement, la distribution de Soliton Idéale (Figure 2.10) proposée par Luby permet de vérifier ces propriétés.

$$\rho(i) = \begin{cases} \frac{1}{K} & \text{for } i = 1 \\ \frac{1}{i(i-1)} & \text{for } 2 \leq i \leq K \end{cases} \quad (2.15)$$

Or, cette distribution n'est pas très efficace en pratique. Le décodage s'interrompt occasionnellement du fait d'un manque de paquets de faible degré. De plus, certains fragments n'apparaissent pas souvent dans les paquets. Pour contrer ces problèmes, Luby a proposé l'amélioration de cette distribution avec la distribution RSD. On assure que l'espérance du nombre de paquets de degré un est approximativement égale à :

$$S \equiv c \ln\left(\frac{K}{\delta}\right) \sqrt{K} \quad (2.16)$$

avec c et δ des nombres réels donnés. De cette manière, la distribution Soliton robuste peut être exprimée comme :

$$\mu(i) = \frac{\rho(i) + \tau(i)}{Z}, \quad (2.17)$$

avec

$$Z = \sum_i \rho(i) + \tau(i), \quad (2.18)$$

et

$$\tau(i) = \begin{cases} \frac{S}{ik} & \text{pour } 1 \leq i \leq \frac{k}{S} - 1, \\ \frac{S \ln\left(\frac{S}{\delta}\right)}{K} & \text{pour } i = \frac{K}{S}, \\ 0 & \text{pour } i > \frac{K}{S}. \end{cases} \quad (2.19)$$

les exemples de distribution de Soliton idéale et de Soliton Robuste sont présentés aux Figures 2.10 et 2.11 respectivement. Nous remarquons particulièrement une valeur spécifique appelée *spike* autour de la valeur de $K \approx 60$. Ces distributions sont présentées pour un code de dimension $K = 100$.

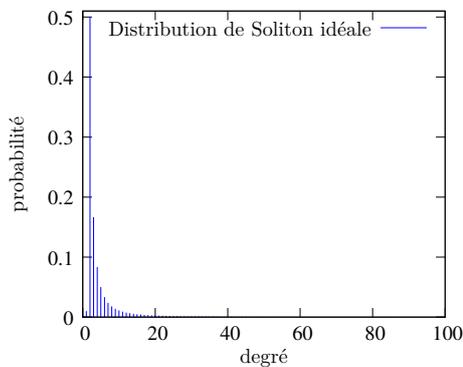


FIGURE 2.10 – Distribution de Soliton idéale

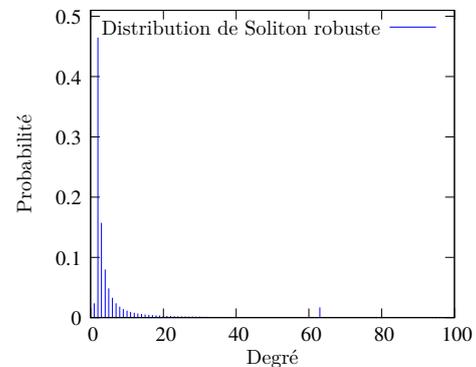


FIGURE 2.11 – Distribution de Soliton robuste

La distribution RSD est asymptotiquement optimale c'est-à-dire que le rendement du code tend vers 1 lorsque la dimension du code K tend vers l'infini. Cependant, pour des longueurs de code petit, la redondance introduite par le code n'est pas négligeable par rapport à la dimension K . Plusieurs techniques pour optimiser la distribution de degré sont proposées pour K petit dans [66, 93]. Nous pouvons également envisager l'adaptation dynamique de degré à l'aide du canal de retour [57].

Décodage du code LT

Le code LT utilise le décodage par propagation de croyance (BP). Cette technique est basée sur le fait que les paquets de degré 1 peuvent être considérés comme décodés. Ainsi, à l'aide des paquets déjà décodés, le décodeur réduit à chaque itération le degré de paquets codés jusqu'à ce que tous les fragments soient décodés. La quantité de paquets nécessaires pour pouvoir décoder sans erreur est de l'ordre de $K + \epsilon$ où ϵ le taux de redondance du code. Cette méthode de décodage BP peut être décrite par l'Algorithme 2 :

Algorithme 2 : Décodage du code LT

tant que Le décodage n'est pas fini et qu'il existe encore dans la mémoire \mathcal{B} au moins un paquet de degré 1 **faire**
 $f_j \leftarrow$ paquet de degré 1 de la mémoire
pour $\forall p \in \mathcal{B}$ **faire**
 si p contient f_j **alors**
 $p \leftarrow p \oplus f_j$
 fin si
fin pour
fin tant que

L'analyse détaillée du processus de décodage par propagation de croyance a été développée dans [58, 117]. Ces études permettent de développer l'expression analytique de la probabilité d'erreur du décodage [72, 106, 101]. Le taux de redondance introduit par le code LT a été quantifié pour des canaux binaires symétriques et des canaux à effacements [45, 77].

Pour mieux comprendre le processus encodage/décodage du code LT, nous considérons un exemple simple dans lequel la source dispose d'un message \mathcal{M} à transmettre au destinataire. Dans un premier temps, la source applique une fragmentation pour créer $K = 3$ fragments $F = \{f_1, f_2, f_3\}$ de même taille. Le bourrage de bit ou *padding* est ensuite appliqué pour avoir des tailles de fragments homogènes. Ensuite, ces fragments sont choisis de manière uniforme pour être combinés par les additions linéaires dans \mathbb{F}_2^K . Le nombre de fragments dans chaque paquet envoyé suit la distribution RSD. Les paquets encodés $P = \{p_1, p_2, \dots, p_N\}$ sont ensuite transmis au destinataire. Ces étapes d'encodage LT sont présentées selon la Figure 2.12.

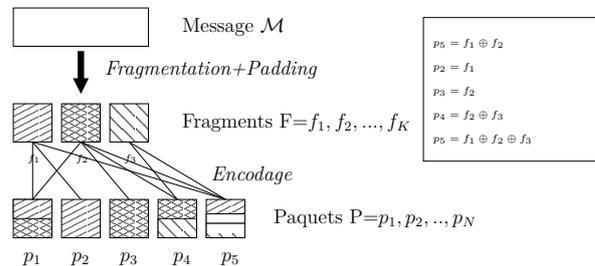
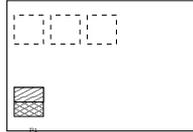


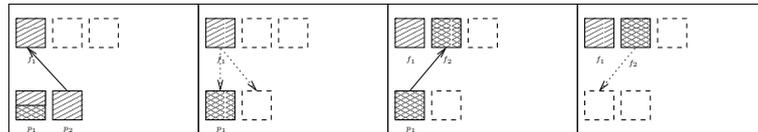
FIGURE 2.12 – L'encodage LT

Le destinataire une fois reçu ces paquets commence le processus de décodage itératif selon les étapes suivantes dès qu'il reçoit le premier paquet :

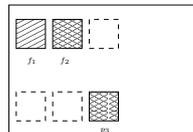
- Dans une première étape, le décodeur reçoit le paquet $p_1 = f_1 \oplus f_2$. Par manque du paquet de degré 1, le décodage ne peut pas poursuivre.



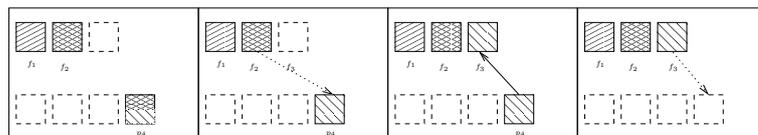
- Ensuite, le décodeur reçoit un deuxième paquet $p_2 = f_1$. Étant de degré 1, ce paquet ne contient que le fragment f_1 . Nous pouvons donc récupérer le fragment f_1 . Ce dernier est ensuite supprimé des autres combinaisons de paquets encodés disponibles par l'addition binaire XOR. Ceci permet de réduire le degré du paquet p_1 à 1 et permet de retrouver le fragment f_2 .



- Le décodeur reçoit ensuite le paquet $p_3 = f_2$, contenant le fragment déjà encodé, ce paquet n'apporte pas de nouvelle information.



- En fin, le quatrième paquet $p_4 = f_2 \oplus f_3$ arrive au décodeur. Sachant qu'on connaît déjà f_1 et f_2 , ce paquet nous permet de décoder le dernier fragment manquant f_3 .



Dans ce petit exemple de décodage, nous remarquons que le décodeur nécessite 4 paquets pour pouvoir décoder 3 fragments initiaux. Nous avons $N = 4$, $K = 3$. Par conséquent, le rendement du code est égal à $R = 3/4$.

2.4.3 Code Raptor

Le code Raptor est une dérivation du code LT qui consiste à améliorer la résistance au bruit du code LT (outer code) par le précodage [143]. Ce précodage est basé sur un code en bloc linéaire à rendement élevé (inner code). Le code LT employé a pour but de maintenir la faible complexité de décodage par BP. Le précode permet d'assister le décodage d'information lorsque le nombre de paquets reçus est inférieur à K . Le choix de précodage diffère selon les applications. Nous trouvons, par exemple, le précodage par les codes LDPC ou les codes de Hamming, etc.

Le code Raptor permet également d'améliorer certaines caractéristiques manquantes des codes LT tels que la propriété systématique [50, 136]. Avec cette propriété, les K premiers fragments sont envoyés sans être encodés, puis les redondances encodées sont envoyées. Ceci permet de réduire la latence de transmission notamment lorsque l'effacement sur le canal est faible car le décodage peut commencer à récupérer les fragments envoyés sans avoir besoin d'attendre la fin de transmission pour décoder. Les complexités de codage et de décodage du code Raptor sont linéaires dans le temps de l'ordre $\mathcal{O}(\log(K))$ opérations par paquets ainsi qu'une meilleure performance pour une petite dimension K .

Le *online code* conçu à partir du code LT concaténé avec un autre code en bloc suit la même principe du code Raptor, qui est éventuellement présenté dans [108].

La comparaison des différents codes fontaine est effectuée dans la Table 2.2 [155] sachant que N représente le nombre de paquets nécessaires reçus par le décodeur pour pouvoir récupérer K fragments d'informations originales. Dans un cas asymptotique $N \approx K$.

Code	Encodage	Décodage	N
Reed-Solomon	$\mathcal{O}(K)$	$\mathcal{O}(K^3)$	K
LDPC	$\mathcal{O}(1)$	$\mathcal{O}(K)$	$K + \epsilon$
RLF	$\mathcal{O}(K)$	$\mathcal{O}(K^3)$	$K + \mathcal{O}(1)$
LT	$\mathcal{O}(\log(K))$	$\mathcal{O}(K \log(K))$	$K + \mathcal{O}(\sqrt{K} \log^2(K))$
Raptor	$\mathcal{O}(1)$	$\mathcal{O}(K)$	$K + \epsilon$

TABLE 2.2 – Coût d'encodage et décodage pour les différent types de codes

2.5 Variantes des codes fontaine

Il existe plusieurs variantes des codes fontaine qui dérivent des principales familles des codes LT et RLF. Le code fontaine Turbo a été proposé dans [152] comme une sous catégorie de code Raptor. Il a été créé par la concaténation du code LT et le précodage utilisant le code Turbo. Dans [157], la technique du code LT est appliquée sur le sous-paquet encodé par le code aléatoire linéaire. Le code LT accumulé (ALT) construit à partir de la concaténation de code LT avec un précodage accumulé est proposé dans [89]. Ce dernier a été optimisé dans [176] en ajoutant

un bit de parité après le codage LT. Les codes LT et Raptor basés sur graphe ont été présentés dans [153].

Nous pouvons noter également plusieurs techniques qui ont pour but d'améliorer la performance d'encodage/décodage avec code fontaine. Dans [137], une méthode pour décoder efficacement en présence de certaines connaissances de l'information originale des deux sources corrélées est proposée par le codage Slepian-Wolf. L'encodage dépend de l'estimation de taux d'effacement dans le canal par la connaissance du nombre de paquets déjà décodés par le décodeur [159]. L'utilisation de code fontaine comme enrobage des codes correcteurs d'erreurs classiques a été étudiée dans [18].

Plusieurs autres variations de code LT peuvent être citées comme le code LT optimisé par la modification de degré LT décalé (Shifted LT Code) [57], le code LT distribué [138], etc.

2.6 Domaine d'application des codes fontaine

Nous considérons dans cette section les différentes solutions de codes fontaine pour la distribution de contenu. Ces codes ont été exploités dans ce contexte dans le but d'optimiser la performance de transmission. Nous pouvons citer plusieurs domaines d'application :

Diffusion multiple de données

Le code d'effacement est une solution intéressante pour assurer la fiabilité de transmission. Préalablement, les codes par répétition ont été employés pour introduire de la redondance d'information permettant de contourner les effacements dans le canal. Dans une diffusion multiples de données, les codes par répétition avec disque à diffusion ont été proposés par Acharya et al. [3]. Cette technique consiste à transmettre en boucle les paquets par la source. Les destinataires reçoivent à partir d'un certain temps d'écoute la totalité des fragments envoyés. Cette technique implique cependant un surcoût en réception non négligeable car si un paquet est perdu, il faut attendre un prochain tour pour récupérer ce paquet. Les études dans [130, 135, 20] ont proposé d'optimiser la transmission dans ce contexte grâce à l'ajout de redondance d'information à l'aide du code correcteurs RS. Cela permet d'exploiter le contenu diffusé dès que la quantité d'information reçue par le récepteur est légèrement supérieure à la quantité d'information de la source.

Les codes correcteurs d'erreurs à effacement ont ensuite été développés comme solution efficace pour la diffusion multiple. Les premiers codes proposés sont les codes RS [64, 129, 130, 135]. Les codes RS sont des familles de code MDS qui permettent d'atteindre la borne de la capacité maximale pour plusieurs topologies du canal : canal à relai [75], multi-relais [76] ou dans un réseau à plus grande échelle [36]. Cependant, le code RS impose une complexité élevée qui devient importante pour une grande dimension du code. Le code Tornado a été proposé comme solution pour réduire la complexité du calcul et l'inefficacité de codage pour une grande quantité de données [99, 25, 20]. Bien que la redondance introduite par le code RS et le code Tornado permet de corriger certains nombres d'effacements, dans le cas où le taux d'erreurs sur le lien dépasse la capacité de correction du code, le mécanisme de demande de retransmission est toujours exigé pour fiabiliser la transmission.

Les codes fontaine numériques tels que le code LT ou le code Raptor permettent de réduire considérablement le nombre de retransmissions exigées et d'adapter le débit de transmission selon les caractéristiques du canal. Ceci est très avantageux pour une communication par diffusion multiple [24, 103].

Compression de données

La compression de données avec les codes d'effacement donne une performance comparable à plusieurs standards de compression de données. Le code LDPC est utilisé dans plusieurs techniques de compressions de données, par exemple, gzip, PPM et BZIP [28, 27, 29]. Le code fontaine concaténé avec le code BWT, qui permet d'obtenir un temps de compression et dé-compression linéaire dans le temps, a été étudié dans [26].

Multimédia et transmission en temps réel

Plusieurs travaux de recherche ont tenté d'adapter l'utilisation de code fontaine dans des communications en temps réel. Dans [146], la notion de fenêtrage a été proposée pour le code RLF. La matrice génératrice du code est choisie consécutivement dans une fenêtre distribuée. Dans un cas général, la taille de fenêtre d'environ $\mathcal{O}(\sqrt{K})$, où K est la taille de fenêtre partitionnée, permet d'obtenir la matrice génératrice assez proche du cas général [158, 159]. L'encodage par code fontaine dans une fenêtre peut être réalisé selon plusieurs techniques optimales proposées dans [156, 160].

L'étude dans [17] a identifié quelques problèmes de transmission avec fenêtrage : les données transmises par cette technique subissent à un délai de transmission supérieur à K . En plus, les erreurs en rafale de taille $> K + 1$ paquets peut conduire à une perte de donnée d'une fenêtre entière. Le premier problème peut être résolu par l'utilisation du code systématique proposé dans [148] qui permet au décodeur de commencer à récupérer les données au fur et à mesure sans attendre la fin de la transmission complète pour décoder la totalité des messages. Nous trouvons également la proposition de la distribution de degré optimisée qui permet d'obtenir une performance de décodage intermédiaire satisfaisante pour réduire la latence dans [134, 147, 148], ceci permet de décoder efficacement l'information au moment où le nombre de paquets reçus par le décodeur est inférieur à K . Enfin, pour éviter les erreurs de transmission en rafales, au lieu de retransmettre successivement la même fenêtre, l'auteur de [40] a proposé l'utilisation de canal de retour de l'ordre de $\mathcal{O}(\sqrt{K})$ pour gérer la perte de données. Le canal de retour employé est destiné à fournir la connaissance du canal à l'émetteur.

La qualité de service du flux multimédia peut être garantie par la répartition variable de la taille de fenêtre [139]. Avec ce mécanisme, les symboles dans la fenêtre de petite taille possèdent une probabilité de protection plus élevée. Cette méthode, également appelée méthode de protection d'erreur non-uniforme UEP est plus adaptée que la méthode proposée dans [127, 90] où l'émetteur assigne la probabilité du choix de symbole à l'entrée avec des probabilités fixes.

Le code Raptor est adopté dans plusieurs standardisation pour l'émission de flux multimédia en temps réel, par exemple, une diffusion multimédia *Multimedia Broadcast and Multicast Services*(MBMS) par l'organisation 3GPP [163], la diffusion multimédia sur une plateforme

UMTS [96] et la diffusion multiple de vidéo numérique *Digital Video Broadcast* (DVB) [161], etc.

Stockage distribué

Le principe de stockage distribué est analogue aux techniques de stockage de disque en miroir RAID. Les autres nœuds dans le réseau sont utilisés comme unités de stockage pour sauvegarder une réplique des données d'un capteur particulier. Lorsqu'il y a une faille, la redondance sauvegardée peut être récupérée.

Le code correcteur d'erreur est introduit dans ce contexte pour améliorer la performance de sauvegarde, assurer la persistance, la fiabilité et la tolérance du système surtout pour les cas de fortes perturbations dans le réseau auquel certains nœuds sont sujets à défaillance. Ceci peut être réalisé selon deux scénarios différents : par le codage centralisé géré par le nœud de contrôle qui s'occupe d'encoder et de diffuser les paquets aux autres nœuds dans le réseau et/ou par le codage distribué où l'ensemble de nœuds s'occupent d'encodage et de décodage des informations à diffuser.

Bien que le stockage centralisé permette d'assurer avec une probabilité élevée la récupération complète du message en choisissant un code approprié [80, 121], il ne permet pas un bon passage à l'échelle. Le codage distribué, quant à lui, propose une solution plus intéressante pour le stockage de donnée dans un réseau à grand échelle pour assurer la persistance, la fiabilité et la résistance aux pannes sans besoin de nœud de contrôle. Nous prenons en exemple le protocole de dissémination de données du cas où K nœuds parmi N capteurs disposent de données à transmettre. Chaque nœud génère un paquet de données. Le but du stockage distribué est de permettre de sauvegarder K paquets dans N nœuds de façon à établir la capacité de reconstruire ces K paquets initiaux à partir de n'importe quels K voisins. Ceci peut être vu comme un codage de réseau linéaire sur le graphe biparti qui relie les données à transmettre et l'index du nœud de stockage. Le premier travail dans un contexte distribué est d'employer le Growth code selon [71]. Ensuite, les solutions développées à l'aide du code RLF ont été proposées pour augmenter la tolérance aux pannes [39, 166, 125, 2]. Il a été montré que le code RLF est suffisant pour résoudre le problème de diffusion multiple [60]. Cependant, d'un point de vue de la complexité de calcul, l'usage de code fontaine LT est plus attirant. Le code LT est employé dans plusieurs protocoles de dissémination comme la technique de marche aléatoire [22, 91, 88]. Cette dernière donne une bonne performance même pour le cas où les informations sur la localisation des capteurs ne sont pas disponibles à l'encodeur [8, 179].

2.7 conclusion

Les codes correcteurs d'erreur permettent de lutter contre l'effacement dans les canaux de transmission. Dans ce chapitre, nous avons fait appel aux définitions et notions fondamentales relatives aux codes correcteurs d'erreurs. Les différents types de codes à effacement ont été présentés : codes à rendement fixe (MDS, LDPC) et les codes sans rendement (RLF, LT). Ce dernier est également connu sous le nom des codes fontaine.

Dans les chapitres qui suivent, nous proposons de nous focaliser autour de l'utilisation des codes fontaine. Le principe des codes fontaine consiste à transmettre différentes combinaisons

d'information jusqu'à ce que l'accusé de réception soit reçu : c'est très intéressant car ces codes permettent de fiabiliser les communications sans avoir besoin de la connaissance de canal à priori. Ainsi ils ont été particulièrement utilisés dans les applications de diffusion de donnée, les transmissions multimédia à temps réel et le stockage distribué.

3

Codes Fontaines dans les Réseaux de Capteurs sans fil

Les réseaux de communication actuels ont la capacité d'interconnecter des milliards de dispositifs (ordinateurs, téléphones, PDA, capteurs, etc.) et permettent ainsi de partager les ressources numériques indépendamment de leur localisation physique. Cette croissance des réseaux a été grandement favorisée par le développement des technologies de transmission sans fil (réseau de téléphonie cellulaires, WLAN, WPAN, WiMax, etc.) qui ont permis d'augmenter à moindre coût de déploiement un grand nombre et différents types de terminaux communicants.

Actuellement, il est possible d'embarquer sur un capteur un module permettant la transmission RF de données mesurées dans l'environnement. Dans ce contexte, il devient possible de déployer des réseaux de capteurs pour fournir un réseau intelligent capable de mesurer et d'interagir avec l'environnement où ledit réseau est déployé. Ce chapitre a pour objet d'introduire la problématique de l'apport des codes fontaine dans les réseaux de capteurs sans fil *Wireless Sensor Network* (WSN). Ces capteurs sont devenus des éléments incontournables dans tous les systèmes où les informations issues de l'environnement extérieur sont nécessaires pour évaluer la situation et agir [56]. Chaque nœud est composé principalement d'un ou de plusieurs émetteurs et récepteurs radio, une unité de traitement et une source d'alimentation. Ces nœuds ont pour but de détecter, surveiller et collecter les informations d'une zone géographique donnée et transférer ces informations à une ou plusieurs entités centrales communément appelées *puits*.

Les WSNs se différencient des autres réseaux de communication sans fil principalement par deux caractéristiques. La première différence est la *contrainte en énergie*. En effet, les capteurs sont souvent alimentés en énergie à l'aide de batteries. De ce fait, il est nécessaire de limiter la consommation énergétique du système pour augmenter sa durée de vie. Une part importante de la consommation est due aux émissions et réceptions des paquets. Ainsi, il est important

de limiter le nombre et la puissance des transmissions dans tout le réseau pour préserver les ressources globales.

La seconde différence est liée à la *fiabilité des transmissions* dans ce type de réseau. De par le mode de déploiement de ce type de réseau, les transmissions sont particulièrement sujettes aux erreurs. En effet, ces réseaux peuvent être déployés dans des environnements contraignants : peu connus et peu maîtrisés, difficiles d'accès, dans des conditions climatiques difficiles (chaleur, humidité, etc.). Ainsi, le type de canal de transmission est difficile à prédire et les capteurs eux-mêmes sont sujets aux erreurs de fonctionnement.

Dans de telles conditions, assurer la fiabilité de la transmission des données tout en préservant l'énergie est le point dur majeur. Beaucoup de solutions protocolaires ont été proposées pour fiabiliser de tels réseaux. Ces solutions ont été développées :

- au niveau routage : comme par exemple le routage robuste [46], le routage par gradient [100], le routage par diffusion [174] ou encore le routage par coordonnées virtuelles [154],
- au niveau MAC : à l'aide d'un protocole d'accès CSMA/CA [48], l'utilisation de schéma de veille/éveil [173, 122], utilisation d'acquittements (ARQ/HARQ [33]), la demande de retransmission par des algorithmes adaptatifs [47, 119, 114], etc.

Dans ce chapitre de thèse, nous proposons d'utiliser les codes fontaine pour palier à ces deux problèmes. En effet, ces codes possèdent la particularité de pouvoir fonctionner avec un acquittement global qui est transmis lorsque l'ensemble des paquets a été correctement décodé. La solution de codage pour fiabiliser la transmission a été préalablement connue sous le nom de système de protection de type *Forward Error Correction* (FEC) utilisant le code à effacement à rendement fixe (cf. code de Golay, code de Hamming, etc.). Or, nous l'avons vu dans le chapitre précédent, les codes fontaine sont sans rendement et adaptent le débit du code aux pertes du réseau. Nous nous intéressons donc à étudier cette nouvelle solution de codage qui nous semble intéressante et qui nous permettra de réduire le nombre de transmissions et la consommation énergétique dans le réseau.

Ce chapitre de thèse est organisé de la façon suivante. Dans un premier temps, nous présentons les caractéristiques des WSNs en général. Puis, après avoir décrit l'état de l'art relatif à l'utilisation des codes fontaines pour les WSNs, nous dégagerons nos principaux enjeux et motivations. Nous étudions ensuite l'utilisation de codes fontaine dans WSNs. Nous discuterons de la performance selon les critères énergie/délai. Selon notre étude réalisée, nous avons relevé durant notre implémentation un problème de débordement du trafic pendant le mécanisme d'accusé de réception.

Les études dans ce chapitre ont contribué à deux principaux résultats de recherche. Ces résultats se concentrent sur les deux points durs dégagés dans notre analyse concernant l'utilisation de codes fontaine pour les faibles quantités de données comme rencontrés dans les WSNs.

- La première étude concerne l'analyse de performance des codes fontaine pour les faibles quantités de données. Le résultat est exprimé en terme de consommation d'énergie et de délai de transmission. Cette étude a donné lieu à une publication dans une conférence nationale GRETSI'2009 [10].
- L'analyse du problème du débordement dans la gestion de l'acquittement globale de transmission a également été présentée et conduit à la publication d'un rapport de recherche INRIA [14]. Ce dernier a été également soumis à une conférence internationale PIMRC'2011.

3.1 Réseau de capteurs

Les réseaux de capteurs sont composés de capteurs spatialement distribués pour détecter, collecter et surveiller les données et les transférer vers un *puits* de collecte. Nous pouvons citer quelques applications de réseaux de capteurs : *les applications militaires* (la localisation des soldats et équipements, la détection d'intrusion), *la surveillance environnementale* (signalisation de feux de forêts, systèmes de détection d'inondation, collecte de données agricole), *les applications médicales* (contrôle des équipements médicaux, surveillance des malades), *les environnement intelligents* (maison intelligente, jeux interactifs, positionnement de véhicules et logistiques, détection d'intrusion et surveillance des installations industrielles), etc. . . [5, 34]. Dans cette partie, nous allons introduire les caractéristiques générales des capteurs, ses principes et leur fonctionnement, ainsi que les différentes topologies de réseau formées à partir de ces capteurs.

3.1.1 Architecture d'un capteur

Un capteur est une petite entité électronique qui est composée d'une *unité capteur* qui s'occupe de mesurer des grandeurs physiques variées (pression, température, intensité lumineuse, etc. . .), d'une *unité de traitement* qui se compose d'un processeur et d'une mémoire interne et d'une *unité de contrôle d'énergie* qui gère la batterie. Ces unités sont intégrées dans un capteur comme présenté dans la Figure 3.1. Chaque nœud capteur possède une capacité d'acquérir, de traiter et d'acheminer les données de la zone de mesure au puits.

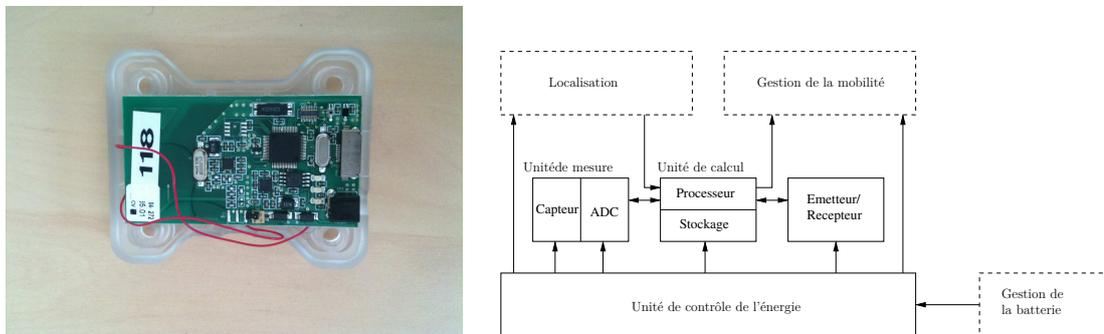


FIGURE 3.1 – Un capteur et son architecture [5]

3.1.2 Types et topologies de réseau

Types de réseaux Les réseaux de capteurs peuvent se décomposer en deux familles par rapport à leur taille :

- **Les réseaux à faible portée** : Ces réseaux sont de taille réduite et la communication se fait le plus souvent directement entre les capteurs et le puits. On peut citer par exemple les réseaux personnels sans fil *Wireless Personal Area Network* (WPAN) , les réseaux corporels *Wireless Body Area Network* (WBAN), etc. Un réseau personnel WPAN (cf. Figure 3.2) est un réseau qui relie des dispositifs habituellement centrés autour d'un

individu. Le réseau WBAN (cf. Figure 3.3) est un réseau spécialisé dans la surveillance médicale qui relie les capteurs attachés à un patient.



FIGURE 3.2 – Réseau personnel

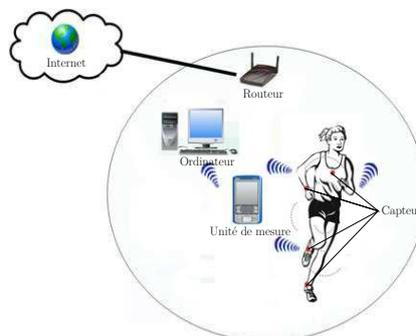


FIGURE 3.3 – Réseau corporel

- **Les réseaux à grande échelle** : Ces réseaux ont pour objet de couvrir une large surface de mesure pour mesurer les quantités physiques ou détecter des événements. Les capteurs s'auto-organisent de manière ad-hoc sans infrastructure fixe. Les communications se font par transmissions multi-sauts pour acheminer les données d'un nœuds donné à une ou plusieurs stations de base qui jouent le rôle d'interface entre les applications et les capteurs selon la Figure 3.4. On peut citer plusieurs exemples du déploiement de capteur dans ce contexte, par exemple, les déploiement de capteurs sur des volcans [168], dans une application militaire [37], etc.

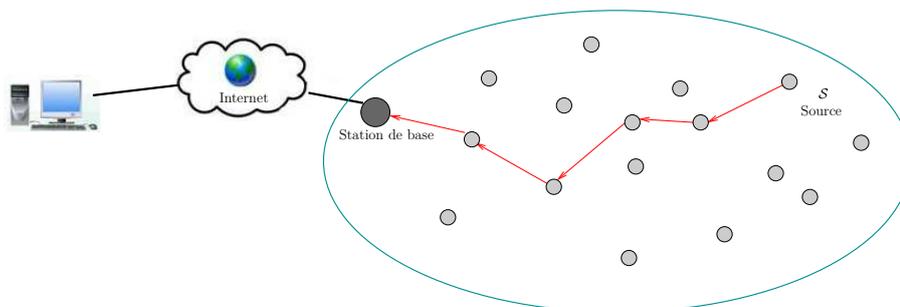


FIGURE 3.4 – Environnement monitoré par des capteurs à grand échelle

Topologies de réseau Selon les différents besoins des applications, les réseaux de capteurs peuvent prendre diverses topologies de réseau. Par exemple, pour des applications de type WPAN ou WBAN, le réseau est de faible taille et présente principalement une topologie en étoile. Pour des applications à plus grande échelle, par exemple, la surveillance géographique, on peut soit concevoir une topologie maillée régulière, soit un déploiement aléatoire de nœuds.

Il est possible de dégager deux types de topologies élémentaires présentes dans un réseau de capteurs :

- **Topologie en étoile** : un système uni-saut où tous les nœuds ne dialoguent qu'avec le puits.
- **Topologie maillée** : un système multi-sauts où les données sont à acheminer depuis la source vers le destinataire en passant par plusieurs nœuds intermédiaires.

3.1.3 Problématique des réseaux de capteurs

Étant donnée leur petite taille, les capteurs disposent généralement de ressources énergétiques limitées et non-remplaçables. Ceci est la raison pour laquelle l'énergie est souvent le critère le plus précieux d'un réseau de capteurs car elle influe directement sur la durée de vie des capteurs, donc du réseau entier. Les travaux de recherche dans les WSNs sont ainsi orientés vers cette problématique importante.

Maximisation de la durée de vie du réseau La maximisation de la durée de vie du réseau peut être obtenue par les différentes techniques de conservation d'énergie existantes. La première technique vise à réduire l'activité radio pour éviter la surconsommation. Ceci peut être implémenté par la mise en place d'un ordonnancement du temps de communication par un mécanisme de veille/éveil par un TDMA, etc. La seconde technique s'oriente vers le traitement de données dans le réseau dans le but de contrôler la quantité de données échangée dans le WSNs. En outre, l'exploitation de la mobilité peut être envisagée pour réduire la consommation du système [126].

Fiabilisation des transmissions Les perturbations dans les canaux sans fil sont dues principalement au bruit, aux défaillances des nœuds, aux interférences, aux évanouissements, etc. Ces perturbations sont des problématiques majeures qui dégradent considérablement la qualité des transmissions. Woo et al. [170] ont caractérisé, par simulation, la caractéristique du lien sans fils (cf. Figure 3.5) ce qui nous permet d'observer le niveau de la fiabilité de transmission dans des WSNs. Pour ce faire, une plateforme de simulation de Berkeley Mica mote avec TinyOS a été exploitée pour simuler une transmission d'un réseau en ligne où les capteurs sont espacés d'une distance d'environ 0.6 m l'un de l'autre. Chaque capteur émet 200 paquets au débit 8 paquets/s à une puissance de transmission fixée. Lorsque les autres nœuds reçoivent des paquets, ils comptent le nombre de réceptions. Nous pouvons visualiser sur la Figure 3.5 le taux de transmission des paquets à chaque distance donnée.

Les résultats obtenus par cette simulation permettant de visualiser la qualité de la transmission selon la Figure 3.5. Cette figure présente le taux de succès de transmission en fonction de la distance entre l'émetteur-récepteur. Nous constatons que la réception obtenue par simulation est probabiliste. Pour une distance supérieur à une distance donnée (ex. 12m), le taux de réception devient quasiment nul. Nous notons que le taux de réception ne peut pas atteindre du 100% même pour une distance très faible (ex. $\leq 3m$).

Cette étude a montré que dans un cas général, la transmission dans les WSNs n'est pas fiable. Il est donc important de faire appel aux techniques qui permettent d'augmenter le taux de succès de transmission. Dans ce chapitre de thèse, nous proposons d'analyser plusieurs techniques qui nous permettront d'assurer la fiabilité de transmission dans des canaux sans fil. Ces techniques

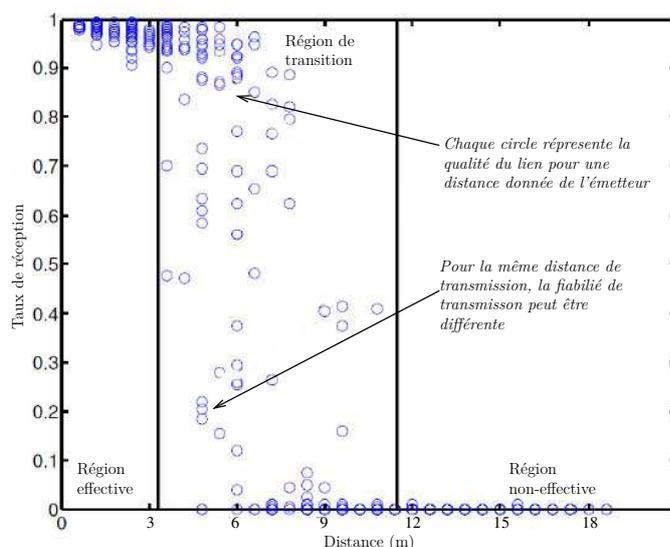


FIGURE 3.5 – La fiabilité de transmissions dans WSN [170]

incluent les solutions protocolaires et les solutions de codage par codes à effacements que nous discuterons ensuite.

Nous pouvons citer ici plusieurs solutions protocolaires qui ont pour objectif de fiabiliser les transmissions. La première approche appelé *Automatic Repeat reQuest* (ARQ) consiste premièrement à détecter les erreurs par un code détecteur d'erreur, puis envoie une requête automatique de répétition ARQ pour demander la retransmission de paquets manquants [150]. La technique ARQ ne nécessite pas de traitement de données supplémentaires au niveau des nœuds intermédiaires dans le réseau. Cependant, il ne passe pas à l'échelle pour un nombre de capteurs élevé. En plus, cette technique implique une grande latence ce qui pose problème dans des applications temps réel et des communications longue distance. D'autres techniques qui permettent d'optimiser la demande de retransmission par algorithmes adaptatifs sont proposées : le protocole SRM, RMTP, etc [47, 119, 114]. Cependant, certaines de ces techniques ne semblent pas adaptables [113], par exemple, lorsque le canal de retour est très limité, les techniques précédemment proposées ne sont pas exploitables.

Pour réduire au minimum l'usage du canal de retour, la solution de codage parait très intéressante. La première solution de codage proposé est appelé *Forward Error Correcting* (FEC) code. Ce code a permis d'obtenir une transmission plus robuste à l'aide de redondance ajoutée au système.

3.2 Codes fontaine pour les réseaux de capteurs sans fil

Du fait de la nécessité de fiabiliser la transmission sur un réseau de capteurs sans fil et grâce à la propriété intéressante qui concerne la gestion des accusés de réception de codes fontaine, nous étudions l'avantage introduit par les codes fontaine pour les WSNs.

3.2.1 Etat de l'art

Dans certains domaines d'application, les codes fontaine ont été montrés avantageux pour contourner les effacements et les erreurs durant une transmission. Ils ont été déployés, par exemple, dans le protocole de diffusion ou dans le stockage distribué de donnée. Nous commençons par donner un état de l'art général de travaux de recherche par rapport à ces contextes d'études.

L'objectif de cette thèse est de fiabiliser la transmission des données pour un réseau WSN. Les codes fontaine sont intéressants dans ce contexte car :

- ils réduisent considérablement le nombre d'acquiescement,
- ils s'adaptent de façon intrinsèque aux pertes dans le réseau.

Dans les WSNs, le code fontaine est notamment utilisé dans un domaine de stockage distribué pour diffuser avec fiabilité les données dans tous les capteurs du réseau. Nous pouvons trouver, par exemple, l'utilisation du code RLF[90], LT[91], *growth code*(une variante du code LT[71]), Raptor[9], etc... Le but du stockage distribué est de répartir K fragments d'information et de les disséminer de façon à permettre de récupérer l'information totale par un nœud quelconque après avoir récupéré $(1 + \epsilon)K$ paquets encodés par ses voisins. Plusieurs topologies de réseau ont également été étudiées comme la topologie en grille [39] ou dans un schéma de décodage décentralisé qui permet d'adapter la distribution de données pour diverses topologies de réseau [91]. Dans [115], un codage conjoint source-canal a été proposé pour améliorer l'efficacité du système de stockage distribué.

En dehors du stockage distribué de données, le code fontaine a également été utilisé pour la reprogrammation des capteurs qui nécessite un taux de fiabilité de transmission élevée. Cette tâche consiste à programmer les nœuds capteurs en distribuant le code source vers ces nœuds. Dans ce contexte, il est important que les programmes soient bien initialisés et que la transmission par diffusion soit fiable. Pour assurer la fiabilité de cette diffusion, le code fontaine a été proposé comme une solution pertinente [132] pour remplacer la méthode d'accusé de réception traditionnelle avec demande de retransmission [63], coûteuse en transmissions et ne permettant pas un bon passage à l'échelle. Plusieurs codes fontaine ont été étudiés dans ce contexte. Nous trouvons, par exemple, l'usage de code RLF et LT dans [132] ainsi que le code online [171]. Ces études ont montré l'intérêt qu'apporte le code fontaine pour garantir la robustesse dans WSNs en termes de nombre de transmissions totales et du compromis intéressant entre le nombre de transmissions et la complexité du décodage.

3.2.2 Motivation de notre étude

Dans cette thèse, nous nous intéressons à caractériser les performances du code fontaine pour la transmission de données de bout-en-bout d'une source vers un destinataire du réseau de capteurs ; où le destinataire est le puits. Plusieurs problématiques se dégagent dans ce contexte :

- Sachant que la taille des paquets émis par les capteurs est en général assez petite, est-ce que le code fontaine est tout de même une solution qui permet d'améliorer les performances? Notons que les familles de codes fontaine ont été modélisées pour être asymptotiquement optimales pour une grande dimension du code. Nous nous intéressons au cas de petite dimension du code K . Nous cherchons donc à voir si dans ce contexte, le rendement du code est toujours bon.
- Alors que les codes fontaine ont été conçus pour une utilisation point à point ou pour de la diffusion, Comment se comporte la transmission d'un flux encodé selon un code fontaine le long d'une route où le nombre de sauts devient important (ex. >5)?
- Peut-on améliorer les performances du code fontaine dans un réseau de capteurs en introduisant dans le réseau des combinaisons de paquets supplémentaires par du codage réseau?
- Comment se comporte un routage opportuniste basé sur la diffusion contrôlée des données avec un flux du type code fontaine?

Cette thèse a pour objet de donner des éléments de réponse à ces questions. Pour cela, nous allons d'abord présenter les outils et modèles considérés dans ces travaux.

3.3 Modèles et Outils

Nous avons fait appel à une plateforme de simulation WSN¹, un simulateur à événement discrets à base de composants qui permet de simuler différentes applications, protocoles de routage et communications radio pour les réseaux de capteurs sans fil [1]. Nous considérons les jeux de composant radio correspondant à un modèle de Chipset CC1100 fabriqué par Texas Instruments, Inc [67]. Des CC1100 sont caractérisés par leur très faible consommation et leur besoin réduit en composants externes, ce qui représente des solutions Radio-Fréquence (RF) peu onéreuses. Dans cette partie, nous étudierons cette plateforme de simulation. Notre outil de simulation et les paramètres de simulation sont ainsi proposés.

3.3.1 Solutions Protocolaires

Des protocoles spécialisés ont été développés pour communiquer à faible portée et à faible consommation. La conception des protocoles de gestion des couches PHY et MAC a été réalisée par l'IEEE en particulier dans le groupe IEEE802.15.4 [48]. Des protocoles de routage et transport ont été développés et également standardisés, par exemple avec Zigbee [6].

La pile protocolaire de chaque capteur est constituée de plusieurs couches du modèle de référence OSI et se présente selon la Figure 3.6. Nous détaillons dans cette partie le principe de chaque couche protocolaire présente dans un capteur. Nous définissons ici les protocoles et paramètres choisis dans cette thèse.

Avant de détailler chaque couche, une des particularités des réseaux de capteurs est la définition de plans de gestion transversaux aux couches. Ces plans de gestion ont pour objet de mieux fiabiliser la transmission, de permettre l'utilisation d'algorithmes économes en énergie et qui passent à l'échelle. Trois plans de gestion ont été définis selon [5] (cf. Figure 3.6). Le premier plan traite de la *gestion de mobilité* du capteur qui gère la localisation et l'ordonnancement des tâches relatives à la mobilité des capteurs. La *gestion des tâches* permet aux capteurs d'adapter leurs actions à leur environnement (trafic, événements, etc. . .). Cette dernière tâche peut aussi fournir des outils pour faire coopérer les capteurs entre eux. Enfin, la *gestion de l'énergie* s'intéresse à l'adaptation de la puissance de transmission et l'activation de diverses fonctions du capteur.

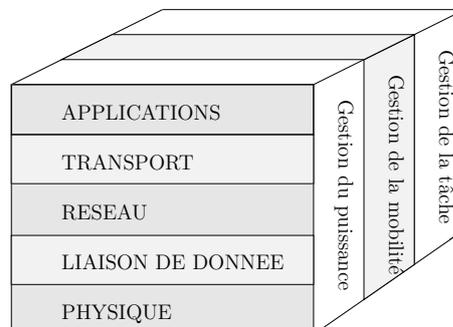


FIGURE 3.6 – Pile protocolaire du capteur

1. <http://wsnet.gforge.inria.fr>

Les capteurs que nous employons durant la thèse ont les caractéristiques suivantes :

La couche physique

La couche physique s'occupe de la transmission et de la réception des paquets sur le canal radio. Elle se compose de l'émetteur RF basé sur la norme IEEE802.15.4 qui a pour tâche le contrôle de la fréquence et la puissance d'émission : la génération de la fréquence porteuse, la détection du signal et la modulation. Selon cette norme, 27 canaux de communication sont alloués sur 3 bandes de fréquences ouvertes destinées au domaine industriel, scientifique, et médical (ISM) : 2.4GHz, 868MHz et 902MHz. Dans notre étude, nous avons choisi de travailler sur la bande de fréquence 868MHz où la norme permet d'utiliser une technique de partage de canal à étalement de spectre à séquence directe *Direct Sequence Spread Spectrum* (DSSS). Nous appliquons la modulation à déplacement de phase binaire *Binary Phase Shift Keying* (BPSK). Le débit de transmission est de 20 Kbit/s. La puissance de transmission est fixée à 10 dBm (10 mW).

La couche liaison de données

La couche liaison de données ou *Medium Access Control* (MAC) gère les accès répartis au médium radio, le multiplexage de flux de données et le contrôle d'erreurs. Il assure la communication point-à-point entre deux entités communicantes. Dans un environnement mobile et bruyé, cette couche doit s'occuper également de l'accès multiple de canal qui peut introduire des interférences et entraîner la perte d'une partie de données transmises. Il s'agit de *collisions* de paquets. La norme IEEE802.15.4 [48] propose deux méthodes d'accès au médium : un mode non coordonné *Distributed Coordination Function* (DCF) qui utilise la technique d'accès au canal du type *Carrier sense multiple access with collision avoidance* (CSMA/CA) et un mode coordonné *Point Coordination Function* (PCF) qui se base sur des trames de balise ou *beacon* pour synchroniser les nœuds du réseau.

La méthode CSMA/CA se base sur l'écoute d'un canal pour éviter les collisions de paquets dues à leur transmission simultanée. Chaque nœud ne transmet que si le canal est libre. S'il détecte le canal comme occupé, le nœud attend une période aléatoire appelée *backoff* avant de tenter de transmettre à nouveau. Si cette tentative d'envoi se déroule sur une échelle de temps qui dépasse une unité de temps défini *timeout* ou si le nombre de tentatives dépasse le seuil prédéfini, l'émetteur rejette le paquet et annule la tentative d'émission.

La couche réseau

Pour les réseaux de capteurs, plusieurs algorithmes de routage dédiés ont été développés. Ils sont souvent spécifiques à l'application du réseau de capteurs. Il existe 3 principales technologies de routage [167] :

- **Orienté données** : Dans un protocole de routage orienté données, le réseau possède une grande base de données interrogée par des requêtes envoyés par le puits. Les protocoles les plus connus sont issus de la *diffusion directe* qui consiste à envoyer un premier message contenant la requête (par ex : $T^{\circ}\text{mesure} > 18^{\circ}\text{C}$) par diffusion. Tous les nœuds ayant une

réponse lui répondent en suivant le chemin le plus intéressant parcouru par la requête. Ce schéma de routage s'inspire du routage réactif.

- **Géographiques** : Dans ce cas, on considère que les nœuds ont une position connue (par GPS, système de localisation ou par localisation relative estimée à partir du graphe de voisinage). Chaque nœud possède une mesure de sa distance au puits. Quand un émetteur envoie un paquet au puits, il inclut sa position/distance dans l'entête du paquet. Tous les nœuds qui reçoivent le paquet le transmettent si leur distance du puits est plus petite. Nous trouvons, par exemple, le protocole de routage *Self-Selective Routing* (SSR), etc.
- **Hiérarchiques** : Le protocole de routage hiérarchique permet d'exploiter l'auto-organisation du réseau. Les capteurs s'organisent pour former des clusters où tous les messages sont envoyés vers un chef de cluster ou *cluster head*. Un deuxième mécanisme de routage entre les chefs de cluster permet d'acheminer les messages vers le puits.

Dans cette thèse, dans un premier temps (Chapitre 3 et 4), nous utiliserons un routage basique figé qui nous permettra d'analyser les performances et l'impact du code fontaine sur un réseau linéaire. Dans le dernier chapitre, nous considérons un réseau à plus grande échelle et nous appuierons pour cela sur des algorithmes de routage par diffusion de la famille de routage par gradient ou *Gradient Broadcasting* selon le routage géographique. Nous montrerons par la suite que cet algorithme a pour avantage de réduire le nombre de paquets de contrôle pendant la transmission. Leur structure est aussi particulièrement favorable pour le code fontaine.

La couche transport

La couche transport est requise dans le cas où les capteurs doivent établir des connexions avec contrôle de transmissions TCP avec Internet ou réseaux externe et gérer les communications de bout en bout. Nous n'utilisons pas cette couche dans cette étude.

La couche application

Chaque nœud récolte l'information et dispose de l'information à acheminer vers le puits selon les différents scénarios de transmission et diverses topologies du réseau. Nous implémentons le code fontaine juste au dessous de cette couche pour fiabiliser la transmission de l'ensemble des données.

3.3.2 Implémentation du codage dans la pile protocolaire

Contrairement au schéma de transmission classique où l'accusé de réception est fait sur chaque paquet individuellement l'accusé de réception avec codes fontaine est déclenché de bout en bout par le destinataire une fois la totalité des symboles initiaux décodés avec succès. Dans ce scénario, ce n'est plus la couche liaison de données mais la couche réseau et application qui s'occupent de la vérification de bout en bout et de l'envoi des acquittements.

La Figure 3.7 donne une vue générale de l'implémentation du processus de codage fontaine aux différents niveaux de la pile protocolaire. La source s'occupe de la fragmentation des données et du processus d'encodage. Quant au destinataire, il s'occupe du processus de décodage ainsi que de l'envoi de l'acquiescement pour mettre fin à la transmission de la source.

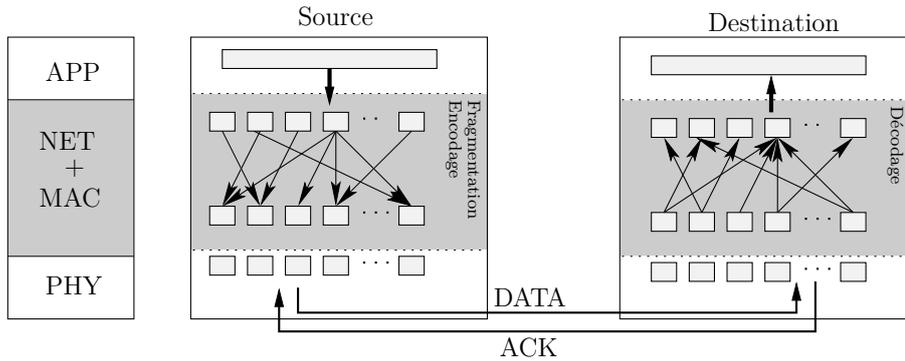


FIGURE 3.7 – Implémentation de la couche de codage sur la pile protocolaire du capteur

Encapsulation

Pour intégrer des codes fontaines dans le réseau, il est nécessaire de modifier l'encapsulation des paquets. Pour que le décodage soit possible au destinataire, il est nécessaire d'ajouter des informations supplémentaires (principalement relatives à la matrice d'encodage) aux paquets codés. Le décodage doit connaître le degré de paquet ainsi que l'ensemble de voisins de chaque paquet encodé (mots de code). Dans ce travail, nous fournissons ces informations sous la forme de K bits supplémentaires ajoutés à chaque paquet encodé. Chaque bit correspond à l'indice du fragment. S'il est présent, ce bit est égal à un, sinon il est égal à zero. En effet, la communication de matrice de contrôle par ce moyen devient coûteuse quand le degré du paquet est faible et que K est élevé. Cependant cette technique apporte quelques avantages et simplifie le mécanisme de codage de réseau qui n'implique que des additions linéaires entre les paquets. Nous discuterons du codage réseau dans le prochain chapitre.

3.3.3 Modélisation du lien radio

Nous modélisons le canal entre deux nœuds qui se situent à une distance d_h mètres l'un de l'autre. Nous calculons d'abord le rapport signal sur bruit SNR (présenté par δ) valide quand aucun signal interfère la transmission.

$$\delta = \frac{G_t \cdot G_r \cdot \lambda^2}{(4\pi)^2 N_o R \cdot L} \cdot P_t \cdot d_h^{-\alpha} \quad (3.1)$$

où G_t et G_r le gain d'antenne de transmission et réception respectivement, λ la longueur d'onde (m), N_o le niveau du bruit blanc gaussien (dBm/Hz), R le taux de transmission (Kbit/s), L la perte de circuit, P_t la puissance de transmission (Watt) et α l'affaiblissement de propagation.

L'information transmise sur ce canal en présence de bruit est sujet aux pertes. Nous pouvons en déduire la probabilité de perte par bit *Bit Error Rate* (BER) selon [54] :

$$\text{BER}(\delta) = \alpha_m Q(\sqrt{\beta_m \cdot \delta}) \quad (3.2)$$

avec la fonction $Q(x) = \int_x^\infty \frac{1}{\sqrt{\pi}} e^{-\frac{u^2}{2}} du$.

α_m et β_m sont des constantes qui varient selon le type et l'ordre de la modulation. Par exemple, pour la modulation de type BPSK, la valeur de $\alpha_m = 1$ et $\beta_m = 2$. Les valeurs de α_m et β_m sont définies dans [177].

Durant cette thèse, nous considérons un canal à effacement où un paquet est considéré comme reçu ou totalement effacé. Nous définissons la probabilité de perte pour chaque paquet comme *Packet Error Rate* (PER). Ceci est obtenu à l'aide de l'expression $PER = 1 - (1 - BER(\delta))^{N_b}$ avec N_b nombre de bits contenus dans le paquet.

Cette expression de PER nous permet de caractériser le lien radio comme γ , la probabilité de succès de transmission d'un paquet sur le lien.

$$\gamma = 1 - PER. \quad (3.3)$$

Il faut noter que la valeur de γ varie selon les différents environnements de propagation. Nous abordons deux environnements d'étude avec la transmission en présence du bruit Gaussien additif *Additive White Gaussian Noise* (AWGN) et en présence d'évanouissement par bloc selon le modèle de Rayleigh. A partir de [177], la modélisation du canal de transmission peut être décrite comme ceci :

- **Canal à effacement AWGN** : Ce modèle de canal prend en compte seulement l'addition linéaire du bruit blanc. Il ne prend pas en compte les phénomènes d'évanouissement, de sélectivité en fréquence, de non-linéarité ou de dispersion. La probabilité du lien (γ_{awgn}) peut être exprimée à l'aide de l'équation (3.2) comme :

$$\gamma_{awgn} = (1 - BER(\delta))^{N_b} \quad (3.4)$$

- **Canal à effacement en présence d'évanouissement par bloc** : En présence d'évanouissement par bloc, l'évanouissement est supposé constant pour tous les bits dans un bloc donné. L'amplitude du signal varie selon la distribution de Rayleigh. La probabilité du lien s'écrit :

$$\gamma_{rayleigh} = \int_{\delta=0}^{\infty} (1 - BER(\delta))^{N_b} \cdot p(\delta|\bar{\delta})d\delta \quad (3.5)$$

avec la distribution de Rayleigh est sous la forme :

$$p(\delta|\bar{\delta}) = \exp(-\delta/\bar{\delta})/\bar{\delta} \quad (3.6)$$

où $\bar{\delta}$ est le rapport signal sur bruit moyen par bit.

La Figure 3.8 présente l'évaluation de la probabilité du lien γ en fonction de la distance d_h entre l'émetteur et le récepteur pour un canal à effacement du type AWGN et Rayleigh par bloc.

3.3.4 Paramètre de simulations

contexte d'étude

Les paramètres de simulation que nous avons utilisés dans ce travail sont résumés dans la Table 3.1.

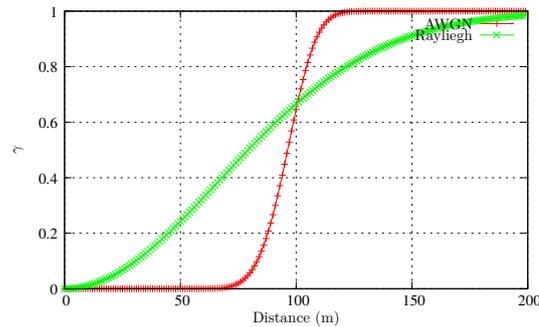


FIGURE 3.8 – La probabilité du lien en présence du bruit AWGN et évanouissement de Rayleigh par bloc

Ajustement du taux d'émission de la source

Objectif : supprimer les collisions pour se rapprocher du modèle en SNR défini précédemment (donc qui ne tient pas compte des interférences SINR)

Un facteur qui joue un rôle important sur les performances de transmission est la période d'envoi des paquets depuis la source (T_s). Un bon réglage de ce paramètre permet de limiter le nombre de collisions dans le système et ramène donc à une optimisation de la consommation d'énergie ainsi que du délai de transmission. Lorsque T_s augmente, le nombre de collisions dans le système se réduit, nous notons une baisse de la consommation d'énergie, mais, une augmentation du délai. Cependant, nous remarquons que la diminution de T_s ne réduit pas forcément le délai. T_s qui est trop petit induit une augmentation de collisions dans le système, ainsi que le nombre de retransmissions, qui ramènent à un gaspillage d'énergie et à un grand délai.

La bonne période d'envoi T_s doit être réglée de façon à allouer suffisamment de temps de propagation d'acquiescement depuis un récepteur vers un émetteur. Or, la valeur de T_s optimale n'est pas toujours évidente à définir. Ce réglage de T_s nécessite la connaissance de canal a priori portant sur le chemin et le nombre de sauts dans lequel les données et les acquiescements se propagent. La transmission avec protocole MAC du type IEEE802.15.4 impose additionnellement des caractéristiques aléatoires dû au temps d'attente aléatoire imposé par le mécanisme d'accès de canal. Pour une stratégie de relaying spécifiée dont le nombre maximum est de h sauts $\{h \in [1, n_h - 1]\}$, il faut allouer approximativement au moins un intervalle de temps optimal ($T_{s_{opt}}$) pour minimiser les pertes et les collisions introduits par la transmission dans les deux sens opposés ce qui est contraignant par la nature semi-duplexe du canal. Ce $T_{s_{opt}}$ peut être exprimé comme la latence totale de la propagation de donnée et de l'acquiescement dans les deux sens de transmission opposés multiplié par le nombre totale du sauts. Cette expression s'écrit :

$$T_{s_{opt}} \approx h \cdot (T_{data} + T_{ack}) \quad (3.7)$$

avec T_{data} et T_{ack} le délai de transmission qui prend en compte le temps de traitement, le temps de propagation et temps d'émission/réception d'un paquet de donnée et d'acquiescement au travers d'un saut. Les valeurs estimées de T_{data} et T_{ack} pour une transmission du protocole IEEE802.15.4 sont proposées dans [84].

Couches	Paramètres de Configuration
Application	Taille du réseau (nombre de sauts) : n_h Distance entre les nœuds = 85m
Réseau	Protocole MAC : IEEE 802.15.4 non slotté avec CSMA/CA Periode d'envoi de la source = 1s Codage : LT code/RLF code Nombre de fragments envoyé (Dimension du code) = K Taille de fragments = 128octets Taille d'acquittement = 36octets Taille de l'entête = Kbits
Radio	dispositif radio : Chipset CC1100 - PtxElec = 82.89mW - PrxElec = 38.88mW - PidleElec = 4.32mW Modulation : BPSK $P_t = 10\text{dBm}$ Débit de Transmission = 20Kbit/s Fréquence porteuse = 868MHz
Environnement	Modèle de propagation : AWGN/Rayleigh Taux d'Atténuation $\alpha = 2$ Bruit Blanc = -111dBm/Hz évanouissement : aucun interférence : orthogonal

TABLE 3.1 – Paramètres de simulation

3.3.5 Critères d'évaluation de performance

Le critère d'évaluation le plus classique pour assurer les performances d'un réseau se focalise sur la mesure du débit de transmission ou la probabilité de coupure. Or, dans cette thèse, nous voulons garantir la fiabilité de transmission dans des WSNs. Pour cela, les transmissions s'effectuent jusqu'à la réception complète de l'information par le destinataire. Les critères d'évaluation de performance sont ainsi exprimés par la complexité de calcul nécessaire au codage/décodage, l'énergie consommée et le délai de transmission. Sachant que l'énergie est le critère le plus important à considérer lors de la conception pour les WSNs, nous allons principalement le mettre en avant.

Énergie

L'énergie consommée pendant une communication avec codes fontaine provient d'une part de l'unité de traitement de données, de l'émission et de la réception de paquets de données et d'acquittement. Ici, le traitement de l'information, le codage et le décodage d'un paquet par exemple, nécessitent des opérations arithmétiques simples de faible complexité (cf. Table 2.2), pour une faible dimension K . Nous négligeons dans notre modèle énergétique cette consommation d'énergie. Cependant, nous comptabilisons les autres surcoûts provenant de l'accroissement

de la taille des paquets du à l'ajout de l'entête contenant l'information de codage nécessaire pour le décodeur. Nous prenons en compte également la redondance du code sachant que le décodeur nécessite en général $K + \epsilon$ paquets pour récupérer K fragments initiaux transmis.

Dans les études relatives à la consommation d'énergie des codes fontaine pour une transmission, à l'aide du modèle énergétique développée dans [177], nous considérons les mesures suivantes de la quantité d'énergie nécessaire à la transmission (E_{tx}), réception de paquet (E_{rx}), ainsi qu'au mode de veille de capteur (E_{idle}).

$$E_{tx} = \frac{N_b}{R} P_{txElec} + \alpha P_t, \quad (3.8)$$

$$E_{rx} = \frac{N_b}{R} P_{rxElec}, \quad (3.9)$$

$$E_{idle} = T_{idle} P_{idleElec} \quad (3.10)$$

avec P_{txElec} et P_{rxElec} la puissance consommée pour émettre et recevoir un bit d'information selon les spécifications de dispositifs radio, $P_{idleElec}$ la puissance consommée pendant l'état de repos, T_{idle} la durée de repos durant la transmission d'un paquet, N_b la taille de paquet, R le débit de transmission (bit/s), α le facteur amplificateur et P_t la puissance de transmission. Nous définissons dans la suite l'énergie consommée comme étant la consommation d'énergie par chaque fragment d'information envoyé.

Délai

Nous définissons le délai de transmission comme l'intervalle de temps nécessaire pour transmettre un message \mathcal{M} divisé en K fragments d'une source S au destinataire D. Ceci est défini par la durée totale de transmission de l'ensemble des $K + \epsilon$ paquets envoyés de la source S au travers du réseau jusqu'à la récupération complète (la réception et le décodage avec succès) des K fragments d'information du destinataire. Afin de comparer avec le cas sans codage, cette durée totale est divisée par le nombre K de fragments initiaux. Le délai de transmission se présente par la suite comme le délai de transmission pour envoyer un fragment d'information et prend en considération le temps d'émission, le temps de propagation, le temps de calcul et le temps de réception d'un paquet.

3.4 Performance d'un code à faible dimension

Etant conçu initialement pour les applications multimédia où la transmission concerne principalement de grandes quantités de données, il a été montré dans [95] que la redondance introduite par un code fontaine est asymptotiquement négligeable par rapport à la dimension du code pour une liaison point à point. Pour atteindre ce type de performances, la dimension du code est de l'ordre de $K \simeq 10000$. Or, pour un réseau de capteurs sans fils dans lequel il s'agit souvent d'envoyer de faibles quantités de données à travers plusieurs sauts avant d'atteindre le puits, l'expression analytique de la redondance n'est plus négligeable par rapport à la dimension K , c'est-à-dire par rapport à la taille des messages. Jusqu'à présent, nous n'avons pas trouvé dans la littérature de travaux qui s'intéressent à quantifier et analyser le taux de redondance du code pour une petite dimension du code. Dans ces travaux, nous considérons principalement un code avec une dimension de $K = 10$ ou $K = 100$, la taille d'un fragment mesurant 128 octets en général.

Dans nos travaux, nous avons en effet bel et bien observé que la redondance introduite par le code devient d'autant plus importante que la dimension du code est petite. Néanmoins, nous avons mis à jour que même dans ce contexte peu favorable aux codes fontaines, l'ensemble des propriétés de ce code permettent tout de même d'améliorer la consommation énergétique d'une transmission par rapport à une fiabilisation basée sur un mode ARQ, et ce pour différentes topologies élémentaires de réseau. En effet, le code en lui-même introduit une redondance ϵ non négligeable mais cette redondance n'est pas aussi importante que la surcharge liée à l'acquittement jusqu'à réception des messages non codés dans le réseau.

Dans toutes les études qui suivront, notre objectif est de garantir la fiabilité des transmissions. En d'autres mots, on veut que *tous* les K paquets transmis par la source arrivent à la destination. Ainsi, nous comparons pour différentes topologies de réseau la performance d'une transmission où les K paquets sont transmis à l'aide d'un code fontaine et la performance d'une transmission classique où la fiabilité est garantie à l'aide d'un protocole ARQ. Nous rappelons que dans ce protocole, chaque message est acquitté saut par saut. Nous rappelons que la transmission d'un code fontaine entraîne une surcharge supplémentaire due :

- à l'élargissement de l'en-tête du paquet ; dans le cas avec codage, au paquet codé de 128 octets s'ajoute un en-tête de K bits. Par exemple, pour $K = 100$, cet en-tête représente environ 9% de la longueur du paquet et pour $K = 10$ elle représente moins de 1%,
- au taux de redondance du code qui est de l'ordre de $O(\epsilon)$ pour un code LT. Ce taux de redondance représente le pourcentage de surcharge induit qui est égal à environ 11.5% et 15% pour $K = 100$ et $K = 10$ respectivement. Pour une grande valeur de dimension du code (ex. $K > 1000$), ce pourcentage de surcharge tend vers 0. Pour le code RLF, le pourcentage de surcharge est très faible (≈ 0) pour tous les valeurs de K (cf. Table 2.2 pp.20).

3.4.1 Topologie Point-à-point

Considérons premièrement un lien point-à-point où A veut envoyer K fragments d'informations au nœud B. Nous comparons deux mécanismes qui ont pour but d'assurer le succès de la

transmission : une solution protocolaire de type ARQ et une solution de codage FEC avec code fontaine RLF et LT.

- **Solution protocolaire (ARQ)** : Avec le mécanisme ARQ, A encapsule chaque fragment dans un paquet de donnée (DATA), et les transmet les uns après les autres au nœud B. Pour chaque paquet transmis, A s'assure que B a bien reçu le paquet avant de transmettre le paquet suivant. Si A n'a pas reçu l'accusé de réception (ACK) dans un intervalle de temps donné, A retransmet ce paquet jusqu'à ce qu'il reçoive l'accusé de réception. Notons que le processus d'acquiescement est réalisé pour chaque paquet de données individuellement. Bien que cette approche ne nécessite pas de traitement et de calcul par les nœuds intermédiaires, il surcharge le canal pour les messages de contrôle. Ceci devient problématique pour la transmission dans un canal asymétrique semi-duplex car ceci engendre des collisions. Pour éviter les retransmissions et l'utilisation du canal de retour, nous analysons l'approche qui consiste à encoder ces fragments par le codage fontaine avant de les envoyer.
- **Solution de codage (FEC)** : Nous nous intéressons particulièrement dans ce chapitre à l'utilisation de code fontaine comme code correcteur d'erreur FEC. Dans ce cas, chaque paquet transmis par A est encodé par les combinaisons linéaires XOR d'un certain nombre de fragments choisis dans K . A envoie consécutivement un nombre illimité de ces paquets encodés à B sans attendre l'accusé de réception pour chaque paquet de donnée (DATA) envoyé. Lorsque le destinataire reçoit suffisamment de paquets (environ $K + \epsilon$) pour pouvoir décoder correctement, il n'envoie qu'un seul paquet pour accuser la réception des K fragments. Le nombre de tentatives de transmissions d'acquiescement est réduit de K à 1 par rapport au cas précédent. En revanche, il faut remarquer que le nombre de paquets de données (DATA) transmis s'élève à $K + \epsilon$ au lieu de K sans compter les pertes de paquets dues à la redondance introduite par les codes.

Même si les codes fontaines permettent de réduire considérablement l'utilisation du canal de retour, un traitement de l'information supplémentaire est exigé soit au niveau de l'émetteur soit du récepteur. Le fait de diviser les données à transmettre en K fragments impose également une latence de $K + \epsilon$ paquets à l'émission.

La Figure 3.9 représente un exemple du scénario de transmission qui nous permet de visualiser la retransmission induite par la perte des paquets d'acquiescement pour deux scénarios de transmission : sans et avec code fontaine. Notons que pour une liaison point-à-point, les paquets de données et d'acquiescement sont transmis avec une probabilité de succès égale à γ_{data} et γ_{ack} respectivement.

Modèle analytique

Pour une liaison point-à-point, le nombre de retransmissions moyennes (\bar{N}_{tx}) pour transmettre un paquet de données avec succès sans considérer l'accusé de réception est égal à :

$$\begin{aligned} \bar{N}_{tx} &= \sum_{n_{tx}=1}^{\infty} n_{tx} \cdot \gamma_{data} \cdot \gamma_{ack} (1 - \gamma_{data} \cdot \gamma_{ack})^{n_{tx}-1} \\ &= \frac{1}{\gamma_{data} \cdot \gamma_{ack}} \end{aligned}$$

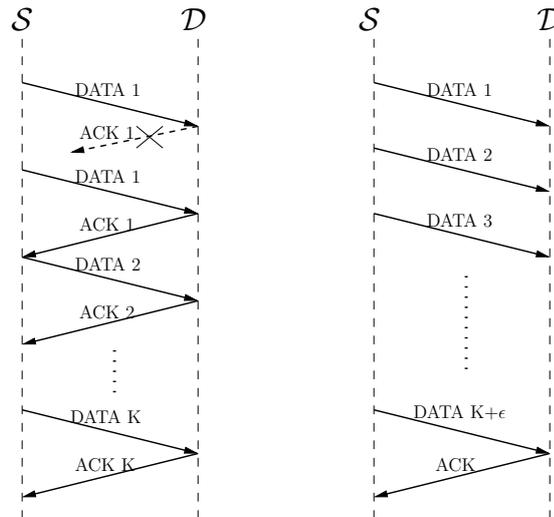


FIGURE 3.9 – Mécanisme d'acquiescement : ARQs vs. Codes fontaine

avec n_{tx} le nombre de transmissions nécessaires au succès, γ_{data} et γ_{ack} les probabilités de succès pour transmettre un paquet de données et accusé de réception au travers d'une liaison point-à-point (cf. équations (3.4) ou (3.5)). Ces derniers ont deux valeurs distinctes car les tailles N_b des paquets de données et d'acquiescement sont différentes.

Nous comparons par la suite deux techniques courantes utilisant le protocole ARQ et le code FEC de type fontaine. L'énergie totale consommée pour transmettre K fragments d'information pour le cas sans codage utilisant un mécanisme ARQ pour fiabiliser la transmission peut être exprimée, à partir des séquences de transmission décrites dans la Figure 3.9, sous la forme :

$$E_{T_S} = K \cdot \frac{1}{\gamma_{ack}} \cdot \left(\left(\frac{1}{\gamma_{data}} \cdot (E_{data} + E_{idle}) \right) + E_{ack} \right) \quad (3.11)$$

E_{data} et E_{ack} sont l'énergie consommée pour émettre et recevoir un paquet de donnée et un acquiescement. Nous avons $E_{data} = E_{tx_{data}} + \gamma_{data} \cdot E_{rx_{data}}$ et $E_{ack} = E_{tx_{ack}} + \gamma_{ack} \cdot E_{rx_{ack}}$. E_{idle} correspond à la consommation de base du capteur durant l'état de repos. Le facteur multiplicateur K/γ_{ack} dans l'équation 3.11 correspond au nombre de retransmission nécessaire pour que K fragments initiaux arrivent au destinataire.

Pour le cas avec codes fontaine, l'expression de la consommation totale d'énergie est égale à E_{T_C} . Dans ce cas, l'acquiescement global est envoyé pour acquiescer la récupération de K fragments initiaux avec succès et mettre fin à la transmission depuis la source. Au cas où l'acquiescement est perdu, la retransmission n'est effectuée que pour le dernier paquet émis. L'expression analytique du cas avec code fontaine est présentée selon l'équation 3.12.

$$E_{T_C} = (K + \epsilon - 1) \cdot \frac{1}{\gamma_{data}} \cdot (E_{data} + E_{idle}) + \frac{1}{\gamma_{ack}} \cdot \left(\frac{1}{\gamma_{data}} \cdot (E_{data} + E_{idle}) + E_{ack} \right) \quad (3.12)$$

Nous pouvons noter que dans le cas avec ARQ, la transmission de l'ACK s'effectue pour tous les K fragments tandis que dans le cas du code FEC, l'ACK n'est transmis que sur le

dernier paquet envoyé. Le facteur multiplicateur est dû au nombre de retransmissions égale à K/γ_{ack} et à $1/\gamma_{ack}$ respectivement.

Résultats de simulations

Nous choisissons les paramètres de réglage de la transmission de façon à minimiser les effets de collision en définissant la valeur de T_s à 1s. A partir des paramètres de simulation de la Table 3.1, avec le code fontaine RLF et LT, nous évaluons sur la Figure 3.10 et 3.11, la valeur moyenne en énergie nécessaire pour transmettre chaque fragment d'information lorsque $K = 10$ et $K = 100$ respectivement. Nous présentons ces résultats de simulation en comparaison avec le cas idéal où le processus d'accusé de réception est supposé parfait : chaque paquet d'accusé est supposé envoyé sans perte ($\gamma_{ack} = 1$). Cette comparaison nous permettra de visualiser les effets de perte de l'accusé sur la retransmission de données en fonction de la distance entre l'émetteur et le récepteur. Nous notons que, pour ce cas idéal, le sur-débit lié au codage (ϵ) est plus consommateur d'énergie que le gain induit. En effet, lorsque le canal de retour est fiable, la redondance introduit par le système est moins utile et introduit un surcôt du nombre de transmission. Cependant, dans un cas réel lorsque l'accusé peut être sujet à des pertes ($\gamma_{ack} \neq 1$), plus la distance est importante (donc plus le lien est dégradé), plus l'utilisation des codes fontaine RLF et LT s'avère intéressante. En effet, la perte d'accusé induit une retransmission. Or, dans le cas sans codage, cela peut se produire pour chaque paquet, tandis que pour les codes fontaine, tout se passe comme si seulement le dernier paquet devait être retransmis. Ainsi, lorsqu'un cas réaliste est considéré, pour une même quantité d'information échangée, les deux codes fontaine RLF et LT deviennent moins gourmands en énergie que le cas sans codage. Sachant que le code RLF introduit moins de redondance grâce au décodage ML efficace, ce code est plus performant que le code LT sous critère d'énergie.

En comparant la Figure 3.10 et la Figure 3.11 pour les valeurs de K égale à 10 et 100, nous constatons que lorsque K augmente ($K = 100$ par rapport à $K = 10$), l'énergie pour transmettre un fragment de donnée est réduit. Ceci provient du taux de redondance qui devient négligeable devant une dimension du code K plus élevée.

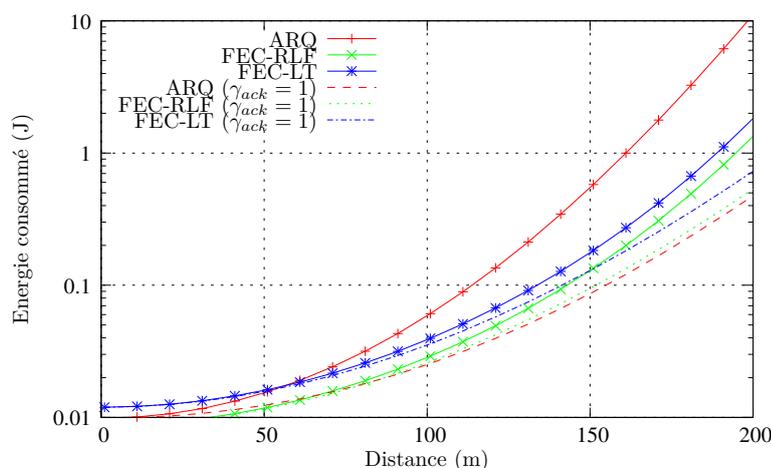


FIGURE 3.10 – La consommation d'énergie pour la transmission point-à-point, $K = 10$

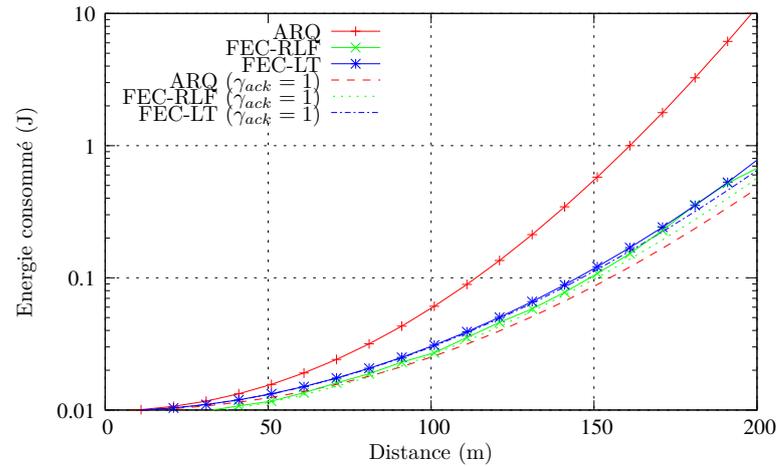


FIGURE 3.11 – La consommation d'énergie pour la transmission point-à-point, $K = 100$

Le niveau de puissance d'émission a également une incidence sur l'énergie consommée. En fixant une distance fixée égale à $d_h = 100m$ avec la dimension du code $K = 100$, en faisant varier la puissance d'émission sur l'intervalle de 0-40 dBm, nous avons cherché sur la Figure 3.12 la puissance d'émission permettant de minimiser la consommation d'énergie avec et sans codage. Lorsque la puissance de transmission est très faible, nous notons que l'énergie consommée est élevée. La courbe décroît jusqu'à un minimum local de la consommation d'énergie et puis croît proportionnellement avec la puissance de transmission. En effet, une faible puissance de transmission entraîne un taux d'effacement élevé dans le canal de transmission tandis qu'une puissance élevée conduit à un gaspillage d'énergie inutile. Ce schéma permet de guider le choix de la puissance de transmission permettant un bon compromis et minimisant l'énergie consommée.

Nous constatons que la solution du codage RLF et LT permet d'avoir une consommation d'énergie optimale à une puissance égale comprise entre [13dBm, 15dBm]. Les codes RLF et LT permettent de réduire la consommation d'énergie environ 34.4% et 10.7% respectivement par rapport au cas sans codage. Le code RLF permet de réduire le nombre d'acquitements et permet d'obtenir une consommation minimale pour tous les intervalles de puissance considérés. Le code LT quant à lui donne une meilleure performance en terme de consommation d'énergie pour la puissance d'émission faible. Dans le cas contraire où le lien est fiable, la redondance ajoutée par le code devient inutile. Nous pouvons en conclure que, bien que les codes fontaines aient été conçus à l'origine (et actuellement considérés dans les travaux de recherche) pour des applications de diffusion ou pour des codes très longs, ces codes permettent aussi de réduire la consommation d'énergie pour une liaison point-à-point pour une transmission avec acquitement. Dans un cas de transmission opportuniste, l'adaptation de la puissance d'émission nous permet d'atteindre ce gain.

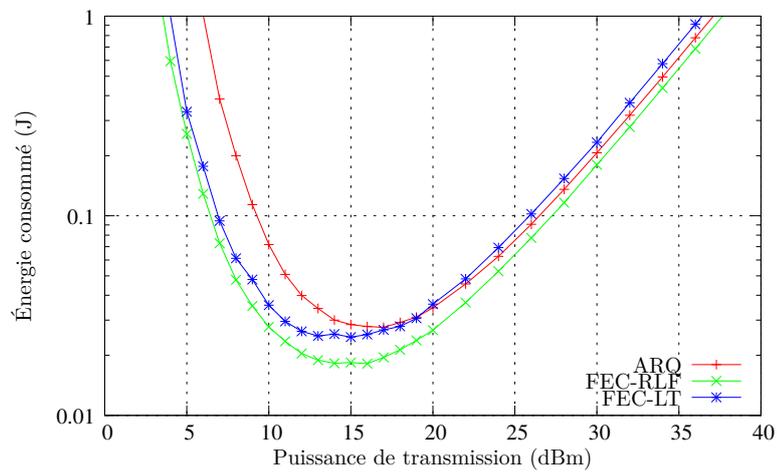


FIGURE 3.12 – La consommation d'énergie en fonction de la puissance d'émission, $K = 100$

3.4.2 Topologie en étoile

Nous réalisons la même étude comparative entre la transmission avec ARQ et le code fontaine pour une topologie en étoile comme illustré à la Figure 3.13. Nous supposons que les sources sont uniformément réparties à une distance de 150 mètre autour du puits.

Résultats de simulations

Sur la Figure 3.14, nous traçons l'évolution de la consommation d'énergie en fonction du nombre de sources activées. Nous constatons que lorsque le nombre de sources augmente, l'énergie consommée augmente qu'il y ait codage ou non. Ainsi, cette augmentation est due aux collisions et à la contention au niveau du puits. Beaucoup de paquets sont perdus de par l'expiration du temps de backoff. L'énergie augmente aussi de par la consommation due à l'écoute du canal qui augmente proportionnellement au trafic du réseau. Cependant, on peut observer qu'un réseau utilisant les codes fontaines (RLF et LT) est plus robuste à l'augmentation du nombre de sources. En effet, le fait que les codes fontaine permettent de réduire le nombre d'acquittements transmis entraîne moins de collisions durant l'accès multiple du canal. Ainsi les codes fontaine se révèlent intéressants dans un contexte non synchronisé, soumis aux collisions.

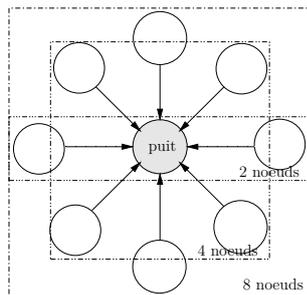


FIGURE 3.13 – Topologie en étoile

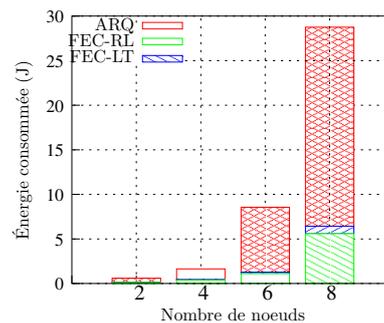


FIGURE 3.14 – La consommation d'énergie pour la topologie en étoile en fonction du nombre de sources

3.4.3 Réseau linéaire

Nous considérons maintenant dans notre étude le réseau de transmission linéaire multi-sauts illustré à la Figure 3.15. Nous pouvons évaluer la transmission qui est réalisée au travers de n_h sauts en passant par $n_h - 1$ nœuds intermédiaires avec une probabilité de passage d'un nœud à l'autre équivalente égale à γ . Ces nœuds intermédiaires sont appelés relais. Supposons que les relais soient espacés d'une distance égale à d_h . Dans cette étude, nous souhaitons mettre en avant l'impact des algorithmes de relayage déployés au niveau des relais. Nous considérons que l'acquittement est prioritaire, selon le protocole IEEE802.15.4, grâce à un temps de contention plus petit affecté aux acquittements.

Nous étudions 3 stratégies de relayage d'un code fontaine que nous comparons à 2 stratégies de relayage classique (sans codage). Nous avons toujours K fragments à transmettre vers D.

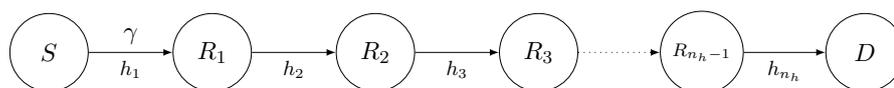


FIGURE 3.15 – Un réseau linéaire multisaut

- **Scénario SB** (sans codage, relayage par bloc) : Ici, la source transmet directement (sans codage) les K fragments. Chaque relai attend d'avoir reçu les K fragments avant de les retransmettre au relai suivant. En cas de perte, la retransmission s'effectue pour chaque fragment jusqu'à ce que l'accusé soit reçu. Avec cette méthode, bien que le nombre de transmissions ainsi que la consommation d'énergie sont minimales, la transmission souffre d'une forte latence de bout en bout.
- **Scénario SP** (sans codage, relayage par paquet) : Chaque nœud transmet les paquets directement après leur réception. Seul le destinataire transmet l'accusé de réception de chaque fragment envoyé. Cependant, en cas d'échec de la transmission de bout en bout, la retransmission doit recommencer à la source ce qui est inefficace d'un point de vue énergétique. Nous pouvons noter qu'une perte d'accusé entraîne la retransmission depuis la source.
- **Scénario CB** (relayage de Code fontaine par Bloc) : Chaque nœud attend la récupération complète des K fragments d'information par le processus de décodage avant de les ré-encoder et de les transmettre au nœud suivant. Les décodage et réencodage sont appliqués au niveau de chaque relai. A chaque saut, un seul accusé est envoyé pour accuser la réception de l'ensemble des K paquets.
- **Scénario CP** (relayage de Code fontaine par Paquet) : La source envoie une séquence encodée à l'aide d'un code fontaine au destinataire. Chaque relai retransmet uniquement les paquets encodés reçus sans les modifier. Le processus de décodage et d'accusé de réception se fait de bout en bout par le destinataire dès qu'il décode et retrouve K fragments initiaux avec succès. Dans ce cas, une perte d'accusé entraîne la retransmission depuis la source. Cependant, en comparaison avec le cas de la transmission passive sans codage, la retransmission ne s'effectue que pour le dernier paquet de donnée reçu.
- **Scénario CPR** (relayage de Code fontaine par Paquet avec Réémission) : Dans le scénario CPR, lorsque le relai ne reçoit pas de paquet, il utilise l'intervalle temporel disponible pour retransmettre un paquet reçu dans le but d'augmenter la fiabilité de la transmission des paquets déjà reçus. Pour cela, quand un paquet n'est pas reçu, le relai envoie une combinaison des paquets présents dans sa mémoire interne. Ceci peut optimiser l'allocation temporelle d'un lien radio et réduire le délai total de transmission.

Résultats de Simulations

Nous présentons à la Figure 3.16, l'énergie consommée en fonction de la distance d_h entre un nœud et son voisin d'un saut. Nous notons d'après cette figure que les codes fontaines (RLF et LT) permettent de réduire le nombre de retransmissions donc conduisent à une meilleure performance d'un point de vue énergétique. Ce gain devient plus important lorsque la distance d_h augmente (lorsqu'on se place dans un cas où chaque lien est dégradé). En comparant ces

différentes stratégies de relayage, nous avons observé que si les relais ont permis de décoder et ré-encoder le paquet (Ex. stratégie SB et CB), la consommation d'énergie est minimisée par rapport aux autres stratégies. En effet, la perte d'acquittement dans certains cas entraîne une retransmission depuis le premier saut (ex. stratégie SP et CP). Ceci est coûteux en pratique avec des pertes dans le canal de transmission. Le schéma de relayage par ré-émission où la retransmission s'effectue à partir de chaque relai sans décodage à chaque relai, ce qui donne la performance intermédiaire en terme d'énergie/complexité par rapport aux deux cas extrêmes précédemment mentionnés.

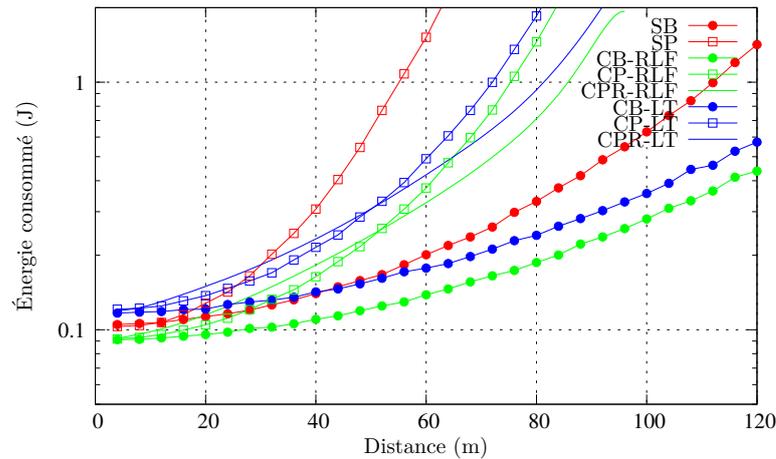


FIGURE 3.16 – La consommation d'énergie pour la transmission multisauts

Nous avons également étudié le délai de transmission pour ces stratégies de relayage en fonction de la distance d_h entre les deux voisins selon la Figure 3.17. Les courbes de délai de transmission évoluent proportionnellement à celle du cas de la consommation d'énergie de la Figure 3.16. Nous soulignons l'avantage de l'usage de code fontaine sur le délai de transmission. En effet, nous pouvons renforcer l'optimisation du délai avec codage fontaine en optimisant la durée d'attente des acquittements T_{ack} pour l'ensemble de paquets transmis avant qu'on espère d'avoir un décodage complet de l'information au destinataire. Il est cependant difficile à l'émetteur de précisément déterminer à quel moment il faut qu'on alloue ou non T_{ack} sans disposer de la connaissance du système à priori.

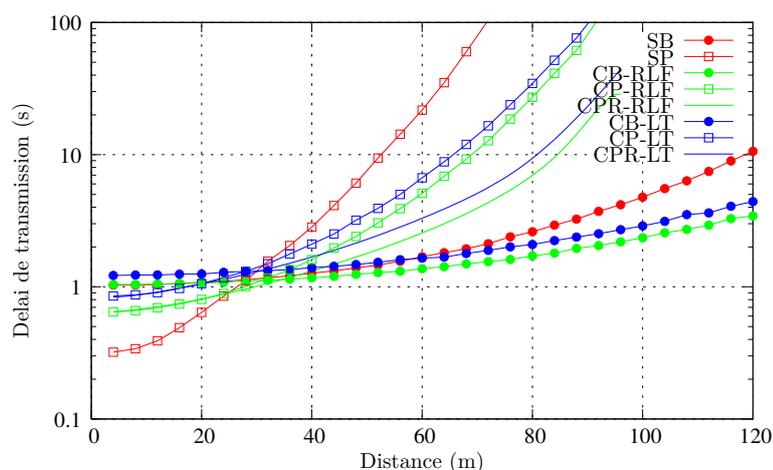


FIGURE 3.17 – Le delai de transmission pour la transmission multi-sauts

3.5 Acquittement et débordement

Nous nous intéressons maintenant à évaluer à quel moment il faut arrêter le flux fontaine. Du fait de la caractéristique du code fontaine qui consiste à transmettre en permanence l'information jusqu'à l'accusé de réception (ou nombre de transmissions maximal est atteint), dans ce cas, nous avons mis à jour un problème inhérent au mode d'acquittement de bout-en-bout des codes fontaine pour une transmission multi-sauts. Contrairement à l'étude précédente où l'on s'intéressait aux transmissions dans un réseau linéaire qui ont lieu *avant* le décodage complet du message, on s'intéresse ici aux émissions de paquets qui ont lieu *après* que la destination ait décodé les K fragments. Une fois que le décodage est réussi, la destination envoie un accusé de réception pour que la source arrête d'envoyer des paquets codés. Nous avons montré que pour une transmission directe, ce mode d'acquittement est particulièrement intéressant et permet de réduire l'énergie consommée par l'émetteur et le récepteur. Maintenant, dans le contexte d'un réseau linéaire, la transmission de cet accusé de réception ne peut pas se faire instantanément. Ainsi, le temps de transmission entre le temps du décodage t_d à la destination et l'arrivée de l'acquittement à la source $t_d + t_{ack}$ engendre une émission de paquets, par la source et les relais intermédiaires, qui sont totalement inutiles au décodage.

Cette surcharge d'émissions, que nous désignons par le terme de *débordement* dans cette thèse, est d'autant plus importante que la longueur du chemin à parcourir est grande. Il est possible d'illustrer ce problème par un petit exemple concret de la vie courante. Imaginons un jardinier qui souhaite remplir un seau à l'aide d'un tuyau d'arrosage au fond de son jardin. Entre le moment où le jardinier décide d'arrêter l'eau et le moment où il arrive au robinet pour le fermer, une certaine quantité d'eau se sera écoulée qui fera déborder le seau. Cette quantité est proportionnelle au temps de trajet que met le jardinier pour aller du fond du jardin au robinet. Bien entendu, si la contenance du seau est très grande et que le jardinier court vite, ce surplus d'eau peut s'avérer négligeable. Par contre, si le saut est petit et le temps de trajet est long, le débordement en eau n'est plus négligeable. Il en est de même pour notre problème de débordement de code fontaine. En effet, si la dimension K est importante et que l'accusé

de réception avance vite, le débordement est négligeable. Par contre, comme notre cas d'étude s'intéresse à une dimension K faible, le débordement peut s'avérer d'autant plus important que l'accusé de réception met du temps à arriver à la source.

Nous nous intéressons dans la suite à l'étude de ce débordement et à la recherche de solutions de relaying permettant de réduire ce phénomène. Pour cela, nous nous plaçons dans le cas où K est petit et nous analysons la surcharge introduite par le code pour différentes stratégies de relaying.

3.5.1 Modélisation

Pour focaliser seulement sur les effets du débordement en négligeant le surcôt de transmission due à la redondance du code pour une transmission donnée, nous adoptons le code RLF dans cette étude et pour une transmission sur canal AWGN. Nous considérons le même réseau linéaire multi-sauts que dans l'étude précédente (cf. Figure 3.15, p.48). La source envoie K fragments encodés par un code fontaine en un ensemble infini de paquets $P = \{p_1, p_2, \dots, p_N\}$ au destinataire périodiquement toutes les T_s secondes. Quand un relai reçoit un paquet codé, il cherche à le retransmettre immédiatement à son prochain voisin. La source s'arrête de transmettre à la réception d'un accusé de réception de la destination. Le destinataire émet un accusé de réception dès que le décodage est réalisé avec succès. L'accusé de réception se propage au travers de n_h sauts vers la source.

La configuration du réseau suit celle présentée précédemment (cf. Table 3.1, pp.39). Les paquets ACK sont traités avec priorité absolue dans la file d'arrivée des paquets. Ainsi, un paquet ACK sera transmis avant tous les paquets codés par le relai de façon à minimiser le temps de remontée de l'accusé de réception.

Definition 3.5.1.1 (débordement). *Nous définissons le débordement comme le nombre de paquets émis par le réseau entre le temps de décodage t_d et le temps de réception de l'accusé de réception à la source t_{ack} .*

Ce débordement se compose de deux types de paquets :

- Les paquets d'accusé de réception (ACK) émis successivement par la destination et tous les relais
- Les paquets de données (DATA) encodés émis par la source et les relais jusqu'à ce qu'ils reçoivent un accusé de réception

La quantité de paquets en débordement dépend de trois facteurs :

- La longueur du réseau n_h en nombre de sauts,
- La progression h d'un paquet d'accusé de réception,
- La stratégie de relaying considérée dans le réseau.

Definition 3.5.1.2 (progression). *Nous définissons la progression h d'un paquet d'accusé de réception comme étant le nombre de sauts qu'un paquet d'accusé de réception parcourt dans le réseau pendant qu'un paquet codé progresse d'un seul saut vers la destination.*

La Figure. 3.18 illustre le débordement pour une progression de $h = 1$. Ici nous considérons qu'un paquet ACK met autant de temps qu'un paquet codé DATA pour arriver au prochain

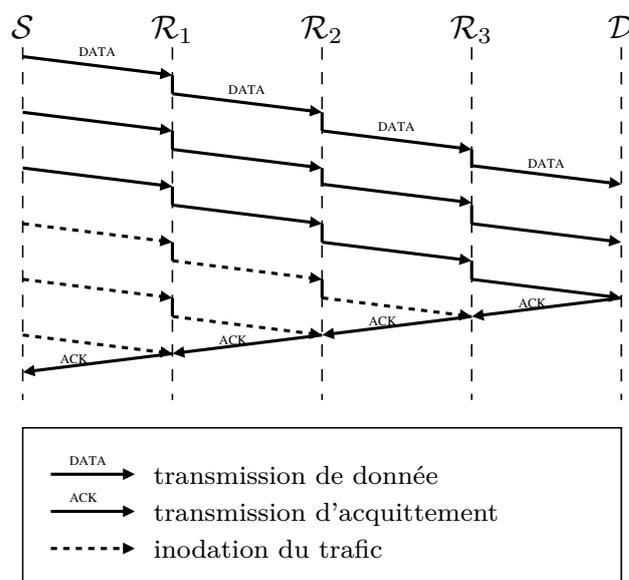


FIGURE 3.18 – Une séquence de transmissions pour $n_h = 4$ et une progression $h = 1$.

relai. Les flèches en pointillés représentent les paquets en débordement qui ne sont pas des acquittements.

La progression d'un ACK peut être supérieure à 1. Par exemple, un paquet ACK peut traverser 2 sauts pendant qu'un paquet de données progresse de 1 saut. En effet, si l'on fait abstraction des collisions, la durée de transmission d'un paquet ainsi que la probabilité d'erreur sur un lien radio diminue avec la taille des paquets. Ainsi, la transmission d'un ACK est plus fiable et plus rapide pour un saut. Comme les relais cherchent à relayer leur ACK dès qu'ils l'ont reçu, un paquet ACK de quelques bits peut parcourir plusieurs sauts.

La prise en compte des collisions rend le problème plus complexe, car la perte d'un acquittement provoque un débordement infini. La valeur de T_s doit être ajusté de façon à limiter ces collisions. Dans notre contexte $T_s = 1s$ est suffisamment grand pour négliger l'interférence et le nombre de collisions qui sont due à la nature semi-duplexe du canal. L'acquittement est ainsi supposé être transmis parfaitement de saut en saut.

La valeur du débordement sera également influencée par la stratégie de relayage. En effet, une stratégie où chaque nœud attend d'avoir décodé les K fragments pour retransmettre (cf. stratégie CB, p.48) présente un débordement minimal égal au nombre de paquets d'acquittement. Une stratégie de relayage passif (cf. stratégie CP, p.48) sera par contre particulièrement sensible au débordement.

Dans la sous-partie suivante, nous nous proposons une mesure du débordement en fonction de la stratégie de transmission. Nous en déduisons une stratégie hybride de relayage de façon à réduire le débordement tout en limitant la complexité de calcul dans les relais.

3.5.2 Surcharge d'acquittement

Nous mesurons l'ordre de grandeur du débordement pour les 4 stratégies de relayage suivantes (cf. p.48 pour plus de détails) :

- **Scénario CB** (relayage de Code fontaine par Bloc) : Chaque relai R_i attend de pouvoir décoder les K fragments reçus de R_{i-1} . Au décodage, il transmet un ACK à R_i et transmet un nouveau code fontaine composé des K fragments vers R_{i+1} .
- **Scénario CP** (relayage de Code fontaine par Paquet) : Chaque nœud relaie un paquet codé dès sa réception. Il n'y a pas d'acquittement saut par saut, mais un seul acquittement global envoyé par la destination quand le message est décodé.
- **Scénario CH** (relayage de Code fontaine Hybride) : Cette technique est une combinaison hybride des stratégies CB+CP. Ici, chaque relai R_i commence par relayer le message reçu vers R_{i+1} et cherche parallèlement à décoder les K fragments. Dès qu'il les a décodés, il envoie un accusé de réception à R_{i-1} .
- **Scénario CPR-XA** (relayage de Code fontaine par XOR Aléatoire) : Chaque nœud relaie à chaque intervalle de temps la combinaison XOR de deux paquets choisis aléatoirement dans sa file. Cette stratégie, appelée *greedy random* a été présentée dans [116]

Pour ces différentes stratégie de transmission, nous pouvons identifier 3 modes de propagation d'acquittement. Le premier consiste à accuser réception du message saut par saut, l'autre de bout-en-bout. Le scénario hybride CH proposé a pour but de profiter de l'avantage des deux cas précédents.

Pour le cas où les transmissions sont acquittées saut par saut, comme dans la stratégie CB, quand le relai R_i a réussi à décoder les K fragments, il les acquitte auprès de la source. Puis, il ré-encode les K fragments et les transmet au relai R_{i+1} , qui acquittera le flux auprès de R_{i-1} uniquement. Après que R_{i-1} ait envoyé son acquittement, il ne peut recevoir qu'une très faible quantité de paquets codés inutiles de R_i . C'est pour cette stratégie que le débordement est minimal. Il est de l'ordre de $O(n_h)$

Pour le cas de l'accusé de réception de bout-en-bout comme dans les stratégies CP et CPR-XA, l'acquittement n'est envoyé que par le destinataire et doit traverser le réseau pour atteindre la source comme présenté sur la Figure. 3.18. Dans ce cas, le débordement est de l'ordre de :

$$n_h \lfloor \frac{n_h}{h} \rfloor - h \cdot \frac{\lfloor \frac{n_h}{h} \rfloor (\lfloor \frac{n_h}{h} \rfloor + 1)}{2}. \quad (3.13)$$

Lorsque l'acquittement a été déjà reçu par r relais ($r < n_h$), il existe $(n_h - rh)$ relais qui continuent d'émettre des paquets. L'équation (3.13) est obtenue en sommant le nombre de paquets $n_h - rh$ émis pour $r \in [1, \lfloor \frac{n_h}{h} \rfloor]$.

3.5.3 Résultats de simulation

Dans cette partie, nous comparons les résultats de simulation issus de notre implémentation qui permet la transmission d'un code fontaine sur le simulateur WSNet. Nous considérons le cas où $n_h \in [2, 15]$ et $K = 10$ et testons les différents scénarios de relayage des codes fontaine RLF et LT. Les résultats suivants ont été obtenus pour 1000 instances de problème.

Quantification du débordement

L'évolution de la taille du débordement en fonction du nombre de sauts n_h pour les stratégies CP et CPR-XA est présentée dans la Figure 3.19. Nous considérons le cas où $n_h \in [2, 15]$ et $K = 10$. Comme nous l'avons introduit précédemment, la valeur du débordement dépend de la vitesse de propagation de l'accusé de réception dans le réseau relativement à la vitesse de propagation des paquets codés vers le destinataire. Or, de par l'impact de la couche MAC du au mécanisme CSMA/CA, il est difficile de dériver théoriquement la valeur de la progression h du paquet. C'est pourquoi nous avons utilisé les mesures de débordement effectuées par simulation pour obtenir la valeur de h pour CP et CPR-XA en réalisant une régression linéaire. Après avoir ajusté la valeur de h par régression, nous constatons que notre estimation théorique (3.13) présente une bonne approximation du débordement.

On observe que la valeur moyenne du débordement est plus importante pour CP que pour CPR-XA. En effet, la perte d'un message DATA ou ACK dans le scénario CP entraîne systématiquement la retransmission depuis la source tandis que dans le scénario CPR-XA, la retransmission s'effectue à l'endroit où le message est perdu. Ceci explique que le nombre de débordement est supérieur dans le scénario de relaying CP par rapport au cas CPR-XA.

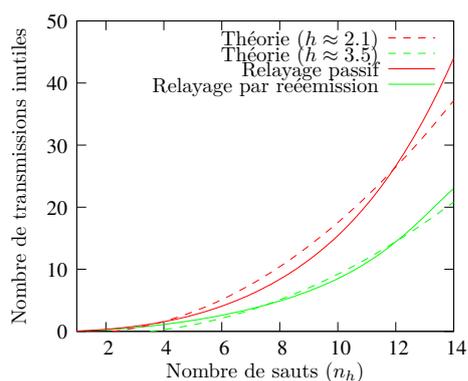


FIGURE 3.19 – Taille du débordement pour les stratégies CP et CPR-XA (simulation et modèle).

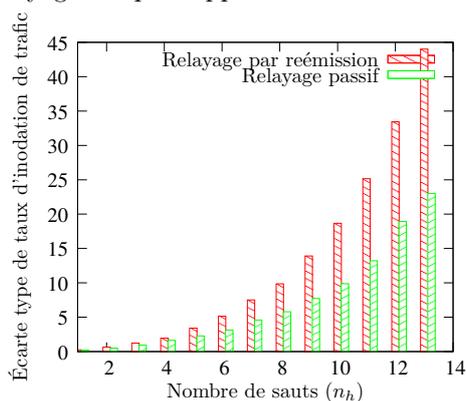


FIGURE 3.20 – Ecart type du débordement pour les stratégies CP et CPR-XA

Dans nos simulations, nous avons observé une forte variabilité de la mesure de ce débordement comme le montre la Figure 3.20. Les écarts types de la mesure du débordement dans nos simulations augmentent rapidement avec n_h . On observe avec environ la même probabilité des instances où la progression $h \approx n_h$ ou $h \approx 1$. Ce phénomène est principalement du aux tirages aléatoires des temps d'attente *backoff* par les nœuds dans le protocole IEEE 802.15.4. Même si nous avons choisi la valeur de T_s de façon à éviter les collisions, l'utilisation de temps d'attente aléatoires modifie la progression de h .

Analyse du débordement

Nous donnons la répartition des types de paquets transmis par les nœuds pour les différents scénarios dans la figure 3.21. Nous avons observé les résultats pour un nombre de sauts $n_h = 5, 10, 15, 20$ et $K = 5$.

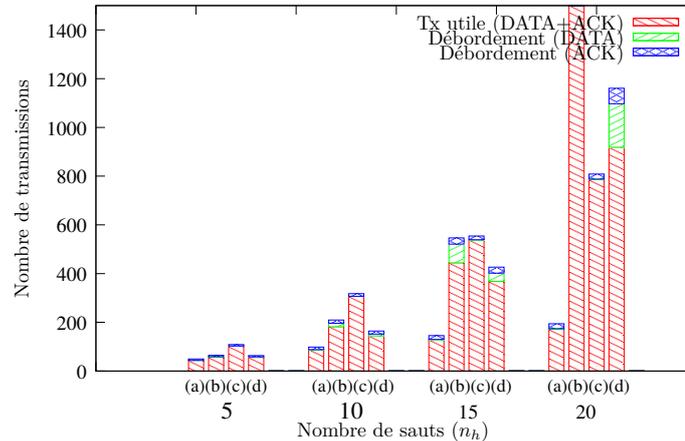


FIGURE 3.21 – Le taux d’inondation du trafic mesuré par divers scénarios de transmissions (a)CB, (b)CP, (c)CH et (d)CPR-XA pour $n_h = 5, 10, 15, 30$ et $K = 5$

Pour cette configuration, le scénario CB offre un meilleur taux de redondance ainsi qu’un meilleur passage à l’échelle. Les cas CP et CPR-XA conduisent à une augmentation exponentielle du débordement. Nous avons observé que pour $n_h = 15$, le débordement du trafic représente 14% du trafic total pour le scénario CP contre 8% du trafic pour le scénario CPR-X. Cependant, lorsque $n_h = 20$, la transmission passive n’est plus pratique du au nombre de redondances introduites.

Le scénario CH qui effectue le décodage et ré-encodage de la totalité de K fragments à chaque saut permet de réduire le problème du débordement lorsque le nombre de saut augmente.

La Figure 3.22 de (a) à (d) décrit le débordement du trafic au niveau de chaque saut pour les différents scénarios de transmissions lorsque $K = 5$ et $n_h = 10$. Nous avons noté que le scénario CPR-XA offre une meilleure performance. Il offre également un meilleur indice d’équité de Jain *Jain’s Fairness*(F) (cf. l’équation 3.14) qui est égale à 0.924. Pour les autres scénarios CB, CP et CH cet indice d’équité de Jain est égale à 0.928, 0.8 et 0.79 respectivement.

$$F(x) = \frac{(\sum x_i)^2}{n_h \cdot \sum x_i^2} \quad (3.14)$$

Nous pouvons en conclure que pour les réseaux tolérants au délai, la transmission du scénario CB, lorsqu’on décode et encode l’information de saut en saut, est préférable car cette stratégie garantit une bande minimale sur le nombre de transmissions et permet de conserver la consommation d’énergie du système. Cependant, pour des réseaux qui disposent de contraintes en terme d’énergie et de délai, les situations ne sont pas évidentes. Si le nombre de sauts est inférieur à environ $n_h \leq 15$, les scénarios CP et CPR-XA où le relayage s’effectue sans décodage complet à chaque saut est préférable. Or, la re-émission entraîne un taux de débordement élevé.

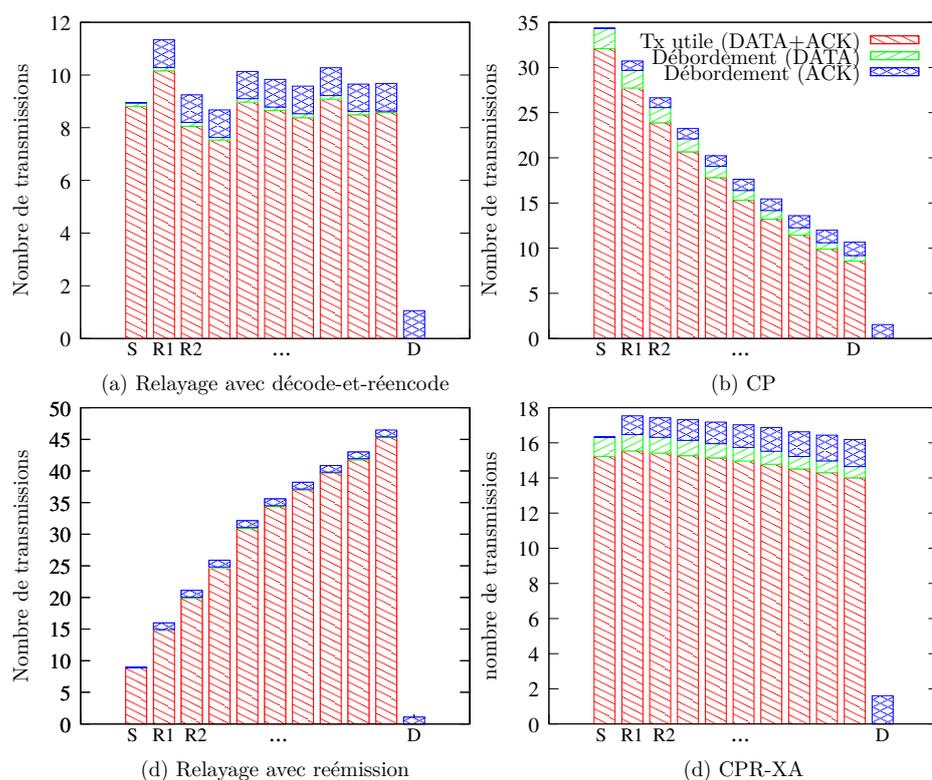


FIGURE 3.22 – Le taux du trafic mesuré par divers scénarios de transmission pour $n_h = 10$ et $K = 5$

Nous pouvons faire appel au schéma hybride CH en imposant à certains nœuds de décoder et encoder complètement l'information avant de transmettre au nœud suivant. Ceci permet d'éviter l'accumulation du nombre de transmissions inutiles au travers d'un grand nombre de sauts.

3.6 Conclusion

Ce chapitre de thèse est destiné à l'étude de la performance et le comportement de codage fontaine dans un réseau de capteur sans fils. Nous avons intégré une couche de codage supplémentaire sur le plateforme de simulation WSNNet. Nous avons abordé l'étude de performances selon plusieurs critères d'évaluation en termes d'énergie et de délai. Nous abordons aussi le problème de la gestion d'acquittements que nous avons relevé pendant notre implémentation dans le schéma de transmission multisauts.

Nous avons considéré un réseau dont la couche de liaison de données et la couche physique sont définies par le protocole IEEE802.15.4. Nous avons montré que la prise en compte des couches protocolaires réalistes et ses imperfections est importante pour l'évaluation des performances de codes fontaine. Nous avons montré que lorsque le canal de retour n'est pas fiable, les codes fontaine (RLF et LT) sont plus robustes face à la perte d'acquittement que dans les cas sans codage même pour une liaison à un saut, et permettent de réduire l'énergie consommée

et donc d'accroître la durée de vie des capteurs. Cet avantage devient plus important lorsque le nombre de sauts augmente. De plus, en présence d'un grand nombre de capteurs dans le réseau, ce codage est avantageux pour réduire l'impact des collisions pour les transmissions avec acquittement. Les résultats de performance se basant sur le délai de transmission ont progressé proportionnellement à la consommation énergétique. Le cas avec codage fontaine est aussi plus avantageux que le cas sans codage si on se base sur le critère du délai.

Bien que le code fontaine permette d'optimiser la gestion des accusés de réception, et ainsi d'augmenter la performance globale du système en terme de consommation d'énergie et de délai de transmission, nous avons souligné un problème du débordement de transmission. Pendant l'intervalle de temps de propagation d'acquiescement vers la source, les transmetteurs continuent toujours à transmettre les paquets. Ce débordement devient important avec un nombre de sauts élevé. Nous avons étudié et analysé le trafic inutilement induit pendant la propagation d'acquiescement pour plusieurs stratégies de relayage et nous en avons conclu que pour les réseaux tolérants aux délais, le relayage de code fontaine saut par saut avec décodage et reencodage complet à chaque saut est préférable car cette stratégie garantit une bande minimale sur le nombre de transmissions et permet de conserver la consommation d'énergie du système. Cependant, pour des réseaux qui disposent de contraintes en terme d'énergie et de délai, le choix est non trivial. Si le nombre de sauts est relativement faible ($n_h \leq 15$ selon nos résultats expérimentaux), le relayage par paquet ne nécessitant aucun traitement aux nœuds intermédiaires est préférable. Dans le cas inverse, le taux de retransmission augmente considérablement et induit un débordement important dans le réseau. Nous pouvons éviter ce problème en introduisant le cas hybride où seuls certains nœuds dans le réseau décodent et ré-encodent les informations au nœud suivant.

4

Diversité d'information par le codage réseau avec code fontaine

Dans les WSNs, du fait de la mobilité des capteurs et de la non-fiabilité du lien radio, le routage de l'information est souvent réalisé de manière opportuniste (cf. Section 3.3.1 pp.34). Dans ce cas, le message émis par la source parcourt généralement différents chemins pour arriver au destinataire. Or, les chemins multiples et le caractère non-déterministe du réseau introduisent une difficulté sur le contrôle de la redondance d'information c'est à dire la réception multiple d'un même paquet. Bien que cette redondance permette d'obtenir une transmission robuste face aux pertes dans le réseau, si aucun mécanisme de contrôle n'est envisagé, ce surplus d'information transmise peut conduire à l'augmentation de la consommation de ressource et réduire la durée de vie du réseau.

Dans ce chapitre, nous nous intéressons à analyser et mettre en place un mécanisme de contrôle qui nous permettra de limiter le taux de redondance d'information. Pour un même nombre de transmissions, cette redondance de paquets peut être minimisée en augmentant les informations utiles dans le contenu de chaque paquet transmis. Ceci peut être réalisé par la transmission de combinaisons de l'information pour introduire ce qu'on appellera par la suite la *diversité d'information*. Cette diversité d'information peut être obtenue grâce à des techniques de codage appropriées par les nœuds à l'intérieur du réseau. Cette technique est également connue sous le nom de *codage réseau*.

Le principe du codage réseau a été abordé par Ahlswede et al. [4] pour améliorer le débit de transmission d'information dans la diffusion multiple. Contrairement au schéma de routage classique où les nœuds intermédiaires transmettent directement l'information sans traitement supplémentaire, nous nous intéressons au cas où on autorise les nœuds intermédiaires situés sur le chemin entre une source et un destinataire quelconque à coder et/ou décoder les paquets avant

de les transmettre. Ainsi, chaque paquet émis par le nœud dans le réseau est une combinaison linéaire des paquets entrant dans sa file.

Dans un système de communication, la capacité du réseau est bornée par la capacité de coupe minimale entre une source et un destinataire quelconque. Cette capacité optimale peut être atteinte lorsque les nœuds intermédiaires situés entre la source et le destinataire (également appelé les *relais*) peuvent traiter l'information en décodant et ré-encodant complètement l'information à chaque relai [49] (mais, ceci peut introduire une grande latence et l'augmentation de la consommation d'énergie) ou en appliquant seulement le ré-encodage par les combinaisons linéaires dans \mathbb{F}_q avec la taille q qui tend vers l'infini. Ce schéma de transmission a montré sa supériorité par rapport à la transmission classique qui s'effectue de bout en bout sans traitement de l'information à l'intérieur du réseau [4, 35, 116]. Li et al. ont montré que de simples combinaisons linéaires dans \mathbb{F}_2 par les nœuds dans le réseau sont suffisantes pour atteindre la capacité maximale du système [87]. L'avantage du codage réseau a été confirmé par Koetter et Médard en terme du débit de transmission [78]. De plus, le codage réseau apporte plusieurs avantages tels qu'un meilleur passage à l'échelle, une optimisation de la complexité, de la sécurité, du délai et du débit de transmission [44, 73, 79, 162, 70].

Plusieurs techniques de codage réseaux ont été proposées. La première technique la plus utilisée est basée sur la technique du code aléatoire linéaire [2, 59, 61, 169]. Kamra et al. a proposé dans [71] des codes dont le nombre de fragments à encoder évoluent dans le temps (growth code) dans le but d'augmenter la robustesse face à la faille. Le code RS a été également adopté dans [86]. D'autres algorithmes de codage/décodage réseau en temps polynomial ont été développés dans [133, 69].

Nous nous intéressons par la suite à étudier le cas où le codage réseau s'applique sur le flux d'information déjà protégé par le code FEC. Les données transmises sont donc codées sur deux niveaux :

- **code correcteur d'erreur** : la protection FEC utilisant le code correcteur d'erreur a pour but de fiabiliser la transmission de bout-en-bout. L'encodage est réalisé par la source et le décodage est réalisé par le destinataire. Nous nous focalisons sur l'utilisation de codes RLF et LT.
- **codage réseau** : le codage réseau est réalisé par les nœuds intermédiaires dans le réseau pour aider la source à relayer efficacement l'information au destinataire. En effet, une façon optimale pour atteindre la borne de capacité du système avec codage réseau consiste à décoder et ré-encoder complètement l'information à chaque relai [49]. Nous considérons le cas où les relais ré-encodent les informations de façon spécifique ou aléatoire, sans décodage à chaque relai.

Ce chapitre de thèse est organisé comme suit. Premièrement, nous abordons une problématique relevée lors du codage réseau sur le flux fontaine LT : la modification du degré. Puis, nous proposons plusieurs techniques de codage réseau qui ont pour objectif à la fois de conserver la distribution RSD tout en augmentant la diversité d'information apportée par le ré-encodage aux nœuds intermédiaires. Nous montrons à la fin du chapitre que notre proposition de stratégie de relayage permet de réduire le nombre de transmissions grâce à la diversité d'information introduite.

La contribution principale dans ce chapitre de thèse peut être résumée en deux parties principales :

- la proposition de techniques de codage réseau qui permettent de conserver la distribution RSD par approche théorique. Cette étude a donné lieu à une publication dans une conférence internationale APCC'2010 [112].
- la proposition de techniques de codage réseau par approche heuristique qui tendent à optimiser la distribution de degré de façon plus réaliste. Nous avons montré que ces techniques proposées permettent de réduire le nombre de transmissions dans le réseau grâce à la diversité d'information obtenue par le code. Cette étude a conduit à la publication dans un journal IEEE Communication Letters'2011 [11] ainsi à la présentation dans une conférence nationale Algotel'2010 [13].

4.1 Enjeux et problématique : code fontaine et codage réseau

Les codes fontaine, étant implémentés dans une couche haute de la pile protocolaire, permettent une adaptation du codage selon les besoins du réseau. Or, ils ont été conçus et modélisés initialement pour une transmission de bout en bout, et leur utilisation optimale dans un réseau à grande échelle n'est pas immédiate. Nous proposons d'exploiter ici le codage réseau pour obtenir un meilleur passage à l'échelle en optimisant la consommation énergétique du système et le délai de transmission.

4.1.1 Problématique

Les réseaux de capteurs sans fil étant limités en ressources, le choix du codage LT est justifié par sa faible complexité de calcul imposée au système. Le code LT est basé sur le décodage par propagation de croyance (BP) qui utilise le décodage itératif basé sur le graphe biparti. Comme expliqué dans le Chapitre 2, pour que le décodage se déroule efficacement, il est nécessaire que l'encodage soit réalisé d'une manière spécifique. La distribution de degré selon la loi de Soliton Robuste (RSD) a été conçue pour maintenir l'efficacité du décodage BP, le but étant d'avoir au moins un degré «1» à chaque étape du décodage.

Cependant, le codage réseau dont le but initial est d'introduire de la diversité dans les paquets transmis, peut aussi affecter la distribution de degré. En effet, l'addition linéaire dans \mathbb{F}_2 due au codage réseau conduit à la création d'une nouvelle combinaison dont les fragments de données peuvent s'ajouter ou s'annuler. Nous cherchons donc dans ce chapitre des solutions pertinentes qui permettront de conserver la distribution du flux encodé LT après les combinaisons linéaires effectuées par le codage réseau.

Principalement, il existe deux solutions pour contourner ce problème :

- la première solution consiste à utiliser un algorithme de décodage qui n'impose pas de contrainte sur la distribution de degré contrairement au décodage BP. Or, le décodage BP a une complexité de calcul faible, ce qui est important dans un capteur pour la limitation de la consommation énergétique. Nous ne considérerons donc pas cette solution.
- la deuxième solution se focalise sur la façon d'encoder les données par les relais pour respecter la distribution de degré imposée par le décodeur

4.1.2 Etat de l'art

L'idée proposée dans [169] qui impose aux nœuds intermédiaires de choisir aléatoirement les paquets à encoder dans sa mémoire avant la retransmission est la plus naturelle. Les nœuds dans ce cas n'ont pas besoin de décoder et ré-encoder complètement l'information ce qui est adapté aux réseaux de capteurs en terme de complexité et latence. Cependant, cette étude considère le code aléatoire linéaire et ne peut pas être appliquée directement au code LT à cause de la contrainte imposée sur la distribution de degré. En effet, la combinaison linéaire entre les paquets encodés génère généralement un ensemble de paquets dont les degrés ne sont plus conformes à la distribution de Soliton Robuste et perturbe fortement le décodage.

Or, jusqu'à présent, il existe assez peu de travaux de recherche sur le codage réseau appliqué aux codes LT, et en particulier sur la conservation de la distribution de degré du flux encodé après d'avoir subi au codage réseau. Dans [123], une nouvelle distribution de degré est proposée par l'encodage des données initiales pour le cas où plusieurs sources veulent communiquer avec un seul nœud intermédiaire. Ce dernier n'offre cependant pas un bon passage d'échelle pour un grand réseau car la technique proposée est liée à une topologie de réseau spécifique. De plus, la source doit connaître les informations sur la topologie du système et l'état du canal avant la transmission ce qui n'est pas pratique en réalité. Une technique pour conserver la distribution de Soliton pour des transmissions coopératives a été proposée dans [55]. Avec cette technique, chaque nœud dans le système agit comme un codeur indépendant. Ainsi, les paquets reçus sont considérés comme des données brutes même si elles ont déjà été codées plusieurs fois. A la réception, le décodage doit enlever successivement toutes les couches de codage. En conséquence, cette technique est trop complexe pour les réseaux de capteurs.

La plupart des études existantes sur l'optimisation de la distribution du degré sont souvent focalisées sur l'optimisation de la distribution pour s'adapter aux contextes d'étude, par exemple, pour la transmission de paquets de petite taille [66], pour la simplification du processus de décodage [153], etc.

Dans [32], un algorithme qui permet de conserver la distribution des degrés, tout en assurant une répartition uniforme des fragments dans les paquets transmis a été proposé. Ces travaux conduisent à des résultats intéressants, mais au prix d'une complexité de calcul élevée dans les relais ainsi qu'une grande mémoire.

Dans ce chapitre de thèse, nous nous intéressons au problème de la conservation de la distribution de degré après le ré-encodage par le relai. Nous proposons plusieurs solutions qui permettent de réaliser cette conservation et nous discutons ensuite des avantages et des inconvénients de chaque solution proposée. Enfin, nous démontrons le compromis nécessaire entre l'optimisation, l'exploitation de ces redondances et l'efficacité du décodage BP. Un point très important à prendre en considération est comment peut-on obtenir ce gain de performance sans introduire trop de complexité et de latence.

4.2 Étude de la distribution de degré obtenue après des combinaisons linéaires XOR

Dans ce paragraphe, nous analysons la statistique du degré obtenu en sortie d'un XOR entre deux paquets. Soit un paquet encodé $p \in \mathbb{F}_2^N$. Il s'écrit $p = \{t|m\}$ et est composé d'un entête $t \in \mathbb{F}_2^K$ décrivant la combinaison linéaire dans \mathbb{F}_2 et $m \in \mathbb{F}_2^{M-K}$ avec M la taille totale du paquet. La combinaison linéaire du message émis m s'écrit :

$$m = \bigoplus_{i=1}^K t_i f_i. \quad (4.1)$$

Nous rappelons la définition du *degré de paquet* d qui correspond au poids de Hamming $W_H(t)$ de t tel que :

$$d = W_H(t) = \sum_{i=1}^K t_i \quad (4.2)$$

Considérons le cas d'une combinaison linéaire XOR entre deux paquets p_1 et p_2 , ces paquets contenant respectivement un entête t_1 et t_2 . Nous appelons d_1 et d_2 les degrés des paquets qui correspondent respectivement à $W_H(t_1)$ et $W_H(t_2)$.

Nous distinguons deux cas. Dans le premier cas qui est très simple, si les deux paquets n'ont pas de fragment en commun, alors, le paquet résultant après l'opération XOR appelé p_{xor} contient l'ensemble des fragments de p_1 et de p_2 . Le degré $d_{xor} = W_H(t_1 \oplus t_2)$ du paquet p_{xor} s'exprime très simplement :

$$d_{xor} = d_1 + d_2 \quad (4.3)$$

Dans le deuxième cas où p_1 et p_2 ont des fragments en commun, le résultat n'est pas aussi simple. De par l'opération XOR dans \mathbb{F}_2 , ces fragments vont s'annuler mutuellement lors de l'addition.

Definition 4.2.0.1. (*Collision*) Nous appelons **collision** la présence du même fragment dans les deux paquets à XORer, conduisant à l'annulation de ces deux fragments lors de l'addition dans \mathbb{F}_2 des paquets. Ainsi, la collision d'un fragment f_i entre des paquets p_1 et p_2 quelconques induit la suppression du fragment f_i après le XOR.

Prenons l'exemple du cas schématisé dans la Figure 4.1 pour $K=5$. Nous considérons que le paquet p_1 contient les fragments f_2 et f_3 et que p_2 contient f_2 et f_5 . Étant donné que nous nous intéressons au degré du paquet résultant, ce sont les bits de l'entête qui sont significatifs et sur lesquels nous allons nous focaliser.

L'addition entre p_1 et p_2 donne :

$$\left. \begin{array}{l} p_1 = f_2 \oplus f_3 \\ p_2 = f_2 \oplus f_5 \end{array} \right\} \implies p_{xor} = f_3 \oplus f_5 \quad (4.4)$$

Selon l'exemple ci-dessus, la présence du fragment f_2 dans p_1 et p_2 conduit à une collision, et à la suppression de ce fragment après XOR. Le degré résultant après l'addition est inférieur à la somme des degrés de p_1 et p_2 . En effet, le degré résultant s'exprime ainsi :

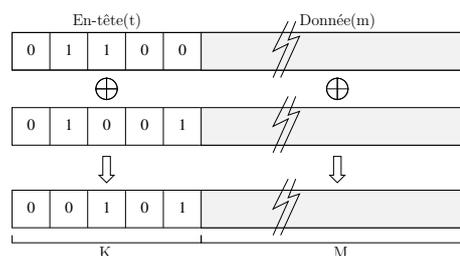


FIGURE 4.1 – Trames de données

$$d_{xor} = d_1 + d_2 - 2o = (d_1 - o) + (d_2 - o), \quad o \leq \min(d_1, d_2) \quad (4.5)$$

avec o le nombre de collisions. Le facteur "2" devant o provient du fait que les fragments en collisions sont supprimés à la fois dans la contribution de p_1 et de p_2 .

4.2.1 Une combinaison linéaire XOR

Nous proposons maintenant d'estimer la statistique du degré résultant. Nous avons vu que le degré résultant après XOR dépend du degré des paquets à XORer ainsi que du nombre de collisions. d_1 et d_2 étant fixés, nous avons donc estimé la probabilité d'avoir des collisions après une combinaison linéaire XOR.

Statistiques des collisions

La probabilité qu'il y ait exactement o collisions après une combinaison XOR entre p_1 et p_2 est :

$$p_o(o|d_1, d_2, K) = \begin{cases} \frac{C_{d_1}^o C_{K-d_1}^{d_2-o}}{C_K^{d_2}} & \text{pour } o \leq \min(d_1, d_2) \\ 0 & \text{sinon} \end{cases} \quad (4.6)$$

Cette formule correspond au dénombrement des choix possibles des fragments dans p_1 et p_2 pour produire o collisions de fragments. En effet, une fois p_1 fixé, $C_{d_1}^o$ est le nombre de cas possibles pour sélectionner les o fragments victimes de collision parmi ceux de p_1 , $C_{K-d_1}^{d_2-o}$ est le nombre de paquets de degré d_2 ayant les o fragments en commun avec ceux de p_1 . Ceci est équivalent à choisir $d_2 - o$ fragments restants dans p_2 parmi $K - d_1$. En effet, les fragments associés aux collisions sont déjà placés, et on peut placer librement les autres. Enfin, $C_K^{d_2}$ est le nombre total de choix de paquets de degré d_2 sans considérer les collisions.

4.2.2 Expression analytique de la probabilité du degré du flux sortant

Pour préciser les résultats précédents, nous cherchons maintenant à déterminer la distribution de degrés en sortie en fonction de la distribution en entrée. Pour cela, nous cherchons maintenant à déterminer l'expression analytique de la probabilité d'obtenir un degré d_{xor} donné après application du codage réseau sur ces paquets d'information.

Etude préliminaire : un seul XOR entre deux paquets

Nous cherchons à étudier l'évolution du degré après les opérations XOR. Nous voulons évaluer si le XOR a tendance à augmenter ou réduire le degré des paquets. Nous nous intéressons à l'analyse de la probabilité que

$$d_{xor} \leq \min(d_1, d_2) \quad (4.7)$$

ou bien

$$W_H(t_1 \oplus t_2) \leq \min(W_H(t_1), W_H(t_2)). \quad (4.8)$$

Pour cela, nous définissons X comme l'événement discret pour lequel les équations (4.7) et (4.8) sont vérifiées.

Pour exprimer la probabilité d'avoir o collisions lors du XOR entre les paquets p_1 et p_2 , nous supposons que $d_1 \leq d_2$. En effet, grâce à la propriété de commutativité de l'addition dans \mathbb{F}_2 , nous obtenons la symétrie : $p_1 \oplus p_2 = p_2 \oplus p_1$. En conséquence, les rôles de p_1 et p_2 sont symétriques et peuvent être permutés pour vérifier $d_1 \leq d_2$.

En supposant que $d_1 \leq d_2$, pour que les équations (4.7) et (4.8) soient vérifiées, il faut que le nombre o de collisions choisies dans d_1 vérifie :

$$d_1 + d_2 - 2o \leq d_1 \implies o \geq d_2/2 \quad (4.9)$$

Nous pouvons ainsi en déduire $P_o(X)$ qui est égal à la somme des événements discrets :

$$P_o(X) = \sum_{o=\lfloor d/2 \rfloor}^K \frac{C_{d_1}^o C_{K-d_1}^{d_2-o}}{C_K^{d_2}}. \quad (4.10)$$

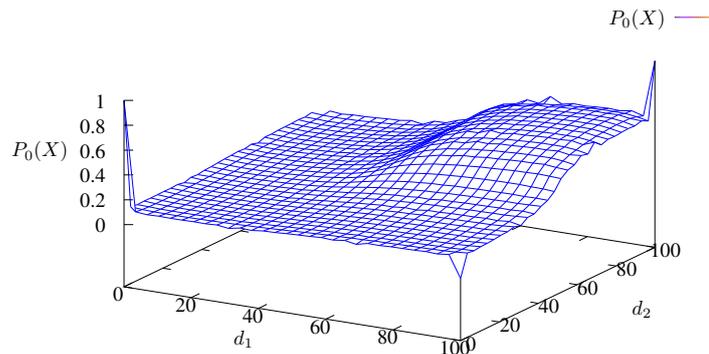


FIGURE 4.2 – Probabilité qu'une combinaison XOR entre deux paquets conduise à la réduction du degré du paquet résultant

Nous pouvons visualiser dans la Figure 4.2 la probabilité que la combinaison linéaire XOR entre p_1 et p_2 résulte de la réduction du degré. Cette figure a été tracée à partir de l'équation 4.10 et en utilisant la connaissance de la propriété symétrique des d_1 et d_2 . La probabilité $P_o(X)$ est ainsi illustrée pour la valeur de d_1 et d_2 quelconque.

Pour la plupart des cas, la probabilité $P_o(X)$ est assez faible (< 0.5) sauf pour des valeurs

spécifiques de d_1 et d_2 très élevées. Cependant, ce dernier cas se produit rarement pour les codes LT car dans le cas général, la distribution du degré en sortie de l'encodeur contient peu de paquets de grand degré.

En conséquence, nous pouvons en conclure que la combinaison linéaire XOR tend à faire translater la distribution de degré, c'est à dire que le degré tend à augmenter après le codage réseau. Ce qui devient problématique pour certains cas, par exemple, le cas où le décodage BP est utilisé au récepteur.

Etude théorique : un seul XOR entre deux paquets

Nous pouvons exprimer la probabilité d'obtenir un degré résultant égal à d_{xor} par la combinaison linéaire XOR entre deux paquets p_1 et p_2 . A l'aide de l'équation (4.5) et (4.6), nous pouvons exprimer la probabilité d'obtenir un degré d_{xor} sachant d_1 , d_2 et o comme la probabilité d'avoir $o = \frac{d_1 + d_2 - d_{xor}}{2}$ collisions entre t_1 et t_2 . Il faut noter que o ne peut prendre qu'une valeur entière. Nous obtenons :

$$p_{d_{xor}}(d_{xor}|d_1, d_2, o) = \begin{cases} p_o(o = \frac{d_1 + d_2 - d_{xor}}{2}|d_1, d_2, K) & \text{si } o \in \mathbb{N} \\ 0 & \text{sinon} \end{cases} \quad (4.11)$$

En appliquant la loi des probabilités totales sur l'ensemble des degrés d_1 et d_2 , l'équation (4.11) devient

$$p_{d_{xor}}(d_{xor}) = \sum_{d_1=1}^K \sum_{d_2=1}^K p_{d_1}(d_1)p_{d_2}(d_2)p_o(o = \frac{d_1 + d_2 - d_{xor}}{2}|d_1, d_2, K) \quad (4.12)$$

$p_{d_1}(d_1)$ et $p_{d_2}(d_2)$ correspondent à la probabilité d'obtenir les degrés d_1 et d_2 pour les paquets initiaux. Cette probabilité dépend de la distribution de degré initiale (par exemple la loi RSD pour le code LT).

Nous présentons sur les Figures 4.3 et 4.4 la distribution de degré du code RLF et LT respectivement. Nous présentons les distributions d'origine, ainsi que la distribution obtenue à la sortie d'un XOR. Nous pouvons observer que les résultats de simulation et théorique se superposent, donc l'expression (4.12) décrit correctement la distribution des degrés. D'autre part, pour le code aléatoire linéaire RL, nous pouvons observer que l'opération XOR ne modifie pas la distribution qui est initialement du type binomiale. En effet, lors du XOR entre 2 paquets, pour chaque fragment, il y a une probabilité de 1/2 que ce fragment soit présent dans le paquet résultant (cas où il est présent dans p_1 uniquement ou dans p_2 uniquement), et une probabilité de 1/2 que ce fragment soit absent (cas où il est présent dans aucun des 2 paquets, ou dans les 2 simultanément). En conséquence, les codes RLF sont naturellement adaptés au codage réseau ce qui justifie les travaux sur ce sujet [2, 59, 61, 169]. A l'inverse, pour le code LT, la distribution de degré après XOR est modifiée et tend à augmenter. Ceci peut être observé selon la Figure 4.4 où les valeurs sont décalées vers la droite.

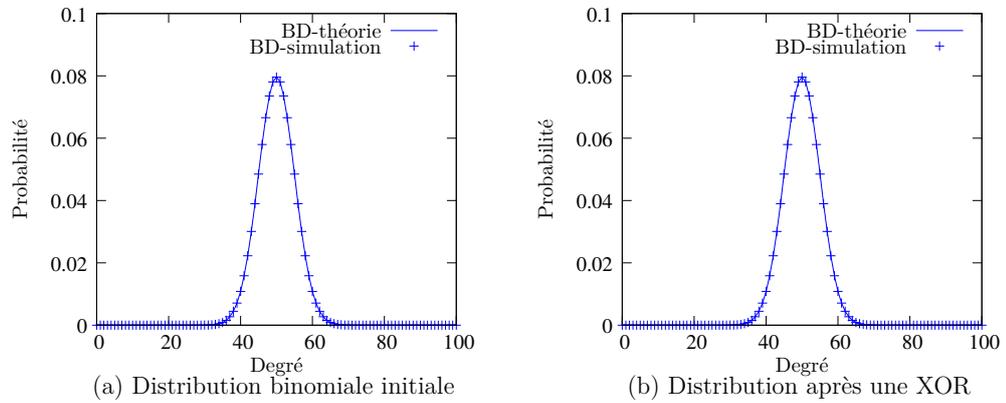


FIGURE 4.3 – Évolution de degré après XORs pour le code RLF

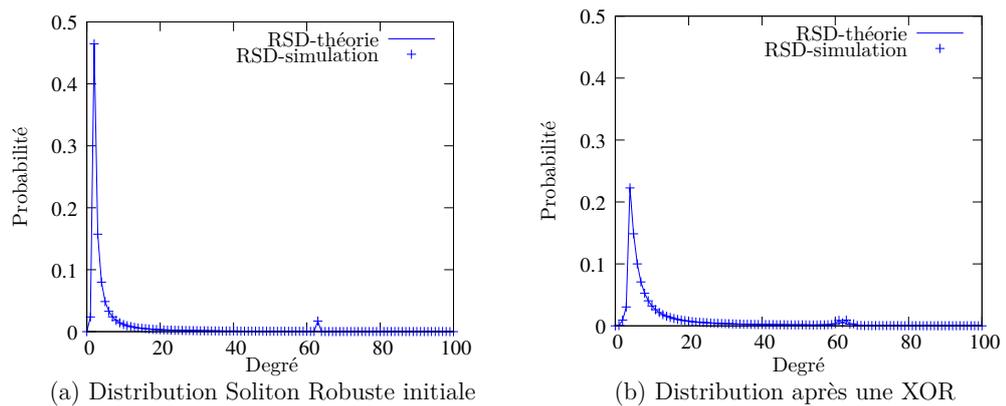


FIGURE 4.4 – Évolution de degré après XORs pour le code LT

Etude théorique : e XORs entre $e + 1$ paquets

Nous souhaitons maintenant généraliser notre étude au cas de XORs entre $e + 1$ paquets. Pour obtenir le degré du paquet résultant, on peut soit :

- considérer les e opérations XORs globalement
- considérer que les e opérations XORs sont réalisées successivement

Afin de simplifier le problème, et d'exploiter le résultat obtenu dans la partie précédente, nous avons choisi la deuxième option. Ainsi, on calcule à chaque étape $e \in \mathbb{N}^+$ le degré résultant entre le paquet $p_{e-1} \oplus p_e$. En conséquence, nous obtenons, après e étapes, la distribution des degrés comme suit d'après l'expression (4.12) :

$$p_{d_{xor_e}}(d_e) = \sum_{d_1=1}^K \sum_{d_0=1}^K p_{d_{xor_{e-1}}}(d_1) p_0(d_0) p_o(o = \frac{d_1 + d_2 - d_{xor}}{2} | d_1, d_0, K) \quad (4.13)$$

où $p_e(d_e)$ et $p_{e-1}(d_1)$ est la distribution des degrés du paquet après e XORs et $p_0(d_0)$ la distribution des degrés initiale sans XOR et que $p_e(d_e) = p_{e-1}(d_1)$ pour $e = 1$.

A l'aide de l'Equation (4.13), nous pouvons comparer l'évolution de la distribution des degrés en fonction du nombre d'étapes e . Ainsi, pour le code aléatoire linéaire RL, nous pouvons observer sur la Figure 4.5(a) que les opérations XORs successives ne modifient pas la distribution qui est initialement du type binomiale (les distributions obtenues après 2 XORs et 30 XORs successifs ont été tracées sur la Figure 4.6 pour montrer qu'il n'y a pas d'effet cumulatif qui apparaît. En effet, elle reste la même pour toutes les valeurs de e , avec une espérance du degré égale à $K/2$. Ceci nous confirme donc la neutralité de l'opération XOR sur les degrés d'un code RL.

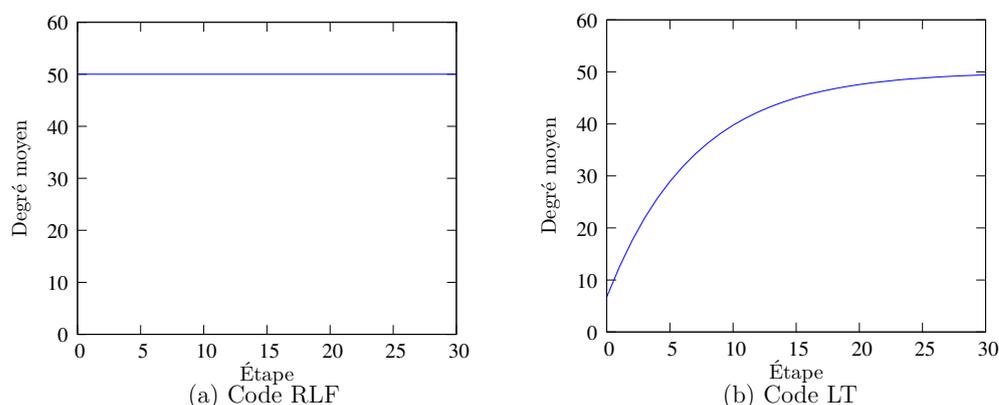


FIGURE 4.5 – Espérance du degré après e étape de XORs successives

A l'inverse, pour le code LT, nous notons une modification de la distribution après e XORs successifs selon la Figure 4.7. Nous pouvons observer que la distribution RSD évolue vers les degrés plus élevés jusqu'à ce que la distribution converge vers la distribution binomiale. D'autre part, l'espérance moyenne du degré de paquet émis après e XORs successifs est tracée sur la Figure 4.5(b). On observe que le degré moyen des paquets augmente jusqu'à $e \approx 30$ où

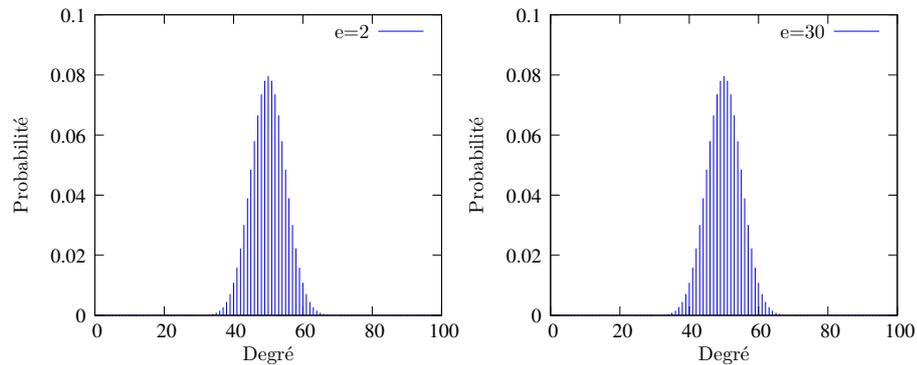


FIGURE 4.6 – Évolution de degré après e étapes de XORs successives pour le code RL

l'espérance du degré reste constante à $K/2$. Ceci est cohérent avec l'algorithme de sélection uniforme des fragments pendant le processus d'encodage. En effet, lorsque le nombre de XOR est suffisamment important, la probabilité qu'un fragment donné soit présent dans un nombre pair de paquets est la même que pour un nombre impair. Ainsi, chaque fragment a une probabilité $1/2$ d'être présent dans le paquet, ce qui correspond à la définition du code RL.

A partir de ces résultats, nous soulevons un problème majeur dû à la modification de la distribution des degrés après les combinaisons linéaires XOR. En effet, nous avons prouvé que la distribution de LT n'est plus conforme à la loi de distribution de Soliton même après avoir subi une seule opération XOR. Ceci rend donc difficile le processus de décodage BP du fait du risque de manque de degré «1» à chaque itération de décodage. Par conséquent, il n'est donc pas pertinent d'implémenter le codage de réseau sans bien prendre en compte de cette contrainte.

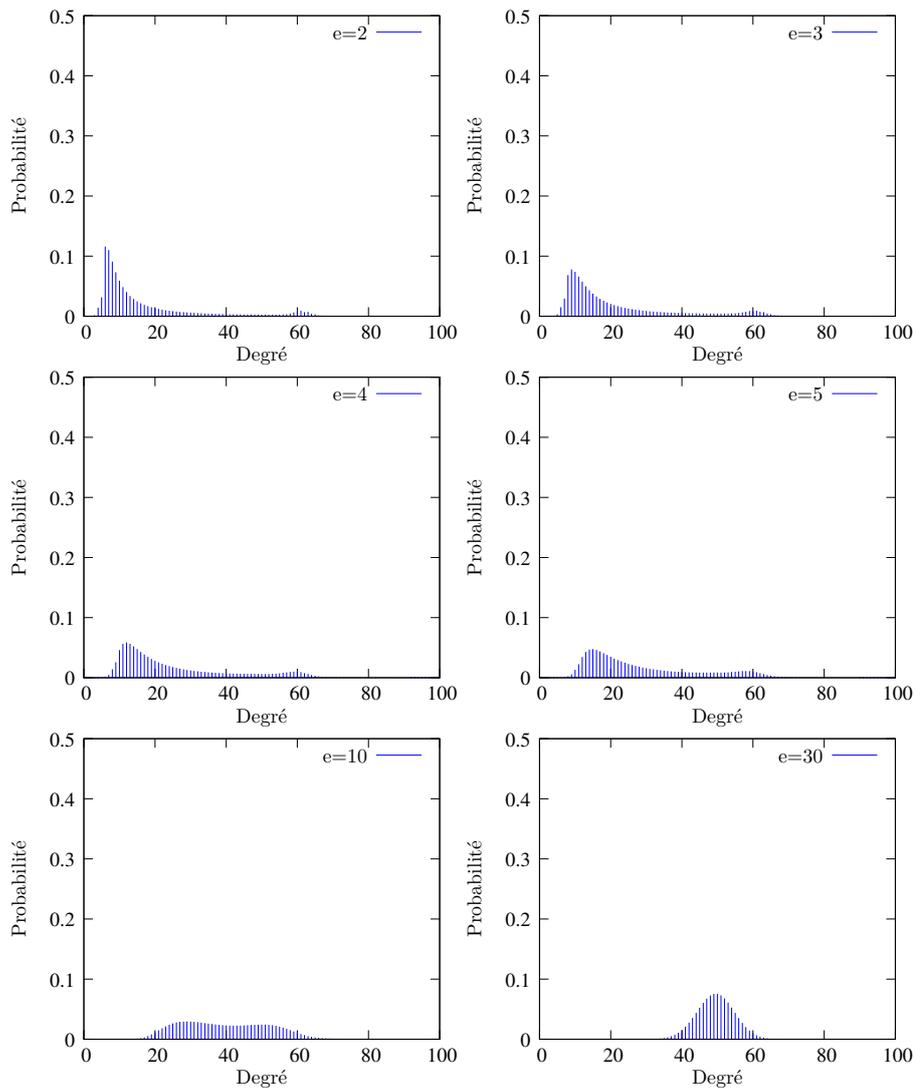


FIGURE 4.7 – Évolution de degré après e étapes de XORs successives pour le code LT

4.3 Comment XORer les paquets : approche théorique

Pour remédier au problème de la modification de la distribution introduite par le codage de réseau, nous proposons d'étudier dans cette partie plusieurs propositions dont le but est de relayer efficacement les données tout en conservant la distribution des degrés de flux encodé fontaine, même après des combinaisons linéaires.

4.3.1 Modélisation théorique

Nous nous intéressons premièrement à l'approche théorique qui consiste à choisir les paquets p_1 et p_2 à XORer de façon à obtenir un degré résultant d_{xor} selon une certaine distribution prédéfinie, grâce à l'équation (4.12). D'une manière générale, il est délicat de trouver la probabilité conjointe au choix de d_1 et d_2 car ceci conduit à résoudre K équations de $2K$ variables inconnues mais les inconnues étant des probabilités leur valeur est contrainte. Pour réduire le nombre d'inconnues, nous proposons de fixer la variable d_1 , puis de déterminer la probabilité de choix de d_2 qui permet d'obtenir la valeur de d_{xor} prédéfinie. En fixant la valeur de d_1 , l'équation (4.12) devient :

$$p_{d_{xor}}(d_{xor}|d_1) = \sum_{d_2=1}^K p_{d_2}(d_2) p_o(o = \frac{d_1 + d_2 - d_{xor}}{2} | d_1, d_2, K) \quad (4.14)$$

avec $\mathbf{P}_{\text{choice}}$, le vecteur des probabilités $p_{\text{choice}}(d_2)$. \mathbf{P}_d un vecteur qui représente la distribution obtenue en sortie du XOR. Étant donné que d_{xor} et d_1 sont fixés, il nous reste à déterminer la probabilité de choix de d_2 pour valider cette condition. Nous appelons cette probabilité $p_{d_2}(d_2) = p_{\text{choice}}(d_2)$. Comme nous l'avons présenté dans [112], la probabilité $p_{\text{choice}}(d_2)$ peut être obtenue à l'aide de la représentation matricielle de la manière suivante.

$$\mathbf{P}_d = \mathbf{M} \cdot \mathbf{P}_{\text{choice}} \quad (4.15)$$

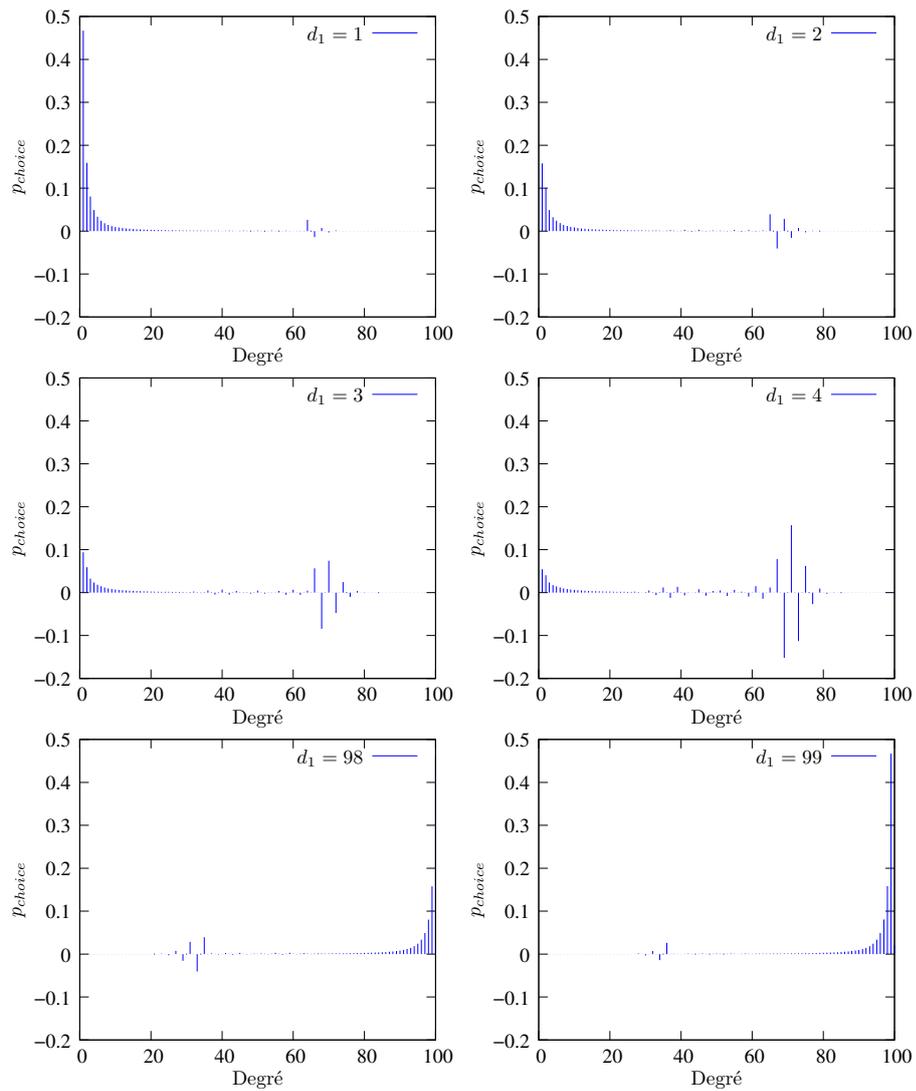
\mathbf{M} est la matrice dont les coefficients sont les probabilités de collisions entre p_1 et p_2 . Chaque coefficient $m_{i,j}$ est défini par $m_{i,j} = p_o(o = \frac{d_1+j-i}{2})$ déterminé par l'équation (4.11). Il faut noter que les éléments de \mathbf{M} dépendent de la valeur prise par d_1 , donc cette matrice est définie pour un degré fixé d_1 . D'autre part, on peut remarquer que la matrice \mathbf{M} est creuse car cette expression n'a un sens que si la collision $o \in \mathbb{N}$ c'est à dire si $d_1 + j - 1$ est pair et positif. Par exemple, pour le cas $d_1 = 2$, l'équation ((4.15)) s'écrit :

$$\begin{bmatrix} p_d(1) \\ p_d(2) \\ p_d(3) \\ \vdots \\ p_d(K) \end{bmatrix} = \begin{bmatrix} m_{1,1} & 0 & m_{1,3} & & 0 \\ 0 & m_{2,2} & 0 & m_{2,4} & \\ m_{3,1} & 0 & m_{3,3} & 0 & m_{3,5} \\ & \ddots & \ddots & \ddots & \ddots \\ & & & 0 & m_{K-2,K} \\ & & & 0 & m_{K-1,K-1} & 0 \\ 0 & & & m_{K,K-2} & 0 & m_{K,K} \end{bmatrix} \cdot \begin{bmatrix} p_{\text{choice}}(d_2 = 1) \\ p_{\text{choice}}(d_2 = 2) \\ \vdots \\ p_{\text{choice}}(d_2 = K) \end{bmatrix} \quad (4.16)$$

Nous faisons appel à la méthode d'approximation des moindres carrées pour résoudre cette

équation et obtenir $\mathbf{P}_{\text{choice}}$. Il faut noter que $\mathbf{P}_{\text{choice}}$ doit vérifier : $\forall i \in \{1, \dots, K\}$, $0 \leq p_{\text{choice}}(i) \leq 1$ et $\sum_{i=1}^K p_{\text{choice}}(i) = 1$. Nous obtenons après la résolution de l'Equation (4.15), la distribution de p_{choice} selon la Figure 4.8.

Nous observons que pour certaines valeurs de d_1 fixées, l'ensemble des solutions trouvées oscille et contient des solutions négatives surtout lorsque d_1 est assez grand. Cela traduit la difficulté de trouver une solution à l'équation tout en respectant les contraintes imposées sur p_{choice} . D'ailleurs, il faut remarquer que ceci est cohérent avec la pratique. En effet, pour une grande valeur de d_1 fixée, il est souvent difficile de trouver un autre paquet p_2 qui permette d'obtenir un degré d_{xor} faible. Cette limitation est accentuée lorsque K est grand car la probabilité de collision entre les fragments est fortement réduite. Nous pouvons en conclure que le système (4.15) ne donne pas systématiquement de solution exploitable. Cependant, puisque pour $d_1 = 1$ et $d_1 = 99$ n'oscille pas excessivement (cf. Figure 4.8), nous allons exploiter la distribution proposée par p_{choice} dans ce cas pour la suite des études dans une implémentation réelle.

FIGURE 4.8 – Distribution de p_{choice} théorique

4.3.2 Pertinence du modèle théorique

Nous souhaitons évaluer la pertinence de la distribution p_{choice} obtenue théoriquement dans la Section 4.3.1 par la simulation pour le code LT.

Nous considérons dans cette partie un système multisauts constitué de trois nœuds : une source S , un nœud intermédiaire R et un destinataire D selon la Figure 4.9. On considère que le routage est fixé. La source S envoie à R . Puis, R transmet ce paquet au destinataire D . Dans ce schéma de transmission, D ne peut pas recevoir de paquet directement de S . Ces nœuds sont reliés par des canaux que nous avons supposés sans effacement, ni interférence, et dont la probabilité γ du lien est égale à 1.

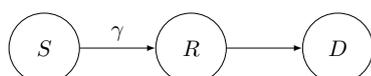


FIGURE 4.9 – Réseau linéaire à 3 sauts

Une mémoire interne de taille infinie est implémentée à chaque nœud du système. En conséquence, le relai R ne supprime aucun des paquets précédemment reçus de la source. Contrairement au cas sans XOR où chaque nœud ne fait qu'une simple retransmission sans traitement, le nœud intermédiaire dans notre contexte applique la combinaison linéaire XOR entre des paquets stockés en mémoire en choisissant les paquets à encoder de la manière suivante : lorsque R reçoit un paquet de la source, il enregistre ce paquet dans sa mémoire interne. Ensuite, il recherche dans sa mémoire le paquet le plus récemment enregistré dont le degré est égal à d_1 . Si aucun paquet de degré d_1 n'est présent dans la mémoire, R n'envoie pas d'information. Sinon, R choisit aléatoirement un autre paquet de degré d_2 où d_2 est généré à l'aide de la distribution de p_{choice} obtenue dans la partie précédente. Il faut vérifier que $p_2 \neq p_1$ pour éviter l'envoi d'un paquet sans données car tous les fragments auraient subi une collision, puis la combinaison linéaire entre p_1 et p_2 est émise. Ainsi, on se place dans le cas où le destinataire reçoit uniquement les paquets ré-encodés provenant du relais R . Ce processus est décrit dans l'algorithme 3.

Algorithme 3 : Encodage basé sur l'approche théorique

```

 $p_1 \leftarrow$  dernier paquet reçu dans la mémoire dont le degré est égal à  $d_1$ 
si  $p_1$  n'existe pas alors
  quitter
sinon
   $d_2 \leftarrow$  degré choisi selon  $p_{choice}$ 
   $p_2 \leftarrow$  paquet de degré  $d_2$  choisi aléatoirement dans la mémoire
  si  $p_2$  n'existe pas alors
    quitter
  fin si
fin si
 $p_{xor} \leftarrow p_1 \text{ XOR } p_2$ 
transmettre  $p_{xor}$ 

```

Nous avons implémenté l'Algorithme 3 sur la plateforme de simulation WSNNet pour $d_1 = 1$ et $K = 100$. Sur la Figure 4.10, avec un nombre d'itérations égal à 100, nous avons tracé la distribution du degré résultant obtenue par le destinataire comparé avec la distribution RSD.

Le résultat obtenu n'est pas aussi bon que ce qu'on aurait pu espérer. En effet, le modèle théorique développé nous permet de nous rapprocher approximativement de la distribution de Soliton.

Cependant, nous observons que les paquets de faible degré possèdent une probabilité plus basse d'être envoyés par cette méthode. Le manque de paquets de faible degré peut entraîner ainsi l'inefficacité du processus de décodage. Ainsi, la Figure 4.11 illustre la probabilité de réception de chaque fragment en fonction de l'indice de fragment reçu par le destinataire. Nous pouvons noter la non-uniformité du choix du fragment. Cela vient du fait que le ré-encodage par R nécessite à chaque fois un paquet de degré 1 pour le paquet p_1 et très souvent un autre paquet p_2 de degré 1 ou 2 (selon la distribution de p_{choice}). Or les degrés des paquets présents dans la mémoire interne du nœud intermédiaire suivent la distribution de Robust Soliton. Par conséquent, les paquets de degré 1 sont une ressource limitée et les mêmes paquets sont régulièrement choisis.

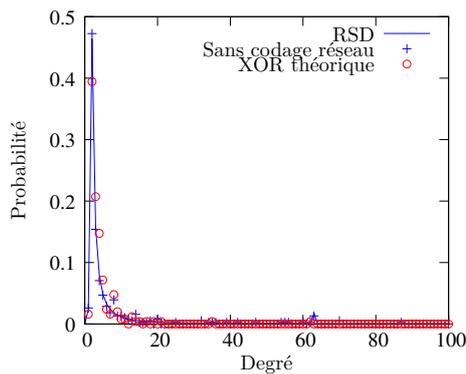


FIGURE 4.10 – Distribution des degrés résultant pour l'approche théorique

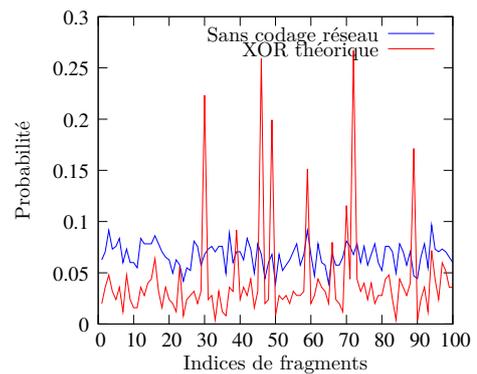


FIGURE 4.11 – Probabilité de répartition des fragments pour l'approche théorique

Ainsi, comme certains fragments ont une faible probabilité de transmission, le décodage prend plus de temps pour décoder tous les fragments. La Figure 4.12 représente le nombre de fragments décodés en fonction du nombre de paquets reçus pour 100 simulations comparé entre le cas sans codage réseau et le cas avec codage réseau par approche théorique. A partir de ces résultats, pour chaque nombre de paquets reçus, nous avons compté le nombre de fois que nous obtenons un décodage avec succès (Figure 4.13). Nous pouvons noter que le nombre de paquets nécessaire pour pouvoir décoder avec succès l'information initiale avec ré-encodage XOR par approche théorique est beaucoup plus élevé qu'au cas sans ré-encodage par le relai. Cette valeur s'élève à 918.28 paquets en moyenne ce qui dépasse largement le cas sans codage de réseau où cette moyenne s'élève à environ 136.3 paquets.

En conclusion, bien que cette modélisation théorique nous permette d'obtenir la distribution des degrés en sortie, en fonction de la distribution en entrée, nous n'avons pas réussi à l'exploiter pour choisir les paquets à ré-encoder au niveau de R de façon à conserver approximativement la

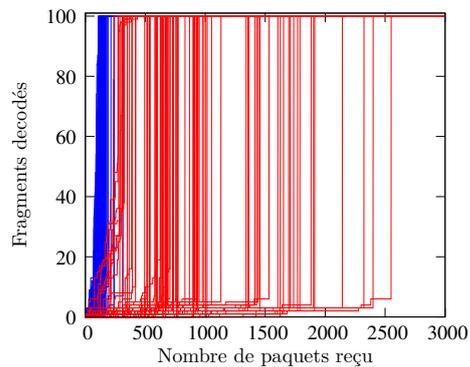


FIGURE 4.12 – Nombre de paquets décodés en fonction du nombre de paquets reçus pour le cas sans XOR (courbes en bleu) en comparaison avec le cas avec XOR réalisé par l’approche théorique (courbes en rouges)

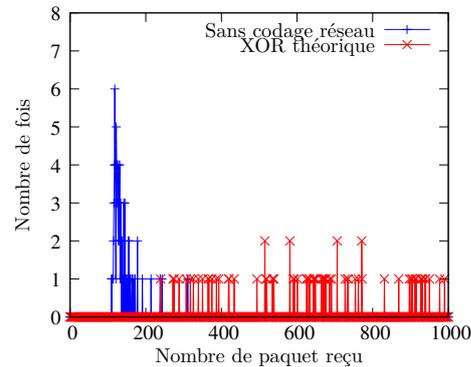


FIGURE 4.13 – Nombre de fois le message est décodé avec succès en fonction du nombre de paquets reçus au décodeur

distribution RSD. En effet, cette méthode ne donne pas de performances satisfaisantes car elle favorise la transmission de certains paquets (ex. paquet de degré d_1 systématiquement choisi au début du ré-encodage). Ce qui fait que les fragments reçus par le décodeur ne sont pas choisis uniformément. De plus, il est souvent difficile d’obtenir des paquets de degré 1 et 2 après la combinaison linéaire. Ce qui impose une grande latence au décodeur. Cette stratégie de relayage qui consiste à faire systématiquement un XOR entre les paquets reçus ne peut donc pas être utilisée telle qu’elle.

4.4 Comment XORer les paquets : approche heuristique

Dans cette partie, nous proposons et analysons plusieurs stratégies de relayage, dont le but est de conserver une distribution des degrés après XOR la plus conforme possible à la distribution RSD, tout en permettant un décodage efficace.

4.4.1 Modèle algorithmique

Nous considérons toujours un système constitué de 3 nœuds : une source S , un nœud intermédiaire R et un destinataire D avec le routage fixe source-relai et relai-destinataire (cf. Figure 4.9, pp.74). Les algorithmes de codage réseau implémentés dans R doivent permettre de choisir de façon intelligente les paquets à combiner pour construire un nouveau paquet dont le degré est cohérent avec la distribution de Soliton et permet d'introduire de la diversité de façon uniforme dans l'information.

Notre idée de base est de ne pas imposer un seul XOR, mais d'essayer de se rapprocher au maximum de la distribution RSD après un nombre adaptatif de XORs. Nous espérons obtenir un degré de paquet résultant qui est le plus proche possible de d prédéfini par la distribution RSD. Nous tirons tout d'abord, selon la distribution des degrés RSD, un degré de paquet d . Ce degré visé est le degré que l'on souhaite obtenir après le processus d'encodage par les relais. Par la suite, nous choisissons le paquet le plus récent dans la file d'attente. Pour un nombre maximal d'itération *maxround* initialement fixé, nous choisissons un paquet aléatoirement dans la file. Si la nouvelle combinaison linéaire avec le paquet récemment tiré de la file permet de s'approcher du degré d , nous gardons cette nouvelle combinaison. Sinon, nous conservons le paquet précédent avant la combinaison XOR. Une fois que le degré du paquet encodé atteint ce degré d visé, nous arrêtons cet algorithme de ré-encodage et le paquet est transmis. Cet algorithme est présenté selon l'Algorithme 4.

Algorithme 4 : Encodage basé sur l'approche heuristique

```

 $d \leftarrow$  degré défini par la loi de Robust Soliton
 $p \leftarrow$  dernier paquet reçu dans la mémoire interne
 $i \leftarrow 0$ 
tant que  $i < \textit{maxround}$  faire
   $p_{rand} \leftarrow$  choisi aléatoirement dans la mémoire
   $p_{xor} \leftarrow p \text{ XOR } p_{rand}$ 
  si degré( $p_{xor}$ ) est plus proche de  $d$  que degré( $p$ ) alors
     $p \leftarrow p_{xor}$ 
    si degré( $p_{xor}$ ) =  $d$  alors
      quitter
    fin si
  fin si
   $i \leftarrow i+1$ 
fin tant que
transmettre  $p_{xor}$ 

```

4.4.2 Validation de l'algorithme de codage réseau

Nous implémentons l'Algorithme 4 dans la plateforme de simulation pour $K = 100$ avec les paramètres de simulations $maxround = 100$ et une taille de mémoire égale à 100. La Figure 4.14 illustre la distribution du degré résultant après XOR. Nous avons vu que la distribution de faible degré est assez difficile à obtenir par cet algorithme mais l'allure de la distribution ressemble à la loi RSD. Nous observons, selon la Figure 4.15, que l'utilisation de l'Algorithme 4 nous permet d'obtenir une répartition des fragments reçus plus uniforme que dans le cas par approche théorique précédemment présentée.

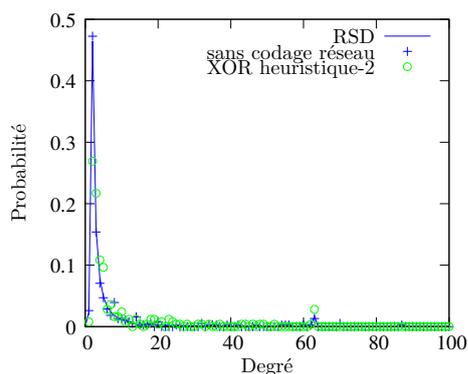


FIGURE 4.14 – Distribution des degrés pour l'approche heuristique

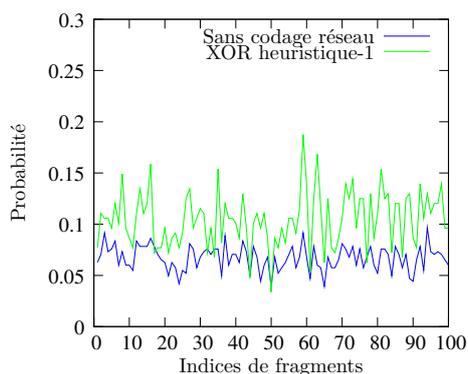


FIGURE 4.15 – Probabilité de répartition de fragments pour l'approche heuristique

Nous avons présenté dans la Figure 4.16 et 4.17 l'évolution de la rapidité du décodage pour le cas de codage réseau de façon heuristique comparé avec le cas sans ré-encodage par le relai. Nous avons observé que par l'approche heuristique proposé, le décodage est fait avec succès à partir d'un nombre moyen de 257.3 paquets reçus par le décodeur. Ceci est beaucoup plus rapide que l'approche théorique précédemment analysée. Cependant, l'obtention du degré 1 et 2 par algorithme 4 reste difficile à atteindre même pour le nombre d'itération $maxround$ très grand. Ceci provient de deux raisons :

- la probabilité de l'obtention d'un degré bas est faible après l'addition dans \mathbb{F}_2 lorsque K est grand,
- étant donné que la mémoire du relai contient un nombre limité de paquets parmi les paquets reçus, le choix du paquet est assez restreint.

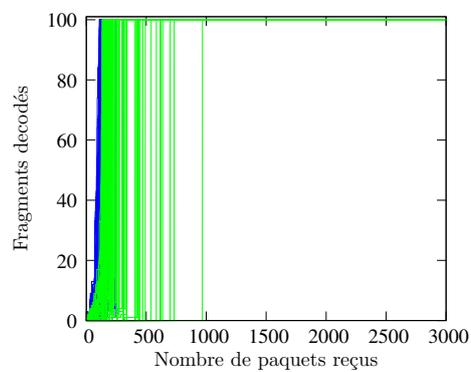


FIGURE 4.16 – Nombre de paquets décodés en fonction du nombre de paquets reçus pour le cas sans XOR (courbes en bleu) en comparaison avec le cas avec XOR réalisé par l'approche heuristique (courbes en verts)

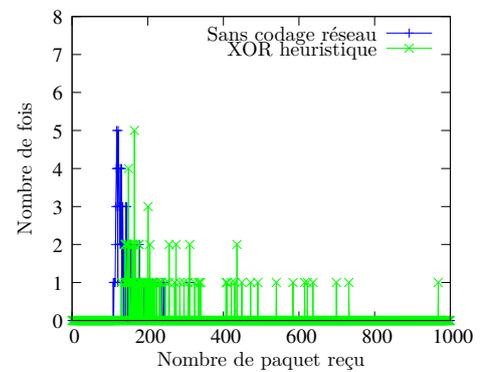


FIGURE 4.17 – Nombre de fois le message est décodé avec succès en fonction du nombre de paquets reçus au décodeur

4.4.3 Optimisation d'algorithme

Nous proposons une optimisation à apporter à l'Algorithme 4 pour favoriser la transmission des paquets de faible degré. Nous faisons appel à l'Algorithme 5 ayant pour but de réduire la probabilité d'appliquer les opérations sur les paquets de degré 1 et 2. Ceci revient à imposer à ces paquets une probabilité ξ d'être soumis à la combinaison linéaire XOR.

Algorithme 5 : Choix du degré pour l'approche heuristique avec optimisation

```

 $p \leftarrow$  dernier paquet reçu dans la mémoire interne
 $\xi \leftarrow$  probabilité de XOR
si degré( $p$ ) = 1 ou 2 alors
  si rand([0,1])  $\leq \xi$  alors
    appliquer l'algorithme 4
  sinon
    retransmettre sans XOR
  fin si
sinon
  appliquer l'algorithme 4
fin si

```

Nous observons, d'après la Figure 4.18 que l'Algorithme 5 apporte une amélioration des performances en terme de nombre de transmissions par rapport à l'Algorithme 4. La distribution du degré est plus conforme à la loi RSD. Nous pouvons considérer, d'après la Figure 4.19, que chaque fragment est plus uniformément transmis au destinataire.

Ensuite, nous présentons l'évaluation du nombre de paquets requis pour pouvoir décoder tous les fragments initiaux en fonction de ξ dans la Figure 4.20. En laissant passer des paquets de faible degré tel qu'ils sont, sans appliquer le codage réseau, le décodage est plus efficace. Nous avons observé que la valeur de $\xi \approx 0$ dans ce cas d'étude permet la meilleure optimisation en terme de nombre de transmissions.

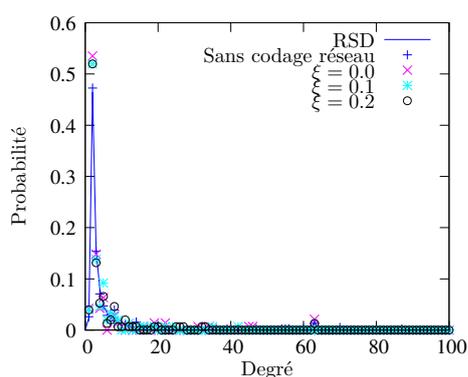


FIGURE 4.18 – Distribution des degrés pour l'approche heuristique avec optimisation

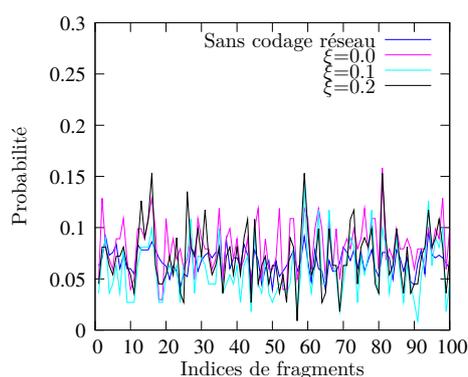


FIGURE 4.19 – Probabilité de répartition de fragments pour l'approche heuristique avec optimisation

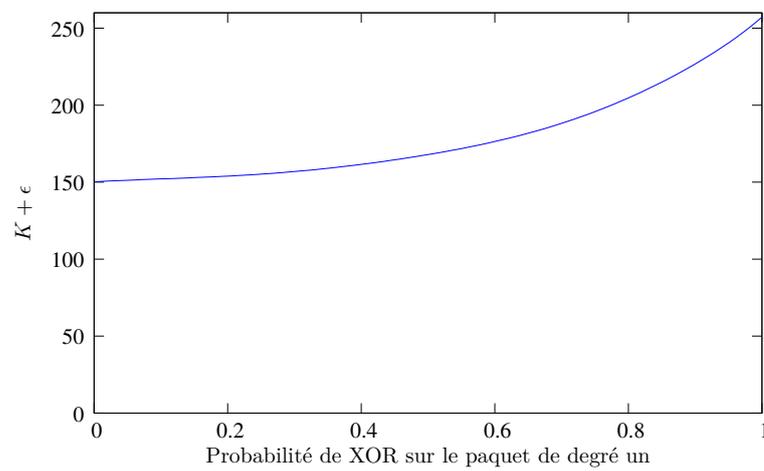


FIGURE 4.20 – Évolution du nombre moyen de paquets requis pour récupérer la totalité des fragments d'information en fonction de la probabilité de XOR.

4.5 Codage réseau pour un système linéaire multisauts

Nous voulons maintenant évaluer que nos stratégies passent à l'échelle et sont efficaces dans un réseau ayant plusieurs sauts. Pour cela, nous comparons dans cette partie le comportement des différentes stratégies de relayage avec codage réseau pour un réseau linéaire multisauts à l'aide des approches théoriques et heuristiques précédemment proposées dans les sections 4.3 et 4.4.

4.5.1 Contexte d'étude

Nous considérons un réseau linéaire composé de n_h sauts selon la Figure 4.21. Les nœuds dans le réseau sont reliés par des liens à effacement caractérisés par une probabilité de lien fixé à $\gamma = 0.4$ qui est supposée la même pour tous les liens. Nous choisissons une probabilité d'erreur assez élevée dans ce cas pour mettre en évidence l'impact de codage réseau. Le routage est fixé et non-opportuniste. Cela veut dire que le nœud ne peut entendre que les données qui lui sont destinées. Dans ce cas d'étude, nous considérons une transmission slottée dans laquelle chaque capteur ne transmet que dans un intervalle de temps qui lui a été attribué. Nous négligeons l'effet des interférences et nous limitons la réception des données à un saut. Nous supposons que le mécanisme d'acquiescement de bout en bout est parfait et que le surcoût provenant de ce mécanisme n'est pas pris en compte dans cette partie (cf. Section 3.5 pp.3.5).

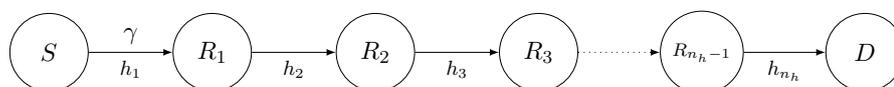


FIGURE 4.21 – Réseau linéaire à n_h sauts

4.5.2 Scénarios d'études

Nous comparons la performance des différents scénarios d'étude suivants :

- **Scénario CB** (relayage de Code fontaine par Bloc) : Chaque nœud décode totalement et ré-encode les informations avant de les transmettre au nœud suivant. Avec cette méthode, bien que le nombre de transmissions ainsi que la consommation d'énergie soient optimisés, la transmission souffre d'un fort délai et nécessite de grosse capacité de calcul dans les capteurs.
- **Scénario CP** (relayage de Code fontaine par Paquet) : Les nœuds intermédiaires retransmettent les paquets au fur à mesure dès leur réception. Le processus de décodage ne se fait qu'au nœud destinataire. Cependant, en cas d'échec, la retransmission doit s'effectuer depuis la source, ce qui est énergétiquement pénalisant.
- **Scénario CPR** (relayage de code fontaine par Paquet avec Réémission) : le dernier paquet reçu est systématiquement transmis dès qu'un intervalle de temps est attribué. Ceci veut dire que la rémission s'effectue à chaque intervalle de temps périodiquement même si le nœud ne reçoit pas de paquet du nœud précédent (dans ce cas, le dernier

paquet reçu est réémis).

Ces trois stratégies ci-dessus sont proposées comme référence sans codage réseau. Pour les 2 première (CB et CP), quand un nœud ne reçoit pas de paquet, il attend simplement le paquet suivant et n'utilise pas l'intervalle temporel qui lui a été attribué. A l'inverse, la stratégie CPR propose de profiter ce cette intervalle de temps pour augmenter la fiabilité de la transmission.

Dans toutes les stratégies suivants de codage réseau avec code fontaine, nous nous basons sur la stratégie CPR. Lorsqu'un nœud reçoit le paquet, il retransmet au nœud suivant le paquet tel qu'il a été reçu. Dans le cas contraire, lorsqu'un paquet n'est pas reçu, il envoie une combinaison linéaire XOR des paquets présents dans sa mémoire interne. Ceci permet d'améliorer l'allocation temporelle d'un lien radio, de réduire le délai et d'introduire de la diversité pour les données contenues dans les paquets. Nous proposons d'observer les stratégies suivantes qui sont destinées à appliquer efficacement le codage de réseau. Le cœur de notre contribution consiste à définir ces stratégies qui permettent à la fois de diversifier les fragments envoyés tout en préservant la distribution de Soliton Robuste des degrés des paquets.

- **Scénario CPR-XR** (relayage avec Réémission de code fontaine par r XORs successifs) : Chaque nœud transmet une combinaison XOR entre les r paquets les plus récents dès qu'un intervalle de temps est disponible. Or, sachant que les combinaisons XOR entre ces paquets entraînent la modification de la distribution des degrés et l'inefficacité de processus de décodage, cette stratégie est présentée comme résultat de référence dans le cas où aucun algorithme de conservation du degré n'est envisagé.
- **Scénario CPR-XT** (relayage avec Réémission de code fontaine par XOR de l'approche Théorique) : Le nœud choisit le paquet le plus récent dans la file qui possède un degré égal à un, puis un autre paquet selon la distribution de p_{choice} définie dans la section 4.3.2, et transmet la combinaison XOR entre ces deux paquets.
- **Scénario CPR-XH** (relayage avec réémission de code fontaine par XOR de l'approche Heuristique) : Cette approche a pour but de conserver la distribution de degré. Nous considérons deux techniques de ré-encodage par les relais (a) lorsqu'on choisit d'une manière spécifique selon l'Algorithme 4 les paquets dans la file de façon à produire à la sortie des paquets ré-encodés dont leur distribution suit la loi RSD. Or, du fait de la limitation de la taille finie de la mémoire, il est parfois difficile d'obtenir la distribution visée du degré. Nous proposons ainsi un autre cas d'étude (b) qui tend à lutter contre cette limitation à l'aide de l'Algorithme 5 dont le but est de favoriser le choix des paquets de faible degré.

Remarque : Nous avons testé et simulé plusieurs méthodes de codage de réseau dans le but de chercher une solution optimale. Nous n'avons présenté ici que les résultats qui donnent les meilleures performances de transmission.

Ainsi, nous avons constaté qu'en choisissant pour le premier paquet, le dernier paquet reçu dans la mémoire puis en appliquant la combinaison avec ce paquet, la performance est supérieure au cas où on choisit aléatoirement tous les paquets dans la mémoire.

4.5.3 Simulations et Résultats

Après avoir implémenté ces différentes techniques de codage réseau sur une plateforme de simulation, nous évaluons les résultats obtenus en terme de nombre minimal de transmissions depuis la source pour obtenir un succès de réception de $K = 100$ fragments au destinataire après i -th saut ($i \in [1, 10]$) selon la Figure 4.21. En effet, ce paramètre de mesure est à la fois proportionnel à l'énergie consommée et au délai de transmission de bout-en-bout du système. Nos résultats de simulation sont présentés sur les Figures 4.22 et 4.23 où nous évaluons le nombre moyen de transmissions qui permettent de recevoir un fragment d'information avec succès. Nous pouvons observer qu'avec un nombre de sauts croissants, le nombre de transmissions croît avec le nombre de sauts.

Nous présentons les résultats pour deux types de décodage : décodage par vraisemblance maximale (ML) et décodage par propagation de croyance (BP). En effet, le code LT a été conçu pour permettre l'utilisation du décodage BP. Cependant, le décodage par ML est considéré ici uniquement pour donner le nombre minimal de transmissions. En effet, le décodage ML permet de savoir à partir de quel moment tous les fragments sont présents et décodables. Ceci revient à chercher le nombre de paquets linéairement indépendants permettant d'avoir une matrice de rang K . Ainsi, le décodage ML permet généralement un décodage bien meilleur que le décodage BP. Les résultats dans cette partie sont présentés pour le cas d'une transmission où les données sont encodées avec le code LT par la source.

Cette étude confirme que le relayage CB lorsqu'on décode et ré-encode à chaque saut donne une borne minimale en terme de nombre total de transmissions par la source. Cependant, ce mécanisme de *décode-et-réencode* impose une grande latence et une complexité de calcul élevés. Nous essayons de nous rapprocher de cette bande avec les autres techniques.

Les résultats dans la Figure 4.22 avec le décodage ML confirment que d'appliquer les opérations XOR réduit le nombre de transmissions requises par la source. En effet, l'addition dans \mathbb{F}_2 par la combinaison linéaire XOR entre les paquets encodés par le code LT a tendance à augmenter l'espérance du degré des paquets codés (cf. Section 4.2.2 pp.68), les transmissions apportent plus d'information car le degré du paquet correspond au nombre moyen de fragments contenus dans le paquet. Ainsi, la diversité d'information contenue dans les paquets est augmentée. Ce dernier peut être affirmé par le résultat de la courbe pour les scénarios CPR-XH optimisés. Lorsqu'on renforce le degré de paquets transmis selon la distribution RSD. La transmission avec XOR donne une meilleure performance que le cas sans XOR.

Cependant, si le décodage BP est utilisé, selon la Figure 4.23, nous pouvons noter que la plupart des transmissions avec codage réseau conduisent à la dégradation des performances (cette dégradation de la distribution des degrés à 10 sauts est présentée dans la figure 4.24). Nous avons remarqué que si aucun mécanisme de protection n'est mis en place, la performance est très dégradée du à l'inefficacité de décodage (ex. scénario de relayage CPR-XR).

Notre stratégie de codage réseau proposée par l'approche théorique ne permet malheureusement pas d'atteindre la distribution RSD après quelques sauts dans la pratique. Ceci est du à la limitation de l'obtention de paquets de faible degré après XOR. A l'inverse, l'approche heuristique proposée permet d'approcher la bande minimale en terme de nombre de transmissions. Il permet également d'obtenir une distribution résultante satisfaisante. Comparé avec la distribution de Soliton Robuste, le relayage (a)CPR, (b)CPR-XT, (c)CPR-XH et (d)CPR-XH optimisé

avec $\xi = 0.2$ donne les erreurs quadratiques moyennes¹ de $3.8 \cdot 10^{-5}$, $2.8 \cdot 10^{-3}$, $8.14 \cdot 10^{-4}$ et $6.25 \cdot 10^{-5}$ respectivement. Nous pouvons voir que le relayage CPR-XH optimisé avec $\xi = 0.2$ permet de conserver la distribution RSD.

Notre algorithme proposé CPR-XH avec optimisation donne un bon compromis entre la diversité introduite et l'efficacité de codage. Cela permet d'approcher la bande minimale en terme de nombre de transmissions (obtenue par le relayage CB) tout en réduisant la complexité de calcul et le traitement à chaque nœud capteur, ainsi qu'en réduisant la latence de transmission globale.

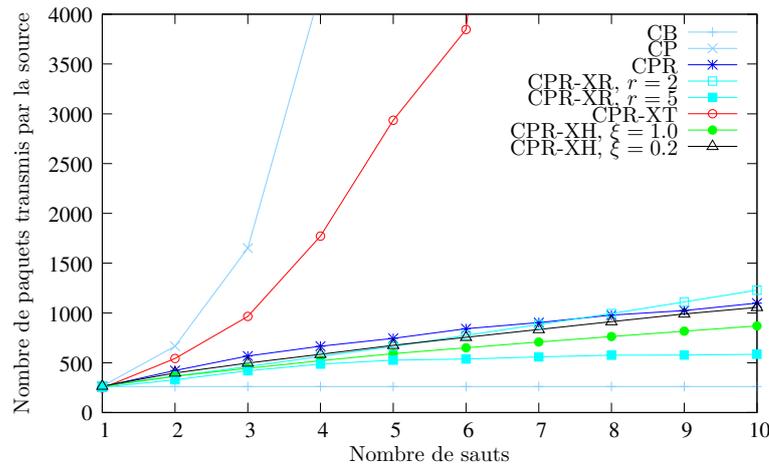


FIGURE 4.22 – Performance du système pour le décodage par vraisemblance maximale (ML)

4.6 Conclusion

Dans ce chapitre, nous avons proposé de nouvelles techniques de codage réseau utilisant les codes LT pour exploiter la diversité d'information dans les réseaux de capteurs sans fil. Avec le codage réseau, l'information contenue dans chaque paquet envoyé est diversifiée. Notre travail est basé sur le codage LT qui est adapté aux réseaux de capteurs grâce à sa faible complexité de codage et décodage. Nous proposons une expression théorique d'évaluation de la distribution des degrés de flux encodé LT après avoir subi au codage réseau par les combinaisons linéaires XOR. Nous soulevons le problème de la modification de la distribution des degrés par processus de ré-encodage par les nœuds intermédiaires entre les chemins. Nous avons montré la nécessité de réaliser un compromis entre la diversité d'information introduite dans les paquets reçus par le codage réseau et l'efficacité de décodage.

Nous avons proposé plusieurs techniques de codage réseau pour le code LT qui tendent à

1. L'erreur quadratique moyenne ou *mean square error* (MSE) est une façon d'estimer les erreurs en calculant la différence moyenne entre la valeur exacte visée θ et $\hat{\theta}$ estimateur du paramètre θ . Cette erreur quadratique moyenne peut être calculée par $MSE = E[(\hat{\theta} - \theta)^2]$. Dans notre étude, θ est le vecteur des degrés.

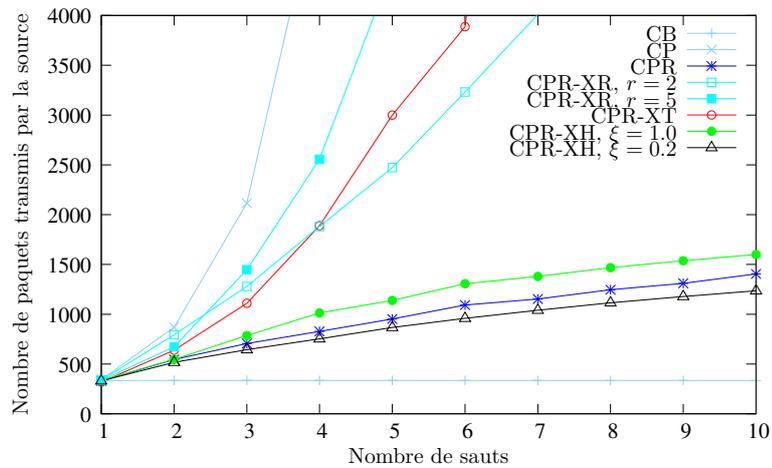


FIGURE 4.23 – Performance du système pour le décodage par propagation de croyance (BP)

préservent la distribution du degré par l'approche théorique, ainsi que par l'approche heuristique. Nous avons montré que nos algorithmes proposés permettent de conserver avec succès la distribution des degrés de paquets encodés après les opérations XOR, ce qui est un paramètre primordial pour que le processus de décodage se poursuive correctement. Cette technique proposée est simple et donne une performance de transmission efficace.

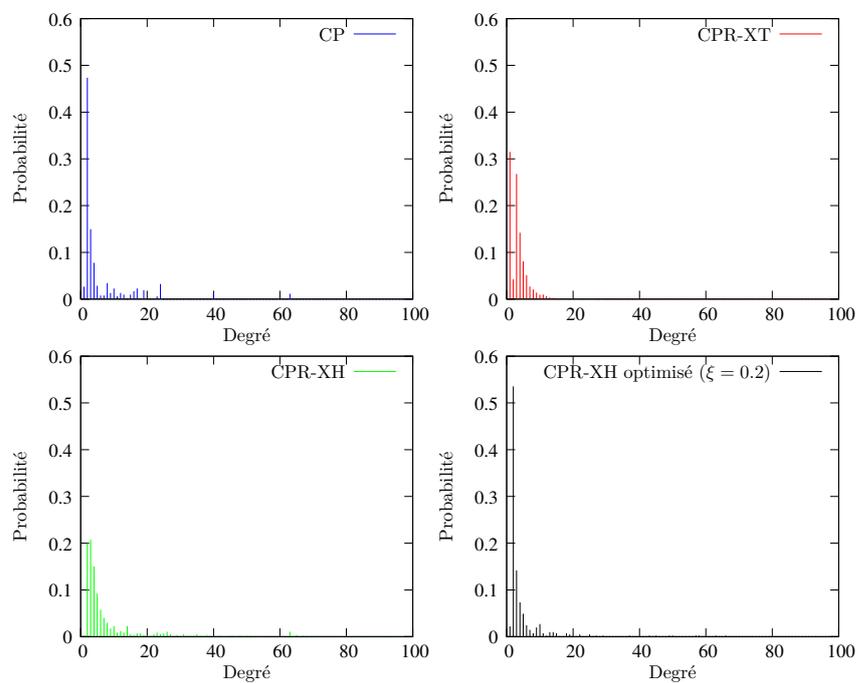


FIGURE 4.24 – Distribution des degrés après 10 sauts pour divers stratégies de transmission : (a) CPR, (b) CPR-XT, (c) CPR-XH et (d) CPR-XH optimisé avec $\xi = 0.2$

5

Code fontaine pour la coopération

Dans le chapitre précédent, nous avons étudié la diversité d'information obtenue à partir du codage réseau pour un flux traversant un chemin unique composé de plusieurs relais. Dans ce chapitre, nous nous intéressons à la transmission à plus large échelle où le flux de données peut être diffusé dans un réseau de capteurs sans fil. Dans ce cas, un même flux peut utiliser plusieurs chemins multi-sauts dans le réseau. Pour cela, nous nous intéressons au cas où des nœuds peuvent coopérer pour relayer les paquets vers la destination. On parlera dans ce cas de *réseau coopératif*. Dans ce type de réseau, il est possible d'augmenter le débit de bout-en-bout tout en optimisant la consommation des ressources dans le système.

Lors d'une communication coopérative, la source diffuse ses informations vers le destinataire. On peut utiliser un routage de type *opportuniste multi-sauts* pour relayer les paquets de proche en proche. Au lieu de jeter un paquet qui ne lui est pas destiné, un relai opportuniste re-diffuse ce message avec ou sans traitement à ses nœuds voisins (voisins à un saut). Avec ce type de retransmission, un message envoyé atteint son destinataire en un certain nombre de sauts et généralement au travers des chemins différents parallèles.

Comme le message parcourt différents chemins dans le réseau, les copies transmises par les différents relais subissent différentes réalisations du canal. Ainsi, on peut tirer parti de cette diversité multi-chemins pour améliorer la fiabilité de la transmission. Cette diversité est d'autant plus avantageuse lorsque le lien direct source-destinataire est dégradé ou impossible. Nous appellerons cette diversité spatiale la *diversité de coopération* ou la *diversité multi-utilisateurs*. Dans un réseau de capteurs, une telle diversité s'avère très intéressante [174, 68] Quant à la limitation de la taille et des ressources des capteurs, cette technique peut être considérée comme une solution alternative à l'intégration d'antennes multiples dans les capteurs qui est très contraignante.

Le prix de cette diversité multi-utilisateurs est la transmission d'un nombre important de

paquets redondants dans le réseau. Des techniques algorithmiques ont été proposées [145, 174, 68] pour améliorer le compromis entre la fiabilité du réseau et la consommation énergétique de l'algorithme de diffusion. Comme il est difficile de réduire cette redondance sans perdre en connectivité dans le réseau, nous proposons pour conclure cette thèse d'introduire un codage fontaine et un codage réseau pour pouvoir augmenter la fiabilité du réseau sans augmenter la redondance dans le réseau, et ainsi améliorer le compromis entre l'énergie et la fiabilité du réseau.

Dans un premier temps, nous nous focalisons sur l'étude de la configuration élémentaire d'un tel réseau coopératif : le canal à relai. Nous analysons analytiquement et par simulation l'apport du codage réseau pour ce canal à relai. Pour cela, nous considérons que le relai transmet des combinaisons linéaires de paquets dans \mathbb{F}_2 qui sont choisis parmi les données brutes non-encodées précédemment reçues et enregistrées dans une file d'attente. Ensuite, nous élargissons notre étude au cas d'un réseau de capteurs à plus grande échelle pour une pile protocolaire réaliste. Dans ce chapitre, nous relaxons l'hypothèse d'un multiplexage temporel pour l'accès au médium radio. En effet, dans un réseau à large échelle, une telle hypothèse n'est plus réalisable. Ainsi, dans ce chapitre, nous considérons un accès de type CSMA/CA pour lequel il est nécessaire de prendre en compte des interférences et des collisions provenant des utilisateurs multiples.

Grâce à la redondance introduite par le code fontaine et la diversité d'information obtenue à partir du codage réseau, nous présentons une solution de relayage pertinente qui nous permet d'obtenir une transmission robuste dans un réseau de capteurs à moindre coût énergétique.

Les publications relatives aux résultats de ce chapitre sont actuellement en cours de rédaction.

5.1 Communication coopérative

Il a été montré dans [83, 140, 141] que la coopération des capteurs permet d'augmenter la performance globale d'une communication. La coopération entre les utilisateurs du système a été proposée comme une alternative à la transmission multi-antennes. Cette coopération impose aux nœuds de relayer les informations pour aider la transmission d'une source vers le destinataire. La diversité peut ainsi être anticipée en fonction du nombre de chemins de propagation. Il existe plusieurs modes de coopération qui permettent d'obtenir ce type de diversité. Laneman et al. ont proposé dans [83] deux techniques de codage qui sont basées sur les deux modes de coopération suivants :

- **Amplify-and-Forward (AF)** : les relais amplifient simplement le signal reçu avant de le transmettre à ses voisins,
- **Decode-and-Forward (DF)** : les relais décodent et ré-encodent le signal avant de le transmettre.

La performance des deux modes de coopération précédents a été analysée dans [175]. Parallèlement à ces deux techniques, les solutions plus récentes pour la transmission coopérative s'orientent vers le déploiement de codes correcteurs d'erreurs pour augmenter la capacité de transmission et réduire la probabilité d'erreur.

Il faut noter qu'il existe plusieurs niveaux de coopération dans la pile protocolaire. Initialement conçue pour la couche physique, la première notion de codage pour la coopération se

concentre sur l'encodage des bits d'information émis sur le médium radio [65, 178, 124]. Le codage coopératif peut être également réalisé par une couche plus haute de la pile protocolaire afin de gérer efficacement les flux provenant de différents utilisateurs. C'est le cas par exemple de la technique du codage réseau présentée dans le chapitre précédent. Un tel codage permet d'augmenter la robustesse de la transmission.

Dans ce chapitre, nous nous intéressons au codage coopératif, et plus particulièrement au codage fontaine et au codage réseau dans le contexte des réseaux de capteurs. Les algorithmes de relayage proposés sont définis au niveau de la couche réseau où les relais peuvent si besoin décoder et ré-encoder des messages reçus.

5.2 Code fontaine pour la coopération

Nous présentons d'abord un état de l'art général sur l'utilisation d'un code fontaine pour la coopération. Nous en déduisons la motivation et les principaux axes de recherche envisagés dans ce chapitre.

5.2.1 Etat de l'art

Les codes sans rendement ou les codes fontaine ont été montré efficaces pour toutes les configurations de canaux pour un système d'un à plusieurs relais [30, 31, 110, 92]. Ces codes permettent d'accumuler l'information transmise et se comportent plus efficacement si le canal est perturbé [110]. Au niveau de la couche physique, la méthode de relayage avec code fontaine nommée *Fountain-Coding-and-Forward* (FCF) a été montrée plus performante que les techniques AF, DF ou que la transmission seule sur un canal direct [124].

Pour les couches réseau et application, la coopération entre les nœuds peut être obtenue par exemple par un routage opportuniste ou le codage réseau. Les techniques de coopération ont pour but de dériver des algorithmes de relayage qui maximisent débit, minimisent le délai de transmission, etc. Par exemple, la coopération par un code réseau appliqué sur un code à effacement a été présentée dans [172, 81]. Dans [172], cette technique permet de réduire le nombre de paquets redondants dans un réseau où deux utilisateurs coopèrent. Dans ce schéma, les deux utilisateurs échangent leur information et mutuellement envoient l'information échangée et encodée au destinataire. Selon [81], cette technique permet également de réduire le nombre total de transmissions et d'augmenter l'utilisation spectrale du système.

Pour le routage dans les réseaux de capteurs, un schéma de stockage distribué coopératif des données dans le réseau a été proposé à l'aide d'un code fontaine. Cette technique est appelée dans [164] *codes fontaines distribués*. Le code fontaine est employé pour distribuer K fragments d'information sur un ensemble de N capteurs du réseau. Contrairement aux codes fontaines distribués, nous nous intéressons dans ce chapitre à l'acheminement des paquets d'un nœud vers le puits au travers de plusieurs sauts. La coopération se définit dans ce cas à l'aide de différents algorithmes de relayage qui ont pour objet de réduire le nombre de transmissions ainsi que la consommation de ressources du système au minimum.

5.2.2 Motivation

Généralement, dans un relayage coopératif utilisant des codes fontaine, la plupart des travaux se focalisent principalement sur les questions suivantes :

- Quels nœuds doivent relayer l'information (ex. le choix pertinent de relais [111]) ?
- Comment relaie-t-on l'information (ex. avec ou sans codage) ?
- A quel moment faut-il relayer l'information ?

La contribution principale de ce chapitre de thèse se focalise plutôt sur les deux dernières questions en proposant des algorithmes efficaces de coopération entre les nœuds dans le réseau. Nous avons vu dans le chapitre précédent que grâce à la diversité d'information introduite par le codage réseau, le nombre de transmissions peut être réduit dans une transmission multi-sauts. Par conséquent, nous adoptons le codage réseau pour améliorer l'efficacité de la coopération dans un réseau de capteurs.

Nous introduisons aussi un troisième degré de liberté qu'est la décision de routage d'un relai. La décision de routage est modélisée à l'aide d'un *taux de relayage* qui n'est autre que la probabilité par un relai de retransmettre un paquet reçu. Dans ce contexte, la transmission n'est plus slottée, il devient nécessaire de prendre en compte les interférences et les collisions provenant des utilisateurs multiples du réseau pour évaluer les performances du réseau. Ce taux de relayage nous permet également de contrôler la consommation énergétique du réseau car il permet de limiter le nombre de re-transmissions introduites par chaque relai. Notre objectif est de trouver un moyen efficace pour réduire ce taux de relayage et réduire ainsi l'énergie consommée tout en gardant une transmission fiable à l'aide de la diversité d'information introduite par le code fontaine et le codage réseau.

5.3 Codage réseau pour le canal à relai

Cette section s'intéresse à l'apport du codage réseau pour le canal à relai.

5.3.1 Caractéristique d'un canal à relai

Le canal à relai a été introduit la première fois par Van der Muelen dans [109]. C'est un système composé de trois nœuds : une source S, un relai R et un destinataire D. La source envoie les données vers le destinataire en passant soit par le lien direct S – D, soit par le nœud intermédiaire en utilisant les liens S – R et R – D, soit en utilisant les deux chemins en parallèle. Le canal à relai permet d'augmenter le débit de transmission, d'augmenter la robustesse et d'améliorer la diversité spatiale de la transmission. Il permet également d'économiser la puissance de transmission pour chaque nœud individuellement et de mieux répartir la puissance consommée par les nœuds dans le réseau. Chaque lien est caractérisé par la probabilité du lien γ modélisée par le taux d'effacement présent sur le lien $\gamma=1$ -PER. Nous définissons ainsi la probabilité de succès du lien S – D, S – R et R – D comme γ_{SD} , γ_{SR} et γ_{RD} respectivement. Nous supposons que tous les nœuds possèdent une seule antenne émettrice et réceptrice et que les nœuds ne peuvent pas émettre et recevoir simultanément (canal semi-duplex).

Notre étude se porte sur le cas où le relai re-combine à l'aide d'un codage réseau sur \mathbb{F}_2 les paquets émis par la source encodés à l'aide d'un code fontaine. Les paquets peuvent être

démodulés et décodés si besoin avant d'être transmis (DF). Le destinataire D reçoit un même paquet de la source à partir de deux chemins de propagation différents. Le schéma du système de transmission est présenté sur la Figure 5.1.

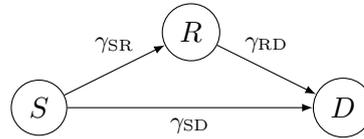


FIGURE 5.1 – Canal à relai

5.3.2 Modélisation de la transmission

Nous désignons une *étape de transmission*, une transmission complète d'un paquet émis depuis la source jusqu'au destinataire. Chaque étape de transmission est composée de deux intervalles de temps consécutifs distincts. Pour l'étude du canal à relai, l'accès au canal est géré par un multiplexage temporel *Time Division Multiple Access* (TDMA). La Table 5.1 présente trois protocoles TDMA qui peuvent être envisagés.

intervalle de temps	protocole I	protocole II	protocole III
1	$S \rightarrow \{R, D\}$	$S \rightarrow \{R, D\}$	$S \rightarrow R$
2	$\{S, R\} \rightarrow D$	$R \rightarrow D$	$\{S, R\} \rightarrow D$

TABLE 5.1 – Les différents protocoles TDMA

Dans les protocoles I et III, le fait que la source transmette en même temps que le relai peut entraîner des interférences et des collisions dans le système. Puisque notre objectif se concentre principalement sur les techniques de relayage par le relai, nous voulons éviter de prendre en considération une technique de réjection d'interférences. Ainsi, nous utilisons le protocole II dans la suite de cette étude où la source transmet au premier intervalle de temps et le relai au second intervalle. Nous considérons que les deux intervalles de temps sont disjoints et que les deux transmissions n'interfèrent pas entre elles. Dans le premier intervalle de temps, la source effectue une diffusion multiple. Le relai et le destinataire reçoivent simultanément ses données. Ensuite, dans un deuxième intervalle de temps, seul le relai peut retransmettre ces informations au destinataire.

Nous nous intéressons ensuite à définir la probabilité du succès équivalente du système à relai. Pour un canal à relai, un paquet est envoyé avec succès sur un système à relai lorsqu'il est transmis avec succès sur le lien S-D avec la probabilité γ_{SD} , sinon transmis avec succès sur le lien à relai avec la probabilité $\gamma_{SR}\gamma_{RD}$. La probabilité du succès équivalent du système est ainsi égale à :

$$\gamma = \gamma_{SD} + (1 - \gamma_{SD})(\gamma_{SR}\gamma_{RD}). \quad (5.1)$$

La performance du système peut être améliorée si le relai peut traiter ses données avant de les relayer. Ainsi, nous proposons d'étudier dans cette partie les performances de quelques

algorithmes de codage réseau par une analyse théorique de la probabilité de succès de la transmission. Celle-ci nous permettra de valider les bénéfices du codage réseau pour un canal à relai.

5.3.3 Etude théorique

Premièrement, nous considérons le cas où la source envoie des paquets non-encodés. Le relai applique le codage entre les paquets reçus avant de les transmettre au destinataire. Supposons que la source S veuille émettre une série de paquets $P = \{p_1, p_2, \dots, p_\infty\}$ à D. Le succès de la transmission de ce paquet dépend à la fois de la qualité du lien direct et de celle du lien de relai. Il dépend également du choix de l'algorithme de codage réseau implanté dans un relai. En fonction de l'algorithme de codage, le paquet présente une probabilité de succès différente. Dans la suite, nous nous intéressons à deux cas d'étude simples qui sont

1. le cas où le relai combine des paquets *émis consécutivement* par la source et
2. le cas où le relai combine des derniers paquets *reçus*.

Dans le premier cas, nous sommes dans le cas où la diversité est maximale car il n'y a pas de pertes sur le lien S-R. Dans le second cas, on a des pertes sur le canal S-R et on ne peut plus appliquer le premier algorithme de relayage.

a) Cas à diversité maximale ($\gamma_{SR} = 1$)

Nous présentons en premier, le cas de la diversité maximale où la probabilité du lien S-R est égale à 1 ($\gamma_{SR} = 1$). Autrement dit, le lien S-R est parfait. Dans ce cas, l'ordre des paquets disponibles au relai est le même que celui des paquets envoyés par la source. Dans ce cas, les deux techniques de combinaison sont équivalentes. A chaque réception de paquet, le relai transmet la combinaison XOR entre les deux derniers paquets reçus dans sa file d'attente. La séquence de transmission de chaque étape E est décrite dans la Table 5.2.

Etape	Lien		
	S – D	S – R	R – D
1	p_1	p_1	–
2	p_2	p_2	p_1
3	p_3	p_3	$p_2 \oplus p_1$
4	p_4	p_4	$p_3 \oplus p_2$
\vdots	\vdots	\vdots	\vdots
i	p_i	p_i	$p_{i-2} \oplus p_{i-1}$
$i + 1$	p_{i+1}	p_{i+1}	$p_{i-1} \oplus p_i$
$i + 2$	p_{i+2}	p_{i+2}	$p_i \oplus p_{i+1}$
$i + 3$	p_{i+3}	p_{i+3}	$p_{i+1} \oplus p_{i+2}$

TABLE 5.2 – Les séquences de transmissions lorsqu'on suppose que le lien S – R est parfait avec $\gamma_{SR} = 1$

Nous cherchons à calculer la probabilité de succès équivalente du canal à relai utilisant cet algorithme de codage réseau. Nous considérons un décodage itératif dans lequel la connaissance

du paquet p_i peut être obtenue de 3 façons différentes :

- Premièrement, lorsque p_i est reçu avec succès sur le lien direct S-D avec la probabilité de succès égale à γ_{SD} .
- Deuxièmement, lorsque la combinaison XOR des paquets $p_i \oplus p_{i-1}$ est reçue avec succès par le destinataire sur le chemin $S - R - D$ à condition que p_{i-1} ait été précédemment reçu et décodé avec succès. Nous parlerons ici d'un *décodage dans le sens croissant* : le cas où l'obtention de p_i se réalise à partir du paquet p_{i-1} . Pour la suite, nous introduisons la notation $\vec{\Gamma}$ pour représenter la probabilité de décodage dans le sens croissant.
- Troisièmement, de la même manière, le paquets p_i peut être récupéré dans le cas où la combinaison XOR des paquets $p_i \oplus p_{i+1}$ est reçue avec succès par le destinataire sur $S - R - D$ à condition que le prochain paquet p_{i+1} soit reçu et décodé. Ceci permettra un décodage en aval de p_i . Nous parlerons ainsi du *décodage dans le sens décroissant* : le cas où un nouveau paquet permet de décoder des paquets jusqu'alors non dé-codables. Nous introduisons également la notation $\overleftarrow{\Gamma}$ pour représenter la probabilité de décodage dans le sens décroissant.

Nous présentons dans la Figure 5.2 les événements possibles qui peuvent se produire pendant la transmission. Le succès de transmission d'un paquet peut être obtenu par le succès de transmission du paquet sur le lien direct, ou bien sur le lien de relaying. La probabilité de succès par le lien de relaying est plus difficile à déterminer car elle dépend à la fois du succès de réception d'un paquet combiné en R et des réceptions aux instants précédents et suivants des paquets combinés en R .

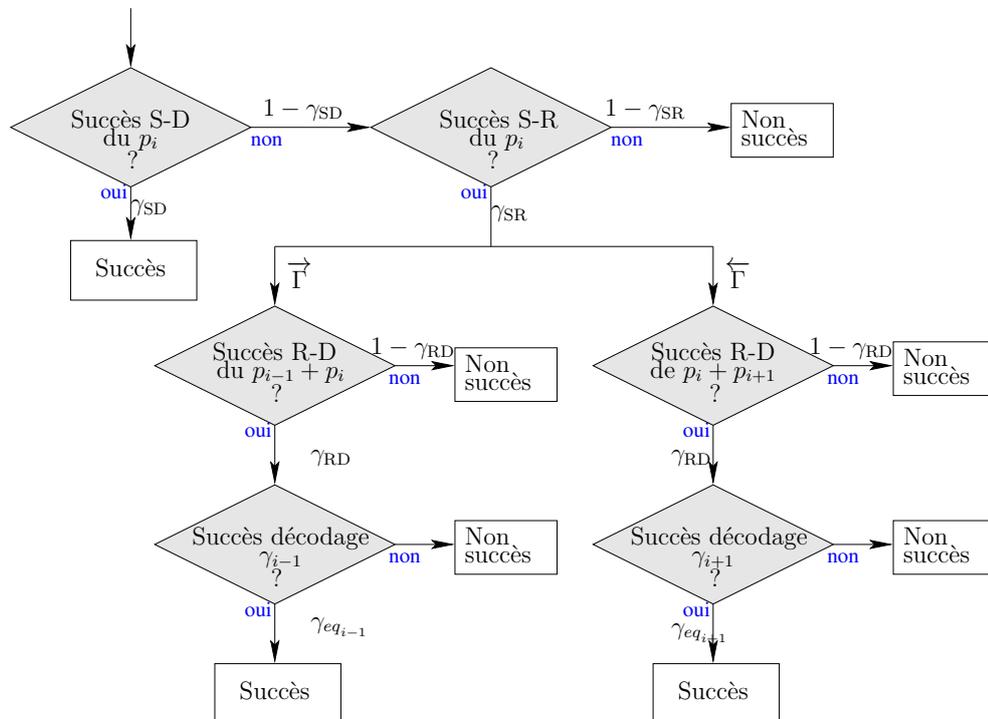


FIGURE 5.2 – Les scénarios des cas possibles qui peuvent se produire pendant la transmission d'un paquet p_i

Nous pouvons ainsi déterminer l'expression théorique de la probabilité équivalente de succès γ_i du système à l'instant i . Cette dernière peut être obtenue à partir de la probabilité de succès de transmission d'un paquet non combiné sur le lien direct, et à partir de la probabilité de succès du relai à condition que les combinaisons des paquets reçus au destinataire puissent être décodées à l'aide de l'ensemble des paquets, combinés ou non, reçus en D . Ces paquets peuvent avoir été reçus avant ($\vec{\Gamma}_i$) ou après ($\overleftarrow{\Gamma}_i$) un paquet courant au décodeur. La probabilité de décodage globale peut être exprimée par la probabilité d'union du cas de décodage dans le sens croissant et décroissant en enlevant la probabilité conjointe des deux cas :

$$\gamma_i = \gamma_{SD} + (1 - \gamma_{SD}) \cdot (\vec{\Gamma}_i + \overleftarrow{\Gamma}_i - \vec{\Gamma}_i \overleftarrow{\Gamma}_i) \quad (5.2)$$

La probabilité dans le sens croissant ($\vec{\Gamma}_i$) est égale à la probabilité où la combinaison des paquets est transmise avec succès au destinataire sur le lien R-D à condition que le paquet précédent p_{i-1} soit reçu et décodé avec succès :

$$\vec{\Gamma}_i = \gamma_{RD} \cdot (\gamma_{SD} + (1 - \gamma_{SD}) \cdot \vec{\Gamma}_{i-1}) \quad (5.3)$$

De la même manière, le terme de la probabilité du succès de décodage dans le sens décroissant ($\overleftarrow{\Gamma}_i$) peut être calculé à partir de la probabilité du succès du paquet p_{i+1} .

$$\overleftarrow{\Gamma}_i = \gamma_{RD} \cdot (\gamma_{SD} + (1 - \gamma_{SD}) \cdot \overleftarrow{\Gamma}_{i+1}). \quad (5.4)$$

La valeur de la probabilité de succès après l'étape E (après l'envoi de E paquets depuis la source) est obtenue par la moyenne de la probabilité de transmission de toutes les étapes. Elle est égale à :

$$\begin{aligned} \gamma_E &= \frac{1}{E} \cdot \sum_{i=1}^E \gamma_i \\ &= \frac{1}{E} \sum_{i=1}^E (\gamma_{SD} + (1 - \gamma_{SD}) \cdot (\vec{\Gamma}_i + \overleftarrow{\Gamma}_i - \vec{\Gamma}_i \overleftarrow{\Gamma}_i)). \end{aligned} \quad (5.5)$$

Pour initialiser les valeurs de $\vec{\Gamma}_i$ et $\overleftarrow{\Gamma}_i$, on doit noter qu'il n'existe pas de paquet dans le relai au début de la transmission et qu'à la dernière étape E de transmission, on ne peut plus recevoir de paquet pour décoder dans le sens décroissant. Ainsi, nous obtenons $\vec{\Gamma}_0 = 0$ et $\overleftarrow{\Gamma}_{E+1} = 0$.

b) Cas général (γ_{SR}) quelconque

Dans un cas plus général, il est possible qu'il existe des effacements sur le lien S-R. Dans ce cas, les paquets disponibles dans la file d'attente du relai sont liés à la probabilité γ_{SR} . Nous considérons ici les deux scénarios du relayage décrits précédemment : celui où le relai envoie deux paquets consécutivement *émis* par la source et celui où le relai envoie deux paquets consécutivement *reçus* dans sa file d'attente.

- Dans le premier cas, le relai envoie systématiquement dans chacun de ses intervalles de temps les deux derniers paquets émis consécutivement par la source (s'ils existent dans

sa file). Les paquets sont choisis par l'ordre d'envoi de la source et déterminés à l'aide des identifiants des paquets.

- Dans le deuxième cas, le relai n'envoie que s'il reçoit un paquet depuis la source. Dans ce cas, la combinaison linéaire du paquet ré-combiné envoyé par le relai est réalisée en additionnant (XOR) les deux derniers paquets reçus dans la file d'attente.

Nous proposons de trouver l'expression théorique de ces probabilités.

Relayage de paquets XORés entre deux paquets consécutivement transmis par la source Nous considérons d'abord le cas où le relai cherche à transmettre à chaque intervalle de temps les combinaisons de paquets consécutivement envoyés par la source. Ceci est équivalent à transmettre, à chaque étape i , le paquet composé de $p_{i-1} \oplus p_{i-2}$. L'algorithme de relayage par le relai suit la démarche présentée dans Algorithme 6. Notons que la condition d'arrêt est que l'acquittement soit reçu par la source pour mettre fin à la transmission ou bien que le nombre de tentatives d'émission soit excédé. Un exemple de séquence de transmission est décrit dans la Table 5.3.

Algorithme 6 : Relayage de la combinaison XOR de deux paquets consécutivement *transmis* par la source

```

j ← 0
tant que la condition d'arrêt n'est pas vérifiée faire
  pj ← dernier paquet reçu dans la file dont l'identifiant est idpj
  pj-1 ← avant dernier paquet reçu dans la file dont l'identifiant est idpj-1
  si (idpj = idpj-1 - 1) alors
    pxor ← pj ⊕ pj-1
  sinon
    pxor ← pj
  fin si
  transmettre pxor
  j = j + 1
fin tant que

```

La probabilité de succès équivalente du système pour ce scénario de relayage peut être calculée à l'aide de l'Equation (5.2) en faisant apparaître le terme γ_{SR} :

$$\gamma_{src_i} = \gamma_{SD} + (1 - \gamma_{SD})\gamma_{SR}(\vec{\Gamma}_{src_i} + \overleftarrow{\Gamma}_{src_i} - \vec{\Gamma}_{src_i} \overleftarrow{\Gamma}_{src_i}). \quad (5.6)$$

La probabilité du succès de transmission dans le sens croissant ($\vec{\Gamma}_{src_i}$) dans ce cas est égale à la probabilité du succès de transmission de la combinaison $p_i \oplus p_{i-1}$ sur le lien R-D, multipliée par la probabilité où le paquet p_{i-1} est reçu avec succès. Cette dernière peut être calculée à partir de la probabilité de succès de la transmission sur le lien direct et sur le lien de relayage (si et seulement si ce paquet réussit à franchir le lien S-R).

$$\vec{\Gamma}_{src_i} = \gamma_{RD}(\gamma_{SD} + (1 - \gamma_{SD})(\gamma_{SR} \vec{\Gamma}_{src_{i-1}} + (1 - \gamma_{SR}))) \quad (5.7)$$

Le terme de la probabilité de succès de décodage dans le sens décroissant $\overleftarrow{\Gamma}_{src_i}$ peut être

Etape	Lien		
	S – D	S – R	R – D
1	p_1	p_1	–
2	p_2	$\cancel{p_2}$	p_1
3	p_3	p_3	$\cancel{p_2} \oplus p_1$
4	p_4	$\cancel{p_4}$	$p_3 \oplus \cancel{p_2}$
\vdots	\vdots	\vdots	\vdots
i	$\boxed{p_i}$	p_i	$p_{i-2} \oplus p_{i-1}$
$i + 1$	p_{i+1}	$\cancel{p_{i+1}}$	$p_{i-1} \oplus \boxed{p_i}$
$i + 2$	p_{i+2}	p_{i+2}	$\boxed{p_i} \oplus \cancel{p_{i+1}}$
$i + 3$	p_{i+3}	p_{i+3}	$\cancel{p_{i+1}} \oplus p_{i+2}$

TABLE 5.3 – Les séquences de transmission du relayage de paquets XORés entre deux paquets consécutivement envoyés par la source pour $\gamma_{S-R} \neq 1$

obtenu par le même raisonnement :

$$\overleftarrow{\Gamma}_{src_i} = \gamma_{RD}(\gamma_{SD} + (1 - \gamma_{SD})(\gamma_{SR}\overleftarrow{\Gamma}_{src_{i+1}} + (1 - \gamma_{SR}))). \quad (5.8)$$

Après E étapes, la probabilité équivalente du système peut être calculée par la moyenne obtenue à chaque étape. Elle est égale à :

$$\gamma_{src_E} = \frac{1}{E} \cdot \sum_{i=1}^E (\gamma_{SD} + (1 - \gamma_{SD})\gamma_{SR}(\overrightarrow{\Gamma}_{src_i} + \overleftarrow{\Gamma}_{src_i} - \overrightarrow{\Gamma}_{src_i}\overleftarrow{\Gamma}_{src_i})). \quad (5.9)$$

La technique de relayage par deux paquets consécutivement envoyés par la source semble pertinente en terme d'implémentation pour un cas simple. Or, en pratique, il est assez difficile d'identifier ces paquets à l'intérieur du réseau surtout lorsque la taille du réseau augmente. De plus, le codage réseau conduit à la création de nouvelles combinaisons de paquets avec divers identifiants. Donc, après plusieurs sauts, cette technique n'est plus applicable. Nous proposons d'analyser donc la deuxième technique de relayage, plus pratique, où on combine deux paquets consécutivement reçus par le relai.

Relayage des paquets XORés entre deux paquets consécutivement reçus au relai

Lorsque le relai reçoit les données depuis la source, il cherche à transmettre la combinaison des deux paquets les plus récents de sa mémoire. Ce scénario de relayage est détaillé dans l'Algorithme 7. Pour $\gamma_{SR} < 1$, le flot de paquets reçus est stoppé momentanément par les effacements du canal. La condition d'arrêt est atteinte lorsque la source n'a plus de paquets à transmettre. Un exemple de séquence de transmission est également présenté dans la Table 5.4.

La probabilité de succès équivalente du système pour ce scénario se déduit de deux cas possibles. Le premier cas est celui où le paquet ne réussit pas à franchir le lien S – R ($\gamma_{SR} = 0$). Dans ce cas, γ_{eq} ne dépend que de la probabilité du succès du lien direct S-D et le relai ne participe pas à la transmission de cette information. Dans le cas contraire, la transmission au niveau du relai se comporte de la même manière que la transmission où on considère qu'il n'y a

Algorithme 7 : Relayage de la combinaison XOR de deux paquets consécutivement reçus par la source

```

j ← 0
tant que la condition d'arrêt n'est pas vérifiée faire
  si R reçoit le paquet de S alors
     $p_{xor} \leftarrow p_j \oplus p_{j-1}$ 
  fin si
  transmettre  $p_{xor}$ 
  j = j + 1
fin tant que

```

Etape	Lien		
	S – D	S – R	R – D
1	p_1	p_1	–
2	p_2	p_2	–
3	p_3	p_3	p_1
4	p_4	p_4	–
⋮	⋮	⋮	⋮
i	p_i	p_i	$p_{i-2} \oplus p_{i-1}$
$i + 1$	p_{i+1}	p_{i+1}	–
$i + 2$	p_{i+2}	p_{i+2}	$p_{i-1} \oplus p_i$
$i + 3$	p_{i+3}	p_{i+3}	$p_i \oplus p_{i+2}$

TABLE 5.4 – Une séquence de relayage des paquets XORés entre deux paquets consécutivement reçus par le relai pour $\gamma_{S-R} \neq 1$

aucun effacement sur le lien S-R que nous avons présenté dans la section précédente. Ainsi, c'est à l'aide des équations (5.2) (5.3) et (5.4) que nous pouvons dériver l'expression de la probabilité équivalente suivante :

$$\gamma_{rlE} = \gamma_{SR} \cdot \frac{1}{\gamma_{SR} \cdot E} \cdot \sum_{i=1}^{\gamma_{SR} \cdot E} \gamma_i + (1 - \gamma_{SR}) \cdot \gamma_{SD} \quad (5.10)$$

5.3.4 Approximation pour des régimes quasi-stationnaires

Dans cette partie, nous proposons d'approximer les termes récurrents qui apparaissent pour l'expression de la probabilité équivalente dans (5.9) et (5.10). Dans le cas général, les expressions analytiques de ces équations peuvent être résolues par récurrence. Or, nous proposons dans cette partie de simplifier ces expressions en éliminant les termes récurrents à l'aide d'une approximation de Régime Quasi-Stationnaire (RQS). Nous supposons que nous disposons d'un intervalle de temps de transmission infini et que le nombre d'étapes de transmission E tend vers l'infini. Ceci nous permet de considérer principalement l'impact du régime stationnaire et de négliger l'impact du régime transitoire.

En considérant que $E \rightarrow \infty$ et le cas où l'opération XOR est réalisée entre deux paquets

consécutivement *envoyés* par la source, nous développons dans un premier temps l'expression analytique de l'approximation en régime quasi-stationnaire de l'Equation (5.9). Ici, nous pouvons proposer une approximation pour la probabilité de succès de décodage dans le sens croissant de l'Equation (5.7) :

$$\vec{\Gamma}_{src_i} = \underbrace{\gamma_{RD}(1-\gamma_{SD})\gamma_{SR}}_{\alpha} \vec{\Gamma}_{src_{i-1}} + \underbrace{\gamma_{RD}\gamma_{SD} + \gamma_{RD}(1-\gamma_{SD})(1-\gamma_{SR})}_{\beta}, \quad (5.11)$$

L'Equation (5.11) est une équation par récurrence linéaire du premier ordre. Nous posons $\vec{\Gamma}_{src_0}$ la condition initiale de la récursion. Nous pouvons réécrire $\vec{\Gamma}_{src_i} = \alpha \vec{\Gamma}_{src_{i-1}} + \beta$ sous la forme¹

$$\vec{\Gamma}_{src_i} = \left(\vec{\Gamma}_{src_0} + \frac{\beta}{\alpha-1}\right)\alpha^i - \frac{\beta}{\alpha-1}, \quad \alpha \neq 1 \quad (5.12)$$

Pour la valeur d'initialisation $\vec{\Gamma}_{src_0} = 0$, l'Equation (5.11) à l'étape E peut être réécrite comme $\beta/(\alpha-1) \cdot (\alpha^E - 1)$:

$$\vec{\Gamma}_{src_E} = \frac{\gamma_{RD}\gamma_{SD} + \gamma_{RD}(1-\gamma_{SD})(1-\gamma_{SR})}{\gamma_{RD}(1-\gamma_{SD})\gamma_{SR} - 1} \cdot ((\gamma_{RD}(1-\gamma_{SD})\gamma_{SR})^E - 1) \quad (5.13)$$

Lorsque $E \rightarrow \infty$, nous disposons de la symétrie du système et des expressions de $\vec{\Gamma}_{src}$ et $\overleftarrow{\Gamma}_{src}$. Nous avons la même probabilité de succès de codage dans le sens croissant et décroissant. Ayant $\Gamma_{src_{E \rightarrow \infty}} = \vec{\Gamma}_{src_{E \rightarrow \infty}} = \overleftarrow{\Gamma}_{src_{E \rightarrow \infty}}$, nous obtenons la probabilité :

$$\Gamma_{src_{E \rightarrow \infty}} = \frac{\gamma_{RD}\gamma_{SD} + \gamma_{RD}(1-\gamma_{SD})(1-\gamma_{SR})}{1 - \gamma_{RD}(1-\gamma_{SD})\gamma_{SR}}. \quad (5.14)$$

En remplaçant l'expression de (5.14) dans l'équation (5.11), nous pouvons en déduire la probabilité du succès d'un paquet qui devient $\forall E \in \mathbb{N}$:

$$\boxed{\gamma_{src_{E \rightarrow \infty}} = \gamma_{SD} + (1-\gamma_{SD})\gamma_{SR}(2\Gamma_{src_{E \rightarrow \infty}} - \Gamma_{src_{E \rightarrow \infty}}^2)}. \quad (5.15)$$

De même manière, pour le cas où le relai transmet la combinaison XOR entre deux paquets consécutivement *reçus* par le relai, nous pouvons en déduire une même approximation de γ_{rI_E} par le régime stationnaire. Nous nous intéressons à l'équation (5.10), sachant que selon (5.3)

1. Considerons l'équation de récurrence linéaire du premier ordre $\gamma_i = \alpha\gamma_{i-1} + \beta$. Nous fixons la valeur initiale γ_0 à une constante près. Pour $i \geq 1$, nous pouvons écrire :

$$\begin{aligned} \Gamma_i &= \alpha\Gamma_{i-1} + \beta \\ &= \alpha^i\Gamma_0 + \beta(\alpha^{i-1} + \alpha^{i-2} + \dots + \alpha^2 + \alpha + 1) \\ &= \alpha^i\Gamma_0 + \beta\left(\frac{\alpha^i - 1}{\alpha - 1}\right) = \alpha^i\left(\Gamma_0 + \frac{\beta}{\alpha - 1}\right) - \frac{\beta}{\alpha - 1} \end{aligned}$$

Cet équation possède une solution unique $\forall i \in \mathbb{N}$. La solution à l'itération i est égale à :

$$\Gamma_i = \begin{cases} \alpha^i\left(\Gamma_0 + \frac{\beta}{\alpha-1}\right) - \frac{\beta}{\alpha-1} & \text{si } \alpha \neq 1 \\ \Gamma_0 + i\beta & \text{si } \alpha = 1 \end{cases}$$

nous avons :

$$\vec{\Gamma}_i = \underbrace{\gamma_{\text{RD}}(1 - \gamma_{\text{SD}})}_{\alpha} \vec{\Gamma}_{i-1} + \underbrace{\gamma_{\text{RD}}\gamma_{\text{SD}}}_{\beta} \quad (5.16)$$

et que

$$\Gamma_i = \alpha \vec{\Gamma}_{i-1} + \beta \Leftrightarrow \vec{\Gamma}_i = \left(\vec{\Gamma}_0 + \frac{\beta}{\alpha - 1} \right) \alpha^i - \frac{\beta}{\alpha - 1}, \quad (5.17)$$

pour la valeur d'initialisation de $\vec{\Gamma}_0 = 0$, la valeur de l'Équation (5.17) pour le cas stationnaire lorsqu'on suppose que l'étape $E \rightarrow \infty$ avec l'égalité $\Gamma_{E \rightarrow \infty} = \vec{\Gamma}_{E \rightarrow \infty} = \overleftarrow{\Gamma}_{E \rightarrow \infty}$, est égal à :

$$\vec{\Gamma}_{E \rightarrow \infty} = \frac{\gamma_{\text{RD}}\gamma_{\text{SD}}}{1 - \gamma_{\text{RD}}(1 - \gamma_{\text{SD}})}. \quad (5.18)$$

En remplaçant l'Équation (5.18) dans (5.10), nous obtenons $\forall E \in \mathbb{N}$:

$$\boxed{\gamma_{r_{E \rightarrow \infty}} = \gamma_{\text{SR}} \cdot (\gamma_{\text{SD}} + (1 - \gamma_{\text{SD}})(2\Gamma_{E \rightarrow \infty} - \Gamma_{E \rightarrow \infty}^2)) + (1 - \gamma_{\text{SR}}) \cdot \gamma_{\text{SD}}.} \quad (5.19)$$

5.3.5 Comparaison du modèle théorique aux simulations

Nous analysons dans cette partie les résultats obtenus par simulation pour vérifier les expressions théoriques précédemment développées. Dans ce scénario de transmission, nous considérons le cas où la source transmet des séquences d'information non encodées. Le relai, quant à lui, applique la combinaison XOR consécutive entre deux paquets consécutivement envoyés par la source et consécutivement reçus par le relai (cf. Algorithme 6 et 7).

La Figure 5.3 présente la probabilité équivalente du système en fonction des probabilités de transmission des liens élémentaires γ_{RD} et γ_{SD} , sachant qu'on fixe $\gamma_{\text{SR}} = 0.5$. Tout d'abord, les résultats de simulation obtenus sont cohérents avec l'expression théorique utilisant la récursion des équations (5.9) et (5.10).

Nous pouvons vérifier que la probabilité de succès équivalente du système évolue proportionnellement à la probabilité de succès des liens γ_{RD} ou γ_{SD} . Pour le relayage entre deux paquets consécutivement envoyés par la source (cf. Figure 5.3(a)), nous obtenons une courbe convexe qui présente une performance de transmission meilleure que pour le cas où deux paquets consécutivement reçus sont combinés (cf. Figure 5.3(b)). Ceci peut être principalement expliqué par les deux raisons suivantes :

1. La transmission de paquets consécutivement envoyés par la source déclenche une tentative d'envoi à chaque période de temps. Un paquet est émis si au moins un des deux paquets $\{p_{i-1}, p_{i-2}\}$ est disponible dans la file du relai. Au contraire, dans le cas où on transmet deux paquets consécutivement reçus au relai, le relayage n'est réalisé que lorsque le relai reçoit un paquet depuis la source. Ainsi, l'envoi de paquets est moins fréquent, ce qui réduit la probabilité de succès.
2. Dans le cas où $\gamma_{\text{SR}} \neq 1$, le relayage entre des paquets consécutifs par la source consiste à transmettre occasionnellement des paquets qui ne contiennent qu'un seul fragment unique. Au contraire, la deuxième technique de relayage consiste à envoyer à chaque émission deux paquets consécutifs combinés par le relai. Cette différence joue un rôle majeur sur la performance du décodage dans le cas où le canal très dégradé ($\gamma_{\text{SD}} \approx 0$).

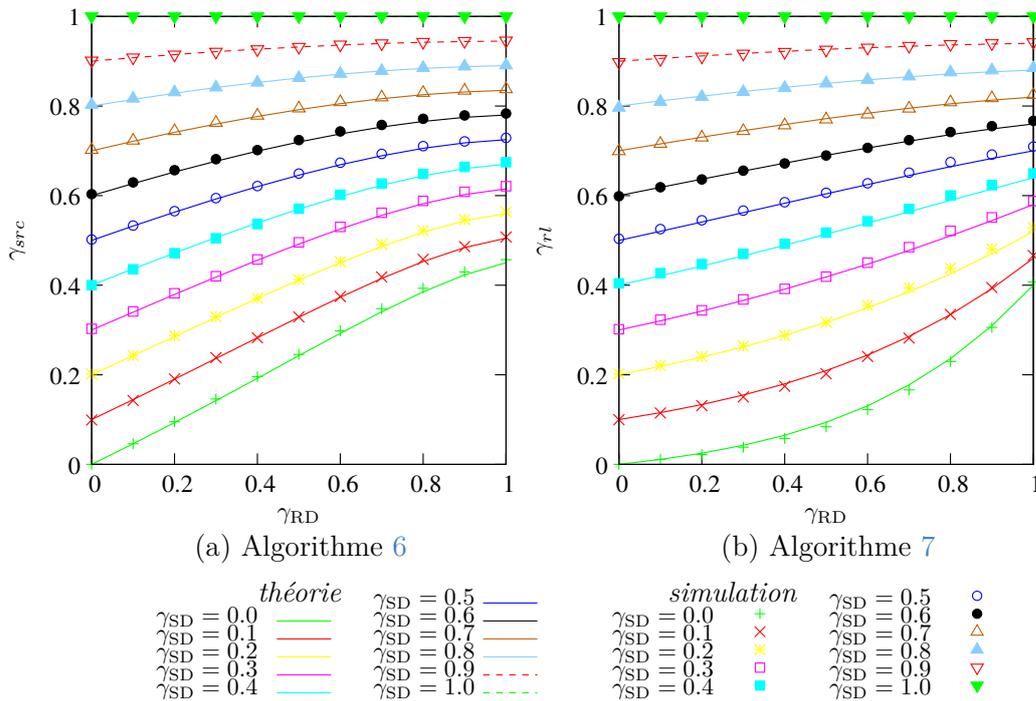


FIGURE 5.3 – Résultats théoriques et par simulation pour les probabilités de succès avec relayage de XOR entre deux paquets consécutivement envoyés par la source (a) et consécutivement reçus par le relai (b) pour $\gamma_{SR} = 0.5, E = 10$

La deuxième scénario est moins performant car le décodeur manque de fragments décodés (i.e. de paquets de degré un) pour décoder l'ensemble du message car il ne peut les obtenir par la transmission directe entre S et D .

La Figure 5.4 compare notre expression théorique développée pour un régime quasi stationnaire (i.e. où E tends vers l'infini) avec les résultats de simulation. Nous pouvons observer que lorsque le nombre d'étapes E augmente, nos résultats de simulation tendent vers l'expression de l'approximation théorique des équations (5.15) et (5.19).

D'après notre étude, nous pouvons conclure que le choix des stratégies de relayage a une incidence sur l'efficacité des transmissions par la source. La première stratégie est plus efficace en terme de la probabilité du succès de transmission. Cependant, pour un réseau plus grand, il n'est pas facile de mettre en place un transmission synchrone (non-slottée). Dans un tel cas, les pertes par collisions sont importantes et la combinaison de deux paquets émis par la source n'est pas toujours possible. De plus, pour un réseau à grande échelle, on ne pourra plus identifier les paquets transmis par la source par ses identifiants puisque le codage réseau aura introduit de nouvelles combinaisons de paquets. Il semble donc plus pratique d'utiliser une stratégie de combinaison qui combine les derniers paquets reçus par un nœud.

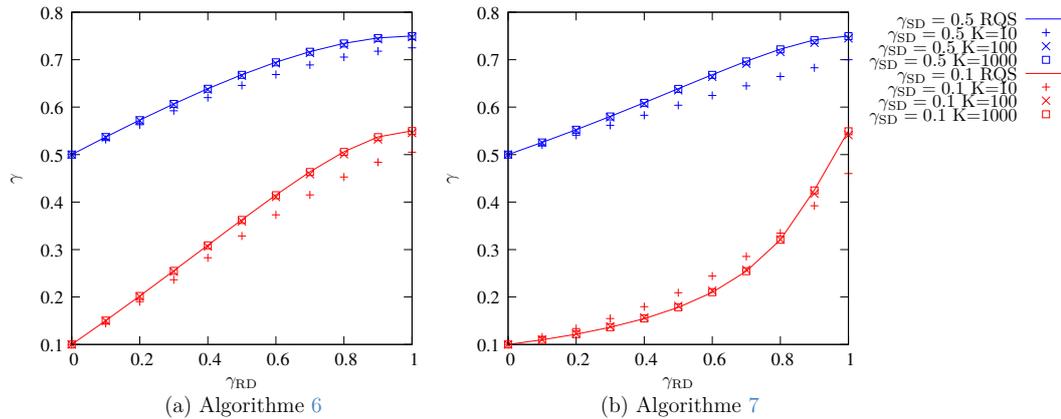


FIGURE 5.4 – Probabilité de succès en fonction de γ_{RD} pour les algorithmes 6 et 7 (avec $\gamma_{SD} = 0.5$ et $\gamma_{SD} = 0.1$) : comparaison du modèle quasi-stationnaire et des simulations pour $E \in \{10, 100, 1000\}$.

5.3.6 Optimisation du débit de relayage

Bien qu'un relayage pertinent par le relai permette d'améliorer la robustesse, la fiabilité et un bon passage à l'échelle, la redondance introduite peut simultanément conduire à des retransmissions inutiles ainsi qu'à des collisions dans le réseau. Un bon compromis est donc nécessaire pour limiter le nombre de paquets redondants en fonction des besoins du système, autrement dit, selon l'état du canal et la qualité des transmissions requise.

Nous cherchons dans cette partie un compromis qui permette à la fois de profiter de l'avantage de la diversité d'information et en même temps de réduire le surcoût de transmission introduit dans le système. En ajoutant un paramètre de contrôle que nous appelons *taux de relayage*, nous cherchons à contrôler voire limiter la redondance dans le réseau en préservant la qualité de transmission.

Definition 5.3.6.1. LE TAUX DE RELAYAGE τ est défini par la probabilité d'un nœud de retransmettre un paquet reçu de n'importe quel voisin. Il prend sa valeur dans l'intervalle $[0, 1]$.

Considérons d'abord le cas où le relai envoie des paquets issus de la combinaison XOR de deux paquets consécutivement transmis par la source. En introduisant la valeur du taux de relayage τ dans l'équation (5.14), nous obtenons :

$$\Gamma_{srcE \rightarrow \infty} = \frac{\tau \cdot \gamma_{RD} \gamma_{SD} + \tau \cdot \gamma_{RD} (1 - \gamma_{SD}) (1 - \gamma_{SR})}{1 - \tau \cdot \gamma_{RD} (1 - \gamma_{SD}) \gamma_{SR}} \quad (5.20)$$

En intégrant (5.20) dans (5.15), nous obtenons la probabilité équivalente du système égale à :

$$\gamma_{srcE \rightarrow \infty} = \gamma_{SD} + (1 - \gamma_{SD})\gamma_{SR} \left(2 \cdot \left(\frac{\tau \cdot \gamma_{RD}\gamma_{SD} + \tau \cdot \gamma_{RD}(1 - \gamma_{SD})(1 - \gamma_{SR})}{1 - \tau \cdot \gamma_{RD}(1 - \gamma_{SD})\gamma_{SR}} \right) - \left(\frac{\tau \cdot \gamma_{RD}\gamma_{SD} + \tau \cdot \gamma_{RD}(1 - \gamma_{SD})(1 - \gamma_{SR})}{1 - \tau \cdot \gamma_{RD}(1 - \gamma_{SD})\gamma_{SR}} \right)^2 \right) \quad (5.21)$$

De la même façon, pour le cas de la combinaison XOR de deux paquets reçus par le relai, nous pouvons modifier l'équation (5.18) pour prendre en compte le débit de relayage τ tel que :

$$\vec{\Gamma}_{E \rightarrow \infty} = \frac{\tau \cdot \gamma_{RD}\gamma_{SD}}{1 - \tau \cdot \gamma_{RD}(1 - \gamma_{SD})}. \quad (5.22)$$

La probabilité équivalente du système à l'aide de l'équation (5.19) devient alors :

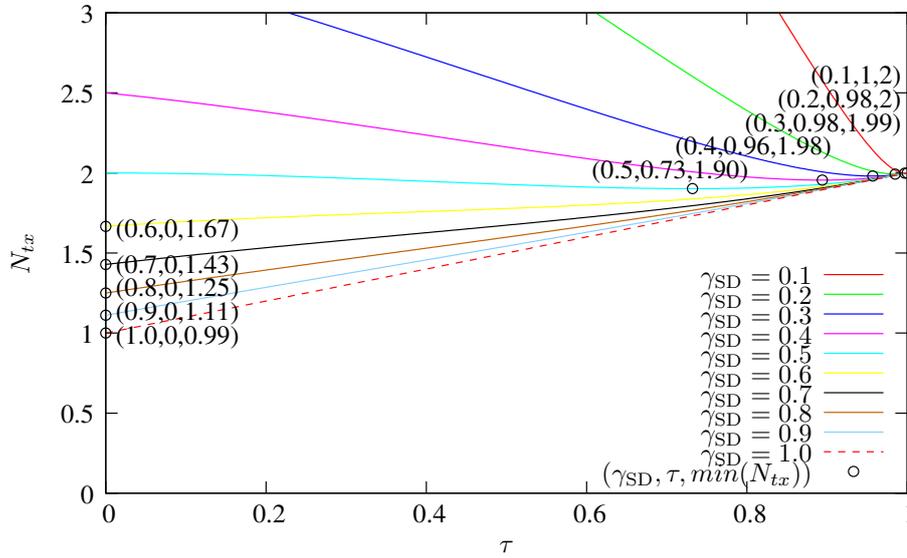


FIGURE 5.5 – Nombre total de transmissions du système en fonction du taux de transmission pour le cas où on applique les opérations XOR entre deux paquets consécutivement envoyés par la source

$$\gamma_{rlE \rightarrow \infty} = \gamma_{SR} \cdot (\gamma_{SD} + (1 - \gamma_{SD}) \left(2 \left(\frac{\tau \cdot \gamma_{RD}\gamma_{SD}}{1 - \tau \cdot \gamma_{RD}(1 - \gamma_{SD})} \right) - \left(\frac{\tau \cdot \gamma_{RD}\gamma_{SD}}{1 - \tau \cdot \gamma_{RD}(1 - \gamma_{SD})} \right)^2 \right)) + (1 - \gamma_{SR}) \cdot \gamma_{SD}. \quad (5.23)$$

A partir de ces expressions, nous cherchons à déterminer *le taux de transmission optimal* τ_{opt} qui va minimiser la consommation énergétique globale du système. Cette consommation est mesurée par le nombre de transmissions moyennes N_{tx} nécessaires à la source pour permettre la récupération complète de tous les fragments au destinataire. A l'aide des expressions de la

probabilité équivalente γ_{eq} du système développée pour plusieurs algorithmes de relayage, le nombre de transmissions moyen par paquet s'écrit :

$$N_{tx} = \frac{1}{\gamma_{eq} \cdot (1 + \gamma_{SR}\tau)} \quad (5.24)$$

En fonction de la probabilité du lien S-D (γ_{SD}), pour les valeurs de γ_{SR} et γ_{RD} fixées, nous traçons dans la figure 5.5 l'évolution du nombre de transmissions requises pour décoder avec succès K paquets envoyés en fonction de τ . Nous en déduisons la consommation minimale $\min(N_{tx})$ ainsi que le taux de transmission optimal τ_{opt} correspondant.

5.3.7 Etude par simulation

Nous présentons dans les Figures 5.6 et 5.7 les résultats de simulation relatifs au *nombre total de transmissions* nécessaire au décodage de K paquets ($K = 100$) en fonction de la qualité γ_{SD} du lien S-D. Les différentes courbes permettent de mesurer l'impact du codage réseau sur le nombre de transmissions dans le canal à relai. Nous rappelons que dans cette étude il n'y a pas de codage fontaine à la source. Les courbes présentant le label «Sans XOR» ont été obtenues sans codage réseau, i.e. le relai retransmet les paquets reçus avec le taux de relayage donné τ . Les courbes avec un label «Algo 6» et «Algo 7» ont été obtenues pour le cas d'un codage réseau utilisant les Algorithmes 6 et 7 respectivement.

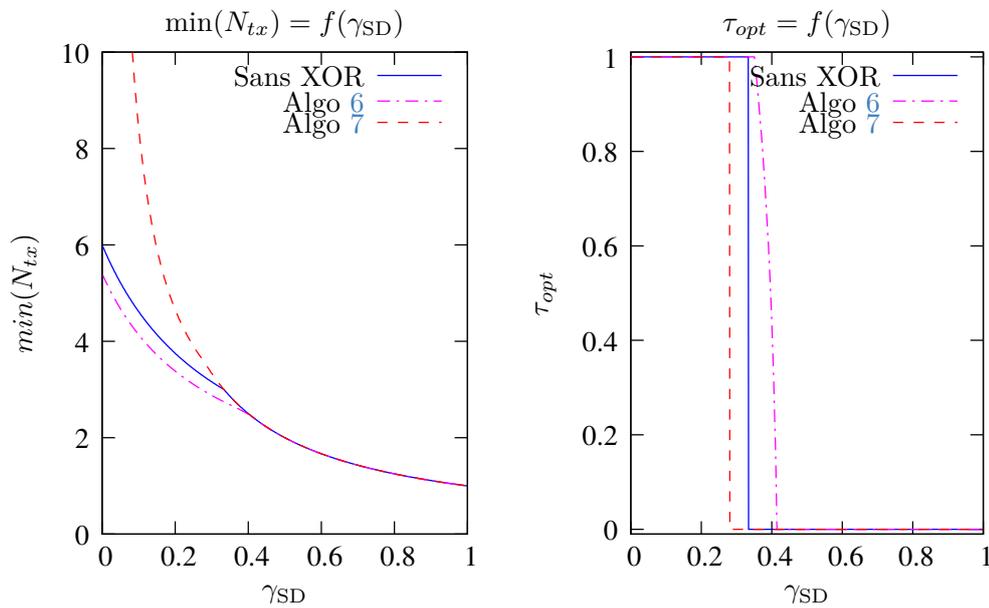


FIGURE 5.6 – Nombre de transmissions minimal et taux de transmission optimal en fonction de γ_{SD} ($\gamma_{SR} = \gamma_{RD} = 0.5$). Les trois courbes représentent le cas sans codage réseau (Sans XOR), le cas où on XOR les deux paquets émis par la source (Algo 6) et le cas où on XOR les deux paquets reçus par le relai (Algo 7).

Selon la Figure 5.7, quand le chemin relai est parfait ($\gamma_{SR} = \gamma_{RD} = 1$), nous avons noté que le codage réseau peut réduire le nombre de transmissions. En effet, quand l'algorithme

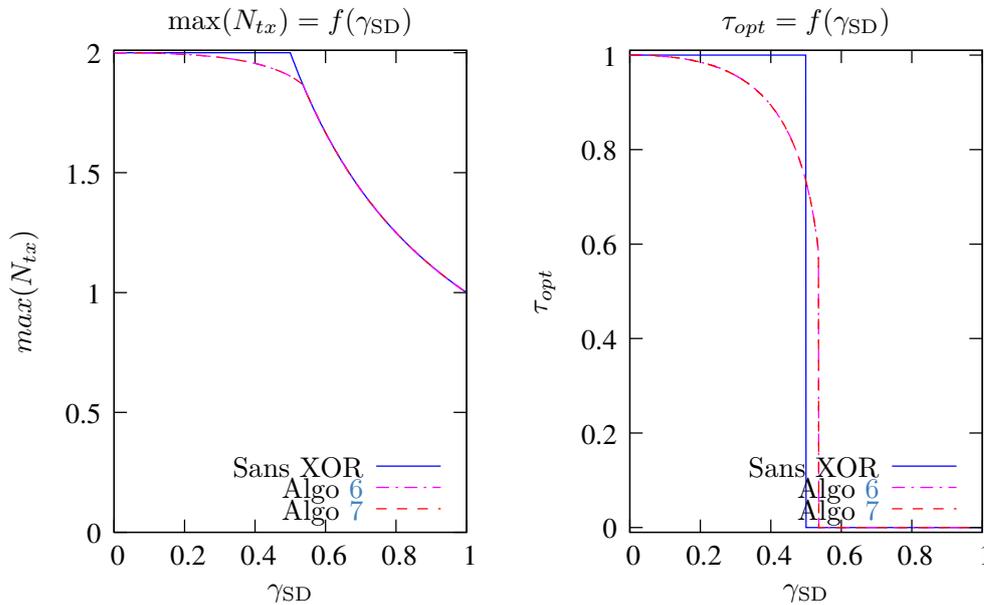


FIGURE 5.7 – Nombre de transmission minimal et taux de transmission optimal en fonction de γ_{SD} ($\gamma_{SR} = \gamma_{RD} = 1$). Les trois courbes représentent le cas sans codage réseau (Sans XOR), le cas où on XOR les deux paquets émis par la source (Algo 6) et le cas où on XOR les deux paquets reçus par le relai (Algo 7).

6 est utilisé (XOR des deux paquets émis par S), on observe une baisse de 10% par rapport au cas sans codage réseau. Pour l'algorithme 7, on observe 25% de transmissions en moins. Si l'effacement est important sur le lien S-R ($\gamma_{SR} = \gamma_{RD} = 0.5$), le fait d'envoyer toujours la combinaison XOR des paquets reçus est défavorable au décodage par propagation de croyance car il manque des fragments de taille 1 pour décoder les paquets combinés. Ceci se traduit par une augmentation du nombre de transmissions nécessaires au décodage. Nous avons noté également que pour n'importe quelle technique de décodage employée, le relayage ne s'avère intéressant que si le lien direct est fortement dégradé : $\gamma_{SD} \lesssim 0.3$.

5.4 Code fontaine distribué pour un canal à relai

L'étude de la partie précédente s'est intéressée à l'apport du codage réseau dans un canal à relai lorsque la source émet l'information non-encodée. Grâce à la diversité d'information introduite par le codage réseau, le taux de relayage peut être réduit et la consommation énergétique globale peut être ainsi optimisée. Nous voulons étudier dans cette partie le choix du taux de relayage optimal τ_{opt} pour le cas où le codage réseau est appliqué sur un flux encodé par un code LT.

5.4.1 Contexte d'étude

Nous nous intéressons dans cette partie à un canal à relai où la source S encode l'information par un code fontaine LT avant de la transmettre. Le destinataire D dispose un décodeur BP. Le relai situé entre la source S et D assiste à la transmission en utilisant un des scénarios de relayage suivants :

- **Scénario CP** (relayage de Code fontaine par Paquet) : le relai retransmet les paquets reçus sans les ré-encoder (pas de codage réseau).
- **Scénario CP-XR** (relayage de Code fontaine par r XORs successifs) : le relai transmet une combinaison XOR entre les r paquets les plus récemment reçus et enregistrés dans sa file d'attente.
- **Scénario CP-XH** (relayage de Code fontaine par XOR, approche Heuristique) : pour respecter la contrainte imposée par le décodeur BP, le relai ré-encode les paquets de façon à conserver la distribution RSD du code LT (cf. Algorithme 5 pp.80).

Chaque relai re-diffuse ses paquets ré-encodés ou non à son voisin avec une probabilité définie par le *taux de relayage* τ introduite précédemment. Nous cherchons dans cette étude à mesurer le nombre de transmissions N_{tx} nécessaire au décodage en fonction de τ .

5.4.2 Simulations et résultats

Les Figures 5.8 et 5.9 représentent le nombre d'émissions totales dans le canal à relai pour transmettre chaque fragment d'information de S à D. On présente le cas où le lien de relayage est parfait ($\gamma_{SR} = \gamma_{RD} = 1$) ou dégradé ($\gamma_{SR} = \gamma_{RD} = 0.5$). Dans chaque figure, nous avons comparé la performance de transmission pour différentes qualités du lien direct S-D ($\gamma_{SD}=0.1, 0.5$ et 1.0).

En fonction de la probabilité du lien S-D, nous avons observé que :

- Lorsque le lien S-D est bon ($\gamma_{SD} \approx 1.0$), le relai devient inutile. Le nombre de transmissions croît rapidement avec τ .
- Lorsque le lien S-D est mauvais ($\gamma_{SD} \approx 0.1$), le relai permet de réduire le nombre de tentatives de transmissions. Le nombre de transmissions décroît rapidement avec τ .

Nous soulignons ainsi les avantages apportés par le relai. La question qu'on se pose ensuite est : quelles sont les techniques efficaces pour relayer les données et : quand est-ce qu'il faut que les relais les mettent en œuvre. Pour les cas étudiés, nous avons observé que les bénéfices du codage réseau sur le flux encodé LT ne sont pas évidents. Sur les Figures 5.8 et 5.9, nous avons observé qu'un relayage par XOR consécutif nécessite un nombre de transmissions plus élevé que le cas sans codage. Ceci provient principalement de la modification de la distribution des degrés

(cf. Chapitre 4). La solution heuristique que nous avons proposée pour corriger le degré des paquets ne permet pas de clairement montrer que le codage réseau est bénéfique pour un code LT. Nous pensons que l'étude réalisée dans cette partie ne permet pas de statuer clairement sur ce point.

Il se peut que le contexte d'étude (le canal à relai) soit trop spécifique et ne nous permette pas de relever les avantages que peuvent apporter le codage réseau pour un code fontaine. Pour cela, nous allons dans la suite élargir l'échelle de notre étude en nous intéressant au cas d'un réseau de capteurs constitué d'un plus grand nombre de nœuds.

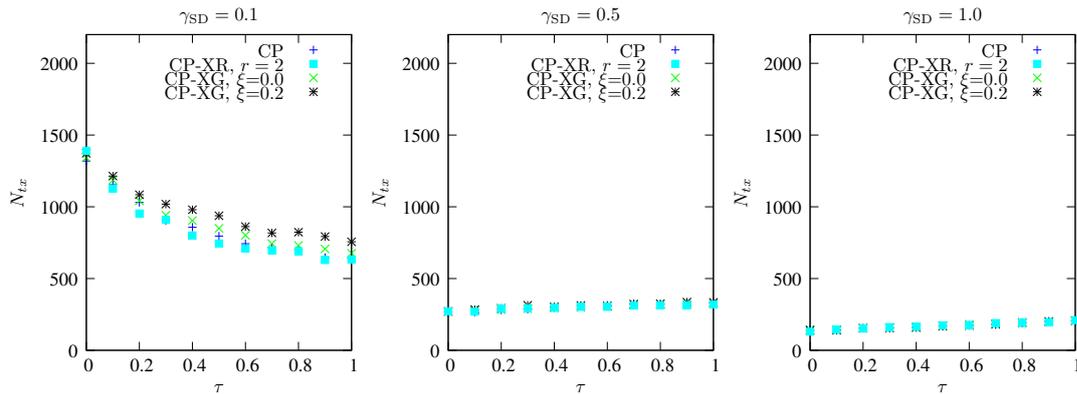


FIGURE 5.8 – Nombre total de transmissions N_{tx} pour différents scénarios de relayage d'un code fontaine. ($\gamma_{SR} = \gamma_{RD} = 0.5$)

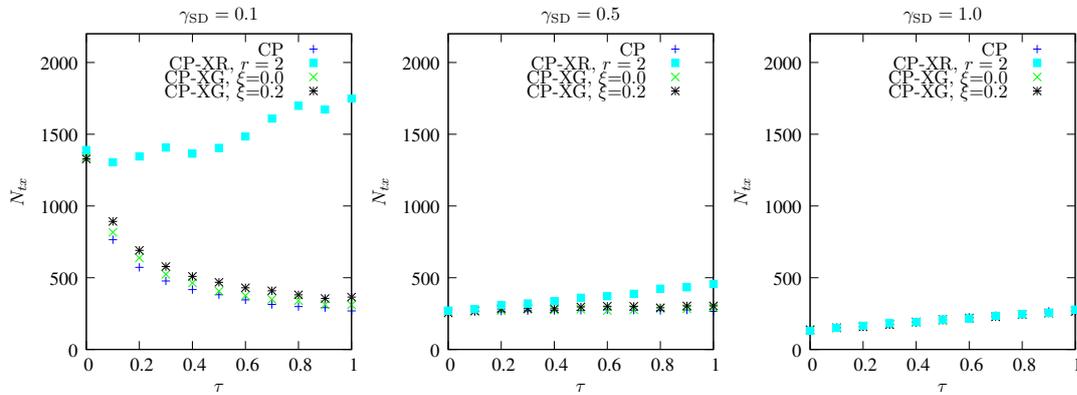


FIGURE 5.9 – Nombre total de transmissions N_{tx} pour différents scénarios de relayage d'un code fontaine. ($\gamma_{SR} = \gamma_{RD} = 1.0$)

5.5 Code fontaine et codage réseau pour la diffusion dans un réseau de capteurs

Après avoir étudié les apports du code fontaine et du codage réseau pour le canal à relai, nous nous intéressons maintenant à étendre cette étude au cas d'un *réseau complet*. Dans ce réseau, on cherche à fiabiliser une transmission vers le puits en réduisant la consommation énergétique et le délai de transmission. Pour cela, nous allons tirer parti de la diversité spatiale d'une diffusion par gradient et de la diversité introduite par la combinaison des paquets obtenus par les codes LT et le codage réseau. Nous utilisons le codage réseau par approche heuristique proposé. Notre but est d'évaluer le codage réseau pour la transmission du code LT pour la diffusion dans un réseau de capteurs.

5.5.1 Diffusion par gradient

La diffusion par gradient est un algorithme de diffusion intéressant pour le routage dans un réseau de capteurs car il ne nécessite qu'une phase d'auto-organisation minimale et permet d'augmenter la robustesse de la transmission dans des environnements difficiles où le taux de pertes est élevé [174, 154, 68]. C'est un routage qui associe un coût Q_i à chaque nœud i du réseau. L'ensemble des coûts forme un champ de gradients décroissants. Le puits se trouve au minimum du champ et possède le coût minimal ($Q_i = 0$). Ce coût peut être obtenu à partir de diverses métriques telles que le nombre de sauts, la distance au puits, etc. Dans notre étude, nous utilisons la distance euclidienne entre le relai i R_i et le destinataire D (i.e. le puits). L'association du coût au nœud est réalisée pendant une phase d'initialisation. Dans un premier temps, le destinataire (ou le puits) diffuse un message ADV avec un coût courant Q_c égal à 0. Le nœud j qui entend ce message initialise son coût Q_j à la somme entre le coût courant Q_c et la distance entre sa position et la position du nœud qui lui a envoyé ce message. Puis, il diffuse ce message aux autres nœuds à sa portée en mettant à jour le coût courant avec son propre coût : $Q_c = Q_j$. Cette étape est répétée jusqu'à ce que tous les nœuds soient associés à leurs coûts.

Dans la phase de transmission des données vers le puits, les paquets d'une source S sont diffusés à l'aide des coûts préalablement établis et suivent la direction des coûts décroissants pour atteindre leur destination. Au début de la phase de communication, chaque nœud diffuse sa data en lui associant un coût dans l'en-tête du paquet Q_p égale à son coût Q_i . Lorsque ses voisins à un saut reçoivent ces paquets, ils les rediffusent si et seulement si leur propre coût est inférieur au coût du paquet : $Q_i < Q_p$. Au moment de la rediffusion par un capteur k , le coût du paquet est remis à jour avec le coût du capteur k : $Q_p = Q_k$. Pour réduire la retransmission multiple d'un même paquet, on impose aux capteurs de ne pas transmettre deux fois un même paquet émis par la source.

Nous prenons l'exemple de la Figure 5.10. Elle présente la valeur des coûts associés aux nœuds du réseau. Dans la phase d'initialisation, le destinataire situé à la position D(100,100) envoie un message ADV pour établir les coûts pour chaque capteur du réseau. La phase de communication commence dès que le champ de coût est établi. Par exemple, un nœud S à la position S(400,400), possédant un coût ≈ 420 cherche à émettre des paquets vers D. Les capteurs dont le coût associé est inférieur à ce coût rediffusent ses données et mettent à jour le

coût courant du paquet. Nous avons représenté dans la Figure 5.10 le nombre de transmissions moyen réalisé par chaque capteur dans la phase de communication. Nous pouvons remarquer que la consommation se concentre au niveau du chemin le plus court entre S et D. Cependant, tous les noeuds à portée de la communication entre S et D participent également à la communication.

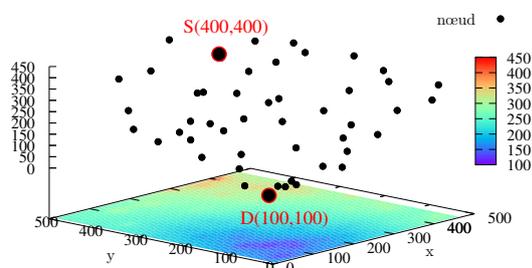


FIGURE 5.10 – Positions et coûts associés aux capteurs par un algorithme à diffusion par gradient

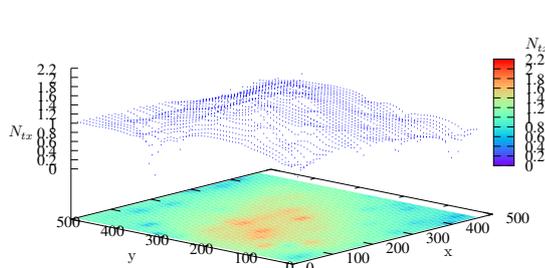


FIGURE 5.11 – Nombre de transmissions par capteur pour une communication entre S et D

5.5.2 Codage fontaine et codage réseau pour la diffusion par gradient

La diffusion par gradient a été proposée pour des réseaux fortement perturbés où la probabilité d'erreur sur les liens est importante et où les capteurs présentent un taux important de pannes dû aux conditions de déploiement (humidité, chaleur, etc...). Elle permet de fiabiliser les transmissions multi-sauts en augmentant la redondance dans le réseau. Ainsi, plus le degré des capteurs (i.e. le nombre de voisins à un saut) est élevé, plus la redondance augmente dans le réseau. Il en résulte qu'un grand nombre de copies d'un même paquet sont reçues à la destination.

Nous proposons dans cette dernière section de tirer parti de cette redondance en diversifiant l'information émise par la source dans les paquets transmis dans le réseau. Pour cela, la source émet un flux encodé par le code LT et les nœuds relaient les paquets à l'aide d'un codage réseau. L'algorithme de codage réseau considéré est l'approche heuristique précédemment proposée (cf. Algorithme 5 pp.80). Notre objectif est de réduire le nombre de transmissions nécessaires à la source pour envoyer ses K paquets de données.

Comme dans la partie précédente, nous introduisons les deux probabilités suivantes pour contrôler le taux de redondance et l'utilisation du codage réseau au moment du relaying :

Definition 5.5.2.1 (taux de relaying). *Le taux de relaying τ d'un nœud est la probabilité de retransmettre un paquet reçu (encodé ou non) à ses voisins.*

Dans notre étude, nous supposons pour l'instant que la valeur de τ est la même pour tous les capteurs.

Definition 5.5.2.2 (probabilité de XOR). *La probabilité de XOR ν est la probabilité qu'un relai ré-encode le paquet reçu avant chaque diffusion.*

L'algorithme de relaying utilisé par un relai est donné dans Algorithme 8.

Algorithme 8 : Macro-algorithme de relaying pour un nœud R

```

 $F_{max} \leftarrow K$ 
si R reçoit un paquet  $p$  alors
  Ajouter  $p$  dans la FIFO de taille  $F_{max}$  ;
  si Test sur  $\tau == \text{vrai}$  alors
    si Test sur  $\nu == \text{vrai}$  alors
      Applique l'algorithme 5
    sinon
       $p_{new} = p$ 
    fin si
  fin si
  transmettre  $p_{new}$ 
fin si

```

La source cherche à transmettre K paquets encodés par le code LT. Chaque nœud possède une file d'attente de type FIFO de taille maximale $F_{max} = K$. Lorsqu'un relai sur le chemin reçoit un paquet, il vérifie si la combinaison des fragments a déjà été reçue ou non. S'il elle a déjà été reçue, il jette le paquet. Sinon, il décide de le retransmettre avec une probabilité égale à τ . Le relai peut décider de retransmettre le paquet tel quel avec la probabilité $1-\nu$ soit de le combiner avec les autres paquets de la file avec la probabilité ν . Dans ce dernier cas, il transmet la combinaison XOR des paquets contenus dans sa file d'attente par l'approche heuristique. Lorsque le destinataire reçoit suffisamment d'information pour décoder le message, le processus de décodage est lancé. Si le décodage est réussi, le destinataire envoie un message d'acquiescement à la source pour mettre fin à la transmission.

Nous avons fait l'hypothèse dans cette partie que le message d'acquiescement, une fois envoyé par le destinataire, ne subit aucune perte. Nous avons donc négligé dans cette étude le surcoût introduit par le mécanisme d'acquiescement. La diffusion du message d'ACK des K paquets à la fin de transmission peut se faire en même temps que la mise à jour du champ de coûts en l'incluant dans le message ADV. Dans le cas où le destinataire ne reçoit pas suffisamment de paquets qui lui permettent de décoder avec succès le message, la source transmet en continu les paquets encodés tant qu'il y a la puissance de disponible.

5.5.3 Simulations

Description des simulations

Nos résultats de simulation sont toujours obtenus avec le simulateur à événements discrets WSNet [1]. Les caractéristiques de capteurs sont définies par la spécification de dispositif radio TI CC1100 [67]. Les protocoles des couches physique et MAC suivent le standard IEEE802.15.4 (cf. Table 3.1 pour plus de détails). Spécifiquement à cette étude, nous avons utilisé des canaux orthogonaux (pas d'interférence entre capteurs) et un modèle de canal AWGN et Rayleigh par bloc. L'accès au médium est géré par le mécanisme CSMA/CA de IEEE802.15.4.

Nous considérons un réseau composé de 50 capteurs distribués uniformément dans un espace de 500x500m. Le nombre moyen de voisins de chaque capteur est d'environ 3 nœuds. Chaque capteur est autonome et indépendant. Les capteurs dans le réseau participent à l'acheminement

des données depuis la source vers le destinataire et utilisent l'Algorithme 8. La diffusion et l'encodage des paquets entre les capteurs permet d'augmenter la fiabilité de la transmission, ce qui devient d'autant plus intéressant que le lien direct source-destinataire est perturbé.

Les résultats de simulation dans cette partie sont obtenus pour la transmission de $M = 50$ messages par une source S vers D (cf. Figure 5.10). Chaque message est divisé en 100 fragments de même taille ($K = 100$) puis encodé avec le code LT. La taille de la file d'attente des capteurs est égale à $K = 100$ paquets également. Les relais appliquent le ré-encodage selon l'algorithme heuristique (cf. CP-XH avec $\xi = 0.0$ détaillé dans la Section 5.4.1 pp.107) entre les paquets reçus.

Métriques de performance

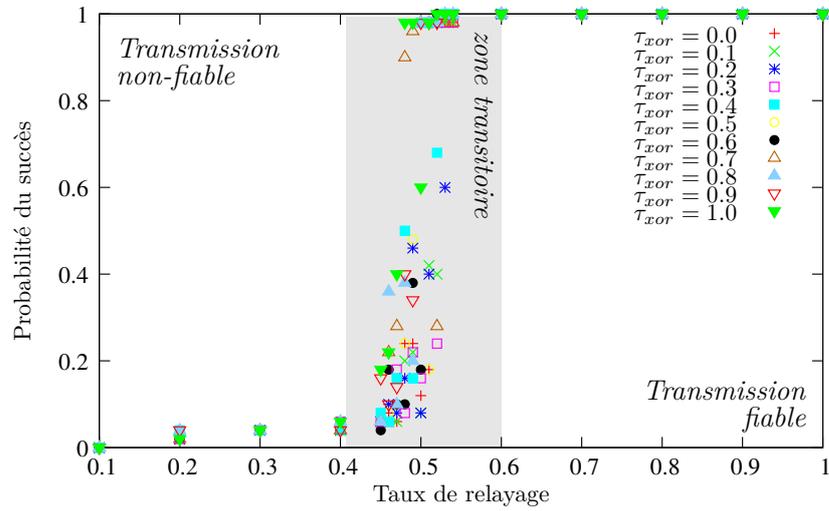
Pour chaque transmission d'un message (composé des K fragments) entre S et D, les métriques de performance suivantes sont calculées :

- **Taux de succès** Le taux de succès de transmission est présenté comme le rapport entre le nombre de fragments décodés avec succès à la destination et le nombre de fragments envoyés K . Comme nous utilisons un code fontaine, le taux de succès est soit de 1 soit de 0. Comme la source transmet M messages, nous présentons le taux de succès moyens pour les M messages.
- **Énergie** La consommation d'énergie est exprimée par la consommation totale d'énergie du réseau pour transmettre tous les M messages au destinataire.
- **Délai** Le délai de transmission est mesurée par la latence moyenne pour transmettre avec succès un message donné au destinataire.
- **Redondance** Nous définissons le taux de redondance μ comme le rapport entre le nombre total de paquets reçus au destinataire et le nombre de paquets utiles au décodage. Cette redondance est également moyennée pour le nombre M de messages envoyés.

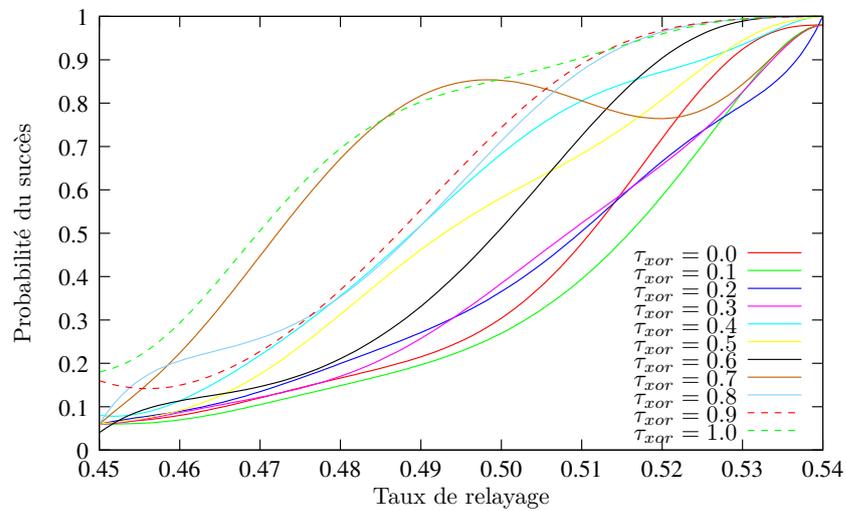
5.5.4 Résultats

Taux de succès

La Figure 5.12 présente le taux de succès en fonction du taux de relayage τ . Nous observons trois zones distinctes sur cette figure. La première zone est la zone non-fiable où le taux de succès est très faible pour $\tau < 0.4$. En effet, dans cette zone, le taux de relayage par le relai n'est pas suffisamment élevé pour assurer la connectivité du réseau. Pour le cas où $\tau > 0.6$, le taux de succès est quasiment égal à 1. Dans ce cas, le taux de relayage est suffisant pour assurer la connectivité du réseau. Nous avons observé une zone de transition autour de la valeur $\tau \approx 0.5$. Dans cette zone, on observe une augmentation rapide du taux de succès. Nous appelons par la suite cette zone *zone transitoire* où la connectivité du réseau est sensible au choix du codage par les relais. Du fait de la non-fiabilité du canal et la variabilité de la connectivité, nous avons observé qu'utiliser un codage réseau systématique ($\nu = 1$) améliore fortement le taux de succès par rapport à $\nu = 0$. Le codage réseau permet bien dans ce contexte d'améliorer la fiabilité du réseau par rapport au cas sans codage.



(a) Probabilité du succès

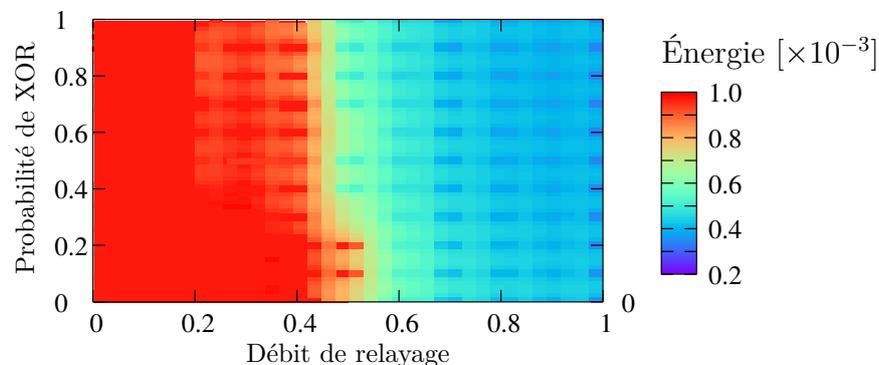


(b) Probabilité du succès dans la zone transitoire obtenue par la méthode de lissage par les courbes de bézier

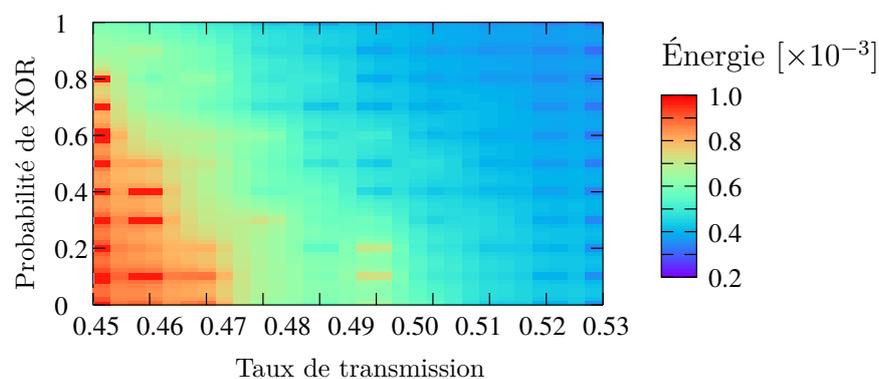
FIGURE 5.12 – Taux de succès de transmission

Énergie

La Figure 5.13(a) illustre la consommation énergétique en fonction du τ et ν .



(a) Consommation d'énergie en fonction du débit de relayage et la probabilité du codage réseau



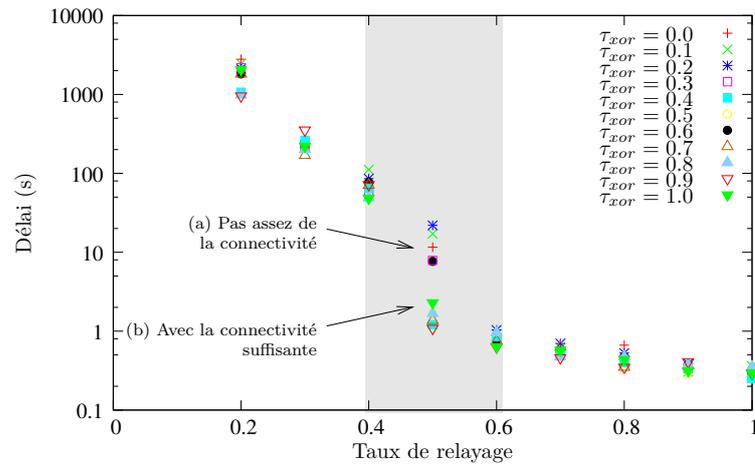
(b) Consommation d'énergie en fonction du débit de relayage et la probabilité du codage réseau pour une zone transitoire

FIGURE 5.13 – La consommation énergétique du réseau

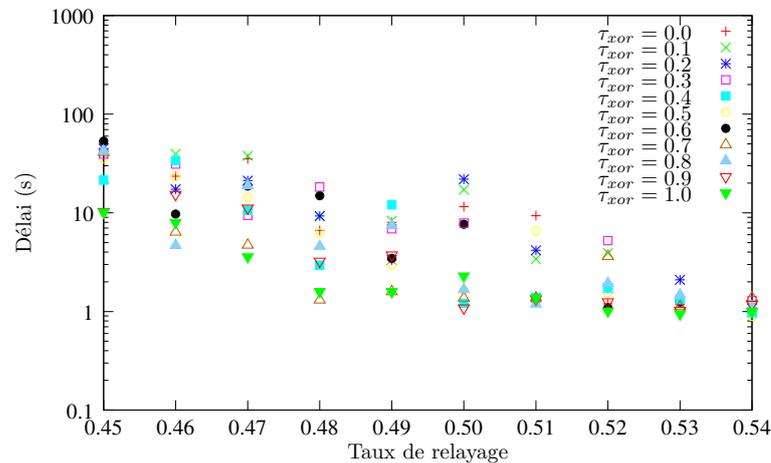
Lorsque le taux de relayage est faible ($\tau \leq 0.4$), nous observons que la consommation d'énergie est très élevée. Ceci est dû à la transmission permanente des paquets encodés LT par la source qui attend une indication d'arrêt (ex. la réception d'acquittement ou le nombre maximal de transmissions atteint). Nous avons observé dans la Figure 5.13(b) la consommation dans le zone de transition où le codage réseau par le relai augmente grandement la fiabilité du réseau. Dans cette zone, la consommation énergétique est plus faible mais néanmoins plus importante que pour le cas où la connectivité est assurée dans le réseau.

Délai

En terme du délai de transmission, nous avons observé sur la Figure 5.14(a) la cohérence de l'évolution du délai par rapport à celle de l'énergie précédemment étudiée. Nous distinguons également les trois zones. Dans le cas où le système est non-fiable ($\tau \leq 0.4$), le délai est très important. Ce délai décroît rapidement lorsque le taux de relayage augmente car le décodage des messages peut se faire de plus en plus tôt. Dans le cas où le canal est très bon ($\tau \geq 0.6$), le délai est assez faible et évolue linéairement avec τ . Dans ces deux cas, nous n'avons pas observé de grand écart entre le délai de transmission pour le cas sans codage et avec codage réseau.



(a) Délai de transmission



(b) Délai de transmission dans la zone transitoire

FIGURE 5.14 – Le délai de transmission

Cependant, c'est dans la zone transitoire où la connectivité est limitée ($\tau \approx 0.5$) que nous avons observé une grande différence entre le cas où le codage réseau est appliqué ou non. Pour le cas où le codage réseau est appliqué avec la probabilité $\nu > 0.5$, nous avons observé que le délai obtenu est assez bon et proche du cas où de la transmission est fiable. Dans le cas contraire, le délai est très élevé. Un zoom sur l'évaluation du délai dans la zone transitoire est présenté dans la Figure 5.14(b).

Nous pouvons donc conclure à l'aide de ces résultats que le codage réseau pour la transmission avec le code LT permet de réduire considérablement le délai de transmission car il permet d'augmenter la fiabilité d'un réseau à faible connectivité.

Redondance

Pour terminer, nous avons mesuré la redondance en fonction du taux de relayage et la probabilité de XOR sur la Figure 5.15. Nous avons observé que le codage réseau permet d'éviter la réception de la même copie du paquet par le destinataire. Ainsi, la redondance diminue avec l'augmentation de la probabilité de XOR car le codage réseau permet de diversifier l'information.

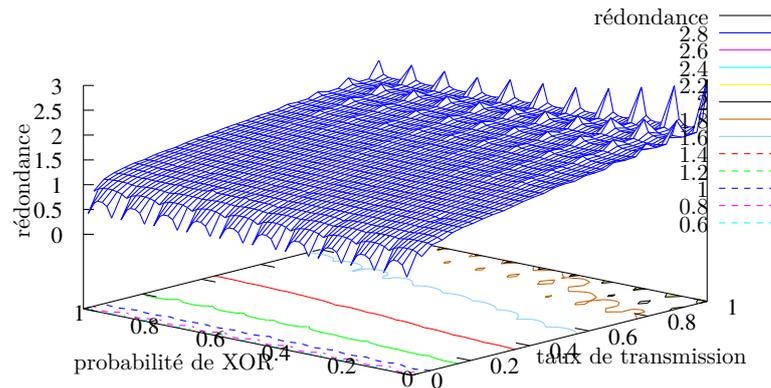


FIGURE 5.15 – Redondance en fonction de la probabilité de XOR et du taux de relayage

5.6 Conclusion

Dans cette dernière partie, nous avons étudié l'impact du taux de relayage et du codage réseau sur la transmission dans un réseau de capteurs. Nous avons pour objet de minimiser le coût de transmission du système à l'aide du codage réseau. Cet avantage a été confirmé, partiellement en théorie pour le canal à relai et par simulation pour un réseau complet. Pour que le codage réseau permette de réduire la consommation globale du système, il faut choisir avec soin l'algorithme de combinaison des paquets. Les résultats obtenus ont confirmé que notre algorithme de relayage par approche heuristique proposé dans le chapitre précédent permet d'obtenir un ré-encodage satisfaisant par le relai et de maintenir la capacité de décodage pour une transmission multi-sauts.

Dans notre étude sur le canal à relai, nous avons montré qu'il est possible de dériver le taux de relayage optimal pour minimiser le nombre de transmissions à la source. Trouver ce point de fonctionnement est particulièrement intéressant mais présente des limites à sa mise en œuvre d'un point de vue pratique. En effet, pour dériver le taux optimal, il faut connaître la qualité des canaux disponibles, ce qui nécessite la mise en place d'algorithmes d'apprentissages du canal coûteux en énergie. De plus, une telle approche peut difficilement s'appliquer à des réseaux à plus grande échelle car la dérivation du taux optimal pour chaque capteur représente un problème d'optimisation centralisé complexe qui nécessiterait pour sa mise en œuvre un grand nombre de messages de contrôle.

Ainsi, pour pouvoir analyser les performances d'un code fontaine et du codage réseau sur un réseau plus grand, nous avons choisi de nous baser sur un algorithme de diffusion par gradient où

tous les capteurs possèdent une même valeur de taux de relaying. Nous avons montré comment la connectivité du réseau évolue avec le taux de relaying et qu'il est possible à l'aide du codage réseau et d'un code LT de réduire le temps nécessaire à la transmission d'un message de bout en bout pour une plus faible valeur de taux de relaying, et ainsi améliorer le compromis entre délai et énergie.

La principale conclusion de ce chapitre est que le meilleur gain en délai et énergie obtenu par le couplage d'un code fontaine et d'un codage réseau se trouve quand le réseau fonctionne en limite de connectivité. Nous avons pu le montrer à la fois dans notre étude du canal à relai où le bénéfice du codage réseau est significatif quand le lien direct dans le canal à relai est fortement dégradé, et à la fois dans notre étude sur le réseau à large échelle. Dans cette deuxième topologie de réseau, le codage fontaine couplé au codage réseau permet clairement de gagner en performance quand la connectivité du réseau est limitée pour un algorithme par diffusion par gradient. Dans ces deux cas, la diversité introduite par le codage réseau permet de réduire le nombre de transmissions et ainsi réduire le délai et l'énergie consommée.

Nous avons montré l'avantage du codage réseau pour certaines configurations de réseau. Lorsque la connectivité du réseau est variable, le codage réseau sur le flux encodé LT permet d'améliorer la robustesse de transmission et ainsi réduire la consommation d'énergie et le délai de transmission dans le réseau. A l'aide de la diversité d'information introduite par le codage réseau, la redondance peut ainsi être limitée.

6

Conclusion et perspectives

Dans cette thèse, nous avons cherché à exploiter les principes des codes fontaine dans le contexte des réseaux de capteurs. Nous nous sommes intéressés à l'association des codes fontaines et du codage en réseau pour améliorer le compromis énergie/fiabilité des transmissions multi-sauts dans les réseaux de capteurs à faible portée, de type WPAN ou WBAN. La contribution principale de cette thèse est la définition de nouvelles propositions dont le but est d'introduire le codage réseau dans les flux d'information multi-sauts encodés par code fontaine. A l'aide du simulateur WSNnet, dans lequel nous avons implémenté un protocole PHY/MAC basé sur le standard IEEE802.15.4, nous avons évalué la performance de ces communications codées selon plusieurs critères, en particulier la consommation énergétique et le délai de transmission, qui nous semblent être les paramètres vitaux à prendre en considération dans la modélisation des réseaux de capteurs. Nous résumons ci-dessous les principales contributions de cette thèse et nous identifions quelques perspectives que nous pensons pertinentes.

6.1 Conclusion

Dans ce travail, nous nous sommes tout d'abord intéressés à assurer la fiabilité et la robustesse des communications entre les nœuds d'un réseau de capteurs. Dans une communication traditionnelle, la fiabilité peut être atteinte à l'aide de différents processus d'accusé de réception. Cependant, ces mécanismes nécessitent un bon canal de retour et introduisent un surcoût important. Dans un système de transmission semi-duplex, les codes fontaine permettent de contourner cette limitation car l'acquiescement peut s'effectuer globalement pour un ensemble de fragments d'information envoyés. Ainsi, dans le troisième chapitre, nous avons observé que les codes fontaines RLF et LT permettent de réduire la consommation d'énergie et le délai de transmission tout en maintenant la fiabilité et la robustesse des communications. Cependant,

comme les codes fontaine engendrent une émission continue de paquets encodés tant que l'accusé de réception final n'est pas reçu, le nombre de transmissions inutiles ou redondantes doit être minimisé par une implémentation adéquate de ce mécanisme d'accusé de réception. Nous avons quantifié ce surcoût imposé au système, et identifié certains scénarios de relayage plus robustes face à ce débordement.

D'autre part, la coopération des nœuds relais permet de faciliter l'acheminement de l'information de la source vers le destinataire, et donc d'améliorer la performance globale d'un système de transmission. Ainsi, dans le quatrième chapitre, nous avons démontré comment l'introduction de codage réseau sur les flux du code fontaine permettait d'améliorer le compromis énergie/fiabilité.

Nous avons cependant soulevé le problème de la dégradation importante de la capacité de décodage par propagation de croyance, lorsqu'on applique un codage réseau sur le flux d'information encodé LT. En effet, le ré-encodage par les nœuds intermédiaires entraîne une modification de la distribution de degré du flux d'information arrivant au destinataire. Afin de conserver la distribution de degré selon la loi RSD qui est la distribution optimale visée, nous avons proposé plusieurs techniques de codage réseau qui permettent le développement conjoint de la conservation de la distribution de degré tout en gardant une bonne *diversité d'information*. Ainsi, la redondance d'information peut être efficacement exploitée. L'algorithme de codage réseau que nous proposons permet d'obtenir une bonne performance de transmission pour un schéma de transmission linéaire multi-sauts.

Enfin, dans le dernier chapitre, nous avons montré que grâce au codage réseau, nous pouvons profiter de la diversité d'information apportée au système pour réduire la consommation totale d'énergie. Les résultats obtenus dans cette partie de thèse ont confirmé que la performance globale du système en terme d'énergie et de délai de transmission peut être optimisée si la technique de relayage est bien choisie. Nous avons modélisé et simulé ces techniques pour mettre en évidence le gain de performance lié à l'association code fontaine et codage en réseau. Une des principales conclusions de ce chapitre est qu'il est possible d'améliorer le compromis entre délai et énergie par le couplage d'un code fontaine et d'un codage réseau quand le réseau fonctionne en limite de connectivité. Nous avons pu le montrer à la fois dans notre étude du canal à relai et à la fois dans notre étude sur un réseau de capteurs à large échelle. Dans ces deux cas, la diversité introduite par le codage réseau permet de réduire le nombre de transmissions et ainsi réduire le délai et l'énergie consommée.

En conclusion, grâce à une analyse théorique et à des simulations, nous avons démontré que, grâce au codage réseau, la *diversité multi-utilisateurs* ou la *diversité par coopération* dans les schémas de communications coopératives peuvent être améliorées, permettant d'optimiser le compromis énergie/délai, en particulier lorsque le réseau est fortement perturbé et que les erreurs de transmission sont importantes.

6.2 Perspectives

Nous proposons plusieurs pistes afin d'améliorer les scénarios de codage réseau que nous avons proposés pour les flux d'information encodés fontaine.

Mécanisme d'accusé de réception ou acquittement La première piste consiste à améliorer le mécanisme d'acquittement dans les flux multi-sauts, pour réduire le taux de débordement. L'image du jardinier que nous avons utilisée pour illustrer le problème montre bien qu'un bon mécanisme nécessite de pouvoir bloquer les re-transmissions à différents endroits dans un flux multi-sauts, et pas seulement par le destinataire. Le problème est que dans ce cas, les nœuds intermédiaires doivent être capables de mesurer la quantité d'information transmise, sans effectuer tout le décodage.

En effet, l'acquittement final est nécessaire dans les transmissions sans rendement prédéfini pour couper le flux lorsque toute l'information est transmise. Cependant, l'envoi d'un message d'acquittement en bout de chaîne n'est pas efficace et de plus doit être prioritaire par rapport aux paquets de données. Dans le cas d'un accès au canal par CSMA, la durée du backoff peut être utilisée pour prioriser les acquittements, comme cela est fait dans 802.11. La vraie question est de savoir comment anticiper sur l'information transmise. Une méthode envisageable est une approche basée sur une mesure de la quantité d'information transmise par chaque relai, pour lui permettre d'acquitter les données vers la source, sans attendre que le destinataire ait tout reçu. Une telle approche va surcharger le travail des relais, et n'est possible que pour un flux mono-trajet. Lorsque les informations sont diffusées, comme dans le cas du routage par gradient étudié au chapitre 5, il n'est plus efficace de stopper les émissions à partir des nœuds relais. En effet, si les nœuds envoient l'acquittement lorsqu'ils ont reçu toute l'information, l'intérêt de la diversité spatiale disparaît, puisqu'il faut qu'au moins un nœud reçoive tout. À l'inverse, il est délicat d'estimer la quantité d'information présente sur l'ensemble des nœuds. On peut aussi envisager que le récepteur anticipe sur les derniers paquets et envoie un acquittement lorsqu'il a reçu un nombre prédéfini de paquets.

Amélioration de codage de réseau La deuxième direction consiste à améliorer la technique de codage réseau avec code LT pour contourner le problème de la distribution de degrés.

Le décodage par propagation de croyance est sous-optimal, mais permet de réduire la complexité de traitement. Si le nœud destinataire est un puits et n'est pas contraint en énergie, il semble préférable d'utiliser un décodage plus performant. Cependant, lorsque les nœuds destinataires sont des nœuds classiques, ou lorsque l'on souhaite que les nœuds relais effectuent un décodage de contrôle, l'approche par propagation de croyance semble particulièrement bien adaptée. Or, les bonnes performances de ce décodage dépendent directement de la distribution des degrés des paquets transmis. Nous avons pu voir dans nos travaux comment l'introduction du codage en réseau modifiait en profondeur la distribution des degrés. Pour permettre d'exploiter toutes les promesses du codage en réseau, il est donc nécessaire de trouver d'autres techniques de décodage moins dépendantes des distributions de degrés. Ainsi, nous pourrions envisager d'autres algorithmes de décodage qui permettent d'avoir un compromis entre la rapidité de décodage et la complexité de calcul.

La deuxième solution consiste à garantir un certain contrôle de ces distributions lors de la mise

en place du codage en réseau pour conserver la capacité de décodage. Nous pourrions envisager l'amélioration de la performance du codage réseau par les relais en considérant, par exemple, un ré-encodage réalisé dans un corps de Galois plus grand \mathbb{F}_q , avec $q = 2^b$ éléments, au lieu de \mathbb{F}_2 . Bien que l'encodage et le décodage dans un dimension du corps plus grand nécessitent une plus grande complexité de calcul, la transmission a été montrée plus robuste avec un meilleur taux d'erreur [62, 144].

Bibliographie

- [1] WSNNet / Worldsens simulator. <http://wsnet.gforge.inria.fr/>.
- [2] Szymon Acedanski, Supratim Deb, Muriel Médard, and Ralf Koetter. How good is random linear coding based distributed networked storage. In *Proc. NetCod Conference*, 2005.
- [3] S. Acharya, M. Franklin, and S. Zdonik. Dissemination-based data delivery using broadcast disks. *IEEE Personal Communications*, 2(6) :50–60, December 1995.
- [4] R. Ahlswede, Ning Cai, S.-Y.R. Li, and R.W. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 46(4) :1204–1216, July 2000.
- [5] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks : a survey. *Comput. Netw.*, 38 :393–422, March 2002.
- [6] Zigbee Alliance. Zigbee. <http://www.zigbee.org/>.
- [7] E. Altman and F. De Pellegrini. Forward correction and fountain codes in delay-tolerant networks. *IEEE/ACM Transactions on Networking*, 19(1) :1–13, February 2011.
- [8] S.A. Aly, Zhenning Kong, and E. Soljanin. Fountain codes based distributed storage algorithms for large-scale wireless sensor networks. In *Proc. International Conference on Information Processing in Sensor Networks (IPSN)*, pages 171–182, April 2008.
- [9] S.A. Aly, Zhenning Kong, and E. Soljanin. Raptor codes based distributed storage algorithms for wireless sensor networks. *IEEE International Symposium on Information Theory*, pages 2051–2055, July 2008.
- [10] Apavatjrut Anya, Goursaud Claire, and Gorce Jean-Marie. Impact des codes fontaine sur la consommation d’énergie dans les réseaux de capteurs avec prise en compte d’une couche mac réaliste. Sep 2009.
- [11] A. Apavatjrut, C. Goursaud, K. Jaffre ? ands Runser, C. Comaniciu, and J. Gorce. Toward increasing packet diversity for relaying lt fountain codes in wireless sensor networks. *Communications Letters, IEEE*, 15(1) :52 –54, 2011.
- [12] Anya Apavatjrut. Fountain codes for exploiting diversity in wireless sensor networks. *A World of Wireless, Mobile and Multimedia Networks, International Symposium on*, 0 :1–2, 2010. (Ph.D.Forum, Best Ph.D. Presentation Award).
- [13] Anya Apavatjrut, Claire Goursaud, Katia Jaffrès-Runser, Cristina Commaniciu, and Jean-Marie Gorce. Exploitation de la diversité par relayage dans les réseaux de capteurs sans fils. *Actes d’Algotel, Belle Dune - Cote d’Opale, France*, Juin 2010.
- [14] Anya Apavatjrut, Katia Jaffrès Runser, Claire Goursaud, and Cédric Lauradoux. On the Flooding Overhead of Fountain Codes in Wireless Sensor Networks. (RR-7397) :16, 09 2010.

- [15] Anya Apavatjirut, Wassim Znaidi, Antoine Fraboulet, Claire Goursaud, Katia Jaffrès-Runser, Cédric Lauradoux, and Marine Minier. Energy efficient authentication strategies for network coding. *to be appeared in Concurrency and Computation : Practice and Experience, John Wiley & Sons*, 2011.
- [16] Anya Apavatjirut, Wassim Znaidi, Antoine Fraboulet, Claire Goursaud, Cedric Lauradoux, and Marine Minier. Energy friendly integrity for network coding in wireless sensor networks. In *Proceedings of the 2010 Fourth International Conference on Network and System Security, NSS '10*, pages 223–230, Washington, DC, USA, 2010. IEEE Computer Society.
- [17] A. Beimel, S. Dolev, and N. Singer. Rt oblivious erasure correcting. In *Proc. IEEE Information Theory Workshop*, pages 236–241, October 2004.
- [18] C. Berger, Shengli Zhou, Yonggang Wen, P. Willett, and K. Pattipati. Optimizing joint erasure and error-correction coding for wireless packet transmissions. *IEEE Transactions on Wireless Communications*, 7(11) :4586–4595, November 2008.
- [19] E. Berlekamp. Nonbinary bch decoding (abstr.). *IEEE Transactions on Information Theory*, 14(2) :242, March 1968.
- [20] Azer Bestavros. *An Adaptive Information Dispersal Algorithm For Time-Critical Reliable Communication*, volume 2. Plenum Publishing Corporation, 1994.
- [21] Sebastien Blanchard, Sofiane Rerbal, Anya Apavatjirut, Claire Goursaud, and Jean-Marie Gorce. Etude du relayage de codes fontaine sur un lien de transmission multi-sauts. *Actes de GRETSI, Dijon, France*, 2009.
- [22] Marcel Busse, Thomas Haenselmann, and Wolfgang Effelsberg. Energy-Efficient Data Dissemination for Wireless Sensor Networks. In *Proc. IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 301–306, White Plains, NY, USA, 2007.
- [23] John W. Byers, Jeffrey Considine, Michael Mitzenmacher, and Stanislav Rost. Informed content delivery across adaptive overlay networks. In *Proc. Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, pages 47–60, Pittsburgh, PA, USA, 2002.
- [24] John W. Byers, Michael Luby, Michael Mitzenmacher, and Ashutosh Rege. A Digital Fountain Approach to Reliable Distribution of Bulk Data. In *Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, pages 56–67, Vancouver, BC, Canada, 1998.
- [25] J.W. Byers, M. Luby, and M. Mitzenmacher. Accessing multiple mirror sites in parallel : using tornado codes to speed up downloads. In *Proc. 18th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 1, pages 275–283, March 1999.
- [26] G. Caire, S. Shamai, A. Shokrollahi, and S. Verdu. Universal variable-length data compression of binary sources using fountain codes. In *Proc. IEEE Information Theory Workshop*, pages 123–128, October 2004.
- [27] G. Caire, S. Shamai, and S. Verdu. Lossless data compression with error correcting codes. In *Proc. IEEE International Symposium on Information Theory*, page 22, 2003.

- [28] G. Caire, S. Shamai, and S. Verdú. A new data compression algorithm for sources with memory based on error correcting codes. In *Proc. IEEE Information Theory Workshop*, pages 291 – 295, 2003.
- [29] Giuseppe Caire, Shlomo Shamai, Amin Shokrollahi, and Sergio Verdú. Fountain codes for lossless data compression. In *Series in Discrete Mathematics and Theoretical Computer Science, Algebraic Coding Theory and Information Theory Workshop (DIMACS)*, volume 68, December 2005.
- [30] J. Castura and Yongyi Mao. Rateless coding for wireless relay channels. In *Proc. International Symposium on Information Theory (ISIT)*, pages 810–814, September 2005.
- [31] J. Castura and Yongyi Mao. Rateless coding for wireless relay channels. In *Proc. International Symposium on Information Theory (ISIT)*, pages 810–814, September 2005.
- [32] M.-L. Changel, K. Huguenin, A.-M. Kermarrec, and N. Le Scouarnec. Lt network codes. In *Proc. IEEE 30th International Conference on Distributed Computing Systems (ICDCS)*, pages 536–546, June 2010.
- [33] R. Comroe and Jr. Costello, D. Arq schemes for data transmission in mobile radio systems. *Selected Areas in Communications, IEEE Journal on*, 2(4) :472 – 481, jul 1984.
- [34] Diane Cook and Sajal Das. *Smart Environments : Technology, Protocols and Applications (Wiley Series on Parallel and Distributed Computing)*. Wiley-Interscience, 2004.
- [35] Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley series in telecommunications. Wiley, New York, NY [u.a.], 1991.
- [36] A.F. Dana, R. Gowaikar, R. Palanki, B. Hassibi, and M. Effros. Capacity of wireless erasure networks. *IEEE Transactions on Information Theory*, 52(3) :789–804, March 2006.
- [37] S.M. Diamond and M.G. Ceruti. Application of wireless sensor network to military information integration. 1 :317–322, June 2007.
- [38] Frédéric Didier. Efficient erasure decoding of reed-solomon codes. *Computing Research Repository (CORR)*, 2009.
- [39] A.G. Dimakis, V. Prabhakaran, and K. Ramchandran. Distributed fountain codes for networked storage. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 5, 2006.
- [40] S. Dolev, B. Fitingof, A. Melkman, and O. Tubman. Smooth and adaptive forward erasure correcting. In *Proc. The 21st IEEE Convention of the Electrical and Electronic Engineers in Israel*, pages 483–486, 2000.
- [41] S.C. Draper, B.J. Frey, and F.R. Kschischang. Efficient variable length channel coding for unknown dmcs. In *Proc. International Symposium on Information Theory (ISIT)*, page 379, 2004.
- [42] A. El Gamal and T.M. Cover. Multiple user information theory. *Proc. of the IEEE*, 68(12) :1466–1483, December 1980.
- [43] Peter Elias. Coding for two noisy channels. In *Proc. London Symposium on Information Theory*, London, UK, 1955.

- [44] Atilla Eryilmaz, Asuman E. Ozdaglar, Muriel Médard, and Ebad Ahmed. On the delay and throughput gains of coding in unreliable networks. *IEEE Transactions on Information Theory*, 54(12) :5511–5524, 2008.
- [45] O. Etesami. Relations between belief propagation on erasure and symmetric channels. In *Proc. International Symposium on Information Theory (ISIT)*, page 209, Chicago, IL, USA, 2004. IEEE.
- [46] Yang Fei. *Reliable and Time-Constrained Communication in Wireless Sensor Networks*. PhD thesis, l’Institut National des Sciences Appliquées de Lyon, 2010.
- [47] S. Floyd, V. Jacobson, C.-G. Liu, S. McCanne, and L. Zhang. A reliable multicast framework for light-weight sessions and application level framing. *IEEE/ACM Transactions on Networking*, 5(6) :784–803, December 1997.
- [48] IEEE Standard for Information technology. IEEE 802.15.4-2006. <http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf>.
- [49] C. Fragouli, J. Widmer, and J.-Y. Le Boudec. Efficient broadcasting using network coding. *IEEE/ACM Transactions on Networking*, 16(2) :450–463, April 2008.
- [50] M. Fresia and L. Vandendorpe. Distributed source coding using raptor codes. In *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, pages 1587–1591, November 2007.
- [51] R. Gallager. Low-density parity-check codes. *IRE Transactions on Information Theory*, 8(1) :21–28, January 1962.
- [52] A.E. Gamal. The capacity of a class of broadcast channels. *IEEE Transactions on Information Theory*, 25(2) :166–169, March 1979.
- [53] J. Gemmell, E. Schooler, and R. Kermode. A scalable multicast architecture for one-to-many telepresentations. In *Proc. IEEE International Conference on Multimedia Computing and Systems*, pages 128–139, June 1998.
- [54] Andrea Goldsmith. *Wireless Communications*. Cambridge University Press, 2005.
- [55] R. Gummadi and R.S. Sreenivas. Relaying a fountain code across multiple nodes. In *Proc. IEEE Information Theory Workshop (ITW)*, pages 149–153, 2008.
- [56] Olivier Salvatori Guy Pujolle and Jacques Nozick. *Les Réseaux*. Eyrolles, 2004.
- [57] A. Hagedorn, S. Agarwal, D. Starobinski, and A. Trachtenberg. Rateless Coding with Feedback. In *Proc. IEEE Computer and Communications Societies (INFOCOMM)*, pages 1791–1799, Rio de Janeiro, Brazil, 2009. IEEE.
- [58] B. Hajek. Connections between network coding and stochastic network theory. In *Proc. Stochastic Networks Conference*, June 2006.
- [59] Tracey Ho, Ralf Koetter, Muriel Médard, David R. Karger, and Michelle Effros. The benefits of coding over routing in a randomized setting. In *Proc. IEEE International Symposium on Information Theory*, 2003.
- [60] Tracey Ho, Muriel Médard, Ralf Koetter, David R. Karger, Michelle Effros, Jun Shi, and Ben Leong. Toward a random operation of networks. *IEEE Transactions on Information Theory*, 2004 :1–8, 2004.

- [61] Tracey Ho, Muriel Medard, Jun Shi, Michelle Effros, and David R. Karger. On randomized network coding. In *Proc. The 41st Annual Allerton Conference on Communication, Control, and Computing*, 2003.
- [62] X.-Yu. Hu and E. Eleftheriou. Binary representation of cycle tanner-graph $gf(2b)$ codes. In *Proc. IEEE International Conference on Communications*, volume 1, pages 528–532, june 2004.
- [63] Jonathan W. Hui and David Culler. The dynamic behavior of a data dissemination protocol for network programming at scale. In *Proc. The 2nd international conference on Embedded networked sensor systems (SenSys)*, pages 81–94, 2004.
- [64] Christian Huitema. The case for packet level fec. In *Proc. The fifth International Workshop on Protocols for High-Speed Networks*, pages 109–120, London, UK, 1997. Chapman & Hall, Ltd.
- [65] Todd E. Hunter. *Coded cooperation : A new framework for user cooperation in wireless networks*. PhD thesis, The University of Texas at Dallas, 2004.
- [66] Esa Hyytiä, Tuomas Tirronen, and Jorma T. Virtamo. Optimal Degree Distribution for LT Codes with Small Message Length. In *Proc. IEEE Computer and Communications Societies (INFOCOMM)*, pages 2576–2580, Anchorage, AK, USA, May 2007. IEEE.
- [67] Texas Instruments Inc. CC1100. <http://focus.ti.com/docs/prod/folders/print/cc1100.html>.
- [68] K. Jaffrès-Runser, C. Comaniciu, J.-M. Gorce, and Ruifeng Zhang. U-grab : A utility-based gradient broadcasting algorithm for wireless sensor networks. In *Proc. IEEE Military Communications Conference (MILCOM)*, pages 1–7, October 2009.
- [69] S. Jaggi, P. Sanders, P.A. Chou, M. Effros, S. Egnér, K. Jain, and L.M.G.M. Tolhuizen. Polynomial time algorithms for multicast network code construction. *IEEE Transactions on Information Theory*, 51(6) :1973–1982, June 2005.
- [70] N. Kadi and K. Al agha. Optimized mpr-based flooding in wireless ad hoc network using network coding. In *Proc. The 1st IFIP Wireless Days (WD)*, pages 1 –5, 2008.
- [71] Abhinav Kamra, Vishal Misra, Jon Feldman, and Dan Rubenstein. Growth codes : maximizing sensor network data persistence. In *Proc. Conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM)*, pages 255–266, New York, NY, USA, 2006.
- [72] R. Karp, M. Luby, and A. Shokrollahi. Finite length analysis of lt codes. In *Proc. International Symposium on Information Theory (ISIT)*, page 39, Chicago, IL, USA, 2004.
- [73] S. Katti, H. Rahul, Wenjun Hu, D. Katabi, M. Medard, and J. Crowcroft. Xors in the air : Practical wireless network coding. *IEEE/ACM Transactions on Networking*, 16(3) :497–510, June 2008.
- [74] Roger G. Kermode. Scoped hybrid automatic repeat request with forward error correction (sharqfec). In *Proc. Conference on Applications, technologies, architectures, and protocols for computer communication (SIGCOMM)*, SIGCOMM '98, pages 278–289, New York, NY, USA, 1998. ACM.

- [75] R. Khalili and K. Salamatian. On the achievability of cut-set bound for a class of erasure relay channels. In *Proc. International Workshop on Wireless Ad-Hoc Networks*, pages 275–280, June 2004.
- [76] R. Khalili and K. Salamatian. On the capacity of erasure relay channel : multi-relay case. In *Proc. IEEE Information Theory Workshop*, page 5, September 2005.
- [77] A. Khandekar. *Graph-based codes and iterative decoding*. PhD thesis, California Institute of Technology, 2002.
- [78] R. Koetter and M. Medard. An algebraic approach to network coding. In *Proc. IEEE International Symposium on Information Theory*, page 104, 2001.
- [79] S. Kocalj-Filipovic, P. Spasojevic, and E. Soljanin. Doped fountain coding for minimum delay data collection in circular networks. *IEEE Journal on Selected Areas in Communications*, 27(5) :673–684, 2009.
- [80] John Kubiawicz, David Bindel, Yan Chen, Steven Czerwinski, Patrick Eaton, Dennis Geels, Ramakrishna Gummadi, Sean Rhea, Hakim Weatherspoon, Chris Wells, and Ben Zhao. Oceanstore : an architecture for global-scale persistent storage. In *Proc. The 9th international ACM conference on Architectural support for programming languages and operating systems*, pages 190–201, New York, NY, USA, 2000.
- [81] E. Kurniawan, S. Sumei, K. Yen, and K.F.E. Chong. Network coded transmission of fountain codes over cooperative relay networks. In *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6, April 2010.
- [82] J. Lacan, V. Roca, J. Peltotalo, and S. Peltotalo. *Reed-Solomon Forward Error Correction (FEC) Schemes*, April 2009. IETF Request for Comments, RFC 5510 (Standards Track/Proposed Standard).
- [83] J. N. Laneman. *Cooperation in Wireless Networks : Principles and Applications*. Cooperative Diversity : Models, Algorithms, and Architectures. 2006.
- [84] Benoît Latré, Pieter De Mil, Ingrid Moerman, Bart Dhoedt, Piet Demeester, and Niek Van Dierdonck. Throughput and delay analysis of unslotted ieee 802.15.4. *JNW*, 1(1) :20–28, 2006.
- [85] Jin Li. The efficient implementation of reed-solomon high rate erasure resilient codes. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005 (ICASSP)*, volume 3, pages iii/1097 – iii/1100 Vol. 3, 2005.
- [86] Li Li, R. Ramjee, M. Buddhikot, and S. Miller. Network coding-based broadcast in mobile ad-hoc networks. In *Proc. The 26th IEEE International Conference on Computer Communications (INFOCOM)*, pages 1739–1747, May 2007.
- [87] S.-Y.R. Li, R.W. Yeung, and Ning Cai. Linear network coding. *IEEE Transactions on Information Theory*, 49(2) :371–381, February 2003.
- [88] J. Liang, J. Wang, and J. Chen. An overhearing-based scheme for improving data persistence in wireless sensor networks. In *IEEE International Conference on Communications (ICC)*, pages 1–5, Cape Town, South Africa, 2010.
- [89] Wei Lin, Baoming Bai, Ying Li, and Xiao Ma. Design of q-ary irregular repeat-accumulate codes. In *Proc. International Conference on Advanced Information Networking and Applications (AINA)*, pages 201–206, May 2009.

- [90] Yunfeng Lin, Baochun Li, and Ben Liang. Differentiated data persistence with priority random linear codes. In *Proc. The 27th International Conference on Distributed Computing Systems (ICDCS)*, page 47, June 2007.
- [91] Yunfeng Lin, Ben Liang, and Baochun Li. Data Persistence in Large-Scale Sensor Networks with Decentralized Fountain Codes. In *Proc. The IEEE International Conference on Computer Communications (INFOCOM)*, pages 1658–1666, Anchorage, Alaska, USA, 2007.
- [92] Xi Liu and Teng Joon Lim. Fountain codes over fading relay channels. *IEEE Transactions on Wireless Communications*, 8(6) :3278–3287, 2009.
- [93] Feng Lu, Chuan Heng Foh, Jianfei Cai, and Liang-Tien Chia. Lt codes decoding : Design and analysis. In *Proc. International Symposium on Information Theory (ISIT)*, pages 2492–2496, Seoul, South Korea, 2009.
- [94] Michael Luby. Tornado codes : Practical erasure codes based on random irregular graphs. In *Proc. Randomization and Approximation Techniques in Computer Science (RANDOM)*, page 171, Barcelona, Spain, 1998. Springer Verlag.
- [95] Michael Luby. LT Codes. In *Proc. The 43rd IEEE Symposium on Foundations of Computer Science (FOCS)*, page 271, Vancouver, BC, Canada, November 2002.
- [96] Michael Luby, Tiago Gasiba, Thomas Stockhammer, and Mark Watson. Reliable Multimedia Download Delivery in Cellular Broadcast Networks. *IEEE Transactions on Broadcasting*, 53(1) :235–246, 2007.
- [97] Michael Luby, Michael Mitzenmacher, and Mohammad Amin Shokrollahi. Analysis of random processes via and-or tree evaluation. In *Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 364–373, San Francisco, CA, USA, 1998.
- [98] Michael Luby, Michael Mitzenmacher, Mohammad Amin Shokrollahi, and Daniel A. Spielman. Efficient erasure correcting codes. *IEEE Transactions on Information Theory*, 47(2) :569–584, 2001.
- [99] Michael G. Luby, Michael Mitzenmacher, M. Amin Shokrollahi, Daniel A. Spielman, and Volker Stemann. Practical loss-resilient codes. In *Proc. of the twenty-ninth annual ACM symposium on Theory of computing (STOC)*, pages 150–159, New York, NY, USA, 1997. ACM.
- [100] M. Lukic, B. Pavkovic, N. Mitton, and I. Stojmenovic. Greedy geographic routing algorithms in real environment. pages 86 –93, December 2009.
- [101] G. Maatouk and A. Shokrollahi. Analysis of the second moment of the LT decoder. In *International Symposium on Information Theory (ISIT)*, pages 2326–2330, Seoul, South Korea, 2009.
- [102] David J. C. MacKay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, New York, NY, USA, 2002.
- [103] David J. C. Mackay. Fountain codes. *IEE Communications*, 152 :1062–1068, 2005.
- [104] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes (North-Holland Mathematical Library)*. North Holland, June 1988.
- [105] F.J. MacWilliams and N.J.A. Sloane. *The theory of error-correcting codes*. 1978.

- [106] E. Maneva and A. Shokrollahi. New model for rigorous analysis of lt-codes. In *Proc. International Symposium on Information Theory (ISIT)*, pages 2677–2679, Seattle, IL, USA, 2006.
- [107] J. Massey. Shift-register synthesis and bch decoding. *IEEE Transactions on Information Theory*, 15(1) :122–127, January 1969.
- [108] Petar Maymounkov. Online codes. Technical report, New York University Technical Report, 2002.
- [109] Edward C. Van Der Meulen. Three-terminal communication channels. *Advances in Applied Probability*, 3(1) :120–154, 1971.
- [110] A.F. Molisch, N.B. Mehta, J.S. Yedidia, and Zhang Jin. Performance of fountain codes in collaborative relay networks. *IEEE Transactions on Wireless Communications*, 6(11) :4108–4119, 2007.
- [111] R. Nikjah and N. Beaulieu. Low complexity selection cooperation techniques using information accumulation in dual-hop relaying networks. *IEEE Transactions on Wireless Communications*, (99) :1–13, 2011.
- [112] Lucie Nodin, Anya Apavatjirut, Claire Goursaud, and Jean-Marie Gorce. Degree distribution of xored fountain codes : Theoretical derivation and analysis. In *Proc. The 16th Asia-Pacific Conference on Communications (APCC)*, November 2010.
- [113] J. Nonnenmacher, M. Lacher, M. Jung, E.W. Biersack, and G. Carle. How bad is reliable multicast without local recovery? In *Proc. The 17th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 3, pages 972–979, April 1998.
- [114] K. Obraczka. Multicast transport protocols : a survey and taxonomy. *IEEE Communications Magazine*, 36(1) :94–102, January 1998.
- [115] A. Oka and L. Lampe. Data extraction from wireless sensor networks using distributed fountain codes. *IEEE Transactions on Communications*, 57(9) :2607–2618, 2009.
- [116] Payam Pakzad, Christina Fragouli, and Amin Shokrollahi. Coding schemes for line networks. In *Proc. IEEE International Symposium on Information Theory (ISIT)*, pages 1853–1857, Adelaide, Australia, 2005. IEEE.
- [117] Payam Pakzad and Amin Shokrollahi. Exit functions for lt and raptor codes, and asymptotic ranks of random matrices. In *Proc. IEEE International Symposium on Information Theory (ISIT)*, pages 411–415, Nice, France, 2007.
- [118] R. Palanki and J.S. Yedidia. Rateless codes on noisy channels. In *Proc. IEEE International Symposium on Information Theory (ISIT)*, page 37, July 2004.
- [119] S. Paul, K.K. Sabnani, J.C.-H. Lin, and S. Bhattacharyya. Reliable multicast transport protocol (rmtp). *IEEE Journal on Selected Areas in Communications*, 15(3) :407–421, April 1997.
- [120] James S. Plank. A tutorial on reed-solomon coding for fault-tolerance in raid-like systems. *Software Practice & Experience*, 27(9) :995–1012, September 1997.
- [121] J.S. Plank and M.G. Thomason. A practical analysis of low-density parity-check erasure codes for wide-area storage applications. In *Proc. International Conference on Dependable Systems and Networks*, pages 115–124, July 2004.

- [122] Joseph Polastre, Jason Hill, and David Culler. Versatile low power media access for wireless sensor networks. In *Proc. The 2nd AMC international conference on Embedded networked sensor systems (SenSys)*, pages 95–107, New York, NY, USA, 2004.
- [123] S. Puducheri, J. Kliewer, and T.E. Fuja. Distributed LT Codes. In *Proc. IEEE International Symposium on Information Theory (ISIT)*, pages 987–991, 2006.
- [124] Shuang Qin, Gang Feng, and Yide Zhang. Capacity bounds of cooperative communications with fountain codes. In *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–4, April 2010.
- [125] M. Rabbat, J. Haupt, A. Singh, and R. Nowak. Decentralized compression and pre-distribution via randomized gossiping. In *Proc. The 5th International Conference on Information Processing in Sensor Networks (IPSN)*, pages 51–59, 2006.
- [126] Kacimi Rahim. Techniques de conservation d'énergie pour les réseaux de capteurs sans fil. Master's thesis, Institut National Polytechnique de Toulouse, 2009.
- [127] Nazanin Rahnavard, Badri N. Vellambi, and Faramarz Fekri. Rateless codes with unequal error protection property. *IEEE Transactions on Information Theory*, 53(4) :1521–1532, April 2007.
- [128] I. S. Reed and G. Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2) :300–304, 1960.
- [129] Luigi Rizzo. Effective erasure codes for reliable computer communication protocols. *ACM SIGCOMM Computer Communication Review*, 27 :24–36, April 1997.
- [130] Luigi Rizzo and Lorenzo Vicisano. A reliable multicast data distribution protocol based on software fec techniques. In *Proc. The 4th IEEE Workshop on High-Performance Communication Systems(HPCS)*, pages 23–25, 1997.
- [131] V. Roca, C. Neumann, and D. Furodet. Low density parity check (ldpc) forward error correction rfc-5170., June 2008.
- [132] M. Rossi, G. Zanca, L. Stabellini, R. Crepaldi, A.F. Harris, and M. Zorzi. Synapse : A network reprogramming protocol for wireless sensor networks using fountain codes. In *Proc. IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 188–196, San Francisco, CA, USA, 2008.
- [133] Peter Sanders, Sebastian Egner, and Ludo Tolhuizen. Polynomial time algorithms for network information flow. In *Proc. The 15th annual ACM symposium on Parallel algorithms and architectures (SPAA)*, pages 286–294, New York, NY, USA, 2003.
- [134] S. Sanghavi. Intermediate performance of rateless codes. In *IEEE Information Theory Workshop (ITW)*, pages 478–482, Lake Tahoe, CA, USA, 2007.
- [135] Eve Schooler, Jim Gemmell, and Redmond Wa. Using multicast fec to solve the midnight madness problem. Technical report, Technical Report, Microsoft research, 1997.
- [136] D. Sejdinovic, R. Piechocki, A. Doufexi, and M. Ismail. Fountain code design for data multicast with side information. *IEEE Transactions on Wireless Communications*, 8(10) :5155–5165, 2009.
- [137] D. Sejdinovic, R. J. Piechocki, A. Doufexi, and M. Ismail. Fountain Coding with Decoder Side Information. In *Proc. IEEE International Conference on Communications (ICC)*, pages 4477–4482, Beijing, China, 2008.

- [138] D. Sejdinovic, R.J. Piechocki, and A. Doufexi. And-or tree analysis of distributed lt codes. In *Proc. IEEE Workshop on Information Theory (ITW)*, pages 261–265, 2009.
- [139] D. Sejdinovic, D. Vukobratovic, A. Doufexi, V. Senk, and R. Piechocki. Expanding window fountain codes for unequal error protection. *IEEE Transactions on Communications*, 57(9) :2510–2516, 2009.
- [140] A. Sendonaris, E. Erkip, and B. Aazhang. User cooperation diversity. part i. system description. *IEEE Transactions on Communications*, 51(11) :1927–1938, November 2003.
- [141] A. Sendonaris, E. Erkip, and B. Aazhang. User cooperation diversity. part ii. implementation aspects and performance analysis. *IEEE Transactions on Communications*, 51(11) :1939–1948, November 2003.
- [142] Claude Elwood Shannon. A mathematical theory of communication. *Bell Systems Technical Journal*, 27 :379–423,623–656, 1948.
- [143] Amin Shokrollahi. Raptor codes. *IEEE Transactions on Information Theory*, 52(6) :2551–2567, 2006.
- [144] C. Spagnol and W.P. Marnane. Performance of ldpc codes over $gf(2^m)$. *IEEE Communications Letters*, 2007.
- [145] I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 13(1) :14–25, January 2002.
- [146] C. Studholme and I. Blake. Windowed erasure codes. In *IEEE International Symposium on Information Theory*, pages 509–513, July 2006.
- [147] Chris Studholme and Ian Blake. Random matrices and codes for the erasure channel. *Algorithmica*, 56 :605–620, 2010.
- [148] V.G. Subramanian and D.J. Leith. On a class of optimal rateless codes. In *Proc. IEEE Annual Allerton Conference on Communication, Control, and Computing*, pages 418–425, 2008.
- [149] Madhu Sudan. List decoding : Algorithms and applications. In *ACM SIGACT News*, volume 31, March 2000.
- [150] Andrew Tanenbaum. *Computer Networks*. Prentice Hall Professional Technical Reference, 4th edition, 2002.
- [151] R. Tanner. A recursive approach to low complexity codes. *IEEE Transactions on Information Theory*, 27(5) :533–547, September 1981.
- [152] A. Tarable and S. Benedetto. Efficiency of precode-only raptor codes and turbo-fountain codes. In *Proc. IEEE Information Theory Workshop (ITW)*, pages 61–65, Punta del Este, Uruguay, 2006. IEEE.
- [153] A. Tarable and S. Benedetto. Graph-based lt and raptor codes. In *IEEE International Symposium on Information Theory (ISIT)*, pages 2061–2065, Toronto, ON, Canada, 2008.
- [154] Watteyne Thomas. *Energy-Efficient Self-Organization for Wireless Sensor Networks*. PhD thesis, Institute National des Sciences appliquées de Lyon, 2008.
- [155] T. Tirronen. *Fountain codes : Performance Analysis and Optimisation*. PhD thesis, Helsinki University of Technology, 2009.

- [156] T. Tirronen. Sliding window-based erasure correction using biased sampling. In *Proc. International Conference on Systems and Networks Communications (ICSNC)*, pages 144–152, 2009.
- [157] T. Tirronen and J. Virtamo. Performance analysis of divided random linear fountain. In *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, pages 520–526, 2007.
- [158] T. Tirronen and J. Virtamo. Finding fountain codes for real-time data by fixed point method. In *Proc. IEEE International Symposium on Information Theory and Its Applications (ISITA)*, pages 1–6, 2008.
- [159] T. Tirronen and J. Virtamo. Greedy approach for efficient packet erasure coding. In *Proc. International Symposium on Turbo Codes and Related Topics*, pages 344–349, 2008.
- [160] T. Tirronen and J. Virtamo. Performance analysis of sliding window based erasure correction for real-time traffic. In *Proc. Next Generation Internet Networks (NGI)*, pages 1–8, July 2009.
- [161] ETSI DVB TM-CBMS1167. Ip datacast over dvb-h : Content delivery protocols draft technical specification, September 2005.
- [162] D. Tuninetti and C. Fragouli. Processing along the way : forwarding vs. coding. In *Proc. IEEE Symposium on Information Theory and its Applications (ISITA)*, Parma, Italy, 2004.
- [163] 3GPP TS 23.246 V6.9.0. Multimedia broadcast/multicast service : Architecture and functional description, December 2005.
- [164] B.N. Vellambi, N. Rahnavard, and F. Fekri. Fts : A distributed energy-efficient broadcasting scheme using fountain codes for multihop wireless networks. *IEEE Transactions on Communications*, 58(12) :3561–3572, December 2010.
- [165] S. Verdú and S. Shamai. Variable-rate channel capacity. *IEEE Transactions on Information Theory*, 56(6) :2651–2667, 2010.
- [166] D. Wang, Qian Zhang, and Jianchuan Liu. Partial network coding : Theory and application for continuous sensor data collection. In *Proc. The 14th IEEE International Workshop on Quality of Service (IWQoS)*, pages 93–101, June 2006.
- [167] T. Watteyne, A. Molinaro, M. Richichi, and M. Dohler. From manet to ietf roll standardization : A paradigm shift in wsn routing protocols. *IEEE Communications Surveys Tutorials*, (99) :1–20, 2010.
- [168] Geoffrey Werner-Allen, Konrad Lorincz, Matt Welsh, Omar Marcillo, Jeff Johnson, Mario Ruiz, and Jonathan Lees. Deploying a wireless sensor network on an active volcano. *IEEE Internet Computing*, 10 :18–25, March 2006.
- [169] H. Wicaksana, S. H. Ting, and Y. L. Guan. Spectral efficient half duplex relaying for fountain code with wireless network coding. In *Proc. IEEE International Conference on Communications Workshops (ICC)*, Beijing, China, May 2008.
- [170] Alec Woo, Terence Tong, and David Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. In *Proc. the 1st AMC international conference on embedded networked Sensor Systems (SenSys)*, pages 14–27, New York, NY, USA, 2003.

- [171] Anthony D. Wood and John A. Stankovic. Online coding for reliable data transfer in lossy wireless sensor networks. In *Proceedings of the 5th IEEE International Conference on Distributed Computing in Sensor Systems*, DCOSS '09, pages 159–172, Berlin, Heidelberg, 2009. Springer-Verlag.
- [172] Jing Yang, Jianping An, Xiangming Li, and Lei Yuan. Combined fountain code with network coding in cooperative communication. In *Proc. The 2nd International Conference on Networks Security Wireless Communications and Trusted Computing (NSWCTC)*, volume 2, pages 24–27, April 2010.
- [173] Fan Ye, G. Zhong, J. Cheng, Songwu Lu, and Lixia Zhang. Peas : a robust energy conserving protocol for long-lived sensor networks. In *Proc. The 23rd International Conference on Distributed Computing Systems*, pages 28–37, May 2003.
- [174] Fan Ye, Gary Zhong, Songwu Lu, and Lixia Zhang. Gradient broadcast : a robust data delivery protocol for large scale sensor networks. *Wireless Network*, 11 :285–298, May 2005.
- [175] Meng Yu and Jing Li. Is amplify-and-forward practically better than decode-and-forward or vice versa? In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 3, pages 365–368, March 2005.
- [176] Xiaojun Yuan and Li Ping. Doped accumulate lt codes. In *Proc. IEEE International Symposium on Information Theory (ISIT)*, pages 2001–2005, Nice, France, 2007.
- [177] Ruifeng ZHANG. *Analysis of Energy-Delay Performance in Multi-Hop Wireless Sensor Networks*. PhD thesis, INSA-LYON, 2009.
- [178] B. Zhao and M.C. Valenti. Distributed turbo coded diversity for relay channel. *Electronics Letters*, 39(10) :786–787, May 2003.
- [179] Kong Zhenning, S. Aly, and E. Soljanin. Decentralized coding algorithms for distributed storage in wireless sensor networks. *IEEE Journal on Selected Areas in Communications*, 28(2) :261–267, 2010.

Liste de publications

Journaux

- ⇒ A. Apavatjrut, W. Znaidi, A. Fraboulet, Cl. Goursaud, K. Jaffrès-Runser, C. Lauradoux and M. Minier, *Energy Efficient Authentication Strategies for Network Coding*, to be appeared in *Concurrency and Computation : Practice and Experience*, John Wiley & Sons 2011.
- ⇒ A. Apavatjrut, C. Goursaud, K. Jaffrès-Runser, C. Comaniciu and J.M.Gorce, *Toward Increasing Packet Diversity for Relaying LT Fountain Codes in Wireless Sensor Networks*, *IEEE Communications Letters*, Jan 2011.

Conférences Internationaux

- ⇒ A. Apavatjrut, K. Jaffrès-Runser, C. Goursaud and C. Lauradoux, *On the Flooding Overhead of Fountain Codes in Wireless Sensor Networks*, in *Proceedings IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Toronto, Canada, September 2011.
- ⇒ L. Nodin, A. Apavatjrut, C. Goursaud and J.M. Gorce, *Degree Distribution of X-Red Fountain Codes : Theoretical Derivation and Analysis*, in *Proceedings Asia-Pacific Conference on Communications (APCC)*, Auckland, New Zealand, November 2010.
- ⇒ A. Apavatjrut, W. Znaidi, A. Fraboulet, Cl. Goursaud, K. Jaffrès-Runser, C. Lauradoux and M. Minier, *Energy Friendly Integrity for Network Coding in Wireless Sensor Networks*, in *Proceedings International Conference on Network and System Security (NSS)*, Melbourne, Australia, September 2010.
- ⇒ A. Apavatjrut, *Fountain codes for exploiting diversity in wireless sensor networks*, in *Proceedings IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp.1-2, Montreal, Canada, June 2010. (Ph.D.Forum, Best Ph.D. Presentation Award)

Conférences Nationaux

- ⇒ A. Apavatjrut, C. Goursaud and J.M. Gorce, *Impact des Codes Fontaine sur la consommation d'énergie dans les réseaux de capteurs avec prise en compte d'une couche MAC réaliste*, Actes de GRETSI, Dijon, France, September 2009.
- ⇒ S. Blanchard, S. Rerbal, A. Apavatjrut, C. Goursaud, J.M. Gorce, *Étude du relayage de codes fontaine sur un lien de transmission multi-sauts*, Actes de GRETSI, Dijon, France, September 2009.

⇒ A. Apavatjrut, C. Goursaud, K. Jaffrès-Runser, C. Comaniciu and J.M.Gorce, *Exploitation de la diversité par relayage dans les réseaux de capteurs sans fils*, Actes d'Algotel, Belle Dune - Cote d'Opale, France, Juin 2010.

Rapports de Recherche

⇒ A. Apavatjrut, C. Goursaud, K. Jaffrès-Runser, C. Comaniciu and J.M.Gorce, *On the Flooding Overhead of Fountain Codes in Wireless Sensor Networks*, RR-7397, INRIA, <http://hal.inria.fr/inria-00521326/fr/>

Liste d'acronymes

BEC	Binary Erasure Channel
BSC	Binary Symetric Channel
CRC	Cyclic Redundancy Check
GF	Galois Field
FEC	Forward Error Correction
XOR	eXclusive OR
MDS	Maximum Distance Seperable
RS	Reed Solomon
LDPC	Low Density Parity Check
AWGN	Additive White Gaussian Noise
RLF	Random Linear Fountaine
ML	Maximum Likelihood
BP	Belief Propagation
RSD	Robust Soliton Distribution
BD	Binomial Distribution
ARQ	Automatic Repeat-reQuest
TDMA	Time Division Multiple Access
GRAB	GRAdient Broadcasting
WPAN	Wireless Personal Area Network
WBAN	Wireless Body Area Network
OSI	Open Systems Interconnection
DSSS	Direct Sequence Spread Spectrum
BPSK	Binary Pharse Shift Keying
MAC	Medium ACcess
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
TCP	Transmission Control Protocol
SNR	Signal to Noise Ratio
PER	Packet Error Rate
BER	Bit Error Rate

Notations

p_e	probabilité d'erreur
γ	probabilité de succès
$GF(q)$	champ de galois à q éléments
\mathbb{F}_q	un corps fini à q éléments
\mathcal{M}	message à émettre
K	dimension du code
N	taille du mot de code
ϵ	facteur de redondance
$F = \{f_1, f_2, f_3, \dots, f_K\}$	l'ensemble de fragments envoyé
$P = \{p_1, p_2, p_3, \dots, p_K\}$	l'ensemble de paquets envoyé
\oplus	addition dans \mathbb{F}_2 (XOR)
N_b	nombre de bits dans un paquet
N_{tx}	nombre de transmission totale
n_{h_n}	nombre du saut
P_{txElec}	puissance d'émission
P_{rxElec}	puissance de réception
$P_{idleElec}$	puissance consommé à l'état de repos
P_t	puissance de transmission
T	période
T_s	période d'envoi de la source
S	nœud source
D	nœud destinataire
R_i	i -ème relai
d_i	degré du i -ème paquet
$W_H(e_i)$	poids du Hamming
o	collision
e	étape
E	nombre totale d'étape
τ	le taux de relayage
τ_{opt}	le taux de relayage optimale
ξ	probabilité d'appliquer les opérations XOR sur un paquet donné
ν	probabilité de ré-encodage par le relai
γ_{eq}	probabilité de succès équivalente du système
h	nombre de saut