



**HAL**  
open science

# Un codage réseau contraint pour les réseaux de capteurs sans fil

Ismail Salhi

► **To cite this version:**

Ismail Salhi. Un codage réseau contraint pour les réseaux de capteurs sans fil. Autre [cs.OH]. Université Paris-Est, 2012. Français. NNT : 2012PEST1103 . pastel-00794618

**HAL Id: pastel-00794618**

**<https://pastel.hal.science/pastel-00794618>**

Submitted on 26 Feb 2013

**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  
Ismail Salhi

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITE PARIS-EST

École Doctorale MSTIC

**Discipline :** *Informatique*

**Date de soutenance :** 4 Avril 2012

# Un codage réseau contraint pour les réseaux de capteurs sans fil

*Sous la direction de* Gilles ROUSSEL

## Jury

M. Marcelo Dias De Amorim	Rapporteur
M. Fabrice Valois	Rapporteur
M. Khaldoun Al Agha	Examineur
M. Jeremie Leguay	Examineur
M. Mario Gerla	Examineur
M. Yacine Ghamri-Doudane	Co-encadrant de thèse
M. Stéphane Lohier	Co-encadrant de thèse
M. Gilles Roussel	Directeur de thèse



## Remerciements

Je tiens à remercier tous les membres de ce jury : M. Marcelo Diaz De Amorim du Laboratoire d'Informatique de Paris 6, et M. Fabrice Valois de l'Institut National des Sciences Appliquées de Lyon qui ont accepté d'être rapporteurs de ma thèse, pour l'intérêt qu'ils ont porté à mes travaux ainsi que pour leurs corrections qui ont contribué à l'amélioration de ce manuscrit. M. Khaldoun Al Agha, Professeur à l'Université Paris-Sud, M. Mario Gerla, Professeur à l'Université de Los Angeles et M. Jérémie Leguay, membre du groupe Thalès Communications pour avoir eu l'amabilité de bien vouloir participer à ce jury.

Je tiens à remercier M. Gilles Roussel pour son encadrement et son soutien tout au long des trois dernières années. Grâce à sa confiance et son aide j'ai pu me focaliser pleinement et dans les meilleures conditions possibles sur mon travail de recherche. Je tiens également à remercier profondément mes encadrants, Messieurs Yacine Ghamri-Doudane et Stéphane Lohier pour la qualité de leur encadrement, leur précieuse assistance et l'enthousiasme qui leur est propre. Au-delà des collaborations que l'on a pu développer durant cette thèse, je garderais en tête un environnement agréable et stimulant dans lequel j'ai pu évoluer à titre personnel et professionnel, en étant toujours très bien écouté et conseillé.

Un grand merci à l'ensemble de l'équipe algorithmique du LIGM et notamment à sa directrice Mme. Marie-Pierre Béal qui n'est jamais avare d'encouragements envers les doctorants du laboratoire. Line Fonfrede, Pascale Souliez à qui j'ai fréquemment rendu visite pour les diverses formalités administratives. Aurélie, Yann, Ibtissem, Nadia, Erwan, Abderrezak, Lamia et tous ceux qui sont passés par le groupe PASNet pour les discussions et l'effervescence qui ont animé mon séjour doctoral dans les murs du bâtiment Copernic. Je tiens en particulier à dire ma gratitude à mon ami et collègue de bureau Sylvain Cherrier, doctorant, ingénieur, enseignant et musicien pour ne citer que quelques-uns de ses innombrables talents. Sans lui, j'en suis persuadé, mon travail serait moins bon et mes journées plus longues. Mon passage par Orange Labs à Lannion m'a aussi énormément appris, surtout sous la direction de M. Senouci avec qui j'ai fait mes premiers pas dans le monde académique. Je l'en remercie.

Je remercie mes parents et toute ma famille. Sans leur aide, soutien et leur affection, je ne serais pas là où j'en suis. Ce qui d'ailleurs ne serait pas si grave, car c'est eux même qui



m'ont appris à relativiser et à savoir donner la même valeur aux succès et aux échecs. Je suis chanceux de les avoir. Je remercie ma famille au sens plus large. Ghania, la femme qui m'a donné les moyens et la motivation de faire ce métier. Un grand merci à mes défunts grands parents qui, loin des hautes sphères du monde académique, ont réussi à mettre un diplôme dans les mains de chacun de leurs enfants.

Je remercie aussi mes amis de tous bords. Merci à mon colocataire et néanmoins camarade Benjamin qui a dû supporter mes états d'âme de fin de thèse durant ces derniers mois. Merci à mon ami Pierre-Ugo Tournoux, l'un des plus talentueux jeunes docteurs de l'Université Française, j'en suis persuadé.

Enfin, je garde une pensée particulière pour Dobrinka, ma source d'énergie au quotidien. Ton soutien, ta compréhension et ton sourire ont fait que les moments difficiles le soient moins. Cette thèse tu y as contribué à ta façon, merci.

Même si il n'aura pris que 3 ans et quelques mois, ce projet aura changé quelque peu ma vie. Surement, grâce aux voyages qu'il m'a permis de faire, aux gens qu'il m'a fait rencontrer, ou peut-être parce qu'il m'a appris à mieux réfléchir, à constamment structurer, référencer et remettre en question mes certitudes pour essayer de forger mes propres idées.

## Résumé

Les réseaux de communication sans fil inter-machines font partie des grandes évolutions de l'informatique contemporaine. Ces réseaux évolutifs promettent une nouvelle gamme de services et une meilleure connaissance du monde qui nous entoure. Pourtant, ils subissent plusieurs limitations de par la nature même des nœuds qui les composent : un débit faible et une fiabilité réduite, des capacités de calcul et de mémoire limitées, une durée de vie finie, ainsi que des délais non bornés souvent causés par des taux de perte élevés. Ces mêmes problématiques constituent, depuis une quinzaine d'années, un frein au déploiement à large échelle des réseaux de capteurs sans fil dans le monde réel.

De récents développements dans la théorie de l'information sont finalement en train de bouleverser l'ordre établi, donnant naissance à une nouvelle approche connue sous le nom de *théorie du codage réseau (network coding)*. Cette nouvelle façon d'acheminer l'information transforme les méthodes d'exploitation, de gestion et même la compréhension des réseaux de communications actuels et futurs. C'est dans ce contexte, que nous proposons dans cette thèse d'étudier l'impact, les bénéfices et la faisabilité du codage réseau dans les réseaux de capteurs sans fil. L'objectif étant de montrer l'intérêt d'exploiter ses concepts pour les différents paradigmes propres aux réseaux inter-machines en général, et aux réseaux de capteurs en particulier.

Il s'agit dans un premier temps d'évaluer les apports prospectifs du codage réseau, qu'il soit linéaire ou opportuniste, strictement en termes de performances (i.e., latence et débit). Dans une seconde partie, nous poursuivrons notre étude en traitant de l'impact du codage réseau sur la fiabilité des réseaux de capteurs, ces derniers étant caractérisés par l'absence de mécanismes de contrôle de flux de bout-en-bout. Dans une dernière contribution, nous explorons la faisabilité sur une plateforme réelle d'un mécanisme de codage réseau contraint spécifiquement conçu pour les réseaux à faible débit. Ce code opportuniste est centré sur l'optimisation mémoire et l'efficacité énergétique. Il permet d'améliorer à la fois les performances du réseau et sa fiabilité.

De manière plus générale, cette thèse se présente comme *une* réponse à la question : « *Quel type de codage, pour quels réseaux de capteurs ?* » Elle ne prétend pas être l'unique réponse, mais propose néanmoins des briques théoriques et technologiques qui illustrent non seulement la faisabilité du codage réseau dans les réseaux contraints mais aussi la nécessité de l'adapter à leurs critères.





## Abstract

Today, it is likely that Machine-to-machine (M2M) architectures are the future of computer networks. These communication systems promise new strains of pervasive services and a better awareness of the world around us. However, they suffer several limitations caused by the very nature of their components: a low data-rate, unreliable transmissions, limited storage and processing capabilities, finite lifespan and unpredictable delays often caused by a high packet loss rate. These are the same issues that are still preventing, since the late eighties, the awaited large-scale deployment of wireless sensor networks in the world of telecommunications.

Recent developments in the Information Theory are finally bringing a wind of change on the common ways of considering data communications, via a new concept known as the Network Coding Theory. This new data delivery approach is transforming the way we operate, manage and even understand existing and future communications networks. In this context, we propose in this thesis to study the impact, benefits and feasibility of network coding in wireless sensor networks. Our goal is to evaluate the cost/benefit value of its practical concepts for different M2M network paradigms in general, and for sensor networks in particular.

Primarily, we assess the potential impact of opportunistic and linear network coding on sensor networks strictly in terms of performance (i.e., data rate and latency). Then we address the problem of reliable communications in coded sensor networks, since such systems are characterized by the absence of end-to-end flow control mechanisms. In the last chapter, we explore the feasibility of network coding in a real testbed. We investigate how an innovative coding technique specifically designed for low-power/low-rate networks can use opportunistic coding to enhance the performance and the reliability of the network.

More broadly, this thesis presents *one* answer to the question: “*Which network coding for what sensor network?*” It does not claim to be the sole answer to this issue, but rather a series of theoretical and technological blocks that denote both the feasibility of network coding in constrained environments and also the need to adapt it to their particular criteria.





# Table des matières

Introduction.....	1
1.1. Contributions .....	3
1.1.1. Codage réseau inter-machines : performances.....	3
1.1.2. Codage réseau fiable : robustesse.....	3
1.1.3. Codage réseau contraint : adaptabilité .....	4
1.2. Organisation du document.....	5
2. Etat de l'art.....	7
2.1. Réseaux de capteurs sans fil.....	7
2.1.1. Applications et architectures .....	8
2.1.2. Caractéristiques réseaux .....	9
2.1.3. Nature du trafic.....	10
2.1.4. Topologies et méthodes d'adressage.....	11
2.1.5. Routage et acheminement .....	11
2.1.6. Distribution de l'information .....	12
2.1.7. Performance et fiabilité .....	13
2.1.8. Digression sur les efforts de standardisation.....	14
2.2. Codage réseau et réseaux sans fil.....	14
2.2.1. Fondamentaux du codage réseau.....	15
2.2.2. Codage réseau linéaire .....	16
2.2.3. Codage réseau opportuniste.....	17
2.2.4. Codage réseau et réseaux de capteurs.....	18
2.3. Conclusion.....	20
3. Codage inter-machines .....	23
3.1. Codage réseau et dissémination de l'information .....	24



3.1.1.	Contexte et applications .....	24
3.1.2.	Description du système.....	25
3.1.3.	Présentation du codage inter-machines linéaire – WSC.....	26
3.1.4.	Présentation du codage inter-machines opportuniste – CNC.....	32
3.1.5.	Discussion .....	38
3.2.	Codage réseau et routage unicast.....	39
3.2.1.	Contexte et applications .....	39
3.2.2.	Description du système.....	39
3.2.3.	Codage opportuniste pour ZigBee - CoZi.....	40
3.2.4.	Discussion .....	44
3.3.	Conclusion.....	44
4.	Codage réseau fiable.....	47
4.1.	Codage Fiable .....	48
4.1.1.	Inférence de topologie .....	49
4.1.2.	Décodabilité.....	49
4.1.3.	Innovativité .....	50
4.1.4.	Optimalité.....	51
4.1.5.	Equité .....	52
4.1.6.	Taux d’acheminement de bout-en-bout .....	53
4.1.7.	Discussion .....	53
4.2.	Codage ZigBee Fiable – ReCoZi .....	53
4.2.1.	Système d’acquittement des paquets .....	54
4.2.2.	Acquittement à état de lien .....	56
4.2.3.	Gestion équitable des files d’attente de codage.....	57
4.2.4.	Analyse de performance.....	57
4.3.	Conclusion.....	61
5.	Codage réseau contraint .....	65
5.1.	Motivation et contexte.....	66
5.2.	Codage par index – InC .....	67
5.2.1.	Le problème de l’incast .....	67

5.2.2.	Pourquoi un codage contraint ? .....	67
5.2.3.	Hypothèses et modélisation du système.....	68
5.2.4.	Codage par index.....	69
5.2.5.	Limites théoriques.....	71
5.2.6.	Discussion .....	75
5.3.	Codage par index dans la pratique – ZInC .....	76
5.3.1.	Description du modèle réseau .....	76
5.3.2.	Présentation de ZInC.....	77
5.3.3.	Exemple canonique .....	80
5.3.4.	Evaluation de performance .....	81
5.4.	Conclusion.....	87
6.	Conclusion & perspectives.....	91
6.1.	Conclusion.....	91
6.2.	Perspectives .....	93
	Publications.....	95
	Références.....	97
	Tables des figures .....	103







## *Chapitre 1*

# Introduction

Aujourd'hui, les réseaux de capteurs sans fil sont suffisamment matures pour être légitimement considéré comme *la* technologie sous-jacente à l'Informatique omniprésente. Leur capacité à générer de la connaissance utile et contextualisée dans une zone géographique donnée, est en train de transformer notre perception des technologies de l'information. En effet, la collecte et le traitement automatique et distribué des données permettent de nouveaux modes d'interactions avec le monde physique, promettant un environnement plus intelligent, plus accessible et moins complexe. Partageant le point de vue de chercheurs et d'industriels qui travaillent sur cette thématique, nous pensons que les réseaux de capteurs sans fil constituent une nouvelle classe de systèmes informatiques et une future révolution technologique et socio-économique. Car, même si l'on se fie aux prévisions les plus pessimistes, il est vraisemblable qu'au cours des vingt prochaines années, le nombre d'hôtes actuellement actifs sur Internet sera infime comparé à la multitude de dispositifs autonomes qui peupleront le réseau mondial. De plus, cette interconnexion sans fil des réseaux inter-machines constituera à terme la majeure partie du trafic d'Internet.

Ce changement radical dans la nature même des nœuds qui composent les réseaux de communication n'est pas sans conséquences. Les caractéristiques des liens qui interconnectent ces hôtes vont eux aussi changer, impliquant de plus forts besoins en termes de débit, de robustesse et de qualité de service. Les liens d'extrémités observeront une plus grande charge de trafic, et les besoins en stockage seront d'autant plus importants. De plus, de nouveaux critères de fiabilité devront être pris en compte, car pour la première fois dans le monde des réseaux, on assistera à une régression des capacités des hôtes de bordure du réseau en charge du routage et de l'acheminement des données. L'efficacité énergétique se trouvera au centre des préoccupations des concepteurs de réseaux contraints, car à l'inverse des protocoles réseaux actuels, les mécanismes de communication inter-machines doivent prendre en compte la contrainte énergétique que ce soit pour les méthodes d'accès au médium, le routage, ou le transport de bout-en-bout.

D'autre part, des développements récents dans la théorie de l'information ont montré que l'utilisation du schéma classique d'acheminement des paquets, à savoir le *store-and-forward*, sous-utilise la bande passante disponible dans la plupart des réseaux de communication, y compris et surtout dans le cas des architectures sans fil tels que les

réseaux de capteurs. Il est important de noter que ce modèle a été conçu et appliqué à l'origine pour des communications point-à-point entre une source et une destination. Avec la popularisation des communications non-unicast, il n'est peut-être pas aberrant de considérer un nouveau modèle de transmission des données qui serait plus adapté aux communications inter-machines. Car cette sous-optimalité inhérente au *store-and-forward* est à l'origine de bien plus que de simples problèmes de débit. En fait, elle a des répercussions sur la fiabilité, les délais et la consommation d'énergie. Cela montre que les protocoles de communication basés sur le *store-and-forward*, de par leur traitement séquentiel et exclusif des paquets, posent un problème d'efficacité, particulièrement dans les réseaux à plusieurs sources et/ou plusieurs destinations.

Afin de remédier à ce problème, des chercheurs du Massachusetts Institutes of Technology ont réfléchi à une nouvelle approche de transmission de l'information dans les réseaux de communication. Leur solution utilise des combinaisons linéaires de messages provenant de différents flux afin d'approcher les limites théoriques d'utilisation de la bande passante. Et cela, que ce soit dans les communications en saut-par-saut ou de bout-en-bout. Depuis, cette théorie du codage réseau (Network Coding) est devenue une discipline à part entière. Sa rupture conceptuelle avec les méthodes traditionnelles d'acheminement de l'information a ouvert une multitude de perspectives de recherche que ce soit du point de vue traitement de signal, du routage, de la correction d'erreurs ou encore du stockage et du calcul distribué. Utilisée d'abord dans les réseaux filaires en multicast, adaptée ensuite aux réseaux unicast puis aux réseaux sans fil, les principes de cette théorie ont complètement révolutionné la manière dont les réseaux de communications sont perçus aujourd'hui. Ses applications concernent des domaines aussi divers que les communications satellitaires, la sécurité ou encore les réseaux d'information quantique.

Dans ce contexte et motivés par les nouveaux défis technologiques propres aux réseaux de capteurs sans fil, nous explorons dans cette thèse une possible et prometteuse évolution des protocoles de communication inter-machines, qui aurait comme base ce nouveau schéma de communication par paquets qu'est le codage réseau. Toutefois, l'application des concepts innovants de cette théorie aux systèmes de communication inter-machines n'est pas triviale. Et pour cause, certaines caractéristiques inhérentes à ce type de réseaux rendrait une utilisation *telle quelle* des techniques de codage existantes inefficace, si ce n'est irréalisable. Parmi elles, 1) des contraintes technologiques (énergie, capacité mémoire et calcul), 2) des propriétés applicatives (type de trafic, taille des données) et 3) des caractéristiques topologiques (réseaux hiérarchiques et connectivité intermittente).

Cette thèse s'articule donc autour de cette alternative conceptuelle qu'est le codage réseau afin d'améliorer le fonctionnement et la résilience des réseaux fortement contraints. Nous y fournissons grâce à trois axes d'analyse différents : *performance*, *fiabilité* et *adaptabilité*, une réponse constructive à la question : *Quel type de codage pour quel réseau inter-machines ?* Nous montrons que même si le codage réseau constitue une alternative réelle et indéniable au modèle *store-and-forward*, son application à des réseaux aussi contraints que les réseaux de capteurs est délicate et nécessite des compromis qui en font en définitive un codage réseau *contraint*.

## 1.1. Contributions

Tous les systèmes décrits dans cette thèse font usage du codage réseau comme modèle alternatif d'acheminement de l'information dans un réseau de capteurs donné. Les deux premières solutions que nous proposons tendent à montrer l'impact en termes de performances de l'utilisation du codage réseau qu'il soit opportuniste ou linéaire sur différents types de réseaux de capteurs. Dans une seconde partie nous définissons la notion de *codage fiable*, un prérequis indispensable pour la réalisation d'algorithmes de codage qui garantissent des communications inter-machines<sup>1</sup> de bout-en-bout robustes. Enfin, dans la troisième partie de ce manuscrit, nous traitons de l'adaptabilité technologique et théorique des mécanismes de codage aux réseaux de capteurs sans fil. Nous introduisons *InC*, un mécanisme de codage contraint qui profite des propriétés intrinsèques des réseaux inter-machines afin de garantir une collecte efficace et fiable de l'information de plusieurs sources vers une seule destination.

### 1.1.1. Codage réseau inter-machines : performances

Dans cette partie, nous tenons à montrer d'abord l'impact prospectif du codage réseau sur différents types de réseaux de capteurs. Il s'agit d'identifier comment les codes linéaires et opportunistes améliorent des critères de performances clés à ces réseaux tels que le débit, le délai et le temps de convergence. Nous proposons trois protocoles d'acheminement basés, l'un (WSC) sur le codage aléatoire linéaire et les autres (CoZi et CNC) sur un codage réseau opportuniste. Contrairement aux mécanismes de codage actuels, les deux derniers codes que nous présentons sont conçus pour exploiter des propriétés spécifiques aux réseaux de capteurs telles que la topologie, la nature du trafic et l'application du réseau. Nous verrons que bien que le codage réseau linéaire assure des performances supérieures à celle du codage opportuniste dans le cas de la dissémination de données, il n'en reste pas moins un mécanisme coûteux, qui ne peut être appliqué que pour une certaine catégorie de réseaux inter-machines avec des capacités supérieures à la normale. Dans le cas de réseaux plus contraints, nous proposons d'utiliser un codage opportuniste qui se base sur la topologie hiérarchique des réseaux de capteurs et qui permet d'améliorer significativement le débit et le délai en comparaison à un modèle d'acheminement classique.

### 1.1.2. Codage réseau fiable : robustesse

Alors que l'étude entreprise en première partie permet de définir différentes stratégies de codage selon le type de réseau inter-machines et ses applications, les performances constatées résultent d'une évaluation dans un environnement idéal dépourvu de pertes. Dans un second travail, nous proposons d'étudier l'impact des pertes de paquets sur l'efficacité des mécanismes de codage réseau. Nous démontrons que les critères de fiabilité usuellement utilisés pour les algorithmes basés sur le *store-and-forward* ne sont pas

---

<sup>1</sup> Dans la suite du document, par réseau *inter-machines* nous ferons implicitement référence au cas général des réseaux de capteurs libérés d'une de leur contraintes majeures.



suffisants pour évaluer des protocoles de codage réseau. En effet, la fiabilisation de flux codés ne répond pas uniquement aux métriques de robustesse telles que le délai et le taux d'acheminement des paquets. Nous complétons donc ces métriques en introduisant la notion de codage fiable : un code est dit fiable s'il permet de coder des paquets de manière suffisamment efficace pour que le maximum de ses destinataires puissent en extraire de l'information utile, quelque soient les conditions du réseau (erreurs, collisions, pannes, etc.).

A partir de cette définition, nous proposons un schéma de retransmissions opportunistes appelé ReCoZi qui permet l'acquittement non pas de paquets reçus, mais de paquets effectivement décodés et dont de l'information utile a pu être extraite. Nous évaluons cette méthode de fiabilisation de flux codés par simulation et montrons que grâce aux nouvelles métriques de codage fiable, il est facile d'évaluer et concevoir des mécanismes de codage robustes qui garantissent une amélioration des performances même lorsque le réseau subit des taux de pertes élevés.

### ***1.1.3. Codage réseau contraint : adaptabilité***

De récents travaux théoriques ont montré la possibilité d'atteindre le débit théorique maximal en utilisant des mécanismes de codage dans les réseaux dits en *incast*. Dans ces réseaux, les communications se font à partir de plusieurs sources vers une seule destination. Il s'agit ici du schéma de trafic le plus commun dans les réseaux de capteurs. Cependant, la conception d'un tel mécanisme de codage n'a jamais pu être constatée dans la littérature. On peut penser intuitivement qu'une adaptation directe des mécanismes de codage existants pourrait résoudre le problème, si ce n'est que la plupart des propositions existantes sont basés sur l'écoute passive de paquets. Ce qui implique qu'elles ne sont pas applicables aux réseaux contraints pour des raisons de conservation d'énergie et de mémoire. De plus il est courant de constater que les algorithmes de codage existants requièrent une surcharge réseau, certes, négligeable dans le cas de réseaux sans fil à haut débit, mais qui s'avère fortement contraignante dans les réseaux de capteurs avec une capacité intrinsèque très réduite.

Dans ce dernier travail, et après avoir signifié l'apport indéniable du codage réseau aux architectures inter-machines en général, nous nous intéressons à son applicabilité dans le cas de réseaux contraints avec très peu de ressources par nœud. Nous montrons que l'amélioration du débit et des délais de bout-en-bout ne peut se faire que via l'utilisation d'une version simplifiée et moins coûteuse du codage réseau opportuniste. Ainsi, nous développons l'idée de codage par index (InC), un code contraint entièrement distribué où les symboles sont codés à la volée et décodés par la destination en une seule réception. Nous apportons une étude théorique sur l'efficacité du code avec pour métriques, le délai, le débit et la surcharge induite par InC. Puis nous évaluons ses performances et son impact sur les ressources grâce une implémentation sur une pile protocolaire IEEE 802.15.4 avec des topologies réalistes et de vrais capteurs sans fil.

## 1.2. Organisation du document

Les chapitres de cette thèse sont organisés dans une logique didactique traitant l'un après l'autre de la performance, de la robustesse et de l'adaptabilité des mécanismes de codage réseau dans les réseaux inter-machines.

Nous commençons par une revue des réseaux inter-machines et des différents efforts de la communauté scientifique (y compris un état de l'art complet de ceux usant du codage réseau) pour y améliorer les performances et la fiabilité.

Dans le chapitre 3, nous décrivons deux solutions différentes basées sur des algorithmes spécifiques issus de la théorie du codage réseau afin d'améliorer les performances de différentes applications des réseaux de capteurs sans fil.

Le chapitre 4 est consacré à ce que l'on a défini comme le codage réseau fiable. Dans un premier temps, une étude de l'impact du taux de perte sur l'efficacité du codage réseau y est détaillée. Puis, nous définissons de nouvelles métriques d'évaluation de la fiabilité des réseaux codés, lesquelles sont utilisées pour améliorer la robustesse de mécanismes de codage opportuniste.

Au chapitre 5, nous présentons InC, un code qui allie des concepts de codage réseau et de gestion des ressources. Nous détaillons les principes de cette nouvelle technique de codage contraint, et étudions ses propriétés théoriques. Nous aborderons ensuite son application dans un réseau de capteurs réels, et montrons qu'il permet d'améliorer sensiblement les performances des réseaux *incast* tout en offrant une consommation de ressources très faible comparée aux mécanismes en *store-and-forward*.

Le dernier chapitre conclut ce travail et introduit des idées et perspectives nouvelles qui pourront succéder à cette thèse.



## Chapitre 2

# 2. Etat de l'art

Dans ce chapitre, comme à travers toute la thèse, nous nous intéressons à l'amélioration des mécanismes d'acheminement de l'information dans les réseaux sans fil inter-machines en termes de performance et de fiabilité. Nous passons en revue les solutions existantes et montrons que de par leur utilisation du *store-and-forward*, elles conduisent à une sous-exploitation de la bande passante disponible dans le réseau. Nous étudions également les principales solutions basées sur le codage réseau et montrons leurs limites.

La section 2.1 pose les bases nécessaires à la compréhension des spécificités des réseaux de capteurs sans fil, que ce soit du point de vue architectural, applicatif ou réseau. Elle rappelle les types de trafic, méthodes d'acheminement et topologies propres à ces réseaux, et liste quelques travaux pionniers dans l'amélioration de leurs performances et leur fiabilité. La section 2.2 traite des fondamentaux du codage réseau sans fil et de ses applications. Elle passe en revue les mécanismes de codage réseau majeurs pour les réseaux sans fil en général ainsi que certaines de ses applications aux réseaux de capteurs sans fil.

### *2.1. Réseaux de capteurs sans fil*

La principale tâche d'un réseau inter-machines est de permettre l'échange d'information entre des entités autonomes et automatisées telles que des capteurs, des effecteurs ou des robots. Le principe étant de collecter de l'information de différentes sources pour l'acheminer vers une ou plusieurs destinations. De manière générale, ces réseaux sont auto-organisés et ne nécessitent pas d'infrastructure fixe. Ils utilisent des algorithmes qui permettent de maintenir une connectivité maillée ou hiérarchique afin d'acheminer des messages en multi-sauts qui transportent des grandeurs physiques caractérisant une zone géographique donnée (température, pression, etc.) L'attrait majeur de ce type de réseau étant la possibilité de création de services robustes, contextualisés et facilement maintenables en un minimum de temps et de coûts de déploiement. Durant ces dernières années, un grand nombre d'implémentations expérimentales et commerciales de ces réseaux ont vu le jour [1], [2] avec l'objectif d'offrir un accès bon marché à des applications jusqu'alors inaccessibles car irréalisables avec des technologies d'infrastructure. Ces applications sont amenées à se généraliser dans l'avenir, promettant des services ubiquitaires que ce soit pour l'industrie automatisée, la gestion des catastrophes, la domotique ou les villes intelligentes.

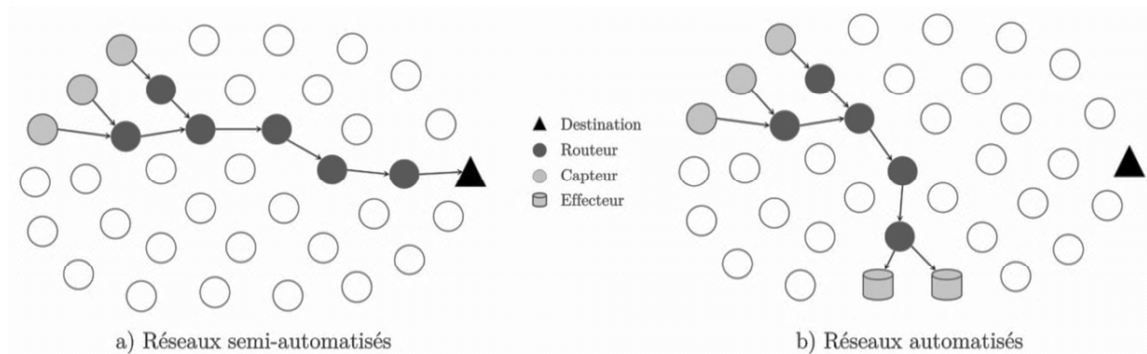
Minimiser la taille, le nombre des messages générés par le réseau et l'énergie nécessaire pour les transmettre, tout en garantissant des performances décentes, est la problématique majeure traitée dans la littérature des réseaux inter-machines [3] et de capteurs sans fil [4]. Depuis presque 30 ans, des solutions à ces problèmes sont proposées, que ce soit par des efforts d'optimisation matérielle ou logicielle [5]. D'autres approches plus théoriques se focalisent sur des améliorations protocolaires optimisant les algorithmes de routage [6], les méthodes d'accès au canal [7] ou encore les mécanismes de fiabilisation du réseau [8].

**Analogie avec les réseaux ad hoc** : de la même façon que pour les réseaux ad hoc sans fil, le déploiement d'un réseau de capteurs s'affranchit par définition de l'utilisation d'infrastructures fixes. Les messages y sont délivrés en utilisant un routage de proche-en-proche et des retransmissions en cas de pertes. Comparés à des réseaux d'infrastructure, ces systèmes peuvent offrir la même couverture réseau en transmettant moins de messages et en garantissant des coûts de déploiement beaucoup plus faibles [9]. Cependant, bien que conceptuellement les réseaux de capteurs puissent être considérés comme des réseaux ad hoc, ils n'en demeurent pas moins un cas très particulier. Le routage y est fortement lié à l'application du réseau et au type de nœuds qui le composent. La nature du trafic et les spécificités des nœuds en font un routage directionnel basé sur des critères de performances tels que l'énergie, la complexité et la tolérance aux pannes. De plus, les communications dans un réseau de capteurs servent à la collecte ou la dissémination de l'information et peuvent utiliser des mécanismes d'agrégation ou de compression des données [10].

### 2.1.1. Applications et architectures

D'un point de vue applicatif, les réseaux de capteurs peuvent être classés en deux types d'architectures génériques[3] :

- a. **Architecture semi-automatisée.** Elle correspond à l'utilisation originelle des réseaux de capteurs (communément appelée WSN). Elle concerne des applications où l'information est collectée puis envoyée vers une passerelle (cf. Figure 1.a) afin de proposer des services de supervision dans un environnement quelconque. Dans ce cas, les communications sont initiées par les nœuds qui collectent les données à la demande ou non du destinataire qui s'occupe du traitement de celles-ci. Utilisés d'abord dans les systèmes de défense, ces services se sont étendus à des usages aussi variés que la supervision d'infrastructure, la télésurveillance ou la protection de l'environnement.
- b. **Architecture automatisée.** représente une nouvelle gamme d'applications des réseaux de capteurs (appelée WSAN) apparue grâce à l'émergence du concept d'actuateur ou effecteur. Ce dernier est un composant matériel (ex. moteurs, robots, interrupteur, etc.) disposant des mêmes caractéristiques de transmission qu'un capteur. Ce type de réseau diffère des architectures classiques de collecte ou de supervision. Ici, le système est capable d'*agir* sur son environnement en faisant coexister et interagir des capteurs et des effecteurs dans le but d'automatiser une action donnée (cf. Figure 1.b).



**Figure 1.** Classes d'architectures des réseaux de capteurs sans fil

La prise de décision basée sur le « *quand* » et le « *où* » accomplir ces actions est définie par les informations collectées sur les capteurs. Ces réseaux sont évidemment utiles dans des applications industrielles telles que les chaînes de productions et les chantiers, mais aussi en automatisation d'infrastructure (ex. domotique, bâtiments intelligents, etc.). Considérées aussi comme la pierre angulaire de l'internet des objets [11], ces architectures promettent une large expansion dans un futur proche comme la littérature prolifique sur le sujet tend à le démontrer.

D'autres classifications plus détaillées ont pu être proposées dans la littérature comme [12] qui prend en compte la mobilité des nœuds ou [13] qui se base sur la densité du réseau. Cependant, afin de garder une certaine genericité, nous considérons les architectures décrites ci-dessus, et traiterons les cas moins génériques comme des cas particuliers.

### 2.1.2. Caractéristiques réseaux

Dans un réseau de capteurs, les nœuds sont équipés de modules de mesure (température, vitesse, pression, etc.), d'un microcontrôleur, d'un système d'alimentation autonome et d'un module de communication RF à faible consommation. Ces nœuds sont pour la plupart caractérisés par des limitations fortes en termes de capacités de calcul, de mémoire et d'énergie. Ce sont ces spécificités qui font des réseaux de capteurs sans fil des systèmes à débit relativement faible ( $\sim 100\text{kb/s}$ ) comparés aux réseaux ad hoc sans fil par exemple.

De plus, les réseaux de capteurs sans fil sont généralement basés sur des méthodes d'accès au canal peu gourmandes en contrôle, telles que CSMA/CA, et des algorithmes de routage réactifs comme AODV [14] ou le routage hiérarchique. Les mécanismes de fiabilisation de flux de proche-en-proche (ex. les messages RTS/CTS) et de bout-en-bout (ex, TCP) y sont évités parce que bien trop pénalisants pour le réseau.

Tout cela fait que la qualité et le coût énergétique des communications sans fil dans ces réseaux dépendent non seulement de leur environnement physique, mais aussi du type de matériel ainsi que de la bande de fréquence et du type de modulation utilisée [1]. Ainsi, lors d'une communication sans fil de bout-en-bout, un acheminement en multi-saut peut parfois s'avérer moins coûteux en termes d'énergie et de qualité de service qu'un acheminement à un seul saut.

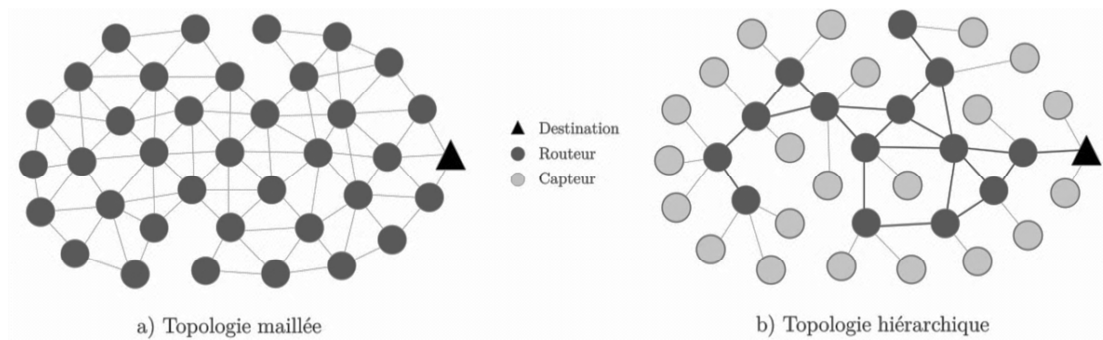


Figure 2. Exemple de topologies des réseaux de capteurs sans fil

### 2.1.3. Nature du trafic

Il existe deux types de trafic dans les réseaux de capteurs qui dépendent de l'application du réseau. Chacun définit la manière dont l'information est générée à la source et influe sur le routage utilisé pour la transmettre à sa destination.

#### 2.1.3.1. Trafic périodique

Dans ces réseaux, les nœuds chargés de la collecte des données transmettent des messages à intervalles réguliers vers leur(s) destinataire(s). Chaque nœud génère donc des flux intermittents de messages qui lui permettent de passer périodiquement en mode sommeil afin de sauvegarder son énergie [21]. Utilisés dans des applications de supervision, les réseaux périodiques permettent, sous réserve d'un échantillonnage adapté, de rassembler suffisamment d'informations sur une zone donnée pour en déduire les tendances et les comportements. Toutefois, la synchronisation des nœuds nécessaire à ces trafics implique l'échange de paquets de contrôle qui peut engendrer une surcharge réseau non-négligeable.

#### 2.1.3.2. Trafic apériodique

Dans un trafic apériodique, la transmission de l'information à partir des nœuds sources peut être déclenchée de deux façons :

- a) *Transmission par évènement (event-driven)* : ici, une source ne transmet des données que lorsqu'un évènement est détecté dans ses environs. Un évènement est décrit comme un changement significatif de la valeur mesurée par le capteur dans le temps. Utilisés dans des applications de surveillance (ex. détection d'incendie et d'intrusion), ce modèle diffère du modèle synchrone parce que le trafic n'y est ni constant, ni prévisible, mais défini par l'apparition aléatoire des évènements et leur distribution géographique dans le réseau.
- b) *Transmission par requêtes (on demand)* : dans ce cas, les capteurs ne transmettent de l'information que lorsqu'un autre nœud du réseau (potentiellement la passerelle) en fait explicitement la requête via un message d'intérêt. Ce modèle en requête/réponse est pratique pour un trafic sporadique, toutefois, quand le nombre de requêtes augmente la surcharge induite par celles-ci peut engendrer une dégradation notable des performances du réseau.

Dans tous ces cas, l'acheminement de l'information dans le réseau est considéré comme un problème plus complexe que dans les autres types de réseau sans fil (ad hoc, ou cellulaires) à cause des caractéristiques inhérentes des capteurs [6].

#### **2.1.4. Topologies et méthodes d'adressage**

Les topologies des réseaux de capteurs sont maillées ou hiérarchiques [2] (cf. Figure 2). La connectivité combinée à l'adressage des nœuds permettent d'établir simplement des routes au-dessus de ces topologies afin de transmettre des messages de proche-en-proche des capteurs sources vers leurs destinataires. On peut distinguer deux approches qui permettent de construire et de maintenir un réseau de capteurs quel que soit sa topologie sous-jacente. Premièrement, via un adressage global des nœuds tel que celui proposé dans [15], où à chaque nœuds est attribuée une adresse unique qui permet de l'atteindre en utilisant un algorithme de routage donné tel que ZigBee [15] ou AODV [14]. Ou deuxièmement, via une approche sans adresse où les données sont collectées grâce à un système d'annonces qui permet aux nœuds intéressés par l'information d'en faire la requête en inondant partiellement ou complètement le réseau [16]. Cette approche est dite *data-centric* car les communications y sont déclenchées par la disponibilité de l'information et non pas par les nœuds eux-mêmes.

Dans la pratique, et bien que des cas de topologies complètement maillées puissent être constatées, la plupart des protocoles standards existants comme [15] ou [17] reposent sur une topologie et un adressage global hiérarchiques. Et ce pour simplifier les mécanismes d'acheminement de l'information en réduisant la taille des tables de routage [18] et en minimisant la consommation d'énergie [18]. Pour preuve, les deux standards les plus répandus dans les réseaux inter-machines reposent sur une topologie hiérarchique et utilisent des schémas d'adressage globaux. En l'occurrence, ZigBee [15] utilise des adresses globales hiérarchiques et 6lowPan [17] une version compressée d'IPv6.

#### **2.1.5. Routage et acheminement**

Dans leur étude quasi-exhaustive [6], Akkaya *et al.* classent, évaluent et discutent les algorithmes de routage existants pour les réseaux de capteurs sans fil et définissent une catégorisation selon leur structure. D'après les auteurs il existe trois types de routage : 1) routage à plat (flat routing), 2) routage géographique et 3) routage hiérarchique.

Dans les algorithmes de routage à plat, les nœuds du réseau sont homogènes et jouent le rôle de routeur lorsque nécessaire. Les paquets y sont transmis en saut-par-saut dans une topologie maillée et ce grâce à des tables de routage construites proactivement ou réactivement selon les besoins du réseau [22]. Le routage géographique [23] quant à lui se base sur la localisation relative ou absolue des nœuds afin de trouver une route vers le(s) destinataire(s). Il s'agit d'utiliser les coordonnées du nœud source et celles de ses voisins afin de transmettre ou de retransmettre un paquet au plus près de sa destination. Même si les mécanismes de localisation impliquent encore un certain coût et complexité pour des réseaux aussi contraints que les réseaux de capteurs, le routage géographique est



particulièrement intéressant dans un contexte de réseaux dynamiques comprenant un très grand nombre de nœuds. Enfin, les algorithmes de routage hiérarchique se basent sur une organisation en arbre ou sur un partitionnement (*clustering*) du réseau en un ensemble de groupes disjoints afin de garantir le passage à l'échelle, la connexité et la prolongation de la durée de vie du réseau. Les communications entre ces différents groupes y sont assurées grâce à un cœur de réseau constitué exclusivement par des *super-nœuds* (*cluster-head* ou nœuds parents) désignés pour prendre en charge le routage intra- et extra-cluster. Contrairement au routage à plat, les protocoles de routage hiérarchique se basent sur des mécanismes très simples tels que la décomposition d'adresse [24] ou le routage pondéré [25] pour acheminer chaque paquet vers sa destination.

### 2.1.6. Distribution de l'information

Dans la mesure où nous nous intéressons d'avantage à la « paquetisation » et à gestion des flux de données routées plutôt qu'aux types et algorithmes de routage, nous ne nous étendrons pas sur le sujet plus que nécessaire. Que ce soit dans un routage à plat, hiérarchique ou géographique, notre objectif est de montrer la sous-optimalité des réseaux en *store-and-forward* en général. Ainsi, plutôt que de passer en revue les mécanismes de routage et d'acheminement par paquets dans les réseaux de capteurs, nous proposons une classification selon le type de distribution constaté dans ces réseaux. Cette classification est suffisante dans notre étude afin d'exposer d'une façon générique les problèmes que nous souhaitons traiter. Nous nous intéressons donc aux trois modèles de distribution de paquets les plus probants [2] dans les réseaux de capteurs sans fil.

#### 2.1.6.1. Distribution point-à-point (*One-to-One*) – *Unicast*

Les requêtes en point-à-point (*unicast*) sont fréquemment utilisées dans les réseaux de capteurs sans fil pour la configuration et le contrôle des nœuds. Elles permettent d'établir des communications en requête/réponse afin de paramétrer à partir d'un nœud source le comportement ou des propriétés d'un nœud précis du réseau (allocation d'adresse, réinitialisation, etc.). Dans un cas plus général, l'*unicast* est souvent utilisé dans les réseaux inter-machines pour établir des relations *observateur-observé* entre un capteur et un effecteur. Ici, la détection d'un évènement par exemple, peut déclencher via une transmission point-à-point une action d'un effecteur à l'autre bout du réseau. Le problème de l'*unicast* est largement traité dans la littérature, plusieurs protocoles ont été proposés pour y améliorer le routage [6] et la fiabilité [8].

#### 2.1.6.2. Dissémination (*One-to-Many*) – *Multicast/Broadcast*

Envoyer de l'information d'une source vers l'ensemble des nœuds du réseau est une pratique courante dans certaines applications des réseaux de capteurs sans fil. Utilisée pour la reprogrammation des nœuds [26] et dans les mécanismes de découverte de route [27] ou de service [28]. Elle génère des problématiques de fiabilité de par son utilisation intensive des canaux de communication. En effet, lors de la dissémination de flux de données, le médium sans fil subit des collisions et interférences (tempêtes de *broadcast*) qui peuvent

altérer la bonne réception des paquets et qui nécessitent l'utilisation de mécanismes spécifiques d'acheminement et de retransmission. Il existe plusieurs méthodes pour pallier ces problèmes en préservant les performances du réseau, que ce soit via la dissémination hiérarchique [29], la diffusion sélective [30] ou la propagation épidémique [31].

### 2.1.6.3. *Distribution centralisée (Many-to-One) – Incast*

La collecte de l'information à partir d'un ensemble de capteurs sources vers une ou plusieurs destinations représente le modèle de distribution le plus répandu dans les réseaux de capteurs. Dans ce cas, l'information est acheminée par un routage dit en *incast*. Ce modèle d'acheminement est utilisé pour la collecte d'information dans des applications de surveillance et de supervision d'infrastructures ou d'environnements donnés. La littérature fournit une batterie de solutions aux problèmes inhérents aux réseaux incast, à savoir l'utilisation optimale de la bande passante [32], la réduction de la consommation d'énergie [33], l'agrégation des données corrélées [34], etc.

### 2.1.7. *Performance et fiabilité*

L'étude de performance et de la fiabilité est au cœur de la majorité des travaux de recherche dans les réseaux de capteurs sans fil. Selon les contraintes imposées par l'application (réactivité, économie d'énergie, robustesse, etc.) et l'environnement du réseau (pertes, interférences, densité, mobilité, etc.), différentes approches peuvent être utilisées afin d'optimiser des métriques de performances telles que le débit ou le délai. Et cela tout en garantissant plus ou moins de robustesse au réseau. Comme les capteurs et le medium sans fil imposent des limitations importantes discutées plus haut, les solutions proposées dans la littérature sont toujours orientées vers un compromis entre les performances et la fiabilité. De plus, les différentes implémentations d'avantage été conçues pour tolérer ces contraintes plutôt qu'à les gérer. Dans la plupart des cas, cette approche par compromis s'avère quand même productive et permet d'améliorer les protocoles de communications.

Par exemple, les protocoles de routage réactifs privilégient la réduction de la surcharge réseau en n'établissant des routes que lorsqu'un nœud en fait explicitement la demande (ex. AODV [14]). Cette approche est efficace dans un réseau statique mais peut dégrader les performances et la fiabilité du réseau si ce dernier subit une connectivité trop intermittente. Alors qu'à l'inverse, les protocoles proactifs résolvent le problème de la dynamique du réseau en établissant des routes qui sont mises-à-jour en fonction de l'état des liens du réseau (ex. OLSR [35]). Evidemment, cette approche nécessite une signalisation intensive qui implique une plus grande charge dans le réseau. Par ailleurs, en terme de fiabilité, lorsqu'une application n'a pas de contraintes de délai par exemple, les pertes peuvent être récupérées par des mécanismes de fiabilisation (ex. ARQ [36], etc.) Cette approche garantie plus de robustesse en relaxant les besoins en latence du réseau.

Il est aussi commun dans des applications de capture scalaire d'utiliser des mécanismes d'agrégation ou de compression de l'information afin d'optimiser les communications en réduisant la charge du réseau et la consommation globale d'énergie. Ces systèmes sont très

efficaces mais s'intéressent uniquement à des cas particuliers où les données collectées sont fortement corrélées ce qui n'est pas le cas dans tous les réseaux de capteurs sans fil.

Aujourd'hui, grâce à l'émergence de la théorie du codage réseau, des solutions sont proposées afin d'éviter certains de ces compromis. En effet, des applications de cette théorie [37] permettent d'améliorer à la fois la performance et la fiabilité des réseaux sans fil justement en exploitant les contraintes qui leurs sont propres telles que la nature partagée du médium, les réceptions redondantes de paquets, la diversité spatiale, ou encore dans certains cas les interférences.

### ***2.1.8. Digression sur les efforts de standardisation***

Nous pensons qu'il est important d'aborder le fossé architectural, technologique et conceptuel qui s'est installé entre les solutions académiques et les standards développées par l'industrie et les instances de normalisation. Alors que la littérature académique traite majoritairement du problème des réseaux de capteurs en tant que réseau maillé, force est de constater que les applications réelles des réseaux de capteurs sont plutôt réalisées via des topologies hiérarchiques. En effet, les deux grands standards qui tendent à s'imposer dans le marché des réseaux de capteurs et effecteurs sans fil se basent sur une topologie en 2-tiers où une fraction des nœuds du réseau joue le rôle de routeurs pendant que l'autre ne s'occupe que de la collecte des données. Dans leur étude de référence [2] Nayak *et al.* indiquent qu'un adressage global tel que IP n'est pas « possible » dans un réseaux de capteurs sans fil. Hors, aujourd'hui, les réseaux 6lowPan basés sur un adressage IPv6 démontrent le contraire et réfutent l'idée ancrée dans beaucoup de travaux académiques, stipulant que le nombre potentiellement élevé des nœuds dans un réseau de capteurs entraîne inadaptabilité des protocoles basés sur l'adressage internet.

Dans ce travail, nous pensons que de la même manière que les réseaux locaux sans fil ont quasiment tous convergé vers un seul standard (i.e., IEEE 802.11 [38]), les réseaux de capteurs vont faire de même en adoptant les deux normes : ZigBee de la ZigBee Alliance [39] et 6lowPan de l'IETF [40]. La première pour des applications industrielles tandis que la seconde pour des applications plutôt grand public. Nous prenons donc dans notre méthodologie d'évaluation des performances le parti pris d'une topologie basée sur la couche physique IEEE 802.15.4 qui représente la méthode d'accès au médium des deux protocoles cités.

## ***2.2. Codage réseau et réseaux sans fil***

Les travaux de Shannon [41] sur la théorie de l'information ont montré qu'il est possible d'approcher la capacité d'un canal de communication tout en garantissant une communication raisonnablement fiable (taux d'erreurs relativement faible). Nous savons aussi grâce au travail d'Ahlswede [42] qu'il est désormais possible via un codage réseau approprié, de transmettre de l'information dans un réseau de communication quelconque tout en atteignant les limites théorique de sa capacité. Pourtant, d'importants challenges subsistent afin d'arriver à une utilisation généralisée du codage réseau : la quantification

des coûts d'exploitation, la formalisation des stratégies de codage et bien sûr la question persistante quant à savoir « *où, quand et comment coder ?* » constituent les derniers problèmes à résoudre avant l'éventuelle popularisation du codage réseau dans les systèmes de communications à large échelle. Des réponses constructives ont été proposées au fil des années pour trouver des solutions à ces problématiques, à la fois par des travaux analytiques et empiriques. En conséquence, plusieurs variantes de l'application originelle [43] du codage réseau ont été développées, et une large littérature a proliféré sur le sujet. Malgré cela, nous arguons qu'il est bien trop prématuré pour affirmer que toutes les optimisations et améliorations de la théorie sont accomplies notamment dans les réseaux de capteurs sans fil, où les techniques de codage réseau n'en sont encore qu'à leurs balbutiements. Dans cette section nous nous intéresserons donc aux méthodes de codage réseau pour les réseaux sans fil en général et pour les réseaux de capteurs en particuliers.

### **2.2.1. Fondamentaux du codage réseau**

Il y a dix ans, la théorie du codage réseau [44] donnait naissance à un nouveau champ d'investigation dans le domaine de la gestion de flux des réseaux de communications par paquets. Les travaux pionniers [45] qui décrivent cette approche inédite de traitement de l'information ont dévoilé le potentiel de ce que nous désignons dans ce manuscrit comme les réseaux en *code-and-forward*. Ils détaillent comment les propriétés théoriques du codage réseau révolutionnent les modèles classiques d'acheminement des données dans les systèmes de communication contemporains.

La formulation initiale de la théorie donnée par Alswhede *et al.* dans l'article de référence : « *Network Information Flow* » [42] définit le codage réseau comme la seule méthode générique qui permet d'utiliser toute la capacité théorique d'un réseau multicast. Les auteurs prouvent que grâce à un codage adapté, il est possible d'étendre l'utilisation de la bande passante d'un réseau multicast quelconque en utilisant des liens que les méthodes d'acheminement classique en *store-and-forward* n'exploitent pas. S'en est suivie une série de travaux de recherche qui illustrent les améliorations significatives en termes de performances et de fiabilité que promettent les codes linéaires [46] et opportunistes [47]. Ces derniers ont été appliqués aux architectures sans fil, démontrant une efficacité sans précédent en termes de débit [48], de sécurité [49], et de tolérance aux pannes [50]. En effet, grâce notamment aux contributions de Katti *et al.* le codage réseau a évolué d'une solution lourde et complexe dédiée uniquement aux systèmes multicast filaires vers une technique générique, simple et qui fait un usage opportuniste de combinaisons linéaires de messages afin d'améliorer les performances des réseaux ad hoc sans fil pour les flux unicast [51], multicast [54] et broadcast [53].

A ce jour, la formalisation des gains du codage réseau sans fil pour des réseaux arbitraires ne serait-ce qu'en termes de débit n'est toujours pas établie. En effet, l'étude des *patterns* de trafic génériques est complexe car les solutions optimales requièrent des tailles de symboles exponentielles et des opérations non-linéaires [54]. Malgré cela, de plus en plus de chercheurs de domaines très variés s'intéressent aux techniques de codage réseau sans fil, à leurs bénéfices, et à leurs méthodes d'implémentation.

### 2.2.2. Codage réseau linéaire

Basé sur le théorème initial de la théorie du codage réseau pour les systèmes multicast, ce code permet aux nœuds destinataires d'exploiter les chemins qui se chevauchent dans un arbre multicast afin d'augmenter l'utilisation de la bande passante du réseau. Il suffit ici de permettre aux nœuds intermédiaires chargés d'acheminer les messages d'effectuer des opérations linéaires, à savoir, des additions et multiplications dans un corps fini, pour que ce débit optimal soit atteignable. Dans la version distribuée du codage réseau linéaire souvent appelée codage réseau *aléatoire*, les nœuds intermédiaires sur la route d'un message combinent linéairement et d'une manière autonome les messages qu'ils reçoivent en leurs affectant des coefficients aléatoires sélectionnés dans un corps fini  $\mathbb{F}_q$ . [55] montre que cela suffit à garantir une probabilité  $P$  de décodage proche de 1 avec un corps fini suffisamment grand ( $P = 1 - |\mathbb{F}_q|^{-k}$ , où  $|\mathbb{F}_q|$  est la cardinalité du corps fini des coefficients).

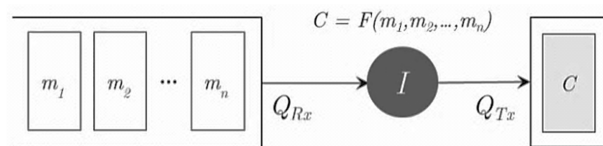


Figure 3. Exemple de codage linéaire

**Exemple : Avalanche.** Le système Avalanche [56] de distribution de fichier à large échelle en pair-à-pair développé dans les laboratoires de Microsoft Research fut parmi les premières applications réelles des techniques de codage linéaire aux systèmes de communications à large échelle. Dans cette solution illustrée sur la Figure 3, au lieu de distribuer des fragments d'un fichier  $m_i$  comme dans n'importe quel système pair-à-pair, chaque nœud produit et transmet une combinaison linéaire  $C=F(m_i)$  des fragments qu'il détient. Lorsqu'un pair reçoit suffisamment de combinaisons de fragments linéairement indépendantes  $C_j$ , il peut les décoder en résolvant le système d'équations constitué par ces combinaisons et ainsi reconstruire le fichier en question. Grâce à cette implémentation, les nœuds du réseau ont la garantie d'acquérir les fragments nécessaire à la reconstruction des fichiers beaucoup plus rapidement que dans une architecture pair-à-pair dépourvue de codage réseau.

**Limites.** Les codes linéaires aléatoires impliquent une amélioration nette des performances réseaux dans le contexte d'applications multicast ou de dissémination des données. Cependant, l'un des problèmes majeurs du codage linéaire qu'il soit simple ou aléatoire réside dans son inadaptabilité aux communications autres que celles à destinations multiples. En effet, les réseaux multicast et broadcast ne représentent qu'une infime partie du trafic d'internet par exemple et sans une adaptation aux trafics point-à-point cette solution ne peut concerner que des applications spécifiques. Autre contrainte que subissent ces architectures est leur faible tolérance aux pannes [57] et leur vulnérabilité aux attaques malveillantes [58]. Que ce soit des liens défectueux, une taille de corps insuffisante ou des nœuds défaillants ou malveillants, la propagation des erreurs dans un réseau codé peut corrompre tout un flux de données lors de la résolution du système linéaire. De récents

travaux [59] traitent ce problème via la combinaison de techniques classiques de détection d'erreur à un saut avec des codes à effacement [60].

### 2.2.3. Codage réseau opportuniste

Contrairement au codage linéaire, le codage opportuniste permet de traiter le problème des communications point-à-point en utilisant un codage à un saut. Ici, chaque routeur qui doit transmettre un message reçu vers une destination le combine avec d'autres messages présents dans sa file d'attente afin d'augmenter leur probabilité de réception *prématurée* au niveau des nœuds voisins. Le routeur transmet donc une combinaison du message en tête de sa file d'attente et d'autres messages afin qu'ils soient décodés par des nœuds autres que la destination à un saut. Le codage opportuniste est souvent appelé codage à *un saut* parce qu'une transmission codée est considérée comme réussie uniquement si la destination à un saut arrive à décoder le message qui lui été adressé. Cette technique permet une amélioration notable du débit et de la latence dans des modèles de propagation de l'information dynamiques sujets aux congestions et aux pertes en rafales.

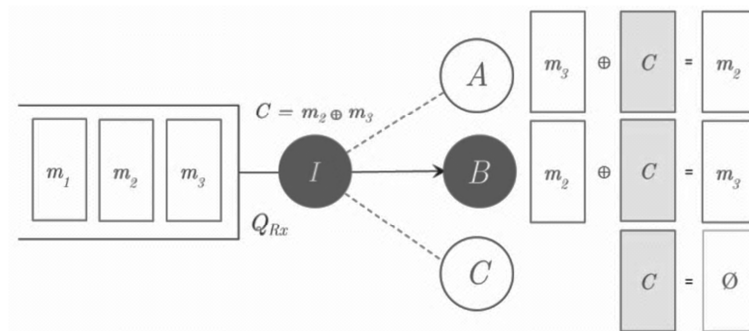


Figure 4. Exemple de codage opportuniste

**Exemple : COPE.** La plus célèbre des utilisations du codage réseau opportuniste a été proposée par Katti *et al.* dans leur article de référence [47] décrivant la première adaptation du codage réseau aux communications sans fil par paquets. Dans leur solution, les nœuds du réseau peuvent détecter les opportunités de codage lorsqu'elles se présentent et qu'elles peuvent apporter une amélioration dans l'exploitation de la bande passante. Comme illustré dans la Figure 4, lorsqu'un nœud intermédiaire  $I$  a un paquet  $m_3$  à transmettre vers  $B$ ,  $I$  peut combiner  $m_3$  avec d'autres paquets (ici  $m_2$ ) sélectionnés dans sa file d'attente, en utilisant un simple XOR. Le choix de ces paquets par  $I$  est défini parce que ses voisins ont déjà reçu ou intercepté<sup>2</sup> lors des dernières transmissions. Sachant que  $B$  a déjà reçu  $m_2$  et que  $A$  a déjà reçu  $m_3$ , il est donc plus efficace pour  $I$  d'envoyer  $m_2 \oplus m_3$  plutôt que d'envoyer simplement  $m_3$ . Ainsi, grâce au codage opportuniste  $A$  et  $B$  reçoivent deux paquets au lieu d'un seul, en une transmission. Il est clair que pour décoder la combinaison  $m_2 \oplus m_3$  il suffit de refaire un XOR entre le(s) paquet(s) déjà reçu et le paquet codé.

<sup>2</sup> Par le terme « intercepté » nous sous-entendons qu'un nœud a sauvegardé un paquet qui ne lui été pas explicitement adressé grâce à la nature partagée du médium sans fil. On parle alors d'*overhearing* en anglais.

**Avantages.** Théoriquement, le codage opportuniste peut réduire le nombre de transmission par deux [48]. Dans la pratique, une implémentation dans un réseau ad hoc sans fil [52] montre que ce facteur peut s'avérer bien plus important. Allant jusqu'à augmenter de l'ordre de 3 à 4 fois le débit du réseau. Cela est dû au fait que le codage a un saut permet de réduire la charge des files d'attente en drainant plus vite le surplus de paquets et évitant le rejet de paquets par des nœuds subissant un flux trop important de paquets. Les réseaux à codage opportuniste sont aussi plus tolérants aux erreurs et aux pannes [62]. La nécessité de pouvoir décoder les messages en un saut rend possible l'utilisation de systèmes de détection d'erreur en saut-par-saut, et permet ainsi de traiter les erreurs dans un message bien avant son arrivée à sa destination finale.

#### **2.2.4. Codage réseau et réseaux de capteurs**

Depuis COPE pour les réseaux sans fil et Avalanche pour les réseaux filaires, plusieurs protocoles de communication tels que CodeCast [63], Secure Network Coding [58] ou I2NC [51] ont fait usage des techniques de codage réseau pour le multicast et la dissémination des données. Aujourd'hui, alors que certaines applications de la théorie du codage réseau sont toujours en cours de développement, initiant régulièrement de nouvelles avancées dans le monde des systèmes informatiques, il existe des cas comme les réseaux multicast [63], les mécanismes de distribution de contenus [56] ou encore dans les réseaux ad hoc sans fil [64] où les techniques de codage réseau sont suffisamment abouties pour être légitimement considérées dans l'exploitation de vrais réseaux.

Dans leur analyse théorique des applications futures du codage réseau dans les systèmes de communication sans fil, les auteurs de [43] évoquent la possibilité d'utiliser le codage réseau pour le multicast, la dissémination et une plus large famille de problèmes incluant les réseaux de capteurs sans fil. Dès lors, plusieurs adaptations des solutions de codage existantes ont été étudiées afin d'améliorer les communications dans les architectures de diffusion de l'information des réseaux de capteurs sans fil. La littérature dans le sujet montre que les solutions proposées pour l'utilisation du codage réseau dans les réseaux de capteurs se basent sur des hypothèses parfois contestables : La première est la présence supposée permanente d'une connectivité totalement fiable et statique entre les nœuds du réseau permettant un codage efficace à chaque transmission, ce qui n'est bien entendu pas le cas des réseaux sans fil soumis au partage de canal et aux interférences. La deuxième consiste à considérer que l'énergie consommée lors de la réception d'un paquet est négligeable comparée à celle consommée pendant une transmission. Ceci n'est pas vérifié dans les capteurs existants, où le coût énergétique d'une transmission équivaut à celui d'une réception [65]. De plus, l'implémentation des solutions de codage est souvent évaluée en simulant des réseaux IEEE 802.11 où la taille des trames est beaucoup plus importante que celles présentes dans les réseaux de capteurs utilisant la couche MAC IEEE 802.15.4.

##### **2.2.4.1. Codage aléatoire pour les réseaux de capteurs**

Afin de prouver l'intérêt d'adopter le codage réseau pour améliorer la dissémination dans les réseaux de capteurs sans fil, [66] compare un protocole basé sur le codage aléatoire avec

un protocole à acheminement probabiliste [67]. Dans leur analyse, le codage réseau démontre une amélioration significative des performances du réseau notamment en termes de débit et de latence. Toutefois, les auteurs présument que la totalité des nœuds transmettent des paquets à tous les autres nœuds du réseau. Ce type d'application n'est pas le plus courant en termes de type de trafic, car ordinairement, la dissémination est déclenchée par un sous ensemble de capteurs (qui détectent un évènement par exemple) et non pas par l'intégralité du réseau. De plus, l'évaluation de la solution est faite sur une couche MAC IEEE 802.11 sans perte qui utilise un mécanisme d'évitement de collision parfait ce qui ne correspond pas à un environnement réaliste. Similairement, le codage linéaire aléatoire a été proposé dans [68] comme solution au problème de la dissémination dans les réseaux à faible débit. Ici, les auteurs proposent une approche inter-couches qui permet de garantir des retransmissions en point-à-point et de bout-en-bout lorsque le réseau subit des pertes non négligeables. Une fois de plus, l'utilisation du codage linéaire s'avère productive augmentant la durée de vie du réseau, tout en garantissant une diminution des transmissions en comparaison à un mécanisme en *store-and-forward*.

#### 2.2.4.2. Codage opportuniste dans les réseaux de capteurs

Bien que les communications point-a-point ne soient pas le type de trafic de prédilection dans les réseaux de capteurs, elles sont souvent utilisées dans des applications de contrôle ou de paramétrage des nœuds. Le codage réseau opportuniste se pose donc comme une méthode d'amélioration des performances au sein de ces architectures. Par exemple, [69] propose un mécanisme de codage opportuniste pour les réseaux de capteurs invoquant sa simplicité, sa décentralisation et son indépendance de la topologie du réseau. Cependant, leur solution requiert l'utilisation constante d'un mécanisme d'inférence de topologie. Or, rappelons que ce dernier nécessite l'écoute constante du medium afin d'augmenter la probabilité de décodage des paquets. Son utilisation peut donc, représenter une source de consommation supplémentaire des ressources énergétiques des nœuds. Dans un autre travail, [70] propose OSCOR un codage à la source combiné avec un code opportuniste afin d'améliorer la collecte d'information dans les réseaux de capteurs. Ici, les données collectées sont compressées à la source puis codées de manière opportuniste afin d'améliorer le débit global du réseau. Même si OSCOR montre des résultats très encourageants et illustre l'intérêt prospectif du codage opportuniste pour la collecte d'information, il n'en demeure pas moins que son évaluation n'a pu se faire que sur un réseau 802.11 où les contraintes liées aux réseaux de capteurs ne sont pas pris en compte.

#### 2.2.4.3. Codage réseau adaptif

Le codage réseau adaptif a été proposé dans [71] sous la forme d'AdapCode, un protocole de codage à état de liens pour la dissémination des données dans les réseaux de capteurs sans fil. AdapCode se base sur un code similaire au codage aléatoire qui permet d'encoder des flux de paquets en utilisant des combinaisons linéaires et de les décoder grâce à des éliminations Gaussiennes, à la différence près qu'il change son mécanisme d'ordonnement de paquet progressivement en s'adaptant aux propriétés du canal en temps réel. Ainsi, lorsque la qualité du lien est dégradée, le nombre de paquets  $N$  inclus dans les combinaisons linéaire est réduit afin d'éviter de générer des codes trop volumineux



plus facilement sujets à l'injection d'erreurs [60]. Afin d'être encore plus réactif, les concepteurs de la solution propose d'adapter aussi  $N$  au nombre de voisins à un saut détecté par le nœud codeur. Cette technique permet d'augmenter significativement la probabilité de décodage d'un paquet dans un environnement perturbé. Même si AdapCode offre des améliorations significatives en termes de débit et de fiabilité, ces résultats ne concernent que des applications de dissémination des données. Par ailleurs, l'étude et l'analyse du protocole ne comprennent aucune référence à des critères de performances clés sur les réseaux de capteurs sans fil tels que la consommation d'énergie ou le délai.

#### *2.2.4.4. Codage à la source et codes à effacement*

Le codage réseau peut être considéré comme une extension ou une généralisation du codage à la source (ex. codes fontaines [72]) dans la mesure où les nœuds sources et destinations codent de l'information. Cependant, il est important de noter que restreindre les opérations de codage uniquement aux extrémités du réseau suppose que les nœuds intermédiaires ne peuvent que retransmettre les messages sans aucun traitement. Ceci implique que quelle que soit la redondance d'un message codé, il sera quand même transmis jusqu'à sa destination. En effet, lors de la présence de cycles dans la topologie, des messages peuvent être dupliqués et des copies multiples du même message peuvent arriver à la destination via différents chemins, diminuant la capacité effective du réseau [73]. En utilisant le codage réseau, en revanche, les messages sont combinés à la volée au niveau de chaque nœud intermédiaire au fur et à mesure qu'ils traversent le réseau. Cette approche permet de réduire drastiquement la probabilité qu'un message soit redondant pour un nœud donné.

Des mécanismes comme les codes Reed-Solomon [74], Digital Fountain [75] ou Raptor [76] ont déjà été proposés pour la dissémination coopérative dans les réseaux de capteurs. Ces approches permettent au destinataire final d'un flux de messages de reconstruire efficacement une information collectée de taille  $n$  à partir d'un sous ensemble d'au moins  $n$  messages arbitraires. Dans des systèmes aussi instables et dynamiques que les réseaux de capteurs, des nœuds peuvent temporairement ou définitivement se déconnecter du réseau. Dans toutes ces solutions, ceci peut impliquer l'indisponibilité de certains messages ou une réduction de leur vitesse de propagation dans le réseau rendant les codes à effacement beaucoup moins efficaces comparé au codage réseau classique [73].

### ***2.3. Conclusion***

Si la théorie du codage réseau a réussi à changer la vision que porte la communauté de recherche sur les méthodes d'acheminement de l'information dans les réseaux de communication, l'appréciation de ses implications sur les différents types de réseaux informatiques est encore à investiguer. Dans des réseaux sans fil ad hoc ou à infrastructure, les applications à contraintes fortes en débits et en délais peuvent utiliser des méthodes de codage réseau telles que COPE, Avalanche ou CodeCast que ce soit pour les communications à destination unique ou multiple. Par contre, dans le cas des réseaux inter-machines où le débit disponible est faible, le taux de perte potentiellement important et les ressources par nœud finies, ces solutions deviennent inadaptées voir inopérables.

Ajuster ces techniques aux contraintes imposées par les réseaux de capteurs est une tâche délicate. En effet, dans la pratique, la gestion de la mémoire, les exigences en termes d'énergie, le faible débit ainsi que la dynamique de ces réseaux sont autant de nouveaux paramètres que les algorithmes de codage réseau doivent dorénavant prendre en compte. Quelques efforts pour traiter cette problématique dans les applications telles que la dissémination ou la collecte des données ont été proposés dans la littérature. Cependant, ce sont pour la plupart des solutions exclusivement théoriques qui ne traitent pas des contraintes technologiques propres aux réseaux de capteurs.

Nous pensons que pour ce type d'architecture, prouver qu'un codage apporte un gain de performance n'est qu'un début, et que sans une étude de sa fiabilité et de son coût en termes de ressources, le problème n'est que partiellement résolu. Ainsi, nous considérons tout au long des chapitres qui suivent, une étude basée sur trois axes d'analyse clés que nous jugeons nécessaires et suffisants pour juger du bénéfice pratique du codage réseau dans les réseaux de capteurs sans fil, à savoir :

- a) *Ses performances* : traite principalement le délai, le débit et le temps de convergence ;
- b) *Sa robustesse* : concerne la tolérance aux pertes ainsi que le taux d'acheminement et de retransmission des paquets en saut-par-saut et de bout-en-bout ;
- c) *Son adaptabilité* : inclut la surcharge réseau, les coûts en mémoire et processeur, ainsi que la consommation d'énergie.



## *Chapitre 3*

# 3. Codage inter-machines

Dans ce chapitre, nous abordons le problème de la sous-optimalité des réseaux en *store-and-forward* en ce qui concerne l'utilisation de la bande passante. Nous présentons des mécanismes de codage réseau qui montrent les réponses des codes linéaires aléatoires et opportunistes à ce problème dans les réseaux de capteurs sans fil. Il est question ici d'identifier comment le codage réseau peut augmenter la bande passante disponible aux nœuds du réseau, améliorant ainsi des critères de performances clés à ces architectures tels que le débit et le délai en tenant compte de leurs contraintes applicatives.

Nous nous intéressons dans cette analyse aux deux types de topologies présentées dans l'état de l'art, à savoir, les réseaux maillés et hiérarchiques. L'objectif est d'améliorer le débit global du réseau tout en gardant des délais raisonnables, que ce soit pour la dissémination ou les communications point-à-point. Cette tâche peut être considérée comme un problème d'ordonnancement connu pour être difficile. Ainsi, nous abordons sa résolution avec une approche qui produit des résultats théoriquement optimaux [46] ainsi que des implémentations qui, dans le pire cas, produisent les mêmes performances qu'un acheminement classique en *store-and-forward* [44]. Cette méthode consiste à enrichir ces techniques classiques de transport de l'information en leur intégrant des capacités de codage de paquets au niveau des nœuds intermédiaires.

Ainsi, nous décrivons d'abord notre première contribution. WSC (Wireless Sensor Coding), un protocole de communication qui permet la dissémination des données dans un réseau de capteurs sans fil basé sur un trafic aperiodique événementiel. Nous montrons que contrairement aux solutions proposées dans la littérature, WSC assure un temps de convergence réduit et permet une propagation rapide des notifications dans le réseau. Dans un second temps nous présentons deux autres contributions. CNC (Clustered Network Coding) et CoZi (Coding for ZigBee), qui sont des mécanismes génériques de codage réseau opportuniste exploitant cette fois-ci la topologie hiérarchique des réseaux de capteurs dans le but d'améliorer le routage et la dissémination des données. Notre étude de performances évalue l'apport de ce codage réseau à un saut en utilisant différents modèles de trafic et de densité des nœuds. Elle montre que grâce à un mécanisme d'inférence de topologie adéquat, le réseau peut connaître une amélioration significative de son débit et du délai d'acheminement de bout-en-bout.

Les principes de WSC, CNC et CoZi sont différents des systèmes de distribution de l'information préalablement présentés (cf. Chapitre 2). Il s'agit dans ces protocoles d'exploiter les réceptions multiples d'un paquet par plusieurs nœuds voisins afin d'améliorer sa diffusion dans le réseau. Cette propriété, spécifique au canal sans fil, permet une meilleure utilisation de la bande passante et un acheminement plus rapide des données transmises par les capteurs. Nous montrons que bien que les codes opportunistes et linéaires assurent des performances supérieures à celle des mécanismes usuels en *store-and-forward*, les deux approches se distinguent de par leur efficacité. En effet, dans le cas de réseaux basés sur des événements, dans lesquels le trafic et la taille des données sont relativement limités, le codage linéaire (donc WSC) s'avère être une meilleure solution que CNC qui se montre beaucoup plus efficace dans des modèles de trafic moins sporadiques comme les réseaux périodiques où un grand nombre de sources délivrent les données collectées en même temps. CoZi pour sa part montre l'utilité du codage réseau sans fil pour l'amélioration des flux unicast.

Ce chapitre est structuré comme suit. La section 3.1 introduit WSC puis CNC. Nous y détaillons le fonctionnement des deux protocoles ainsi que leurs études de performances respectives en comparaison avec d'autres systèmes analogues. La section 3.2 quant à elle évalue l'apport du codage opportuniste via l'étude de CoZi, un protocole de routage hiérarchique basé sur l'inférence locale de topologie. Finalement la section 3.3 discute la faisabilité de ces protocoles pour différents contextes applicatifs des réseaux inter-machines.

### ***3.1. Codage réseau et dissémination de l'information***

#### ***3.1.1. Contexte et applications***

La dissémination des données est le trafic de prédilection des réseaux de capteurs aperiodiques conçus pour la supervision d'événements [77]. Ces réseaux diffèrent des architectures classiques de collecte d'informations où les communications sont déclenchées périodiquement ou par requêtes. Le but du déploiement d'un tel réseau est de fournir au plus vite et en « temps réel » une vue d'ensemble des événements survenant dans une zone donnée (cf. Figure 5). Dans ce type d'application, les puits de données peuvent être multiples et mobiles. Les communications vers ces puits sont déclenchées lorsqu'un événement est détecté par un ou plusieurs nœuds chargés de la supervision de la zone.

Parce qu'ils présentent, entre autres, une meilleure efficacité énergétique, ces réseaux ont fait l'objet de nombreux travaux. En effet, l'approche par événement nécessite beaucoup moins de transmissions de messages qu'une approche périodique ou à la demande. Les auteurs de [78] évaluent sur une plateforme comportant un grand nombre de capteurs leur système de diffusion d'événements dans un scénario de détection d'intrusion. Dans [79], les auteurs proposent une nouvelle architecture qui détecte et classe des événements aléatoire, rares et éphémères pour des applications de détection d'incidents. Les deux études concluent que la dissémination est le modèle d'acheminement le plus adéquat pour ce type de réseaux.

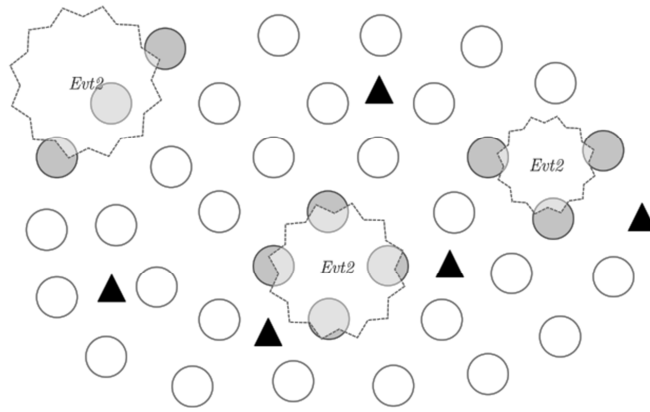


Figure 5. Vue d'ensemble d'un réseau de capteurs basé sur des événements

Dans ce type de réseau de capteurs dits apériodiques, les puits de données peuvent être de différents types. Il peut s'agir d'unités de collecte et d'affichage électronique (panneau dans une ville, un aéroport, etc.), d'utilisateurs nomades (smartphone, ordinateurs, etc.) ou encore d'une simple passerelle internet. Chacun peut souscrire à des événements différents (ex. place de parking disponible, état des files d'attentes dans un magasin, seuils du niveau de pollution, etc.) et reçoit les notifications à chaque fois qu'un sous ensemble de nœuds les détectent. La dissémination est donc un moyen simple d'offrir des informations à jour aux différents utilisateurs en répliquant la notification d'un événement sur le réseau. Ainsi, quel que soit l'endroit où un hôte se connecte il peut rapidement être au courant de l'occurrence des événements qui l'intéressent. En découlent les problèmes inhérents à la dissémination dans les réseaux sans fil [26] tels que l'implosion de trafic ou la famine.

### 3.1.2. Description du système

Dans notre cas, nous partons d'un système à topologie maillée ou hiérarchique constituée de  $n$  capteurs uniformément positionnés dans une surface carrée et plane. Nous désignons par  $S = \{1, 2, \dots, n\}$  l'ensemble des capteurs présents dans le réseau,  $E$  le nombre d'événements qui peuvent avoir lieu et être simultanément détectés par des nœuds dans cette zone. Nous définissons la fréquence d'occurrence des événements  $\mu$  qui décrit le nombre moyen d'événements qui ont lieu à chaque unité de temps.  $\mu$  peut suivre différents processus aléatoires suivant l'application du réseau étudiée. Chaque événement peut être détecté par  $m$  capteurs distincts sélectionnés dans  $S$  avec une probabilité  $P$ . Ce sous ensemble de  $m$  nœuds constitue  $D_i = \{1, 2, \dots, m\}$ , l'ensemble des nœuds qui détectent un événement  $i$ . Ainsi, un nœud  $j$  détecte un événement  $i$  selon un processus aléatoire à débit  $x_j \geq 0$ ,  $j \in D_i$  avec  $\sum x_j = \mu$ . Dans la suite de ce travail, nous désignons par *nœuds sources* les nœuds qui détectent un événement. Aussi, tout événement survenant dans une zone donnée peut être détecté par un ou plusieurs capteurs. Cette redondance est essentielle pour garantir une propagation sans perte de l'information et une meilleure disponibilité des données. Dans certains cas, ce surplus d'information peut également être utilisé pour confirmer l'incidence d'un événement. Par exemple, dans des applications critiques telles que la détection d'incendies, les faux positifs doivent être éliminés notamment grâce à la validation de l'occurrence d'un événement à partir de sources multiples.

### 3.1.3. Présentation du codage inter-machines linéaire – WSC

Dans un protocole de dissémination des données basé sur le codage réseau, des combinaisons linéaires de tous les messages disponibles au niveau d'un nœud sont générées en utilisant des coefficients sélectionnés aléatoirement dans un corps fini assez grand [44]. Lorsqu'un destinataire reçoit assez de messages codés linéairement indépendants, il peut reconstruire les messages originaux en utilisant des éliminations Gaussiennes. Dans le cas de réseaux de capteurs basés sur des événements, nous proposons WSC (Wireless Sensor Coding, ou codage sans fil inter-machines), une technique de dissémination similaire à Avalanche proposé par Gkantsidis *et al* dans [56] pour améliorer les performances des protocoles de distribution de fichiers pair-à-pair sur internet. Dans notre approche, nous proposons d'exploiter les combinaisons aléatoires de messages et la propagation coopérative inhérente au codage linéaire afin de rapidement disséminer les informations collectées par les capteurs. Ce codage est réalisé afin d'éviter le problème d'implosion de trafic en réduisant le nombre de transmissions. Le système de dissémination dans WSC diffère de celui proposé dans [56] où les messages sont des blocs d'un seul fichier qui a pour source un nœud unique. Dans WSC, les messages sont générés par plusieurs nœuds qui détectent le même événement. Contrairement à Avalanche, il n'est pas nécessaire pour les destinataires de connaître le nombre de messages (blocs dans le cas de [56]) générés par la source.

En bref, WSC est un système de dissémination collaboratif basé sur un code linéaire aléatoire [80]. Il permet la distribution d'éléments d'information sur l'occurrence d'événements dans une zone donnée. L'objectif étant d'améliorer le débit global, de réduire le nombre de transmissions, ainsi que la latence en combinant aléatoirement des paquets pour augmenter la capacité de chaque transmission. Afin d'évaluer notre solution, nous effectuons une analyse détaillée des performances de WSC. Nous le comparons avec trois protocoles de dissémination, et cela via différents scénarios incluant plusieurs charges de trafic et densités du réseau.

#### 3.1.3.1. Propagation des messages

Supposons que chaque nœud source  $j \in D_i$  génère un message  $M_i(j)$ , i.e., chaque événement détecté par un ensemble de nœuds  $D_i$  déclenche la transmission d'un message de chacun des nœuds. Le message  $M_i(j)$  contient des informations concernant l'événement  $i$  (identifiant, timestamp, métadonnées, etc.). L'objectif de WSC après la génération du message est de disséminer  $M_i(j)$  à l'ensemble des nœuds du réseau. Nous désignons par  $In(k)$  l'ensemble de messages déjà reçus par un nœud  $k$ .  $T_c(i)$  désigne le *temps de convergence* d'un événement  $i$ , i.e., la durée nécessaire à tous les messages sources générés après la détection de  $i$  pour qu'ils puissent être reçus par l'intégralité des nœuds du réseau (cf. Equation 1).

$$\forall j \in D_i, \forall k \in S, M_i(j) \in In(k) \quad (1)$$

Avant de transmettre un message et afin d'éviter les collisions au moment de la détection d'un événement  $i$ . Chaque nœud source se met en attente de transmission pendant une durée aléatoire bornée.

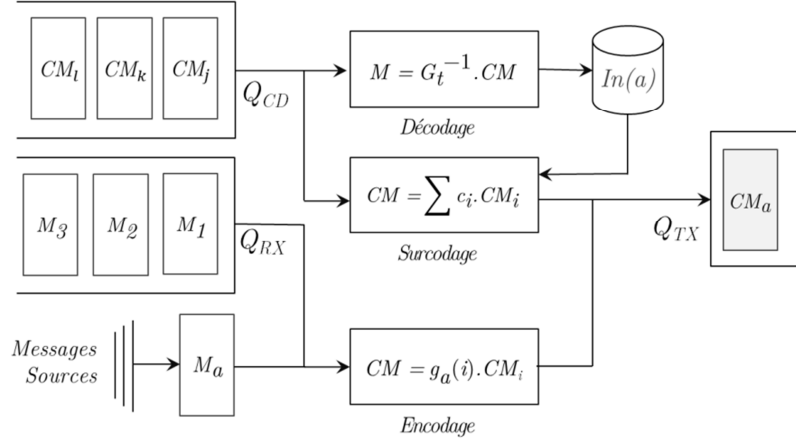


Figure 6. Système de codage WSC

Chaque message contient des informations sur l'évènement qui vient d'avoir lieu et possède un identifiant unique. Afin de garantir un temps de propagation rapide, l'information contenue dans ces messages doit être répliquée au niveau de tous les nœuds du réseau aussi vite que possible, en évitant les problèmes d'implosion et de famine. C'est dans cette optique que s'inscrit le schéma de codage décrit sur la Figure 6 et qui est utilisé à la réception de n'importe quel paquet au niveau d'un nœud  $i$ .

### 3.1.3.2. Encodage

Lorsqu'un nœud  $a \in S$  reçoit un nouveau message  $M_i(j)$  à partir d'un nœud source  $j$  ( $j \neq a$ ,  $j \in D_i$ ), si ce message n'a pas encore été codé alors  $a$  le met dans  $Q_{RX}(a)$  pour un éventuel encodage. Si  $a$  a déjà reçu ce message ( $M_i(j) \in In(a)$ ) alors il ne le traite pas. Cependant, si le message est déjà codé,  $a$  le met dans ce que l'on appelle sa *file de codage*  $Q_{CD}(a)$  comme illustré sur la Figure 6 et procède à ce que l'on appelle une opération de *surcodage*, laquelle est détaillée plus loin dans ce chapitre. Pour simplifier la modélisation, nous considérons les messages transmis comme des vecteurs de longueur  $l(M) = N$  bits, et éléments d'un corps fini  $\mathbb{F}_2^N$ . La longueur de ces vecteurs est identique pour chaque transmission et quel que soit l'évènement détecté. Après une durée  $T_{buf}$  de mise en tampon, tous les messages présents dans la file de codage du nœud sont codés selon l'Equation 2. Le résultat  $CM_a$  est ensuite transmis en broadcast à l'ensemble des nœuds voisins.

$$\forall M_i \in Q_{rx}(a), CM_a = \sum_{i=0}^{|Q_{rx}(a)|} g_a(i).M_i \quad (2)$$

Nous désignons par  $\vec{g} = (g_a(i))$ , le vecteur de codage du message  $CM_a$ . Dans une approche similaire à celle proposée dans [56], nous faisons usage de méthode de propagation aléatoire où chaque nœud sélectionne d'une manière autonome les coefficients linéaires du vecteur de codage  $g_a(i)$  à partir d'un corps fini  $\mathbb{F}_2^N$ . Chaque message codé contient en plus des données combinées  $CM_a$  : un identifiant (ID), la liste des identifiants des messages encodés (CODED\_IDS) pour permettre le décodage à la réception, le vecteur de codage, l'adresse de la source ainsi que la durée de vie du message (cf. Figure 7).



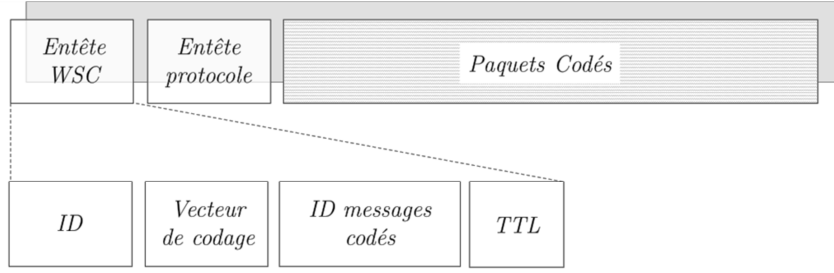


Figure 7. Structure des messages codés de WSC

Dans le cas où la file de codage est vide, le nœud réinitialise simplement la durée de mise en tampon et attend jusqu'au tour suivant. Alors que si  $Q_{rx}(a)$  contient un seul message, le nœud le transmet une seule fois en broadcast. Il est à noter que le nombre de messages disponibles dans  $Q_{rx}(a)$  est fortement corrélé à la durée de mise en tampon : si  $T_{buf}$  est trop courte alors le nœud n'aura pas assez de temps pour recevoir des messages additionnels à coder. Dans ce cas, WSC fonctionne exactement comme un mécanisme classique d'acheminement en *store-and-forward*. Evidemment, lorsque la valeur de  $T_{buf}$  est trop grande, la file de codage peut devenir surchargée, les opérations de codage prennent plus de temps et des messages peuvent être perdus. Finalement, selon la valeur estimée de la fréquence d'occurrence d'évènement  $\mu$ , il est nécessaire de choisir des valeurs appropriées pour  $T_{buf}$  qui permettent d'éviter ce problème.

### 3.1.3.3. Décodage

Le processus de décodage dans WSC est effectué à la réception d'un paquet codé. Il consiste en la résolution d'un système d'équations linéaire à  $u$  inconnues. Nous définissons la file de *décodage*  $Q_{dx}(a)$  pour un nœud  $a$  arbitraire dans  $S$ .  $Q_{dx}(a)$  contient l'ensemble des messages codés que  $a$  a pu recevoir (cf. Figure 6), et  $u$  est le nombre d'identifiants distincts présents dans le champ CODED\_IDS de chaque message dans  $Q_{dx}(a)$ . Pour que le système d'équation puisse être résolu, chaque nœud reçoit des messages qui sont des combinaisons linéaires des messages sources et les sauvegardes dans une matrice avec leurs vecteurs de codage respectifs comme décrit dans l'Equation 3. Ici,  $G_t$  symbolise la matrice des vecteurs de codage, et  $h$  le nombre de messages codés reçus et à décoder. Comme démontré dans [55], il est possible de recouvrer tous les messages codés  $M_0, \dots, M_h$  si la matrice  $G_t$  est de plein rang  $h$ .

$$\underbrace{\begin{bmatrix} CM_0 \\ \vdots \\ CM_h \end{bmatrix}}_{Q_{in}(a)} = \underbrace{\begin{bmatrix} g_0(0) & \cdots & g_h(0) \\ \vdots & \ddots & \vdots \\ g_0(h) & \cdots & g_h(h) \end{bmatrix}}_{G_t} \begin{bmatrix} M_0 \\ \vdots \\ M_h \end{bmatrix} \Rightarrow \begin{bmatrix} M_0 \\ \vdots \\ M_h \end{bmatrix} = G_t^{-1} \begin{bmatrix} CM_0 \\ \vdots \\ CM_h \end{bmatrix} \quad (3)$$

De plus, selon [55], la matrice de décodage  $G_t$  est inversible avec une probabilité proche de 1 si les coefficients  $c_i$  des vecteurs de codage sont aléatoire et sélectionnés dans un corps fini  $\mathbb{F}_2^N$  assez grand. Les auteurs de [81] affirment dans leur étude qu'avec une taille maximum de  $2^{16}$  la probabilité d'inverser la matrice  $G_t$  est de 0.996. Ce qui rend le codage linéaire envisageable dans les réseaux de capteurs sans fil.

## 3.1.3.4. Surcodage

Dans plusieurs situations, les nœuds peuvent recevoir deux ou plusieurs messages déjà codés. Si un nœud choisit (comme le nœud  $k$  dans la Figure 8) d'acheminer uniquement l'un des deux messages ( $CM_i$  dans cet exemple) et ne pas procéder à une opération de recodage, les voisins comme le nœud  $m$  peuvent recevoir le même message codé de différentes sources ( $l$  et  $k$  dans ce cas) ce qui peut réduire la nature aléatoire du réseau et à terme la vitesse de décodage. Pour remédier à ce problème, nous proposons de recoder ces messages afin d'augmenter la probabilité que les messages codés retransmis soient linéairement indépendants. Ainsi, les nœuds qui effectuent cette opération de recodage (ou surcodage) créent plus de paquets innovants qui augmentent le nombre d'équations linéairement indépendantes au niveau des récepteurs.

La Figure 9 illustre un cas où WSC fait usage du surcodage. Initialement, un évènement est détecté par les nœuds  $1, 2, \dots, m$ . Chacun de ces nœuds génère et transmet en broadcast un message  $M_1, M_2, \dots, M_m$  à ces voisins. Lorsque  $i$  et  $j$  reçoivent  $M_1, M_2$  et  $M_4$ , ils choisissent respectivement  $\vec{c}^i = (c_1^i, c_2^i, c_4^i)$  et  $\vec{c}^j = (c_1^j, c_2^j, c_4^j)$  comme vecteurs de codage pour ensuite encoder les messages sources en  $CM_i$  et  $CM_j$  avant de les transmettre. Quand  $k$  reçoit  $CM_i$  et  $CM_j$ , au lieu de n'en transmettre qu'un seul des deux car ils contiennent les mêmes informations, le nœud  $k$  choisit deux coefficients aléatoires  $c_1^k, c_2^k$  et génère une combinaison linéaire des deux messages codés sous forme d'un « surcode »  $CM_k$  (cf. Equation 4). Avant de relayer ce nouveau code aux prochains sauts,  $k$  inclut le vecteur de codage décrit par l'Equation 5

$$CM_k = c_1^k \cdot CM_i + c_2^k \cdot CM_j \quad (4)$$

$$\vec{c}^k = c_1^k \cdot \vec{c}^i + c_2^k \cdot \vec{c}^j \quad (5)$$

Il est à noter que le processus de décodage reste le même dans ce cas, car les vecteurs de codage correspondants sont modifiés à chaque fois qu'une opération de surcodage a lieu.

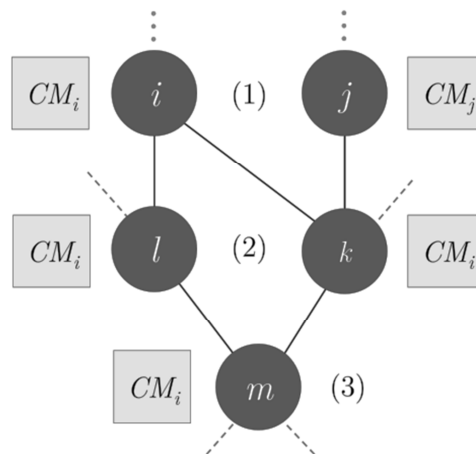


Figure 8. Problème de dépendance linéaire

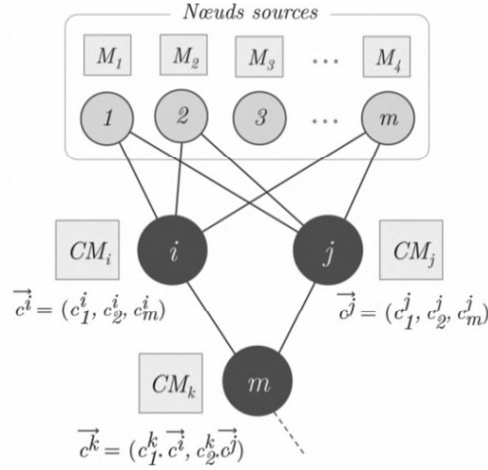


Figure 9. Exemple de surcodage

### 3.1.3.5. Analyse de performance

Nous évaluons les performances de WSC et comparons les résultats avec d'autres systèmes de dissémination existants, à savoir : SPIN [82] qui se base sur la négociation entre les nœuds pour décider du relai ou non d'un message, Gossiping [83] qui utilise la propagation épidémique pour éviter un trop grande surcharge du réseau et enfin, un mécanisme simple de propagation par backoff où les nœuds ne transmettent leurs données qu'après avoir attendu une période aléatoire pour éviter les collisions et les tempêtes de broadcast. Pour ce faire, nous utilisons l'environnement de simulation Qualnet 4.5 [84]. Nous nous intéressons à l'étude des principaux paramètres qui peuvent influencer sur l'acheminement de l'information de la source jusqu'aux destinataires : 1) la fréquence d'apparition des événements  $\mu$ , 2) le nombre de capteurs  $|S|$ , et 3) le nombre moyen des nœuds sources. Pour évaluer les performances de WSC et le comparer aux autres systèmes de dissémination, nous avons considéré deux métriques clés des réseaux de capteurs sans fil :

- i. *La durée de convergence moyenne* : le temps de convergence  $T_c$  est une métrique très importante dans les réseaux de capteurs aperiodiques. Elle indique la durée pendant laquelle les informations concernant un événement sont distribuées à l'intégralité des nœuds du réseau. Nous étudions donc l'impact de  $\mu$ ,  $|S|$  et du nombre de nœuds sources sur  $T_c$  en utilisant WSC, Gossiping, SPIN et la propagation par backoff.
- ii. *Le nombre de transmissions* : afin de déterminer la surcharge réseau propre à chaque solution, nous calculons le nombre de transmissions  $N_{msg}$  pour l'ensemble des nœuds du réseau, et comparons les résultats afin de déduire l'efficacité de l'exploitation de la bande passante de chaque système.

Afin d'implémenter les différentes solutions étudiées, nous avons conçu pour chaque protocole un module lié à la couche MAC 802.15.4 du simulateur. De plus, nous avons développé un module de déclenchement aléatoire d'évènement à travers le réseau selon une fréquence d'occurrence  $\mu$  donnée. Ainsi, nous utilisons un réseau sans fil maillé connexe, où 150 nœuds sont disposés sur une surface de 1500 m<sup>2</sup>. Le rayon de transmission d'un nœud varie selon le modèle de propagation utilisé, mais sa valeur moyenne est de 50 mètres. Le reste des caractéristiques de la simulation est résumé dans le tableau 1.

Tableau 1. Paramètres de simulation

Durée de simulation	600 secondes
Topologie	maillé – connexe
Protocole de couche MAC	IEEE 802.15.4
Capacité	250 kbps
Rayon de transmission	~35 mètres
Fréquence $\mu$	0.1   0.5   1   5   40 evt./s
Nombre moyen de nœuds sources	2.0   6.0
Période de mise en tampon $T_{buf}$	adaptative
Taille des files de codage $Q_{rx}/Q_{dec}$	500 paquets
Taille des paquets	64 kb

**Temps de convergence.** Premièrement, nous calculons pour chaque solution la valeur de la durée de convergence  $T_c$  en tenant compte des valeurs de  $\mu$  (voir Tableau 1). Ainsi, nous pouvons déduire quel algorithme permet la propagation la plus rapide des données. Dans le cas de WSC, nous paramétrons la valeur de  $T_{buf}$  en fonction de la fréquence  $\mu$ . Il est à noter que la combinaison de  $\mu$  et du nombre de nœuds sources détermine la charge du trafic réseau. C'est d'ailleurs grâce à ces deux valeurs que l'on peut inférer le passage à l'échelle de chaque solution. La Figure 10 montre les résultats de cette première analyse (pour  $D=2$  et  $D=6$ ) et confirme bel et bien le bénéfice de WSC. En effet, le temps de convergence de SPIN est plus élevé que WSC. Ceci est due au fait que SPIN utilise un système de communication en 3 étapes avec les séquences de messages ADV→REQ→DATA pour la propagation de chaque information dans le réseau. Dans le système de propagation en *store-and-forward*, le temps de convergence est aussi plus élevé, principalement parce que plusieurs messages redondants sont transmis et que la nature partagée du canal sans fil n'est pas exploitée comme c'est le cas avec le codage linéaire. Gossiping produit le temps de convergence le plus long car la propagation qu'il assure est épidémique et de ce fait très lente. Il existe cependant un cas où le temps de convergence produit par WSC est plus long que celui induit par SPIN : lorsque  $\mu = 0.1$ . Ce cas particulier s'explique par le fait que la période de mise en tampon  $T_{buf}$  est plus longue qu'une séquence de SPIN. Cependant, dans une application réaliste des réseaux de capteurs on peut envisager que la période de mise en tampon soit adaptée dynamiquement à la fréquence d'occurrence d'événements.

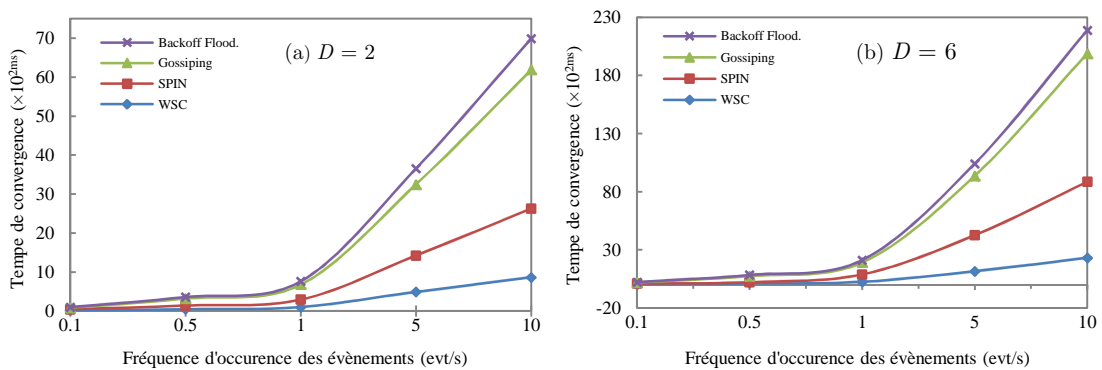


Figure 10. Temps de convergence et fréquence d'apparition des événements

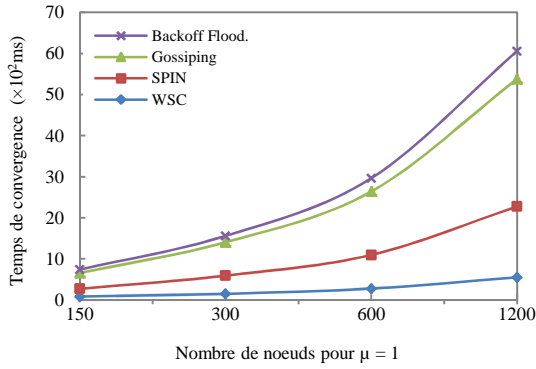


Figure 11. Temps de convergence et fréquence d'apparition des événements

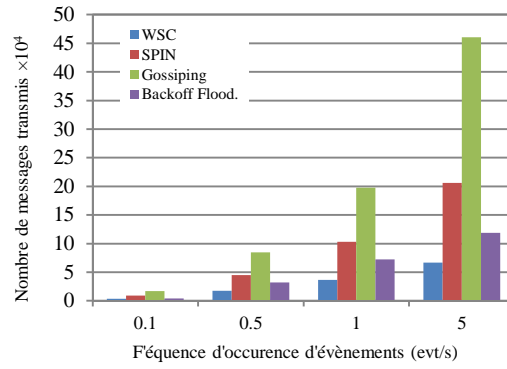


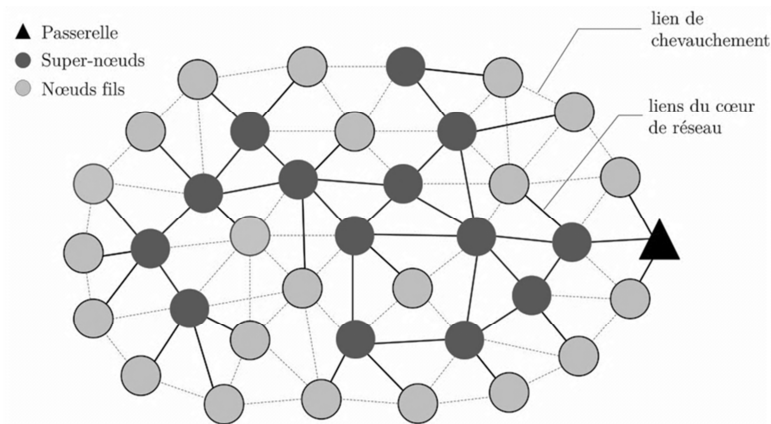
Figure 12. Nombre de messages vs. Fréquence d'apparition des événements

Comme le montre la Figure 11, le temps de convergence de WSC reste bas en comparaison avec les autres protocoles même si le nombre moyen de nœuds sources  $D$  augmente. Ceci s'explique par l'utilisation des mécanismes de propagation collaborative au sein d'une seule transmission propres au codage réseau linéaire. Nous pouvons conclure de ces premiers résultats que WSC reste efficace précisément lorsque la charge de trafic augmente dans le réseau. Ce qui n'est pas le cas des protocoles de dissémination en *store-and-forward*.

**Surcharge réseau.** Pour déterminer quel mécanisme implique le moins de surcharge réseau, nous calculons le nombre de transmissions  $N_{msg}$  par nœud pour chaque solution. La figure 12 montre que la valeur de  $N_{msg}$  avec WSC est très faible en comparaison à SPIN ce qui est prévisible car SPIN utilise des messages de négociation qui nécessitent plus de transmissions. Dans Gossiping, les nœuds choisissent de relayer un message à un seul de leurs voisins mais continuent à faire de même jusqu'à ce que tous les nœuds du réseau aient reçu le message, ce qui engendre une augmentation du nombre de transmissions par nœud et explique la différence avec WSC. Enfin, les messages additionnels dans le mécanisme par *backoff* sont dus au traitement séquentiel des paquets même lorsque des opportunités de codage se présentent et permettent d'envoyer plusieurs messages en une seule transmission.

### 3.1.4. Présentation du codage inter-machines opportuniste – CNC

Alors que les topologies hiérarchiques assurent plus de robustesse, de stabilité et une plus grande évolutivité aux réseaux sans fil, elles souffrent d'une sous-optimalité au niveau du débit global offert par le réseau par rapport aux topologies maillées. Cette sous-optimalité est une conséquence du partitionnement du réseau en groupes disjoints car en général, cette technique restreint l'utilisation de certains liens du réseau et impliquent une réduction de la bande passante théorique disponible [85]. Dans la littérature, on peut distinguer plusieurs critères (mobilité, énergie, etc.) utilisés pour permettre cette hiérarchisation du réseau. Bien que d'un point de vue abstrait les nœuds sont affectés à des clusters distincts, ces derniers peuvent se chevaucher et en réalité des nœuds peuvent fréquemment recevoir des paquets d'autres nœuds qui ne sont pas forcément dans le même cluster (*overhearing*). Dans ce travail, nous utilisons les termes *liens d'écoute* ou de *chevauchement* pour désigner ces liens (cf. Figure 13). Ainsi, nous proposons de les exploiter pour améliorer les méthodes d'acheminement dans les réseaux de capteurs sans fil hiérarchiques.



**Figure 13.** Liens de chevauchement dans un réseau hiérarchique

Notre objectif est de continuer à bénéficier des avantages des réseaux en clusters mais en évitant la perte en débit due à la séparation logique des nœuds. L'idée consiste à utiliser un codage opportuniste afin de combiner à chaque transmission le paquet à envoyer avec d'autres paquets écoutés sur les liens de chevauchement. En faisant cela, nous améliorons les performances de la dissémination à la fois en termes de débit et de délai.

Dans une topologie maillée le codage réseau linéaire prend tout son sens car la nature partagée des communications en broadcast permet les réceptions multiples d'un paquet par plusieurs nœuds qui peuvent en extraire des données utiles. Par contre, dans une topologie hiérarchique, le codage linéaire devient plus contraignant car il implique l'utilisation de mécanismes plus complexes de gestion de flux (maintenance de la hiérarchie, routage, etc.). En effet, dans un réseau hiérarchique où la dissémination se fait entre des clusters, les transmissions n'ont lieu que via le cœur de réseau constitué par les super nœuds. Ceci implique que le modèle de propagation change en comparaison avec une topologie maillée où typiquement tout le monde retransmet ce qu'il a reçu à ces voisins en évitant les redondances. Ainsi, nous proposons CNC (Clustered Network Coding – ou Codage réseau hiérarchique) un mécanisme de codage réseau basé sur l'inférence de topologie locale qui permet l'utilisation du codage opportuniste pour encoder différents flux de paquets à partir des clusters adjacents en exploitant les liens de chevauchement naturellement présents dans les réseaux hiérarchiques. Nous évaluons CNC via une analyse de performance qui inclut différents schémas de trafic et densités des nœuds. Les résultats montrent une amélioration du débit global du réseau spécialement lorsqu'il subit une charge de trafic importante.

#### 3.1.4.1. Etablissement de la topologie hiérarchique

Le principal but d'un algorithme de construction de topologie hiérarchique est d'apporter une certaine stabilité des liens tout en maximisant la durée de vie de chaque cluster avec une faible surcharge réseau. Tout cela en un temps de convergence aussi court que possible. Ainsi, ces algorithmes [82] incluent souvent des mécanismes d'auto-organisation synchrones ou asynchrones pour faire face à la dynamique inhérente aux réseaux de capteurs sans fil. Durant la transmission des données le cœur de réseau peut se comporter différemment selon les caractéristiques des nœuds et l'application du réseau.

Dans la mesure où nous abordons les algorithmes hiérarchiques en général, quelques propriétés génériques doivent être définies afin de satisfaire n'importe quelle configuration réseau. Nous supposons que les nœuds sont capables d'écouter les liens de chevauchement et présumons qu'ils sont également en mesure d'effectuer des opérations logiques atomiques telles que le XOR pour l'encodage des paquets. Nous partons du même contexte système décrit en section 3.1.2 mais adapté à une topologie hiérarchique. Ici, les  $n$  capteurs du réseau sont organisés en  $K$  clusters distincts. Chaque cluster contient  $k$  nœuds spéciaux dénommés *super-nœuds* (SN) qui sont choisis au début de la première phase d'établissement de la topologie hiérarchique. A la fin de cette phase, les nœuds restants doivent se lier avec un super-nœud avant de commencer à transmettre leurs données. Il est à noter qu'un nœud lié à un ou plusieurs SN ne peut transmettre ses données qu'à ce(s) nœud(s) précis. Nous définissons  $T_c$ ,  $T_e$  et  $T_f$  tel que  $T_c = T_f + T_e$ .  $T_c$  est le temps de convergence de l'algorithme d'établissement de la topologie hiérarchique. Il inclut la durée d'élection des super-nœuds ( $T_e$ ) ainsi que la durée de formation du cluster ( $T_f$ ) durant laquelle les nœuds se lient à leurs super-nœud(s) respectif(s). Nous désignons par  $C_i = \{n_j\}$  l'ensemble des nœuds  $n_j$  qui sont membres du même cluster  $C_i$  (avec  $i = 1, \dots, k$  et  $j = 1, \dots, n$ ). Noter qu'un nœud ne peut être membre que d'un seul cluster (cf. Equation 6).

$$\forall l, m \in k, \quad l \neq m \Rightarrow C_l \cap C_m = \emptyset \quad (6)$$

Une hypothèse importante sur laquelle nous nous basons, est le fait que tous les algorithmes d'établissement de topologie hiérarchique utilisent une maintenance des clusters basée sur l'échange de message de contrôle<sup>3</sup> et passent au moins par trois phases :

- 1) *Election des super-nœuds (SN)* : à la fin de cette étape de durée  $T_e$ , chaque nœud doit déterminer s'il est super-nœud ou pas. Cette décision peut se faire en utilisant des techniques probabilistes [87], basées sur l'identifiant du nœud [88] ou sur une pondération quelconque [89] (énergie résiduelle, nombre de saut, etc.). Chacune correspond à un certain type d'application. Par exemple, les approches centrées sur l'identifiant du nœud sont très efficaces dans des réseaux mobiles où les super-nœuds doivent être élus rapidement, mais peuvent pénaliser certains nœuds en termes de consommation de ressources.
- 2) *Formation des clusters* : une fois les SN élus, la façon la plus rapide d'en informer les nœuds voisins et de transmettre un paquet de contrôle ADV qui annonce que le nœud  $i$  ( $\forall i \in M$ ) est un SN. Chaque nœud non élu peut ensuite envoyer un message d'acquiescement au super-nœud auquel il désire s'associer et établir ainsi une relation *parent-enfant*. A noter que selon l'algorithme, d'autres messages de contrôle peuvent être transmis entre le SN et les membres de son cluster pour organiser la prochaine phase (ordonnancement en slots de temps, paramètres de cryptage, etc.). L'algorithme de CNC présume qu'après  $T_c$  les  $K$  clusters sont construits.

---

<sup>3</sup> La maintenance de la topologie dans ce contexte consiste en l'échange de messages *hello* courts qui contiennent l'identifiant du transmetteur (potentiellement parmi d'autres informations) et cela afin de construire et de maintenir la structure en cluster.

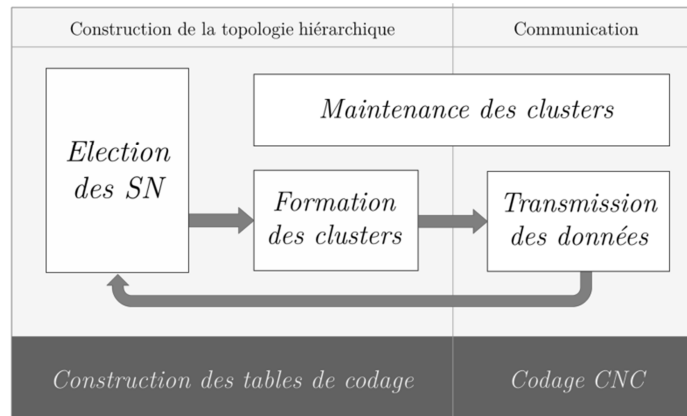


Figure 14. Vue schématique du fonctionnement de CNC dans le temps

- 3) *Dissémination des données* : cette phase permet la transmission des données effectives d'une source vers l'ensemble du réseau de façon hiérarchique. L'information collectée par les nœuds est disséminée d'un nœud vers un autre selon sa position dans la hiérarchie. Ainsi, chaque nœud envoie son paquet à son super-nœud parent direct, ce dernier le relaiera de proche-en-proche jusqu'à atteindre tous les SN du réseau qui transmettrons les données à leurs enfants respectifs.

Cette organisation chronologique des algorithmes de construction de topologie hiérarchiques est assez générique pour modéliser les différentes approches de clustering que nous avons pu identifier dans la littérature.

#### 3.1.4.2. Système d'ordonnement

Comme illustré sur la Figure 14, CNC s'exécute en deux parties. Premièrement, il exploite les messages de contrôle échangés pendant la phase de construction des clusters afin d'inférer la topologie local des nœuds et construire ainsi leur table de codage. Ensuite, CNC utilise ces tables pour effectuer la décision de codage la plus optimale à chaque transmission. Pour ce faire, les nœuds intermédiaires combinent (en utilisant un simple XOR) le paquet à transmettre avec les paquets écoutés sur les liens de chevauchement avec pour seule contrainte qu'un maximum de nœuds qui reçoivent le paquet codé puissent en extraire de l'information utile.

#### 3.1.4.3. Inférence de topologie locale et encodage

Dans COPE [47] le problème de l'inférence de topologie est partiellement résolu grâce à l'échange de rapports de réception à intervalles de temps réguliers. En effet, pour pouvoir réussir la transmission d'un paquet codé, il est nécessaire d'être au courant de ce que les voisins à un saut ont déjà reçus pour savoir quel paquet inclure ou pas dans le code à transmettre. Le problème n'est que partiellement résolu car dans le cas de présence de pertes non-négligeables, des rapports de réception peuvent être perdus et des erreurs dans le codage peuvent apparaître. De plus, cette solution implique une surcharge considérable sur les liens du réseau liée à l'envoi constant de ces rapports de réception.



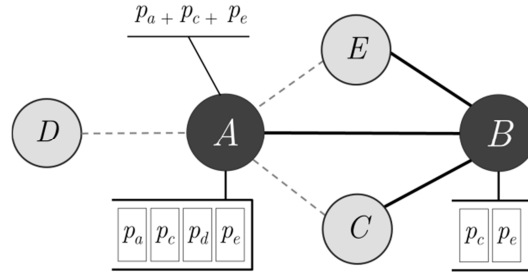


Figure 15. Exemple d'un codage opportuniste

Afin d'apporter une réponse moins couteuse à la question : *qui a reçu quoi ?* CNC exploite dans sa phase d'apprentissage (cf. Figure 14) les messages de contrôle déjà utilisés pour construire et maintenir la topologie hiérarchique du réseau. Ces messages sont en effet utilisés afin de construire localement des tables de voisinage similaires à celles proposées dans COPE. En analysant ces messages les nœuds peuvent lister l'ensemble des liens de chevauchement de leurs voisins et ainsi savoir quel paquet ces dernier peuvent décoder ou pas. Considérons par exemple, le nœud A dans la Figure 15. A la fin de la phase de formation des clusters, A sait qu'il peut écouter des paquets provenant des nœuds B, C, D et E grâce aux messages de contrôle qui se sont échangés depuis l'initialisation du réseau. Il peut ainsi transmettre à ces voisins un paquet de contrôle qui contient les identifiants des nœuds qu'il peut écouter. Il est à noter que cette opération n'a lieu qu'après la phase de formation du cluster, et de ce fait n'implique pas autant de surcharge réseau que les rapports de réception utilisés dans COPE. Les opérations atomiques de codage et décodage de paquets restent les même que dans [47], i.e., quand un ensemble de  $N$  paquets sont codés, le nœud les combine en utilisant un XOR et transmet le résultat en broadcast. Lors du décodage, la même opération de XOR est effectuée entre les  $(N - 1)$  paquets reçus/écoutés et le paquet codé en question afin de recouvrer le paquet original.

#### 3.1.4.4. Stratégie de codage

Lors de la propagation de données dans une topologie hiérarchique, parce que les clusters se chevauchent parfois, une fraction du trafic écouté peut être rejetée car elle provient de clusters différents. Avec CNC ces paquets non exploités ne sont plus rejetés mais combinés au trafic courant de chaque nœud intermédiaire et transmis au reste du réseau permettant à un maximum de nœud d'en extraire des données utiles. Prenons l'exemple canonique suivant qui illustre une communication inter-cluster entre quatre super-nœuds voisins A, B, C et D. La configuration des files d'attente de chacun est illustrée dans ce qui suit :

$$Q_{rx}(A) = \{p_a, p_b, p_c, p_d\}; Q_{ovh}(B) = \{p_b, p_d\}$$

$$Q_{ovh}(C) = \{p_a, p_b\}; Q_{ovh}(D) = \{p_a, p_c\}$$

Où  $p_a$ ,  $p_b$ ,  $p_c$  et  $p_d$  sont des paquets qui doivent tous être délivrés aux nœuds A, B, C et D. Disséminer des paquets implique d'essayer d'envoyer le plus rapidement possible le maximum de paquets à un maximum de destinations. Chaque fois qu'un nœud  $i$  reçoit un nouveau paquet  $p$ , si ce dernier est destiné à tous les nœuds du réseau, une variante du codage opportuniste classique est effectuée. En effet, au lieu de combiner la tête de la file de transmission avec les paquets écoutés, le nœud sélectionne dans sa table de codage la

combinaison de paquets qui sera décodée par le plus grand nombre de nœuds et transmet la combinaison optimale. Dans la figure 15 par exemple, il est bien plus efficace pour A de transmettre  $p_c \oplus p_b$  au lieu d'une combinaison en codage réseau opportuniste classique où ce serait  $p_a \oplus p_b$ . Grâce à CNC et à cette stratégie de codage optimisée pour la dissémination, l'intégralité des nœuds voisins peuvent extraire de l'information utile du paquet codé. Que ce soit B et C qui décotent  $p_c$  ou D qui décode  $p_b$ .

#### 3.1.4.5. Analyse de performance

Dans cette partie du travail, nous évaluons les performances de CNC. Nous utilisons l'environnement de simulation Qualnet 4.5 en considérant différents schémas de trafic et des densités variables. Nous calculons la vitesse de propagation, la surcharge réseau et montrons que CNC n'affecte que peu le temps de convergence des algorithmes d'établissement de topologie hiérarchiques. Nous comparons le résultat avec une approche sans codage et avec COPE qui effectue un simple codage opportuniste sans prendre en compte la topologie hiérarchique. Nous examinons deux scénarios avec 50 et 100 nœuds placés dans une surface plane de 1000 m<sup>2</sup>. Nous avons modifié le mécanisme d'ordonnancement de paquets du module IEEE 802.15.4 présent dans le simulateur pour permettre l'implémentation du codage opportuniste de CNC et COPE. L'ensemble des attributs de la simulation sont présentés dans le tableau suivant.

**Tableau 2.** Paramètres de simulation

<b>Durée de simulation</b>	600 s
<b>Topologie</b>	Hiérarchique (ZigBee) BO = 15   SO = 15
<b>Protocole de couche MAC</b>	IEEE 802.15.4
<b>Capacité</b>	250 kbps
<b>Rayon de transmission</b>	~35 mètres
<b>Fréquence <math>\mu</math></b>	0.1   0.5   1   5   40 evt./s
<b>Nombre moyen de nœuds sources</b>	2.0   6.0
<b>Période de mise en tampon <math>T_{buf}</math></b>	500 ms
<b>Taille des files de codage <math>Q_{rx}/Q_{dec}</math></b>	500 paquets
<b>Taille des paquets</b>	116 octets

**Temps de convergence.** La Figure 16 représente l'évolution du taux d'acheminement moyen pour un trafic à fréquence  $\mu$  qui doit être disséminé à l'intégralité du réseau avec un taux de perte moyen de 10%. Elle montre que grâce à l'utilisation d'un mécanisme de codage opportuniste spécialement conçu pour la dissémination de l'information, CNC surpasse COPE en termes de vitesse de propagation. Cela est dû au fait que la politique d'ordonnancement des paquets de CNC permet de générer des combinaisons de paquets bien plus optimales que celle générées par COPE. Evidemment, COPE et CNC offrent des temps de convergence plus courts comparé à un mécanisme d'acheminement sans codage (routage ZigBee [39] dans ce cas).

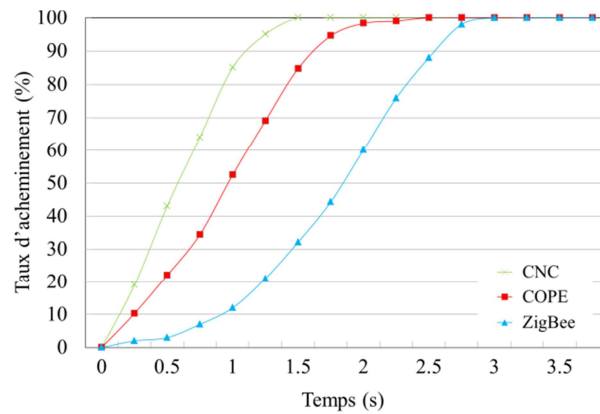


Figure 16. Evolution du taux d'acheminement des paquets dans le temps

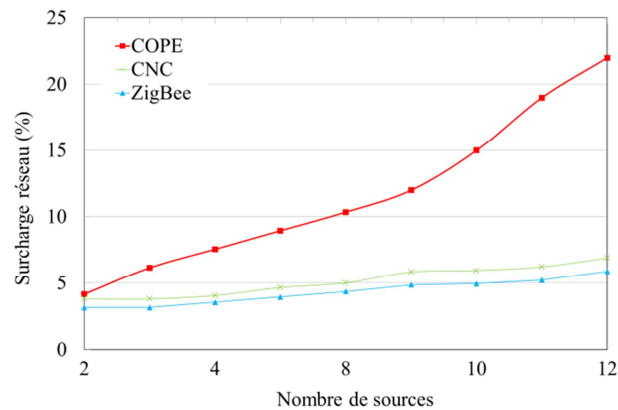


Figure 17. Nombre de messages transmis en fonction du nombre de sources

**Surcharge réseau.** La Figure 17 montre l'évolution, en fonction du nombre de sources, du nombre de messages de contrôle transmis afin de maintenir la topologie hiérarchique du réseau ainsi que les messages utilisés pour le codage réseau le cas échéant. Cette moyenne représente la fraction de la bande passante utilisée pour un usage autre que la transmission de données. Comme prévu, les résultats indiquent le faible impact de CNC en termes de surcharge réseau en comparaison avec COPE et ce car le système que nous proposons est beaucoup moins invasif que COPE, grâce à son mécanisme d'inférence de topologie qui exploite les messages déjà utilisés lors de la formation des clusters et leurs maintenance.

### 3.1.5. Discussion

Dans cette analyse des apports du codage réseau à la dissémination des données dans les réseaux de capteurs, nous avons proposé deux mécanismes de codage, l'un linéaire (WSC) et l'autre opportuniste (CNC), qui ont permis d'améliorer l'acheminement de l'information pour des applications contraintes et à fortes exigences en temps de réponse. Nous avons montré que grâce à un codage réseau approprié, il est possible de s'adapter aux propriétés applicatives des réseaux de capteurs, ainsi qu'à leurs topologies. Ainsi, le codage linéaire (WSC) permet d'améliorer l'utilisation de la bande passante dans des topologies maillées alors que le codage opportuniste (CNC) convient plus aux topologies hiérarchiques. Il est somme toute important de noter que même si le codage linéaire offre une vitesse de propagation plus rapide que celle d'un codage opportuniste, il est clair que c'est un système

assez lourd qui subit les mêmes limitations que les codes à effacements ou les codes à la sources, à savoir, un manque de fiabilité lors de la présence de pertes dans le réseau. Après avoir traité le problème de la sous-optimalité des méthodes de dissémination dans les réseaux de capteurs sans fil, nous allons procéder dans la suite de ce chapitre à l'étude des apports prospectifs du codage réseau dans les communications point-à-point.

## 3.2. Codage réseau et routage unicast

### 3.2.1. Contexte et applications

Les nœuds dans un réseau inter-machines basé sur l'orchestration [90] communiquent généralement grâce à des flux unicast (cf. Figure 18). Largement utilisées dans les architectures automatisées pour coordonner des capteurs avec des effecteurs [2], ces applications connaissent un intérêt particulier surtout depuis la popularisation des thématiques d'informatique ubiquitaire et d'internet des objets [91]. Ici, les efforts de la communauté se concentrent sur la réduction de la consommation d'énergie, la fiabilisation des communications de bout-en-bout, mais surtout sur l'amélioration des délais d'exécution entre capteurs et effecteurs orchestrés. L'unicast est aussi utilisé dans les réseaux de capteurs à large échelle pour des applications en requête/réponse. Que ce soit pour l'adressage, la réinitialisation ou simplement l'interrogation d'un nœud, ce type de trafic nécessite des débits élevés et peu variables [2] pour assurer des temps de réponses satisfaisants tout en consommant le moins de ressources possible.

### 3.2.2. Description du système

Dans cette étude, nous faisons l'hypothèse d'un réseau de capteurs à large échelle où des communications en unicast ont lieu entre des couples de nœuds sélectionnés aléatoirement. Nous prenons cette fois-ci l'hypothèse d'une topologie hiérarchique uniquement car les systèmes en cluster s'avèrent moins coûteux en termes de ressources pour ce type de trafic routé [86]. En effet, dans un réseau où les données sont disséminées, la topologie peut être maillée car il n'est pas nécessaire d'utiliser des mécanismes d'établissement et de maintien de route qui consomment des ressources mémoire (tables de routages), des ressources processeur (calcul des chemins) et de l'énergie (maintien de la connectivité).

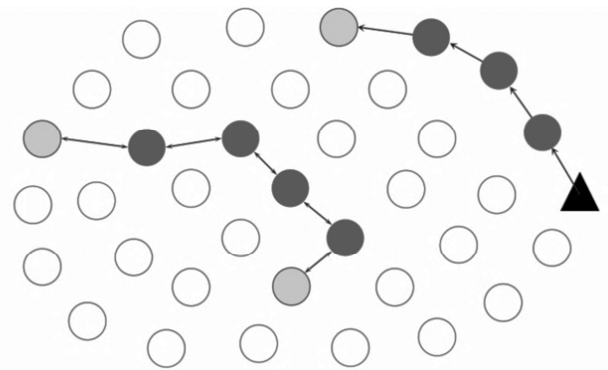


Figure 18. Vue d'ensemble d'un réseau de capteurs basé orchestration

Par contre, dans un réseau de capteurs où le routage de paquets est utilisé pour atteindre la destination (ici l'unicast), la réduction du coût protocolaire ne peut se faire que via une organisation aussi simple et peu coûteuse que la topologie hiérarchique. Nous partons donc d'un système hiérarchique similaire à celui décrit dans la section 3.1.4.1, mais cette fois-ci conçu pour la topologie hiérarchique proposée dans les réseaux ZigBee/IEEE 802.15.4. En effet, nous supposons un réseau de capteurs sans fil, constitué de  $n$  nœuds organisés en  $K$  clusters distincts. Dans la mesure où nous traitons le problème de l'unicast où les flux sont acheminés selon le routage hiérarchique, notre étude aura comme point de comparaison le routage ZigBee qui organise le réseau en une topologie à deux tiers :

- i.* *Les routeurs* : ce sont les nœuds parents chargés du routage de l'information mais aussi d'assurer un certain nombre de services comme la synchronisation des nœuds, l'agrégation des données, ou l'allocation d'adresses. Ces routeurs combinés au puits de données constituent le cœur de réseau de la topologie hiérarchique sur lequel sont routés tous les flux unicast.
- ii.* *Les nœuds terminaux* : ce sont les nœuds enfants qui ne peuvent communiquer qu'à travers leurs parents (routeurs) respectifs. Ils transmettent leurs flux unicast lorsqu'ils ont des données à envoyer et se mettent en veille le reste du temps.

Nous considérons chaque ensemble de parent-enfant(s) associés comme un cluster dans la topologie. Ainsi, nous pouvons intégrer des mécanismes de codage réseau opportuniste pour combiner des flux unicast provenant de sources différentes afin d'améliorer le débit et la latence au sein du cœur de réseau chargé du routage hiérarchique.

### 3.2.3. Codage opportuniste pour ZigBee - CoZi

Le choix du codage opportuniste s'impose lors de l'utilisation de flux unicast. En effet, utiliser un code linéaire pour améliorer le routage point-à-point est une tâche délicate [51] car contrairement à la dissémination, chaque paquet transmis doit prendre un chemin précis sur le réseau pour pouvoir arriver rapidement à sa destination. Le problème du codage réseau linéaire réside dans sa nature aléatoire qui ne permet pas (contrairement au codage opportuniste) d'assurer un décodage en saut par saut nécessaire pour garantir que le paquet à transmettre traverse bien la route prévue.

#### 3.2.3.1. Ordonnancement des paquets

Comme décrit sur la Figure 19 le mécanisme d'ordonnancement derrière CoZi requiert l'utilisation de trois files d'attente.  $Q_{ovh}$  contient les paquets provenant de clusters différents,  $Q_{out}$  est la file des paquets unicast reçus ou générés par le nœud lui-même, tandis que  $Q_{in}$  représente la file des paquets à transmettre qu'ils soient codés ou non. L'Algorithme 1 montre qu'à chaque fois qu'un nœud intermédiaire  $A$  reçoit un paquet  $p$ , il vérifie si  $p$  est non codé, auquel cas  $A$  acquitte sa réception s'il lui est destiné, ou le sauvegarde dans  $Q_{ovh}$  sinon. Dans l'éventualité où  $p$  serait codé et que  $A$  parvient à en extraire un paquet en utilisant le contenu de  $Q_{ovh}$  et de  $Q_{in}$ , un message d'acquiescement ACK est envoyé au transmetteur si  $A$  est la destination. Par contre, si  $A$  n'est pas la

destination du paquet extrait, ce dernier est sauvegardé dans  $Q_{ovh}$  pour être utilisé dans de futures opérations de décodage.

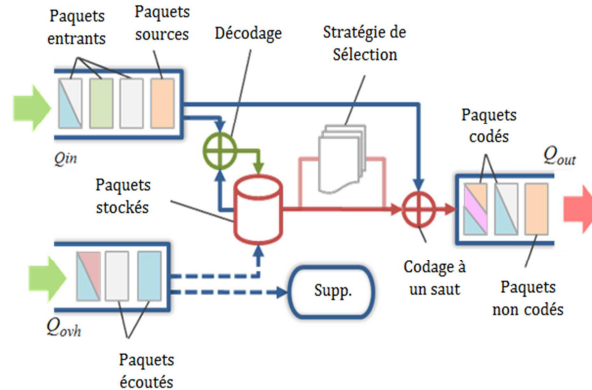


Figure 19. Mécanisme d'ordonnancement de CoZi

On peut s'interroger sur le surcout en usage mémoire induit par l'écoute de paquets au niveau des nœuds intermédiaires, et sur les délais liés aux opérations de codage et de décodage. En fait, ces délais et besoins en mémoire sont négligeables. Par exemple, la sauvegarde de 117 paquets ZigBee qui correspond à une période d'écoute de 500 ms nécessite une quantité mémoire d'environ 15 kilo-octets. Cette quantité est largement disponible dans la plupart des capteurs compatibles ZigBee. Concernant le coût des opérations de codage, il est lui aussi négligeable car le codage opportuniste se base uniquement sur des opérations XOR atomiques au niveau processeur qui n'altère pas le temps de traitement des paquets.

---

**Algorithme 1.** Réception d'un paquet par un routeur  $A$

---

```

1: for each incoming packet  $p$  in  $Q_{in}(A)$  do
2:   if  $p$  is non-coded then
3:     if  $p$  is destined to  $A$  then
4:        $A$  sends an  $ACK$  to the sender
5:     else /*  $A$  overheard  $p$  */
6:        $p$  is stored in  $Q_{ovh}(A)$ 
7:     end if
8:   else /*  $p$  is coded */
9:     if  $p$  is decodable then
10:       $p_{dec} = Decode(p)$ 
11:      if  $p_{dec}$  is destined to  $A$  then
12:         $A$  sends an  $ACK$  to the sender
13:      else /*  $A$  overheard  $p$  */
14:         $p_{dec}$  is stored in  $Q_{ovh}(A)$ 
15:      end if
16:    else /*  $p$  is not decodable */
17:       $p$  is dropped from  $Q_{in}(A)$ 
18:    end if
19:  end if
20: end for

```

---

2	2	2	1	1	(94 to 110) - MCP
Frame control	Destination Address	Source Address	Radius	Sequence Number	Packet Payload
	Routing Fields				
NWK Header					NWK Payload

Figure 20. Structure d'une trame 802.15.4

2	1	4 to 20	(102 to 118) - MCP	MCP	2
Frame control	Sequence Number	Addressing Fields	Data Payload	Coded Packet IDs	FCS
	MAC Fields			MAC Fields	
MAC Header			MAC Payload	MAC Footer	

Figure 21. Structure d'une trame 802.15.4 codée

### 3.2.3.2. Codage

Sachant que les nœuds intermédiaires sur la route d'un paquet dans un réseau ZigBee sont nécessairement des routeurs, nous supposons que les opérations de codage réseau sont exclusivement effectuées au niveau de ces nœuds avant n'importe quelle transmission. Nous supposons le cas où un nœud  $A$  reçoit  $N$  paquets dans  $Q_{in}$  et doit les retransmettre à  $M$  nœuds. Quand un paquet  $p$  en tête de  $Q_{out}$  doit être transmis à son prochain saut  $B$ , le nœud  $A$  choisi parmi  $Q_{out}$  les paquets qui sont supposés être déjà reçus ou écoutés par  $B$  et les combine avec  $p$  pour générer un paquet codé encapsulé dans une trame spéciale 802.15.4 (cf. Figure 21). Cette opération permet à certains nœuds voisins de recevoir un nouveau paquet en parallèle à la transmission de  $p$ . Il est à noter que la taille des identifiants des paquets codés (nécessaires au décodage) est définie par  $MCP$ , le nombre maximum de paquets à coder. La valeur de  $MCP$  peut être sélectionnée selon les caractéristiques du réseau (nature du trafic, densité, topologie, etc.). Dans notre cas, une valeur de 5 s'est avérée suffisante pour garantir une amélioration des performances réseaux.

### 3.2.3.3. Décodage

Lorsqu'un nœud  $A$  reçoit ou écoute un paquet codé  $q$ , il vérifie le champ  $MCP$  qui contient les identifiants des paquets codés pour vérifier s'il est en possession (dans  $Q_{rx}$  ou  $Q_{ovh}$ ) des paquets nécessaires à son décodage. Si tel est le cas, le nœud XOR  $q$  avec les paquets correspondants et en extrait le paquet original  $p$ . Après cette opération,  $A$  vérifie le champ *adresse de destination* de  $p$  (cf. Figure 20). Si le paquet lui est destiné,  $A$  valide son décodage et sa réception avec un paquet court d'acquiescement (ACK). Par contre, si pour une raison ou pour une autre  $A$  n'a pas les paquets nécessaires au décodage de  $p$ , et que ce dernier lui était destiné, alors une retransmission sera effectuée, potentiellement codée avec un nouvel ensemble de paquets.

### 3.2.3.4. Analyse de performance

Dans cette partie, nous évaluons les performances de CoZi dans différents scénarios avec deux densités différentes de réseau, et cela pour quantifier le débit global et la latence pour des communications en point-à-point. Pour ce faire, nous utilisons une plateforme de simulation (Qualnet 4.5) dans laquelle 50 à 100 nœuds sont déployés dans une zone plane.

Le coordinateur ZigBee est placé au centre du réseau et le reste des nœuds sont soit des routeurs, soit des nœuds terminaux. Nous utilisons un module d'extension qui permet la sélection aléatoire de couples de nœuds source et destination avec un trafic constant (CBR). L'ensemble des attributs de la simulation est résumé dans la Tableau 3.

Tableau 3. Paramètres de simulation

Durée de simulation	600 s
Topologie	Hierarchique (ZigBee) BO = 15   SO = 15
Routage	ZigBee cluster-tree routing
Protocole de couche MAC	IEEE 802.15.4
Capacité	250 kbps
Rayon de transmission	~35 mètres
Taux de pertes moyen	10%
MCP (nombre max. de paquets codés)	5
Type de trafic	CBR variable
Période de mise en tampon $T_{buf}$	500 ms
Taille des files de codage $Q_{rx}/Q_{dec}$	500 paquets
Taille des paquets	116 octets
Nombre de routeurs	80%
Nombre de nœud terminaux	19%

**Débit.** Nous calculons d'abord le débit par rapport à la charge de trafic pour un réseau de 50 et de 100 nœuds. Ce calcul est le résultat du rapport entre le débit utile constaté à la réception (hors retransmissions) et celui du CBR généré à la transmission. La Figure 22 montre que le débit atteint reste stable (80-95%) en utilisant CoZi, alors qu'il décroît significativement lorsque le routage ZigBee sans codage est utilisé. Ces performances sont dues aux opérations de codage opportuniste qui permettent d'envoyer plus de données en moins de transmissions. Les résultats montrent donc que CoZi peut améliorer l'utilisation de la bande passante lors de communications dans les réseaux routés, particulièrement quand ces derniers subissent des charges de trafic élevées pendant lesquelles les opportunités de codage sont plus fréquentes.

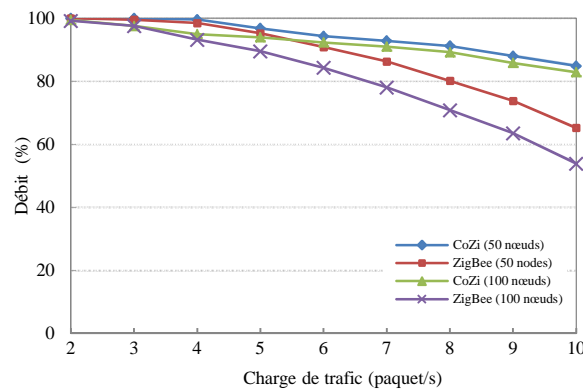
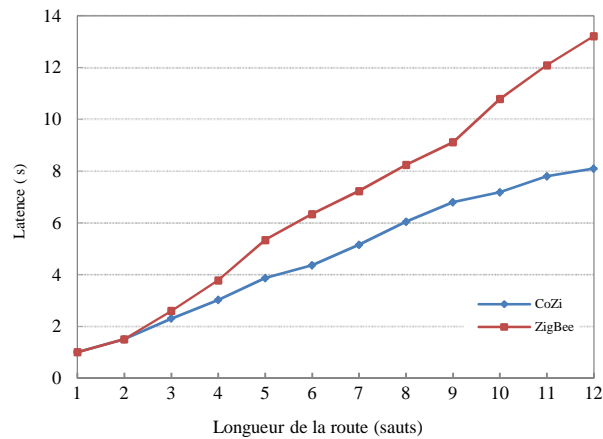


Figure 22. Débit par rapport à la charge de trafic





**Figure 23.** Latence en fonction de la longueur de la route

**Latence.** Afin d'estimer le délai de bout-en-bout, un trafic constant de 8 paquets par seconde est appliqué au chemin reliant deux nœuds terminaux aléatoirement sélectionnés. Nous calculons en suite la latence de chaque communication. Le résultat est présenté dans la Figure 23 (le nombre de nœuds dans ce cas est de 100) et confirme l'avantage de l'utilisation de CoZi pour des applications à fortes contraintes en délai. En effet, grâce au mécanisme de codage opportuniste, les nœuds peuvent recevoir des paquets plutôt qu'avec un routage classique, ce qui raccourcit certaines routes et apporte ainsi une réduction substantielle de la latence subie par le réseau.

### 3.2.4. Discussion

Les nœuds dans les réseaux de capteurs ZigBee n'exploitent pas la nature partagée du médium sans fil. Ce travail présente CoZi, un mécanisme de codage distribué basé sur l'application de la théorie du codage réseau qui exploite ces liens partagés à travers le réseau. Grâce à l'algorithme de construction de la topologie hiérarchique propre à ZigBee, CoZi offre une meilleure utilisation de la bande passante avec une faible surcharge réseau. Il utilise un mécanisme d'inférence de topologie locale à partir des messages de contrôle ZigBee qui permet d'effectuer des opérations de codage opportuniste. CoZi aide à optimiser les stratégies de codage au niveau des nœuds intermédiaires, et cela pour garantir une plus grande probabilité de décodage de paquets.

### 3.3. Conclusion

Durant ce chapitre, nous avons étudié les apports en termes de performances des techniques de codage réseau linéaire et opportuniste pour la dissémination ainsi que le routage unicast dans les réseaux inter-machines. Ce premier travail consiste à enrichir des mécanismes d'acheminement de l'information en *store-and-forward* avec des méthodes de codage pour améliorer l'utilisation de la bande passante et assurer un meilleur délai de bout-en-bout dans ces réseaux.

Nous avons montré via WSC, que le codage linéaire est particulièrement adéquat dans le cas de réseaux inter-machines basés sur la dissémination des données sur des topologies maillées. De plus, grâce à CNC et CoZi nous avons montré l'avantage d'utiliser le codage opportuniste pour les topologies hiérarchiques courantes dans les réseaux inter-machines plus contraints. En effet, que ce soit pour la dissémination ou pour l'unicast, le codage opportuniste est plus approprié aux contraintes imposées par ces réseaux car il garantit un décodage systématique en saut par saut sur la route d'un paquet codé, ce qui n'est pas possible en utilisant un codage linéaire.

De manière plus générale, les diverses analyses de performances que nous avons effectué tout au long de ce chapitre indiquent une nette amélioration du débit et du délai lors de l'utilisation de techniques de codage réseau dans les différentes applications des réseaux de capteurs sans fil. Cette évaluation constitue la première étape dans notre étude de faisabilité du codage réseau sans fil dans les réseaux inter-machines.

Bien qu'il révèle des résultats intéressants et prometteurs, ce chapitre détaille une étude de performance dans un environnement idéal dénué de pertes. Dans la partie suivante, nous allons passer à l'analyse du comportement et de l'efficacité des mécanismes de codage réseau dans un environnement sans fil plus réaliste. Il est question dans ce prochain chapitre d'évaluer la tolérance des mécanismes de codage aux pertes et aux pannes avec comme point de comparaison la fiabilité des réseaux en *store-and-forward*.



## Chapitre 4

# 4. Codage réseau fiable

De nos jours, la théorie du codage réseau est de plus en plus présente dans des domaines très variés de l'informatique contemporaine. Bien que la vision que porte la communauté sur les réseaux de communication ait changé grâce à cette théorie, ses implications sur la façon dont nous étudions les architectures en *code-and-forward* est toujours sous-estimée. Plus précisément, nous constatons que même si beaucoup de chercheurs dans les réseaux informatiques sont maintenant familiers avec les avantages des codes opportunistes tels que COPE [47], leurs approches pour valider leurs travaux dans ce domaine restent très proche des méthodologies propres aux réseaux *store-and-forward*. Cela est dû à l'absence dans la littérature d'un cadre théorique dédié qui permette l'évaluation et la validation des systèmes de codage réseau.

De fait, il est indéniable que dans un environnement idéal le codage réseau opportuniste permet d'améliorer l'utilisation de la bande passante et de réduire la latence des réseaux maillés et hiérarchiques. Pourtant, il subit les mêmes défaillances que d'autres systèmes de codage basés sur des méthodes d'acquittement et de retransmissions venues du monde du *store-and-forward*. Ces défaillances existent parce que les codes proposés dans la littérature, qu'ils soient linéaires ou opportunistes, ne sont conçus que pour garantir une fiabilité dite « classique » des communications au lieu d'adopter des méthodes spécialement conçues pour le codage réseau.

Aussi, avant d'entamer une quelconque étude de fiabilité du codage inter-machines, nous devons dans un premier temps savoir si les méthodes d'appréciation de la fiabilité utilisées dans le *store-and-forward* sont suffisantes pour accomplir cette tâche. Concrètement, dans un réseau non-codé, garantir la fiabilité des communications consiste à maximiser le taux d'acheminement à un saut et de bout-en-bout tout en gardant des délais raisonnables et une certaine équité. Cela peut s'évaluer simplement en mesurant les délais et les taux de pertes de paquets de proche-en-proche et de bout-en-bout. Cependant, dans un système codé, l'acheminement réussi d'un paquet ne rime pas forcément avec succès de la communication, car un paquet codé qui arrive à son destinataire n'a de valeur que s'il peut être décodé.

Afin de remédier à ces insuffisances nous définissons dans ce chapitre la notion de *codage fiable*. Nous affirmons que ce concept constitue un élément important dans l'évaluation et la conception de nouveaux mécanismes de codage réseau. Étonnement, il n'a jamais (ou pas explicitement) été pris en compte dans la littérature actuelle, même s'il peut avoir un impact négatif sur l'efficacité des algorithmes de codage conçus pour les réseaux sans fil, où le médium est par nature très peu fiable. L'élément clé du codage fiable est de fournir un premier ensemble d'outils formels qui contribuent à combler le fossé entre l'implémentation du codage réseau et sa fiabilisation, et cela en répondant à des questions cruciales. Notamment :

1. Est-ce que les mécanismes de codage réseau subissent la même dégradation de performances que ceux en *store-and-forward* en présence de pertes de paquets ?
2. Est-ce que les métriques classiques de performances (débit, délai, taux d'acheminement) suffisent à évaluer les réseaux codés ?
3. Et finalement, peut-on évaluer la fiabilité des algorithmes de codage de la même manière que celle des systèmes en *store-and-forward* ?

Dans un second temps, nous proposons d'illustrer comment un codage *non fiable* peut réduire les performances des réseaux de capteurs sans fil et comment il est possible d'y remédier. Comme exemple empirique, nous montrons la façon dont les interférences et les collisions peuvent influencer sur le comportement de CoZi, le mécanisme de codage opportuniste proposé dans le chapitre précédent. Puis nous présentons ReCoZi une solution simple, et pourtant efficace, qui se base sur un mécanisme d'acquiescement par écho [91] et sur l'inférence de topologie locale pour améliorer la robustesse du codage réseau lors de présence de pertes non-négligeables. Enfin, nous montrons que ReCoZi surpasse nettement le routage en *store-and-forward* ZigBee en termes de taux d'acheminement avec cependant une légère baisse en termes de latence dans certains cas.

Ce chapitre est organisé comme suit. Dans la section 4.1 nous abordons le problème du codage fiable dans un réseau de capteurs sans fil. Nous y étudions le comportement des algorithmes de codage opportunistes face aux pertes de paquets et définissons la notion de codage fiable. Nous montrons que ce dernier permet d'offrir des communications plus robustes que ce soit de proche-en-proche ou de bout-en-bout. Nous développons ensuite dans la section 4.2 l'évaluation de ReCoZi un mécanisme de retransmission spécialement conçu pour assurer un codage réseau fiable quelques soient les conditions du médium dans un réseau de capteurs sans fil. Nous y montrons les performances et discutons son apport pour le codage opportuniste inter-machines. Enfin, la section 4.3 conclut le chapitre.

### 4.1. Codage Fiable

Il est important de comprendre que ce que nous définissons comme codage fiable diffère de la notion classique de fiabilité dans les réseaux de communication par paquets. En effet, la fiabilité dans les réseaux sans fil par exemple fait référence à une maximisation du taux d'acheminement des données en utilisant des mécanismes de retransmission pour permettre aux paquets de traverser le réseau et d'arriver sans erreurs à leur destination [93].

Un codage réseau fiable concerne plutôt la capacité d'un nœud à encoder des paquets de manière suffisamment efficace pour qu'un nombre maximum de ses prochains sauts puissent en extraire de l'information utile, et cela quelques soient les conditions du réseau (interférences, collisions, défaillances de nœuds, etc.). Ainsi, nous proposons les métriques suivantes qui définissent les principaux critères qu'un algorithme de codage doit satisfaire pour garantir des communications fiables et efficaces.

#### 4.1.1. Inférence de topologie

Cette métrique représente l'un des paramètres clés dans la fiabilisation des algorithmes de codage réseau, car une prédiction erronée de la disponibilité des paquets dans le voisinage d'un nœud peut conduire au choix d'un code sous-optimal voir à une combinaison de paquets non-decodable. La Figure 24 (a) montre un exemple dans lequel un nœud intermédiaire  $i$  fait une prédiction exacte de ce que ses voisins à un saut ont déjà reçu. Dans ce cas de figure, le nœud effectue une opération de codage optimale qui permet à l'intégralité des nœuds voisins de décoder le paquet. Inversement, les Figure 24 (b) et (c) illustrent des situations où  $i$  encode des paquets en se basant sur une inférence de topologie locale incorrecte causée par la perte de paquets de *feedback*, ce qui conduit respectivement, à un décodage partiel dans (b) et impossible dans (c).

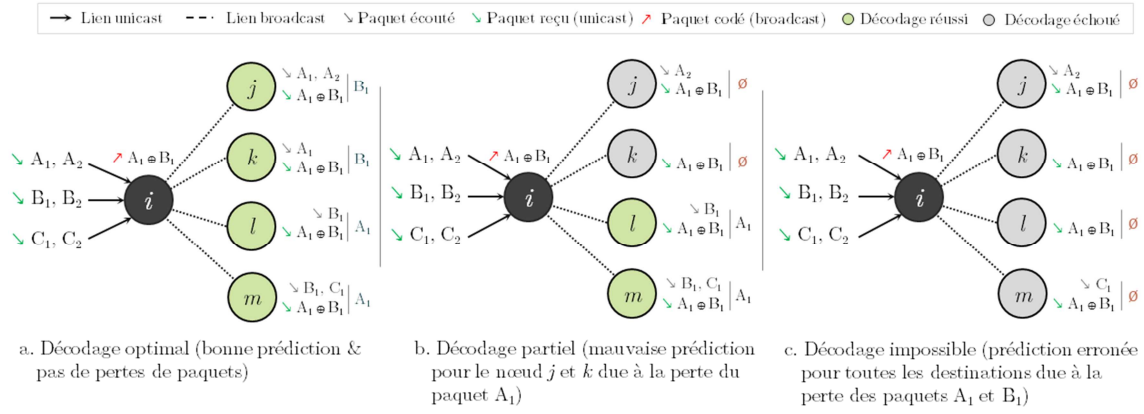
Plusieurs mécanismes existent pour permettre une inférence de topologie en quasi temps-réel dans un réseau de communication par paquets. Dans les réseaux statiques, la topologie environnante d'un nœud peut être facilement prédite à l'aide de méthodes telles que [51] ou [63] qui utilisent de simples heuristiques pour « deviner » ce que les nœuds voisins ont déjà reçu ou écouté. Ces techniques peuvent rapidement devenir inefficaces chaque fois que le canal de communication subit des pertes trop importantes qui font que le nœud intermédiaire n'arrive pas à faire des prédictions précises. Dans [47], les auteurs optent pour l'utilisation de rapports de réception afin d'informer un nœud codeur de l'ensemble de paquets disponibles au niveau de ses voisins. Cette approche est efficace parce que l'inférence de topologie est réalisée en temps réel. Toutefois, elle peut impliquer une surcharge de paquets de contrôle élevée, ce qui n'est pas souhaitable pour les réseaux de capteurs où le nombre de transmissions doit être minimisé.

#### 4.1.2. Décodabilité

Nous appelons décodabilité la probabilité  $Dec(c)$  qu'un code  $c$  sélectionné par un nœud  $s$  soit décodé par toutes ses destinations. L'équation suivante décrit la valeur de  $Dec$ .

$$Dec(c) = \begin{cases} \sum_{i=0}^{N(s)} \frac{P_{dec_i}(c)}{N(s)}, & N(s) \neq 0 \\ 0, & \text{sinon} \end{cases} \quad (7)$$

Avec  $P_{dec_i}(c) = P[dec_i(c) \neq \emptyset]$  la probabilité qu'un nœud  $i$  puisse extraire au moins un paquet lors du décodage de  $c$  et  $dec_i(c)$  qui est l'ensemble des paquets décodés par un nœud  $i$  à partir de  $c$ .  $N(s)$  est le nombre de nœuds qui sont présumés par  $s$  comme étant capables de décoder  $c$  (et non le nombre total de voisins).



**Figure 24.** Illustration de l'impact du taux de pertes sur l'efficacité du codage réseau

Les travaux antérieurs dans le codage réseau ont principalement concentré leurs efforts dans l'amélioration de la décodabilité. [94] par exemple étudie l'impact de la taille de la file d'écoute des paquets sur l'efficacité du codage et [95] analyse celui de l'ordonnancement sur le rendement des codes opportunistes. Même si la décodabilité est souvent désignée dans la littérature comme l'efficacité du codage, elle ne suffit pas à affirmer que le choix d'un code est optimal. En effet, une décodabilité élevée ne signifie pas nécessairement un codage réseau plus fiable.

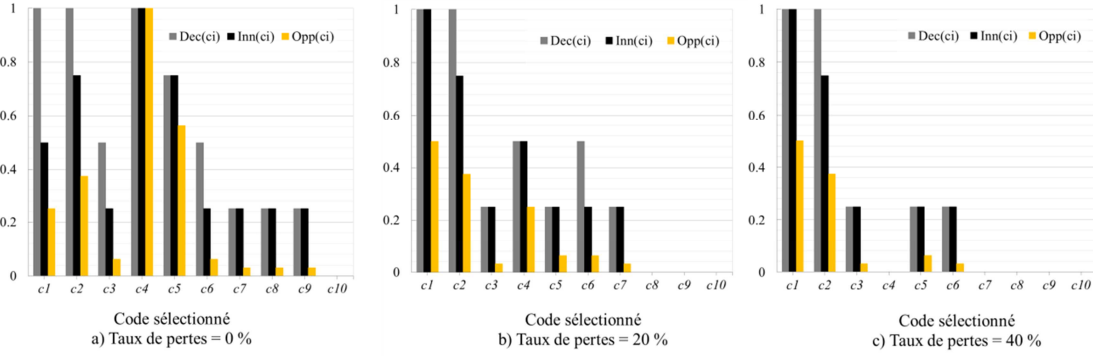
### 4.1.3. Innovativité

Par innovativité nous désignons la probabilité qu'un paquet codé  $c$  produise au moins un paquet utile chez toutes ses destinations. Cette valeur est définie dans la formule suivante :

$$Inn(c) = \begin{cases} \sum_{i=0}^{N(s)} \frac{P_{inn_i}(c)}{N(s)}, & N(s) \neq 0 \\ 0, & \text{sinon} \end{cases} \quad (8)$$

Avec  $P_{inn_i}(c) = P[dec_i(c) \notin (Q_{in}(i) \cup Q_{ovh}(i))]$ .  $P_{inn_i}(c)$  est la probabilité qu'un paquet  $c$  contienne de l'information utile pour un nœud  $i$ , i.e., que le paquet extrait de  $c$  n'existe ni dans sa file des paquets reçus  $Q_{in}(i)$ , ni dans celle des paquets écoutés  $Q_{ovh}(i)$ .

Un paquet est considéré comme utile s'il n'a jamais été reçu ou écouté par son destinataire auparavant. Utilisée pour augmenter le rang de la matrice de décodage dans le codage linéaire, l'innovativité est souvent considérée comme triviale dans le codage opportuniste car les nœuds reçoivent des rapports de réception de leurs voisins. Pourtant, lorsque les défaillances de liens augmentent et que des rapports de réceptions sont perdus, les messages codés contiennent moins de paquets innovants ce qui dégrade les performances du processus de codage. Ainsi, avant qu'un nœud intermédiaire ne décide quel paquet et à partir de quel flux l'inclure dans le prochain code à envoyer, il doit savoir si ce paquet sera innovant après son décodage à sa (ses) destination(s). Ceci peut être garanti si ce nœud sait (ou estime) ce que ses voisins ont déjà reçu ou écouté. Notons que plus grande est la valeur de l'innovativité, plus grande doit être la probabilité pour qu'un paquet soit inclus dans le prochain code à transmettre.



**Figure 25.** Illustration de l'impact du taux de pertes sur la fiabilité d'un code

L'exemple illustré dans la Figure 24 montre des situations où le même paquet codé différemment a plusieurs valeurs d'innovativité selon le taux de pertes.

Figure 24 (a) :  $Inn(p_a) = 1$ , lorsqu'il n'y a pas de pertes (0%).

Figure 24 (b) :  $Inn(p_b) = \frac{1}{2}$ , quand le paquet  $A_1$  est perdu (20%).

Figure 24 (c) :  $Inn(p_c) = 0$ , quand  $A_1$  et  $B_1$  sont perdus (40%).

#### 4.1.4. Optimalité

Cette valeur représente, pour un code donné, la conjonction de sa décodabilité et de son innovativité, i.e., un code  $c$  (une combinaison de paquets) est dit optimal si et seulement si ce dernier permet le décodage d'un nombre maximum de paquets et qu'il offre des valeurs maximales de décodabilité et d'innovativité. Un paquet codé peut en effet être décodable par toutes ses destinations et pourtant n'apporter aucune innovativité à aucune d'entre-elles, ou au contraire, il peut contenir des paquets innovants pour ses destinations, mais non décodables par ces dernières, affectant ainsi la fiabilité du codage. L'équation suivante donne une formalisation de l'optimalité d'un code quelconque  $c_i$ .

$$Opt(c_i) = Max \left[ Dec(c_i) \times Inn(c_i) \times \frac{N_{dec_i}}{N_{pos_i}} \right] \quad (9)$$

Où  $N_{dec_i}$  est le nombre de paquets distincts effectivement décodés à partir de  $c_i$  et  $N_{pos_i}$  le nombre de paquets potentiellement décodables à partir de  $c_i$ . Améliorer l'optimalité d'un code consiste donc en la maximisation d' $Opt(c_i)$ , c'est à dire permettre à l'ensemble de paquets sélectionnés d'être décodé par un maximum de destinataires, et que ces derniers puissent en extraire de l'information utile.

Par souci de clarté, nous énumérons dans le tableau 4 différentes valeurs de décodabilité et d'innovativité pour  $s = \{c_i\}$ , un ensemble de 10 stratégies de codage établies par un nœud intermédiaire quelconque. Ces valeurs correspondent à la topologie présentée dans l'exemple canonique de la Figure 24, avec  $L_0$ ,  $L_1$  et  $L_2$  qui représentent les taux de pertes des configurations illustrées dans les Figure 24 (a), (b) et (c). La Figure 25 révèle un constat important, à savoir, l'impact du taux de pertes sur la fiabilité du codage réseau. En effet, on note une baisse moyenne de plus de 50% de la valeur de  $Opt$  lorsque le taux de pertes atteint 20%. Ceci suggère que l'application de décisions de codage qui ne prennent



pas en compte les conditions du médium peut s'avérer inefficace car la meilleure stratégie de codage n'est pas forcément de transmettre le code avec l'optimalité *idéale* sans pertes.

**Tableau 4.** Taux de pertes et optimalité du codage

Possibilité de codage	$Dec(c_i)$			$Inn(c_i)$		
	$L_0$	$L_1$	$L_2$	$L_0$	$L_1$	$L_2$
$c_1 = A_1$	1	1	1	$\frac{1}{2}$	1	1
$c_2 = A_2$	1	1	1	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{3}{4}$
$c_3 = A_1 \oplus A_2$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$
$c_4 = A_1 \oplus B_1$	1	$\frac{1}{2}$	0	1	$\frac{1}{2}$	0
$c_5 = A_1 \oplus C_1$	$\frac{3}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{3}{4}$	$\frac{1}{4}$	$\frac{1}{4}$
$c_6 = B_1 \oplus C_1$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$
$c_7 = A_1 \oplus B_1 \oplus C_1$	$\frac{1}{4}$	$\frac{1}{4}$	0	$\frac{1}{4}$	$\frac{1}{4}$	0
$c_8 = A_1 \oplus A_2 \oplus B_1$	$\frac{1}{4}$	0	0	$\frac{1}{4}$	0	0
$c_9 = A_1 \oplus A_2 \oplus C_1$	$\frac{1}{4}$	0	0	$\frac{1}{4}$	0	0
$c_{10} = A_1 \oplus B_2 \oplus C_2$	0	0	0	0	0	0

#### 4.1.5. Équité

Alors que le codage réseau améliore sans équivoque l'utilisation de la bande passante, il peut aussi pénaliser les flux selon la façon dont l'algorithme de sélection de paquets se comporte. Par exemple, si un nœud intermédiaire se concentre uniquement sur l'optimalité d'un code, il peut en découler une détérioration de l'équité par rapport aux différents flux de données qu'il se doit d'acheminer. En fait, en sélectionnant systématiquement des paquets à partir des mêmes flux de données à chaque opportunité de codage, le nœud peut retarder la transmission d'autres flux affectant ainsi considérablement la latence de certaines communications de bout-en-bout. Par conséquent, nous définissons  $Fair_i(f_j)$  la probabilité que le prochain paquet codé  $c$  à transmettre à partir du nœud  $i$  contienne au moins un paquet du flux de données  $f_j$ .

$$\forall i \in N, j \in InFlows(i),$$

$$Fair_i(f_j) = P[c \in Q_{out}(i) \mid \exists p \in f_j, p \oplus c = c - p] \quad (10)$$

Où  $Q_{out}(i)$  est l'ensemble des paquets à transmettre par le nœud  $i$ ,  $InFlows(i)$  l'ensemble des flux entrants  $f_j$  et  $p$  un paquet du flux  $f_j$ .

Il est à noter que lorsqu'un algorithme de codage effectue une sélection optimale de paquets, plus la valeur de  $Fair_i$  est élevée pour un flux  $f_j$ , plus grand est le débit auquel il doit s'attendre. Garantir un accès équitable au médium est usuellement effectué par de simples mécanismes de gestion de files d'attente. Pourtant, dans un réseau codé, comme l'ordonnancement des paquets dépend de la stratégie de codage, l'équité des flux de données nécessite des solutions plus sophistiquées. En effet, si un simple système FIFO est souvent suffisant pour garantir un accès équitable à la file d'attente de transmission dans les réseaux *store-and-forward*, il est impossible à mettre en œuvre pour le codage réseau. Cela est dû au fait que dans un réseau codé, les paquets ne sont pas sélectionnés dans leur ordre d'arrivée, mais plutôt en fonction des opportunités de codage optimal. A notre connaissance, le sujet de l'équité dans le codage réseau n'est pas traité dans la littérature,

bien que cette métrique puisse considérablement dégrader la latence de bout-en-bout dans les systèmes sans fil.

#### 4.1.6. Taux d'acheminement de bout-en-bout

Cette métrique, de la même manière que l'équité, concerne les communications de bout-en-bout. Il s'agit d'un critère classique d'évaluation de la fiabilité des réseaux de communication en général. Il implique que les algorithmes de codage doivent assurer un ratio minimum d'acheminement des paquets. L'équation suivante détaille ce ratio :

$$r(i, j) = \frac{\text{Nombre de paquets reçus par } j}{\text{Nombre de paquets transmis par } i} \quad (11)$$

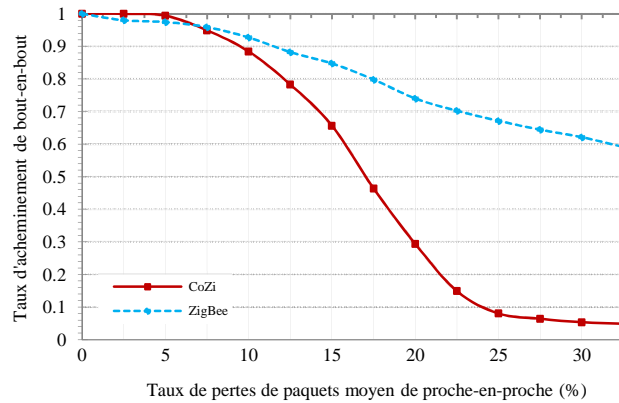
Où  $i$  est la source et  $j$  la destination lors d'une session de communication de bout-en-bout entre les nœuds  $i$  et  $j$ . Notons que selon l'application du réseau, les protocoles peuvent être contraints d'assurer un taux d'acheminement minimum  $r$  afin de garantir une certaine qualité de service.

#### 4.1.7. Discussion

Même si elles peuvent comporter quelques métriques similaires comme le taux d'acheminement de bout-en-bout ou l'équité inter-flux, la notion de fiabilité classique reste assez différente du concept de codage fiable introduit dans cette première partie. Notre analyse confirme cette affirmation et met en exergue la nécessité de prévoir des mécanismes spécifiques pour la fiabilisation des réseaux codés. Puisque le concept de codage fiable n'a pas encore été traité dans la littérature, nous présentons dans ce premier travail une série de solutions prévues pour améliorer la fiabilité des réseaux de capteurs sans fil basés sur le codage opportuniste. Par ailleurs, les exigences de ces algorithmes de codage en termes de fiabilité diffèrent non seulement en fonction de l'application du réseau, mais aussi selon les composantes spécifiques à l'algorithme, telles que son mécanisme d'inférence de topologie, les stratégies de codage ou encore le système de retransmission. Ainsi, les solutions que nous proposons se doivent d'être adaptées en tenant compte de ces composantes. Un exemple de cela est donné avec ReCoZi, un mécanisme de codage fiable qui étend le codage opportuniste ZigBee (CoZi) proposé dans le chapitre précédent.

## 4.2. Codage ZigBee Fiable – ReCoZi

Afin d'effectuer un acheminement fiable des paquets, des mécanismes de retransmission sont souvent nécessaires dans les réseaux sans fil en général et dans les réseaux de capteurs en particulier. Ces approches ont montré une efficacité remarquable dans plusieurs protocoles de communication (802.15.4 [20], ZigBee [15], B-Mac [96], etc.). Toutefois, lorsque le codage réseau est utilisé, ces mécanismes de retransmission par acquittement peuvent devenir insuffisants en raison des caractéristiques spécifiques des algorithmes de codage. En effet, dans ces systèmes les paquets ne peuvent pas être simplement acquittés à chaque fois qu'ils sont reçus ou écoutés, parce que des opérations supplémentaires doivent être effectuées pour décider si le paquet est exploitable ou pas.



**Figure 26.** Taux d'acheminement vs. taux de pertes

Il est clair grâce à la Figure 26, qui évalue via simulation comment le taux de pertes impacte CoZi, que notre mécanisme de codage opportuniste souffre d'un déclin de performances bien plus rapide que ZigBee lorsque le canal radio devient erratique. Cela est dû au fait que le système d'acquiescement de CoZi est conçu pour assurer une fiabilité classique des communications et non pour garantir un codage fiable tel que défini dans la section précédente. Par exemple, un paquet codé qui a été décodé par un sous-ensemble de ses destinataires mais pas par le prochain saut sur sa route peut être considéré comme acheminé (car les paquets ACK confirment sa réception) et n'est donc jamais retransmis provoquant un chemin *sans issue* pour un flux de paquets donné. Avec l'augmentation du taux de pertes, ce phénomène devient de plus en plus fréquent menant à la prolifération de ces chemins sans issue et affectant de façon drastique le taux d'acheminement du réseau. De plus, CoZi effectue souvent des décisions de codage erronées à cause d'une mauvaise inférence de topologie causée par la perte de paquets. En fait, les nœuds codeurs ne possèdent aucun moyen pour savoir si un paquet a été perdu à cause des conditions du médium ou d'une mauvaise stratégie de codage, ce qui affecte la décodabilité et l'innovativité des codes sélectionnés à chaque transmission.

Ainsi, nous considérons comme cas d'étude et durant le reste de ce chapitre, le problème du codage fiable dans les réseaux de capteurs ZigBee basés sur la couche MAC IEEE 802.15.4 et le codage opportuniste CoZi. Nous proposons ReCoZi, une série de solutions pour faire face aux défaillances de codage provoquées par l'instabilité du canal de communication. Basé sur notre définition du codage fiable, nous montrons comment grâce à un mécanisme d'acquiescement adapté et à une plus grande connaissance de l'état de liens, ReCoZi conserve à la fois le gain de performance du codage réseau et la fiabilité des communications lors de la présence de pertes.

#### 4.2.1. Système d'acquiescement des paquets

Dans la section précédente nous avons défini une série de paramètres pour évaluer la fiabilité des algorithmes de codage réseau. En conséquence, et afin d'atténuer l'impact que peut avoir un canal de communication non fiable sur l'efficacité du codage, nous présentons un système de retransmission/recodage par acquiescement adapté aux algorithmes de codage opportuniste dans les réseaux ZigBee.

---

**Algorithme 2.** Réception d'un paquet par un routeur  $A$ 

---

```

1: for each incoming packet  $p$  in  $Q_{in}(A)$  do
2:   if  $p$  is non-coded then
3:     if  $A$  is dest. of  $p$  and is not the final hop then
4:        $A$  adds  $p$  to its next echo-ACK packet
5:     else
6:       if  $A$  is dest. of  $p$  and is the final hop then
7:          $A$  sends an ACK to the sender
8:       else /*  $A$  overheard  $p$  */
9:          $p$  is stored in  $Q_{ovh}(A)$ 
10:      end if
11:    else /*  $p$  is coded */
12:      if  $p$  is decodable then
13:         $p_{dec} = Decode(p)$ 
14:        if  $A$  is dest. of  $p_{dec}$  and is the final hop then
15:           $A$  sends an ACK to the sender
16:        else
17:          if  $A$  is dest. of  $p_{dec}$  and is not the final hop then
18:             $A$  adds  $p_{dec}$  to its next echo-ACK packet
19:          else /*  $A$  overheard  $p$  */
20:             $p_{dec}$  is stored in  $Q_{ovh}(A)$ 
21:             $A$  adds  $p_{dec}$  to its next echo-ACK packet
22:          end if
23:        else /*  $p$  is not decodable */
24:           $p$  is dropped from  $Q_{in}(A)$ 
25:        end if
26:      end if
27:    end for

```

---

Au lieu d'utiliser le même système d'acquittement pour l'ensemble des paquets, nous proposons de modifier CoZi de sorte que seules les réceptions importantes soient explicitement acquittées (i.e. les paquets qui ont atteint leur destination finale) tandis que celles au niveau des nœuds intermédiaires ne sont que passivement acquittées en utilisant un système d'acquittement par écho<sup>4</sup> [97]. Ainsi, afin d'éviter toute détérioration significative de la fiabilité due à la perte de paquet d'acquittement, nous ajoutons un champ *echo-feedback* dans les paquets codés sortants qui permet d'informer les nœuds voisins à la fois de tout paquet reçu ou écouté et aussi des paquets éventuellement perdus ou impossible à décoder. L'algorithme 2 décrit plus en détail ce processus de retransmission. Les lignes 4, 18 et 21 montrent que l'acquittement par écho est utilisé à chaque fois qu'un paquet est décodé par un nœud intermédiaire (qu'il soit reçu ou écouté). Cette approche passive d'acquittement consiste à valider la réception d'un paquet par sa destination à un saut en remplissant le champ ACK dans l'entête du paquet suivant à transmettre, à l'exception du dernier nœud sur la route du paquet où un acquittement explicite doit être généré (cf. lignes 7 et 15).

---

<sup>4</sup> L'acquittement par écho est possible grâce à la propriété omnidirectionnelle du broadcast dans les liens sans fil

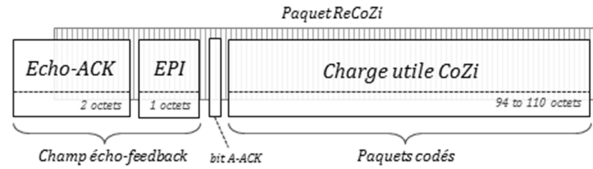


Figure 27. Champs supplémentaires nécessaires à l’acquittement par écho

Comme illustré sur la Figure 27, nous faisons usage du concept de *piggybacking* pour notifier les nœuds transmetteurs du prochain paquet attendu par leurs voisins. En effet, grâce à l’incorporation dans l’entête ReCoZi du champ EPI (Expected Packet Id) qui contient le numéro de séquence du prochain paquet attendu par le nœud, il est possible pour un nœud intermédiaire d’ajouter l’identifiant du prochain paquet qu’il s’attend à recevoir. Ainsi, ses voisins pourront effectuer des décisions de codage plus précises en vérifiant ce champ et en intégrant dans leur code le paquet manquant, s’il est disponible dans leur file d’attente. Par exemple, considérons trois nœuds intermédiaires  $i$ ,  $j$  et  $k$  sur la route d’un paquet codé  $p$ . Après réception de  $p_i$  par  $j$  et avant qu’il ne le relaye à  $k$  si le prochain paquet attendu par le nœud  $j$  est  $p_{i+1}$ , le champ EPI contiendra le numéro de séquence de  $p_{i+1}$  de sorte que le nœud émetteur  $i$  soit au courant en écoutant le canal que  $p_{i+1}$  n’est pas encore reçu par le nœud  $j$ .

Cette technique permet d’effectuer une meilleure inférence de topologie locale, améliorant la décodabilité et l’innovativité des codes sélectionnés. L’utilisation du *piggybacking* permet aussi la diminution de la surcharge réseau induite par l’envoi de paquets ACK à chaque phase de décodage. En conséquence, les nœuds codeurs peuvent constamment savoir quand un paquet est perdu, et détecter quels codes n’ont pas pu être décodés par quels nœuds, ce qui permet une meilleure gestion des retransmissions et du recodage des paquets, et qui résout en partie le problème des chemins sans issue dans les communications de bout-en-bout. Par ailleurs, un vrai gain sur le nombre de transmissions peut être constaté car beaucoup moins de paquets sont dupliqués utilement lors de la présence d’un fort taux de pertes (pertes de paquets de données ou pertes d’ACK).

#### 4.2.2. Acquittement à état de lien

La préservation du taux d’acheminement des données dans les réseaux codés lors de la présence de pertes non négligeables ne peut se faire que grâce à des mécanismes supplémentaires spécialement conçu pour le codage réseau. Cependant, lorsque les performances du codage opportuniste sont satisfaisantes parce que le canal est fiable, il n’est nullement nécessaire de recourir aux mécanismes d’acquittement décrit plus haut, et cela pour optimiser l’utilisation de la bande passante disponible. Par contre, à mesure que la fiabilité du réseau se dégrade et que l’inférence de topologie devient moins précise, les mécanismes d’acheminement doivent être enrichis avec des méthodes d’acquittement qui assurent un codage réseau plus fiable dans un environnement perturbé. L’utilisation du champ RSSI<sup>5</sup> dans la trame 802.15.4 combiné au taux d’erreur mesuré (BER) à la

<sup>5</sup> Received Signal Strength Indication, ou indicateur de la puissance de signal reçu

réception permet à chaque nœud intermédiaire d'évaluer localement la qualité de ses liens. Grâce à ces valeurs, un nœud peut facilement déclencher l'utilisation des mécanismes d'acquittement avancés afin de faire face à une éventuelle dégradation du canal de communication. Dans le cas de ReCoZi, cela est effectué en attribuant la valeur 1 au bit A-ACK (cf. Figure 27) à chaque fois que les valeurs du *RSSI* et du *BER* descendent sous un seuil prédéfini, signifiant aux autres nœuds que les mécanismes d'acquittement avancés sont activés. Bien évidemment, afin d'éviter toute activation fortuite de ce mode, les nœuds calculent une valeur moyenne pour un nombre  $N$  de paquets reçus avant de vérifier si le seuil  $T$  a été atteint. L'équation suivante détaille ce procédé :

$$\left[ c_1 \cdot \left( \frac{\sum_N RSSI_i}{N} \right) + c_2 \cdot \left( \frac{\sum_N BER_i}{N} \right) \right] < T \Rightarrow A\_ACK = 1 \quad (12)$$

Où  $c_1$  et  $c_2$  sont des coefficients de pondération utilisés pour donner plus ou moins d'importance soit au *RSSI* ou au *BER* dans l'estimation de l'état de liens. Grâce à cette solution simple, ReCoZi fournit des estimations des conditions du canal de communication plus réactives de sorte que la fiabilité soit maintenue en utilisant les mécanismes d'acquittement aussi longtemps que le medium sans fil reste perturbé.

#### 4.2.3. Gestion équitable des files d'attente de codage

Comme mentionné précédemment, pour être considéré comme fiable, un mécanisme de codage réseau doit fournir un taux d'acheminement élevé ainsi qu'une équité entre les différents flux de données. Nous avons pu noter lors de l'évaluation de CoZi un écart important entre les délais propres à chaque communication de bout-en-bout, et cela parce que les mécanismes de codage opportuniste ne font aucune distinction entre les flux puisque souvent l'efficacité du codage prévaut. En effet, il arrive fréquemment que l'algorithme d'ordonnancement sélectionne des paquets du même sous-ensemble de flux, en négligeant d'autres communications de bout-en-bout (surtout lors de la présence de pertes) menant à l'expiration de la durée de vie du paquet en nombre de saut (*Radius* dans ZigBee), ce qui affecte clairement le taux d'acheminement moyen du réseau. Pour faire face à ce problème, ReCoZi comprend un mécanisme de gestion de file d'attente de codage grâce auquel chaque nœud supervise sa file de transmission, vérifiant si les paquets d'un flux donné sont restés une période de temps déterminée sans jamais être intégrés dans un code et de ce fait, transmis. Dans ce cas, le nœud en question suspend toutes les opérations de codage en cours afin d'envoyer ces paquets lors de la prochaine opportunité de transmission. Cette solution évite d'appliquer des stratégies de codage inéquitables qui réduisent le taux d'acheminement de certaines communications de bout-en-bout.

#### 4.2.4. Analyse de performance

Dans cette section nous évaluons la fiabilité du codage ReCoZi dans différentes conditions radio. Nous quantifions l'optimalité de l'algorithme et le taux d'acheminement qu'il offre. Puis nous comparons ses performances (délai et débit) avec CoZi et un routage ZigBee standard. Enfin, nous montrons la surcharge de paquets de contrôle induite par les mécanismes de retransmissions propres à ces trois systèmes.

Tableau 5. Paramètres de simulation

<b>Durée de simulation</b>	600 s
<b>Topologie</b>	Hierarchique (ZigBee) BO = 15   SO = 15
<b>Routage</b>	ZigBee cluster-tree routing
<b>Protocole de couche MAC</b>	IEEE 802.15.4
<b>Capacité</b>	250 kbps
<b>Modèle de propagation</b>	Lognormal shadowing (M = 4.5 dB)
<b>Modèle d'atténuation</b>	Rayleigh/ Longley-Rice
<b>Rayon de transmission</b>	~35 mètres
<b>Type de trafic</b>	CBR variable
<b>Période de mise en tampon</b>	500 ms
<b>Taille des paquets</b>	116 octets
<b>Nombre de routeurs</b>	80%
<b>Nombre de nœud terminaux</b>	19%

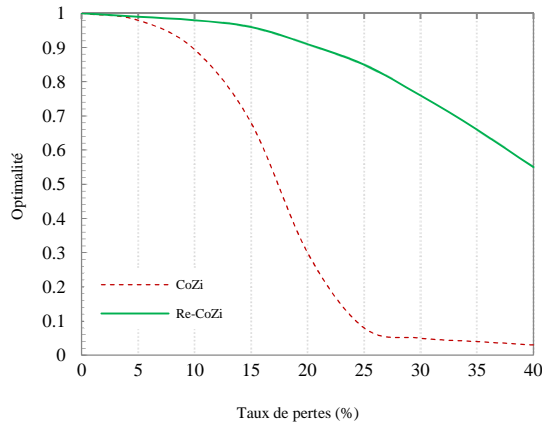
#### 4.2.4.1. Environnement de simulation

Nous utilisons la plateforme de simulation Qualnet 4.5 [84] pour évaluer notre solution. Où ReCoZi, CoZi, ZigBee et la couche MAC 802.15.4 sont implémentés. Nous considérons un réseau de 100 nœuds uniformément positionnés dans une surface de 150x150m. Le coordinateur ZigBee est placé au centre du réseau et le reste des nœuds sont soit des routeurs soit des nœuds terminaux. Chaque communication à un débit qui varie de 25 à 100 *paquets/s*. Tous les paramètres de simulation sont résumés dans le tableau 5.

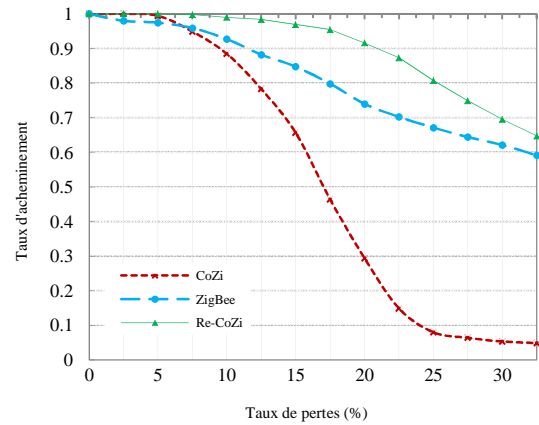
Afin d'évaluer la fiabilité de ReCoZi et de le comparer à CoZi et ZigBee, nous appliquons des taux de pertes différents au réseau et calculons les résultats sur la base des métriques suivantes : l'optimalité du codage, le taux d'acheminement des paquets, le délai, le débit et la surcharge du réseau.

#### 4.2.4.2. Optimalité

Dans un premier temps, nous calculons l'optimalité par rapport au taux de pertes de paquets. Ce calcul se fait en utilisant la décodabilité, l'innovativité et le nombre de paquets décodés pour un code donné à chaque nœud intermédiaire. Nous pouvons constater sur la Figure 28 que ReCoZi améliore considérablement l'optimalité des codes sélectionnés, même lorsque le canal devient très erratique. Cela est dû à l'utilisation des mécanismes de retransmission décrits plus haut, où les décisions de codage sont plus précises grâce notamment à l'activation de l'acquiescement par écho qui permet une inférence de topologie plus réaliste du nœud en question. Il est à noter que nous ne calculons pas la valeur de *Opt* pour ZigBee car aucun codage n'est effectué dans ce cas.



**Figure 28.** Valeur de l'optimalité et taux de pertes de paquets



**Figure 29.** Taux d'acheminement et taux de pertes de paquets

#### 4.2.4.3. Taux d'acheminement

Nous pouvons voir sur la Figure 29 que le taux d'acheminement fourni par ReCoZi surpasse à la fois ZigBee et CoZi à mesure que le canal subit plus de pertes. En fait, avec l'activation des méthodes d'acquiescement avancés, ReCoZi effectue des décisions de codage plus précises qui limitent le phénomène des chemins sans issue et permettent de retransmettre efficacement les paquets codés perdus en vérifiant le champ Echo-ACK des paquets reçus et écoutés. Notons également que ReCoZi fournit un taux d'acheminement plus élevé que celui offert par ZigBee. Ceci est une conséquence directe de l'utilisation du codage réseau au niveau des nœuds intermédiaires du chemin, puisque l'utilisation de la bande passante est optimisée avec plusieurs paquets relayés en une seule transmission. Néanmoins, lorsque le taux de pertes de paquets atteint un certain seuil et que les opportunités de codage se raréfient, les valeurs du taux d'acheminement constatées avec ReCoZi et ZigBee convergent ce qui est normal car les deux systèmes fonctionnent similairement dans ces conditions.

#### 4.2.4.4. Délai de bout-en-bout et équité

La Figure 30 montre que la latence moyenne des communications lors de l'utilisation de ReCoZi est légèrement supérieure à celle de ZigBee mais toutefois nettement moins importante que celle offerte par CoZi et cela lorsque le réseau endure des pertes élevées de paquets. Si ReCoZi assure un délai de bout-en-bout inférieur à CoZi, c'est parce que via sa gestion différente de la file de codage, il limite l'apparition de chemins sans issue causés par les systèmes d'acquiescement inadéquats au codage réseau et qu'il assure plus d'équité que son homologue : ces deux problèmes sont les causes directes de la présence de délais supplémentaires dans les réseaux codés. En effet, ils retardent significativement certains flux de paquets affectant la latence moyenne de bout-en-bout dans le réseau. Il est aussi à noter que dans un contexte plus idéal (sans pertes), ReCoZi conserve les mêmes performances que CoZi.



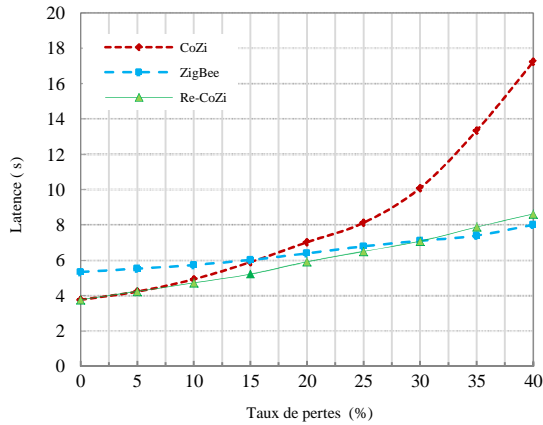


Figure 30. Latence et taux de pertes

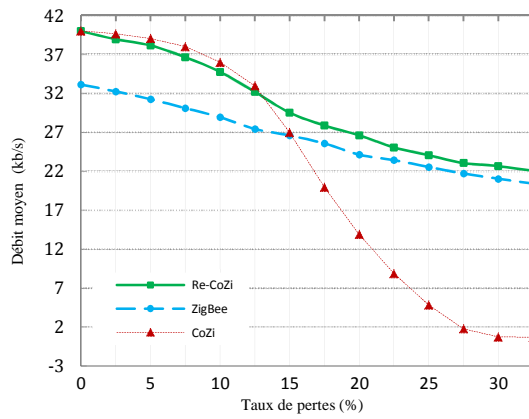


Figure 31. Débit et taux de pertes

#### 4.2.4.5. Débit effectif

Le débit effectif<sup>6</sup> global est augmenté grâce à l'amélioration de l'optimalité du codage. En fait, comme le montre la Figure 31, ReCoZi évite que le débit ne s'effondre comme c'est le cas dans CoZi, grâce à l'activation de l'acquittement à état de lien qui permet d'éviter les mauvaises décisions de codage. De plus, nous remarquons que le débit global obtenu par ReCoZi est presque identique à celui de ZigBee lorsque le taux de perte dépasse 30%. Cela est dû au fait que l'optimalité des codes sélectionnés diminue drastiquement avec l'augmentation des pertes de paquets, ce qui réduit l'ensemble des opportunités de codage possibles et donc ne permet d'envoyer, dans la plupart des cas, que des paquets non codés.

#### 4.2.4.6. Surcharge réseau

La Figure 32 illustre la croissance du nombre de paquets de contrôle nécessaires pour ZigBee, CoZi et ReCoZi. Cette surcharge réseau inclut les messages d'acquittement explicites et les paquets de feedback nécessaires à l'inférence de topologie. Elle représente la fraction de la bande passante utilisée par ces paquets de contrôle et les en-têtes 802.15.4. Puisque ReCoZi n'utilise que peu de paquets d'acquittement explicites, il requiert beaucoup moins de paquets de contrôle que CoZi et ZigBee. Cependant, comme il se base sur des en-têtes spéciaux au niveau de la trame 802.15.4, la charge utile des paquets ReCoZi est plus petite que dans CoZi. Ceci explique le fait que le débit de ReCoZi est légèrement en deçà de celui de CoZi dans la Figure 31 (dans les cas où le taux de pertes est inférieur à 12.5%).

Il est aussi à noter que la réduction de la surcharge réseau est liée à la réduction du nombre global de transmissions grâce au codage réseau, ce qui est en soi une amélioration importante dans la mesure où dans les réseaux de capteurs sans fil l'utilisation du module radio doit être minimisée.

<sup>6</sup> Le débit effectif est calculé au niveau des nœuds destinataires, il n'inclut pas les retransmissions, les paquets ACK et les paquets redondants.

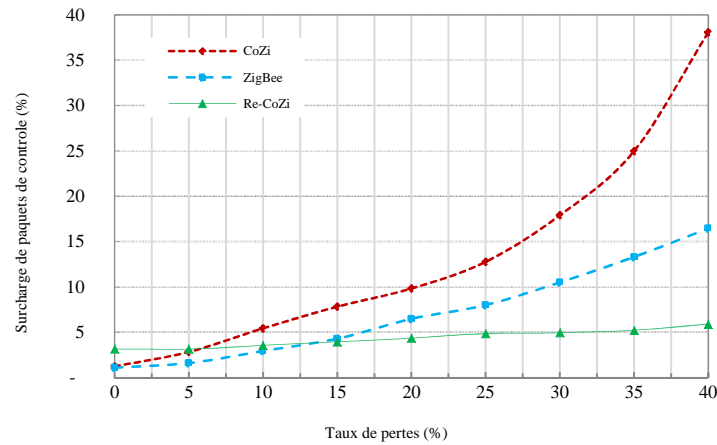


Figure 32. Surcharge réseau et taux de pertes

### 4.3. Conclusion

Dans ce chapitre, nous montrons que les méthodes de fiabilisation inspirées des systèmes *store-and-forward* ne sont pas adaptées au codage réseau en raison de ses spécificités (ordonnancement, écoute de paquets, décodabilité, etc.). Afin de résoudre ce problème, nous introduisons la notion de codage fiable : une série de métriques qui permettent la conception de systèmes de codage réseau plus robustes, i.e. qui garantissent un traitement équitable des flux de données, un taux d'acheminement optimisé et un ordonnancement efficace des paquets quel que soit l'état du médium sans fil. Afin notamment de démontrer l'intérêt de ces métriques, nous proposons ReCoZi, un mécanisme de codage fiable pour les réseaux de capteurs ZigBee. Ce code opportuniste utilise une méthode de retransmission basée sur l'état des liens afin de garantir des communications codées plus tolérantes aux pertes. Enfin, nous évaluons les performances et la robustesse de ReCoZi et montrons qu'il pallie à la problématique de vulnérabilité aux pertes inhérente aux protocoles de codage réseau sans fil en général.

Les résultats de notre étude de performance et de fiabilité montrent que ReCoZi répond à l'objectif fixé qui est de garder la valeur ajoutée du codage réseau même lorsque le canal de communication devient erratique. Grâce à de simples mécanismes de retransmissions adaptés aux communications codées, ReCoZi permet d'effectuer un codage robuste plus tolérant aux pertes, avec une faible surcharge réseau et un accès équitable au médium. En gardant la valeur ajoutée du codage réseau, il permet aussi de réduire le nombre de transmissions, de mieux exploiter la bande passante disponible et ainsi de réduire les délais de bout-en-bout.

Cet effort dans l'amélioration de la conception, de la validation et de l'évaluation des techniques de codage fiables s'inscrit comme une première formalisation qui permet aux concepteurs de systèmes de codage réseau de mieux appréhender l'étude de performance et de fiabilité de leur solution. Grâce à cette formalisation, il devient plus facile de quantifier à la fois les apports et les défaillances des protocoles de communication basés sur le codage réseau sans fil.

Toutefois, la fiabilisation des mécanismes de codage réseau n'est pas le seul problème auquel les réseaux de capteurs sans fil sont confrontés. L'adaptabilité des protocoles de communication dans ces architectures, qu'ils soient codés ou pas, reste l'élément clé qui permet d'établir leur viabilité dans un environnement aussi contraint que celui des capteurs sans fil. En effet, un protocole de communication pour les réseaux inter-machines n'est efficace que s'il est fiable, qu'il produit des performances élevées et qu'il peut fonctionner en consommant le moins de ressources possible au niveau des nœuds qui composent le réseau.

Ainsi, le chapitre suivant continue dans cet effort d'adaptation du codage réseau aux contraintes des réseaux de capteurs sans fil. Nous y proposons un système opportuniste de codage par index pour des réseaux très contraints et étudions son rendement par deux analyses, la première théorique qui prouve l'efficacité de son algorithme de codage, et la deuxième empirique, qui montre son adaptabilité dans un vrai réseau de capteurs sans fil.





## *Chapitre 5*

# 5. Codage réseau contraint

Les principaux défis lors de la conception et du déploiement d'un réseau de capteurs sans fil restent le prolongement de la durée de vie du réseau et l'amélioration de sa fiabilité tout en gardant des performances décentes en termes de débit et de latence. Les deux chapitres précédents ont montré que de tels défis peuvent être abordés avec l'utilisation du codage réseau sans fil, précisément dans le cas des applications multi-sources ou multi-destinations. Pourtant, sans une adaptation réfléchie aux contraintes en ressources et en débit des réseaux de capteurs, le bilan coûts/bénéfices du codage réseau peut s'avérer négatif.

Aujourd'hui, l'application de la théorie du codage réseau fournit un ensemble d'outils qui permettent une réduction importante des transmissions dans un réseau sans fil arbitraire [37]. Les deux chapitres précédents en plus de la littérature existante sur le sujet tendent à le démontrer. Une façon attrayante de considérer les problématiques susmentionnées et d'améliorer les communications des réseaux contraints serait donc d'exploiter les concepts innovants du codage réseau. Les progrès accomplis dans l'application de ces concepts ont attiré l'attention des chercheurs et concepteurs de réseaux informatiques parce qu'ils promettent une amélioration de l'utilisation de la bande passante en résolvant le problème de la sous-optimalité inhérente aux protocoles traditionnels en *store-and-forward* [57]. Cela dit, l'application des techniques de codage réseau à des architectures aussi contraintes en mémoire et en calcul est loin d'être une tâche triviale. En fait, une utilisation *telle quelle* de techniques de codage existantes telles que COPE [47] ou le codage aléatoire [46] dans les réseaux de capteurs serait inefficace, voire impossible en pratique, car elles requièrent une surcharge par paquet non négligeable et un impact substantiel sur l'espace mémoire nécessaire à la pile protocolaire de chaque nœud.

Dans ce chapitre, nous abordons ce problème d'adaptabilité du codage réseau aux limitations des réseaux de capteurs. Nous y définissons un mécanisme de codage réseau contraint appelé codage par index (InC) qui améliore les communications à partir de plusieurs sources vers une seule destination. Dans un tel type de trafic, plusieurs sources transmettent des données potentiellement corrélées à un nœud puits. Notre contribution consiste en une couche de codage qui combine des paquets entrants à partir de ces sources dans un paquet unique après un calcul d'indice qui permet de récupérer à la destination chaque paquet ainsi que l'adresse de son émetteur. Le codage par index diffère des algorithmes de codage réseau classiques car

il est conçu pour fonctionner sur des capteurs très contraints, disposant d'une taille mémoire faible et d'un niveau d'énergie fini. Le principe d'indexation d'InC consiste à exprimer au sein du paquet transmis l'adresse des nœuds sources grâce à un indice attribué par leurs voisins à un saut. Grâce à la propriété d'invariance des sous-espaces vectoriels, le nœud destinataire peut recouvrir les données sources et l'adresse de leur émetteur respectif. Afin d'évaluer InC, nous calculons d'abord les limites théoriques de ses performances en utilisant différentes métriques essentielles aux réseaux de capteurs telles que le nombre de transmissions, le délai de bout-en-bout et la consommation énergétique.

Dans la deuxième partie du chapitre, nous abordons l'étude pratique de InC. Pour ce faire, nous concevons le protocole ZInC (ZigBee Index-Coding). Ce dernier est une application des concepts du codage par index adaptée aux réseaux ZigBee. Cette adaptation a pour but de montrer l'opérabilité de InC dans une plateforme d'expérimentation constituée par de vrais capteurs. Ainsi, nous exposons les avantages considérables que peut fournir InC dans un environnement réaliste, à savoir, une mémoire RAM très faible, une gestion des buffers limitées et un canal de communication non fiable. Mieux encore, cette expérience dénote la flexibilité du codage par index, car il permet aux concepteurs de réseaux de capteurs d'adapter InC à différents protocoles de routage et d'accès au médium.

La suite de ce chapitre est organisée comme suit. La section 5.1 motive ce travail et définit son contexte. Dans la section 5.2, nous définissons InC, notre solution au problème d'adaptabilité du codage réseau. Nous y détaillons sa modélisation et montrons les limites théoriques de ses performances. Nous décrivons ensuite dans la section 5.3 l'implémentation du codage par index via ZInC. Ce protocole évalué sur une plateforme réelle de réseau de capteurs démontre non seulement les performances du codage par index, mais aussi sa capacité d'intégration dans un réseau contraint. Enfin, la section 5.4 conclut le chapitre.

### ***5.1. Motivation et contexte***

L'objectif principal des réseaux de capteurs sans fil est de collecter des informations à partir de plusieurs sources vers une ou plusieurs destinations. Cette utilisation reste la plus répandue dans les applications commerciales et industrielles de cette technologie. Mis à part le cas des réseaux de capteurs multimédias, la taille des données transmises dans de tels réseaux ne dépasse généralement pas quelques octets par nœud. Par ailleurs, le processus d'acheminement des messages de proche-en-proche nécessite une surcharge non négligeable de données d'encapsulation par paquet. D'autre part, dans les réseaux de capteurs sans fil, les architectures matérielles relèvent souvent de l'informatique embarquée, impliquant des limitations en termes de mémoire, de calcul et d'énergie qui ne sont pas forcément compatibles avec les techniques de codage existantes. De plus, Les schémas de trafic différent de ceux utilisés dans des réseaux sans fil plus génériques. Par exemple, alors que les approches de codage réseau proposées dans la littérature pour les réseaux sans fil sont basées sur de l'unicast ou du multicast, dans les réseaux de capteurs, la dissémination et les communications en *incast* (i.e. multipoints-à-point) s'avèrent les plus courantes.

L'adaptation à toutes ces contraintes, la minimisation de la surcharge par paquet et du nombre de transmissions au niveau de chaque nœud est l'une des problématiques importantes

des réseaux inter-machines. En effet, cela permet de réduire la complexité de la pile protocolaire et de limiter la consommation d'énergie des nœuds prolongeant ainsi le cycle de vie du réseau. D'où la nécessité de définir un système de codage plus contraint qui soit suffisamment simple et efficace pour permettre d'améliorer les communications dans ces réseaux, tout en garantissant une utilisation minimale des ressources.

## ***5.2. Codage par index – InC***

### ***5.2.1. Le problème de l'incast***

Le problème des communications en incast est assez différent de ce qui peut sembler être son contraire : *le multicast*. Dans un réseau multicast l'unique transmetteur envoie la même information à un ensemble de destinataires, tandis que dans des communications en incast c'est un ensemble de nœuds qui transmettent des informations, cette fois distinctes, à un destinataire unique. [98] démontre que dans un réseau M2O (many-to-one) le débit maximum disponible peut être atteint via un processus de routage adéquat, en trouvant un chemin disjoint de chaque source vers la destination. Cette solution théoriquement optimale est difficile dans la réalité, particulièrement dans une topologie hiérarchique, où le réseau est sujet aux pertes de paquets comme cela est le cas dans un réseau de capteurs.

Alors que le problème des réseaux incast a été traité théoriquement dans la littérature [99] et que les solutions optimales ont été approchées via des algorithmes de codage réseau basés sur des heuristiques [100], aucune solution pratique n'a pu fournir de réponse applicable à la nature contrainte des réseaux de capteurs sans fil. Même si quelques efforts ont été faits afin d'adapter le codage réseau aux réseaux de capteurs, la plupart des travaux portent uniquement sur son impact théorique sur les performances et la fiabilité [64]. De plus, très peu d'initiatives ont considéré le coût réel de leur solution, négligeant quelques critères essentiels tels que la surcharge par paquet, l'efficacité énergétique et l'utilisation de la mémoire. Dans ce dernier travail, nous nous concentrons donc à la fois sur l'efficacité du codage mais aussi sur son impact concret sur le réseau et ses composants.

### ***5.2.2. Pourquoi un codage contraint ?***

L'idée de concevoir InC a émergé de notre constat que la taille de la charge utile dans les réseaux de capteurs est particulièrement réduite par rapport à d'autres réseaux sans fil. Par exemple, une trame de données IEEE 802.15.4 est environ 20 fois plus petite qu'une trame 802.11. De plus, la taille des données récoltées par les nœuds est souvent très inférieure à la taille maximum de la charge utile du paquet prévu pour les transporter. Dans plusieurs cas, la taille des en-têtes réseau et MAC est plus grande que celle des données elles-mêmes. En fait, dans les applications scalaires des réseaux de capteurs sans fil, les nœuds sources transmettent des informations telles que la température, la pression atmosphérique ou la luminosité. Ce type de données peut être encodé en un vecteur de quelques bits ou octets seulement. Par exemple, pour envoyer la température ambiante dans une zone donnée, la valeur détectée peut être encodée en un entier court, ce qui équivaut à 1 octet. Dans les



protocoles de routage standards tels que ZigBee ou 6lowPan, cette valeur représenterait moins de 1% de la taille totale de la trame de donnée.

Ceci suggère que les systèmes de codage réseau tels que [47] et [52] où la taille de l'entête est fonction du nombre de paquets codés doivent être évités pour réduire la surcharge par paquet. Qui plus est, et contrairement à la plupart des mécanismes de codage réseau dans la littérature, nous pensons qu'il est important que notre solution ne soit pas basée sur l'écoute active de paquets. Et pour cause, cette technique implique un surplus important de consommation d'énergie et une trop grande utilisation de la mémoire due à la gestion des différents buffers nécessaires aux opérations de codage et de décodage. Par ailleurs, dans les réseaux de capteurs, les délais pour accéder au médium sans fil peuvent être longs lorsque le réseau subit des congestions ou des pertes. Ceci est due à l'utilisation de temps d'attente progressifs et aléatoires afin d'éviter les collisions. Cette contrainte dénote la nécessité d'envoyer le maximum de données lorsque le médium est disponible.

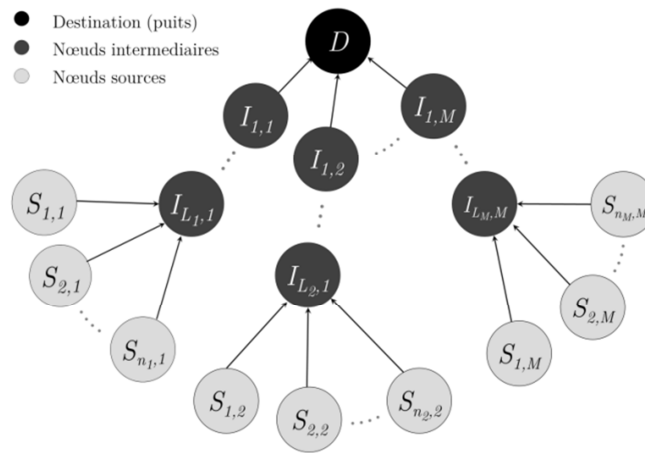


Figure 33. Les nœuds  $S_{1,1}, \dots, S_{n_M, M}$  transmettent leurs données vers  $S$  via  $M$  routes différentes

### 5.2.3. Hypothèses et modélisation du système

Nous considérons un réseau de capteurs scalaires où un ensemble de nœuds transmettent de manière synchrone leurs données à un nœud puits. Dans les protocoles de routage courants, ces communications en multi-sauts sont effectuées par un acheminement maillé ou hiérarchique. Dans les deux cas, nous supposons que des communications de bout-en-bout ont lieu entre  $N$  sources et une destination via des nœuds intermédiaires qui constituent un maximum de  $M$  routes distinctes (cf. Figure 33). Nous proposons dans cette partie d'utiliser les concepts du codage en sous-espace vectoriel [101] car nous estimons que cette approche correspond plus au paradigme des réseaux de capteurs sans fil, comparée à l'approche classique basée sur des vecteurs qui serait trop coûteuse. La Figure 33 montre un réseau dans lequel  $N$  nœuds doivent transmettre des données à la destination  $D$  via  $M$  routes différentes. Nous désignons par  $R_j$  une route constituée de  $L_j$  sauts, établie par des nœuds intermédiaires  $I_{i,k}$  ( $i=1, \dots, L_j$  et  $k=1, \dots, M$ ). Au début d'une route quelconque  $R_j$ , il n'est possible qu'à un sous ensemble  $S'_j = \{S_{1,j}, \dots, S_{n_j,j}\}$  (avec  $S'_j \subseteq S$ ) de  $n_j$  nœuds sources ( $\sum_{1,M} n_j = N$ ) de transmettre des données vers la destination en utilisant  $R_j$ . Nous supposons que chaque information  $m_{i,j}$  collectée par le nœud  $i$  est un vecteur de taille  $\eta$  qui appartient à un espace vectoriel  $V = \mathbb{F}_q^\ell$

à  $\ell$ -dimensions. Nous prenons aussi l'hypothèse que l'ensemble des données potentiellement générées par les nœuds sources au début d'une route  $j$  est  $\mathcal{M}_{i,j}$  pour n'importe quelle transmission. Nous désignons par  $v$  le sous-espace de  $V$  et utilisons la *distance* entre deux sous-espaces  $v_1$  et  $v_2$ , tel que :

$$d(v_1, v_2) \triangleq \dim(v_1 + v_2) - \dim(v_1 \cap v_2)$$

Afin de permettre à la destination de recouvrer tous les messages sources, des sous-espaces distinct<sup>7</sup> doivent être générés à chaque combinaison linéaire effectuée au niveau des nœuds intermédiaires. Ainsi, pour pouvoir appliquer InC (*Index Coding*) à des messages arbitraires  $m_{i,j}$  relayés sur une route  $R_j$  à partir de  $n_j$  sources différentes, nous considérons un code de longueur  $n_j \times \eta$ , de distance minimum  $d_{min} = n_j \eta + 1$  et une redondance  $\ell$ .

### 5.2.4. Codage par index

#### 5.2.4.1. Décalage à la source

Pour permettre le codage par index, chaque source active dans  $S'_j$  se doit d'appliquer une opération de décalage de symbole (ici un bit) sur l'information générée avant n'importe quelle transmission. Nous formalisons cette opération par un espace vectoriel  $\mathcal{O}_{i,j}$  qui fait correspondre chaque vecteur de données dans  $\mathcal{M}_{i,j}$  avec sa version décalée, qui est un sous-espace de  $d$ -dimensions dans  $\mathbb{F}_q^\ell$ . Pour chaque message  $m_{i,j}$  issu d'un nœud  $i$ , correspond une représentation dans  $\mathcal{O}_{i,j}$  qui sera transmise au prochain saut sur la route.

$$\mathcal{O}_{i,j}: \mathcal{M}_{i,j} \rightarrow \{v_{k,j}^{(i)}: v_{k,j}^{(i)} \subseteq \mathbb{F}_q^\ell, \dim(v_{k,j}^{(i)}) = d, 1 \leq k \leq |\mathcal{M}_{i,j}|\}, \quad i = 1, \dots, n_j, j = 1, \dots, M$$

Où le message décalé à la source  $i$  au commencement de la route  $j$  est  $M_{i,j}$ . Ce dernier est défini comme un vecteur qui recouvre  $v$  dans  $\mathcal{O}_{i,j}$ . Noter qu'un nœud source  $i$  peut générer un maximum de  $|\mathcal{M}_{i,j}| = q^{n_j}$  valeurs différentes à chaque transmission. Ainsi, InC assigne à chaque nœud source  $\eta$  colonnes dans une matrice  $(\ell \times n_j \eta)$  de contrôle de parité H [102] et qui contient les sous-espaces correspondant de  $\mathcal{O}_{i,j}$  et assigne des zéros dans le reste des colonnes. En d'autres termes, InC assigne à chaque nœud source  $i$  un sous-espace recouvert par la sous-matrice définie par les colonnes  $(i-1)\eta + 1$  jusqu'à  $\eta i$  dans H.

#### 5.2.4.2. Encodage

Lorsqu'un nœud intermédiaire  $I_{L,j}$  sur une route  $R_j$  reçoit un paquet des nœuds sources  $S'_j$ , il combine leurs vecteurs avec le sien en  $d$  vecteurs différents, et transmet ces derniers au prochain saut sur  $R_j$ . D'une manière générale, quel que soit un nœud intermédiaire  $I_{L,j}$ , il encode les  $(d \times n_j)$  vecteurs entrants selon l'équation suivante :

$$\forall j \in [1, M] \wedge \forall i \in [1, n_j], c_{L,j}^{(k)} = u_{L,j}^{(k)} + \sum_{i=1}^{n_j} v_{i,j}^{(k)} \quad 1 \leq k \leq d \quad (13)$$

Où,  $u_{L,j}^{(k)}$  sont les  $d$  vecteurs générés par  $I_{L,j}$ . Comme indiqué précédemment, InC concerne uniquement les réseaux de capteurs scalaires où de petites quantités de données sont reportées par les nœuds. En effet, à mesure que la taille des données transmises augmente, le

<sup>7</sup> Deux sous-espaces sont *distinct* s'ils contiennent au moins une dimension différents l'un par rapport à l'autre

bénéfice du codage par index décroît. Cela est lié au fait que plus la taille des vecteurs est grande, plus leurs versions décalées sera grande, ce qui augmente la surcharge réseau.

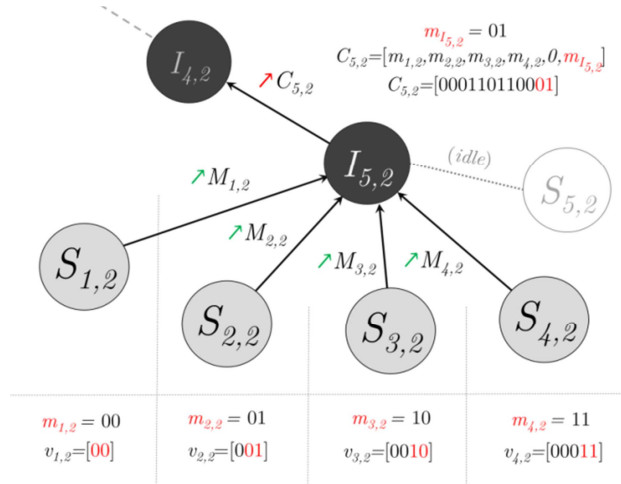
#### 5.2.4.3. Décodage

Dans le codage par index, la destination reçoit des combinaisons de vecteurs générés par les nœuds sources (par exemple,  $v_{1,j} + \dots + v_{n_i,j} + u_{L_{i,j}}$  quand tous les nœuds envoient leurs données). En utilisant l'ensemble des espaces vectoriels  $\{\overline{\mathcal{M}}_{j,i}\}$  et les vecteurs reçus, la destination peut connaître l'identité des sources ainsi que le contenu de leurs messages. Soit  $\{Y_j\}$  l'ensemble des  $d$  vecteurs reçus et  $X_{i,j}$  la matrice de vecteurs sources transmis par un nœud  $i$  sur la route  $R_j$ . Nous désignons par  $C_{i,j}$  la matrice qui représente toutes les opérations de codage effectuées sur  $X_{i,j}$  au niveau de chaque nœud intermédiaire. Parce que les sous-espaces vectoriels ne peuvent pas être altérés par des opérations linéaires, et du moment que tous les messages sont représentés par des sous-espaces vectoriels, aucune connaissance sur  $C_{i,j}$  n'est requise ni à la source, ni au niveau de la destination. En conséquence, quand un message codé arrive à sa destination, celle-ci peut en décoder les vecteurs  $v_{i,j}$  en recouvrant chaque donnée de taille  $\ell \times \eta$  encodée entre les colonnes  $(i-1)\eta + 1$  et  $i\eta$  de  $C_{i,j}$ .

De plus, et comme indiqué dans [102], ce type de codage garanti que les  $d_{min} - 1$  colonnes de  $H$  sont toujours linéairement indépendantes. Vu qu'au maximum  $n_j$  vecteurs peuvent être possiblement combinés, la destination reçoit des vecteurs qui sont des combinaisons linéaires de  $\ell \times n_j \eta$  colonnes de  $H$ . Sachant que  $d_{min} > n_j \eta$ , il devient impossible de se trouver face à une combinaison linéaire de plus de  $n_j \eta$  colonnes à la destination. Cela signifie que même compte tenu de toutes les combinaisons linéaires subies par les vecteurs générés par  $\mathcal{S}'_j$  au niveau des nœuds intermédiaires, il est toujours possible d'assurer que le puits de données reçoive des sous-espaces *distincts* comme défini plus haut. Par conséquent, le codage par index permet toujours à des sources arbitraires de générer des messages décalés qui peuvent être décodés individuellement à la destination.

#### 5.2.4.4. Exemple canonique

Par souci de clarté, prenons l'exemple illustré dans la Figure 34. Dans cette partie du réseau, 4 des 5 nœuds sources  $\mathcal{S}_2 = \{S_{1,2}, S_{2,2}, S_{3,2}, S_{4,2}\}$  au début de la route  $R_2$  transmettent leur informations  $m_{i,j}$  (codée sur 2 bits) au nœud intermédiaire  $I_{5,2}$ . Dans ce cas, l'espace vectoriel est  $\mathbb{F}_2^8$  et chaque vecteur transmis représente un sous-espace à 10 dimensions de  $\mathbb{F}_2^8$ . InC sélectionne pour chaque source  $i$  son sous-espace correspondant  $v_{k,2}^{(i)}$  à partir de  $\mathcal{O}_{i,2}$  (avec  $1 \leq k \leq 5$ ). Par exemple, InC assigne les colonnes de  $(i-1)\eta + 1 = 1$  à  $i\eta = 2$  au nœud  $S_{1,2}$  afin qu'il encode  $m_{1,2}$  dans son vecteur décalé correspondant. De la même manière, au nœud  $S_{2,2}$  sont assignées les colonnes 3 et 4 et ainsi de suite.



**Figure 34.** Exemple de InC, où les nœuds actifs envoient les vecteurs décalés  $M_{i,2}$  à  $D$  sur  $R_2$  via le nœud intermédiaire  $I_{5,2}$ .  $I_{5,2}$  combine ensuite les  $M_{i,2}$  et transmet le résultat à son prochain saut  $I_{4,2}$ .

$$M_{1,2} = [m_{1,2}^{(1)} \ m_{1,2}^{(1)} \ 0 \ \dots \ 0], M_{2,2} = [0 \ 0 \ m_{2,2}^{(1)} \ m_{2,2}^{(2)} \ 0 \ \dots \ 0], \dots$$

Lorsque  $I_{5,2}$  reçoit les messages  $M_{i,2}$ , il les combine comme décrit dans la section précédente, en incluant son propre vecteur dans le message codé  $C_{5,2}$  et transmet ce dernier au prochain saut dans  $R_2$  comme décrit dans la Figure 34. Enfin, lorsque le paquet codé arrive à sa destination, celle-ci pourra recouvrer à partir de chaque  $2i$  colonnes de  $C_{5,2}$ , à la fois la source et le contenu des messages originaux  $m_{i,j}$ . On peut noter qu'un code spécial doit être défini pour différencier les nœuds actifs et ceux qui ne le sont pas. Par exemple, dans le cas d'un nœud source inactif (comme c'est le cas pour  $S_{5,2}$  dans l'exemple), on peut choisir d'utiliser le vecteur  $[0 \ 0]$  pour indiquer qu'il ne transmet aucune donnée. Cela permettra d'éviter à la destination de décoder des informations ambiguës du nœud en question.

### 5.2.5. Limites théoriques

#### 5.2.5.1. Nombre de transmissions

Dans un réseau en *store-and-forward*, les nœuds intermédiaires relayent simplement les  $N$  messages transmis par  $S$  jusqu'à ce qu'ils atteignent  $D$ . Dans un environnement idéal dénué de pertes, cette opération impliquerait la transmission de  $N_{tx}$  message, avec :

$$N_{tx} = \sum_{i=1}^M |S'_i| \times (L_i + 1) \quad (14)$$

Pour arriver à transmettre moins de messages, sachant qu'il existe une sous-exploitation de la bande passante du réseau, InC propose que chaque nœud intermédiaire au début d'une route  $R_i$  combine les messages provenant des nœuds dans  $S'_j$  en un seul et unique message. Un tel codage réduirait la valeur de  $N_{tx}$  à  $\hat{N}_{tx}$ , où :

$$\hat{N}_{tx} = \sum_{i=1}^M |S'_i| + L_i \quad (15)$$

La Figure 35, qui présente le tracé des équations 14 et 15, montre que  $\exists i \in [1, M], |S'_i| \neq 1 \wedge L_i \neq 0 \Rightarrow \hat{N}_{tx} < N_{tx}$ , i.e. le nombre de transmissions est amélioré lorsqu'il y a plus d'une source qui transmet des données à plus d'un saut de la destination.

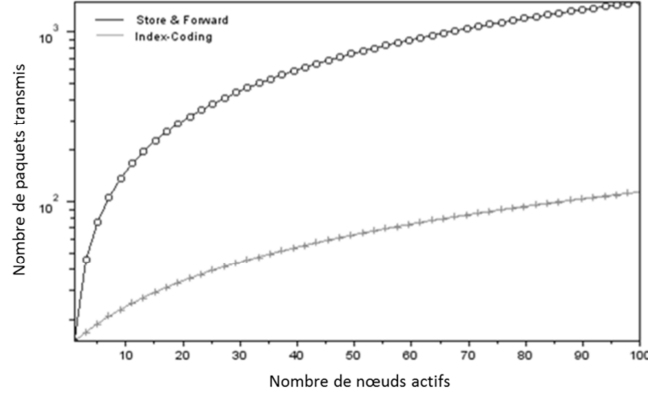


Figure 35. Nombre de transmissions par rapport au nombre de nœuds sources

Notons que  $\hat{N}_{tx} = N_{tx} \Rightarrow L_i = 0 \vee |S'_i| = 1, \forall i \in [1, M]$ , i.e. l'utilisation des opérations de codage donnerait, dans le pire des cas, des performances équivalentes à un routage *store-and-forward* en ce qui concerne le nombre de transmissions. InC est donc équivalent à un système d'acheminement classique, soit quand tous les nœuds sources sont directement connectés au puits de données, soit lorsque toutes les communications commencent à partir d'un seul nœud source. Ces deux situations sont bien évidemment des cas spéciaux qui ne caractérisent en rien un réseau de capteurs sans fil réel, puisque l'un implique l'absence du multi-saut et l'autre dénote un déploiement non optimisé.

#### 5.2.5.2. Délai de bout-en-bout

Une métrique importante dans les réseaux de capteurs sans fil est la latence de bout-en-bout des communications en incast, i.e. le délai durant lequel l'intégralité des messages sources est correctement délivrée à la destination. Dans un réseau *store-and-forward* cette latence est fortement corrélée au nombre de nœuds actifs et au taux de pertes enduré par le réseau. Grâce au codage par index, nous pouvons réduire l'impact du taux de pertes sur la latence car les nœuds transmettent en général considérablement moins de messages et de ce fait, il devient possible de réduire significativement la probabilité de perte d'un message dues aux collisions endurées le long de la route jusqu'au destinataire. Pourtant, il est difficile de formaliser analytiquement cette latence dans un réseau *store-and-forward* à sources multiples car cela impliquerait la modélisation des différentes files d'attente au niveau de chaque nœud intermédiaire. Ainsi, nous donnons une estimation de cette valeur en supposant que les messages subissent un temps d'attente  $d_q$  au niveau des nœuds intermédiaires.  $d_q$  dépend du taux de pertes et du nombre de messages déjà présents dans la file de réception. Les équations 16 et 17 montrent, respectivement pour le *store-and-forward* puis pour InC, comment le taux de pertes influe sur la latence dans un réseau incast. La latence  $D$  est calculée comme le délai d'acheminement d'un message de  $N$  nœuds à une destination en incluant les retards dus aux retransmissions et aux files d'attente.

$$D \approx \sum_{i=1}^M [(|S'_i| + L_i - 1) \cdot d_{tx} + |S'_i| \cdot (L_i - 1) \cdot d_q] \quad (16)$$

$$\hat{D} \approx \sum_{i=1}^M [(|S'_i| + L_i - 1) \cdot d_{tx} + (L_i - 1) \cdot d_q] \quad (17)$$

Avec  $d_{tx} = \prod_1^{N_{rt}} p_{ok} \times (d_{data} + d_{ack} + \varepsilon)$ .

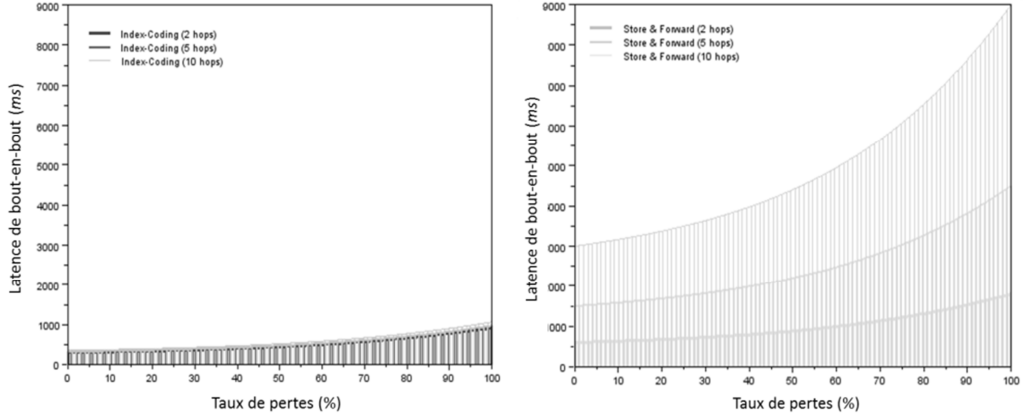


Figure 36. Délai de bout-en-bout par rapport au taux de pertes de messages

La Figure 36 représente une application numérique des équations 16 et 17. Elle montre que la progression de la latence de bout-en-bout à mesure que le taux de pertes augmente est quasiment indépendante du nombre de nœuds dans InC, alors qu'elle suit une progression quasi linéaire dans le cas du *store-and-forward*. De plus, le délai moyen en utilisant le codage par index reste relativement bas en comparaison avec un routage classique, car les délais dans les files d'attente ne sont pas linéaires au nombre de nœuds actifs dans ce cas.

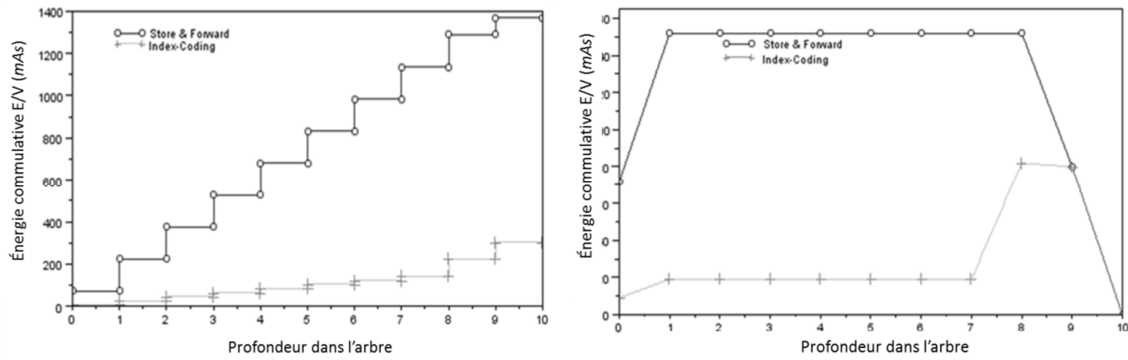
### 5.2.5.3. Consommation d'énergie

Les métriques les plus importantes dans un réseau de capteurs sans fil sont la consommation d'énergie par nœud et la consommation énergétique globale du réseau. Dans cette partie, au lieu d'inférer la consommation d'énergie à partir du nombre de transmissions, nous calculons celle-ci en utilisant la méthode d'estimation fournie par Dunkels *et al.* dans [65], et selon la profondeur des sources dans le réseau. Dans notre cas, nous supposons que les nœuds ont un cycle de durée  $T_{cycle}$  pendant lequel ils sont soit endormis pendant  $t_s$  ou actifs. Un nœud est actif s'il transmet (durant  $t_{tx}$ ), reçoit (durant  $t_{rx}$ ) ou traite (durant  $t_p$ ) des données. L'équation 18 calcule cette estimation.

$$\frac{E_d}{V} = \begin{cases} I_p \cdot t_p + I_{rx} \cdot t_{rx} & d = 0 \\ I_p \cdot t_p + I_s \cdot t_s + I_{tx} \cdot t_{tx} + I_{rx} \cdot t_{rx} & 0 < d < d_{max} \\ I_p \cdot t_p + I_s \cdot t_s + I_{tx} \cdot t_{tx} & d = d_{max} \end{cases} \quad (18)$$

Avec  $t_s = T_{cycle} - t_p - t_{tx} - t_{rx}$ ,  $t_{tx} = N_{tx} \cdot d_{tx}$  et  $t_{rx} = N_{rx} \cdot d_{rx}$

Où  $E_d$  est la consommation cumulée de l'énergie d'un nœud dont la profondeur réseau est  $d$ .  $V$  est la tension d'alimentation et  $I_m$  l'intensité fournie au microprocesseur lorsqu'il est actif.  $t_m$  est le temps durant lequel le microprocesseur a été actif.  $I_{tx}$  et  $t_{tx}$  représentent respectivement, l'intensité consommée et la durée d'utilisation du processeur pendant une transmission,  $I_{rx}$  et  $t_{rx}$  désignent ces mêmes valeurs en mode réception. Il est à noter qu'il n'y a pas besoin de procéder au calcul de la tension  $V$ , puisque l'ensemble des nœuds fonctionnent avec la même valeur. Ainsi, les valeurs de  $E_d/V$  peuvent être comparées directement. La Figure 37 (a) montre les résultats de l'application numérique du modèle en appliquant des valeurs qui correspondent aux paramètres des dispositifs matériels utilisés dans la deuxième partie de ce chapitre (capteurs ZigBee).



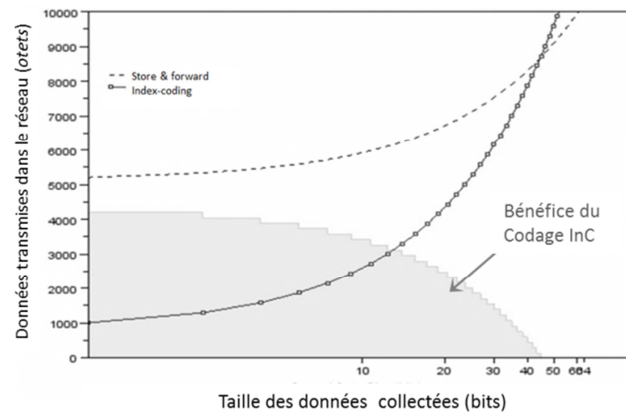
**Figure 37.** Consommation d'énergie et distribution de la consommation par rapport à la profondeur du réseau.

Il est clair que comparé à un routage en *store-and-forward*, le codage par index assure une plus faible consommation d'énergie par nœud. Cela est possible car InC permet des durées d'inactivité plus longues au niveau des nœuds intermédiaires. En effet, ces derniers relayent beaucoup moins de messages et de ce fait ne sont plus obligés d'être constamment actifs. Notons que pour effectuer ces estimations, nous calculons la consommation d'énergie moyenne en supposant un réseau de 15 nœuds qui transmettent des données à la destination via une route de longueur variable (1 à 10 sauts).

Par ailleurs, la Figure 37 (b) illustre comment InC permet une meilleure distribution de la consommation d'énergie sur un arbre incast. En fait, un problème typique des réseaux incast est que les nœuds intermédiaires relayent un nombre très important de messages, relativement au nombre de sources. En effet, pour chaque communication de bout-en-bout, un nœud intermédiaire est obligé de relayer autant de messages qu'il y a de sources. Lors de l'utilisation de InC, la consommation d'énergie est équitablement répartie sur les nœuds de l'arbre incast, dans la mesure où quand  $|S'_i|$  sources transmettent un message, les nœuds intermédiaires transmettent eux aussi un seul message au lieu de  $|S'_i|$ . Si l'on considère l'exemple canonique décrit plus haut (cf. Figure 34), à chaque fois que l'un des 5 nœuds sources transmet un paquet, tous les nœuds intermédiaires auront à relayer  $|S'_i| = 5$  messages en amont vers la destination. Ceci implique que leurs module de communication doit être actif plus longtemps pour garantir la réception de chaque message. Tandis qu'avec l'utilisation de InC, les nœuds intermédiaires n'auraient à relayer qu'un seul message, ce qui leur permettra de passer plus vite et plus longtemps en mode inactif.

#### 5.2.5.4. Surcharge réseau

Alors que nous avons montré le bénéfice, sur le nombre de transmissions, de l'utilisation du codage par index, il est également important de déterminer son coût en évaluant son excédent par paquet. En effet, l'efficacité de InC repose non seulement sur le nombre de nœuds actifs qui transmettent des données, mais aussi sur la taille de ces dernières. Ainsi, quand le nombre de nœuds actifs diminue il devient moins intéressant d'effectuer des opérations de codage par index, puisqu'elles impliquent l'utilisation du décalage à la source et donc l'augmentation de la taille des messages.



**Figure 38.** Bénéfice du codage par index en fonction de la taille des données

Néanmoins, dans la mesure où nous nous intéressons à des communications à sources multiples, nous pouvons supposer qu'en général, une communication de bout-en-bout à lieu au moins entre deux sources et une destination, autrement, il s'agirait d'un réseau unicast.

Concernant l'impact de la taille des données transmises, la Figure 38 montre que InC implique une surcharge jusqu'à 5 fois inférieure à celle du *store-and-forward*, pour transmettre la même quantité de données. Particulièrement lorsque la taille de l'information collectée est réduite (moins de 4 octet dans cet exemple<sup>8</sup>). Cependant, à mesure que la taille des données augmente, le bénéfice<sup>9</sup> du codage par index diminue. Ceci est dû au fait que les messages sources décalés deviennent trop grands et par conséquent, augmente la charge du réseau. C'est la raison principale qui fait que le codage par index ne concerne que les réseaux de capteurs scalaires où de petites quantités de données sont collectées par chaque capteur.

### 5.2.6. Discussion

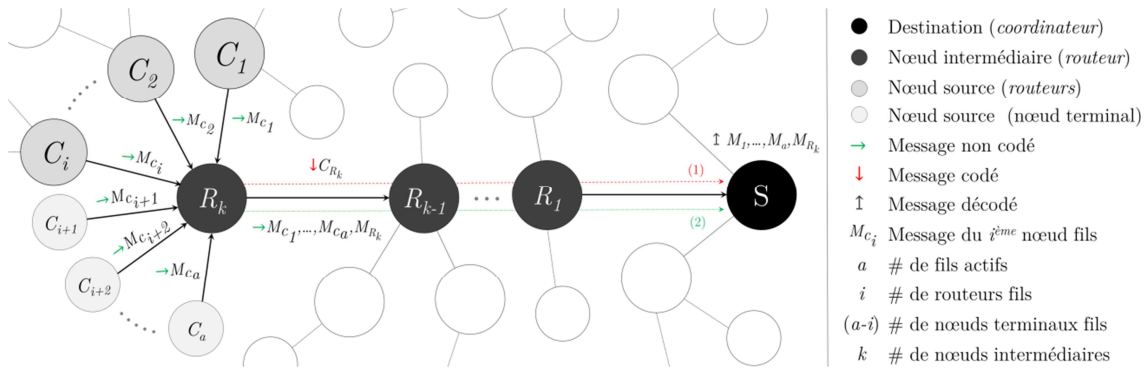
Dans cette première partie, nous avons étudié à la fois les fondements théoriques et les avantages prospectifs du codage par index. Nous avons démontré que quelles que soient les décisions de codage prises au niveau des nœuds intermédiaires, la destination sera toujours capable de décoder et d'identifier les messages sources. Nous avons également identifié le cadre dans lequel InC assure des gains notables, à savoir des communications incast dans un réseau de capteurs scalaire. De plus, nous montrons que le codage par index augure des résultats très satisfaisants en termes de consommation d'énergie et de délai de bout-en-bout, tout en gardant une surcharge réseau raisonnable.

Cette étude a pour but de démontrer la faisabilité du codage par index dans les communications incast sans fil en général. En continuité de ce travail, la section suivante décrit l'application pratique de InC au sein d'une plateforme de test constituée de vrais capteurs sans fil, et qui démontre l'efficacité et les performances du codage par index pour des applications réelles des réseaux de capteurs sans fil.

<sup>8</sup> Nous utilisons dans cette estimation, une topologie dans laquelle 10 nœuds sources transmettent des données à une destination via une route constituée de 8 nœuds intermédiaires.

<sup>9</sup> Différence entre la quantité des données transmises en store-and-forward et celles avec InC.





**Figure 39.** Les nœuds  $C_1, \dots, C_a$  envoient leur données à  $S$  via une route de  $k$  sauts soit en utilisant le routage ZigBee (en vert) ou ZInC (en rouge)

### 5.3. Codage par index dans la pratique – ZInC

Dans cette partie empirique de l'étude, nous décrivons la mise en œuvre du codage par index dans un protocole de communication standard pour les réseaux de capteurs et intermachines, à savoir celui utilisé dans ZigBee. Ainsi, nous proposons ZInC (ZigBee Index Coding), un protocole de communication pour les réseaux de capteurs, qui applique les concepts du codage par index pour améliorer la collecte périodique de données à partir d'un ensemble de capteurs sources. Nous montrons que ZInC est adapté pour des applications réalistes d'acheminement en incast, et qu'il garantit une amélioration significative des performances tout en assurant une plus longue durée de vie du réseau. Plus important encore, cette application dans un réseau réel est sensée illustrer l'adaptabilité de InC dans un environnement contraint, et montrer sa faible consommation mémoire, son efficacité énergétique et sa surcharge réseau négligeable.

#### 5.3.1. Description du modèle réseau

L'algorithme de routage utilisé dans ZigBee est hiérarchique ou maillé. Le routage hiérarchique permet l'acheminement de paquets dans l'arbre ZigBee en utilisant un système d'allocation de blocs d'adresses hiérarchiques basé sur une fonction de décalage nommée  $C_{\text{skip}}$  [15]. Il permet à chaque nœud parent de disposer d'un intervalle d'adresses qu'il peut attribuer à ses nœuds enfants. Le système maillé basé sur les tables de routage utilise quant à lui AODV [14] pour établir la connectivité ad hoc. Dans cette partie, nous nous concentrons sur la topologie en cluster avec un routage hiérarchique qui représente les paramètres par défaut utilisés dans ZigBee. Néanmoins, ZInC peut être facilement étendu pour des applications complément maillées. Nous supposons donc un réseau de capteurs ZigBee avec les paramètres de configuration [15]  $C_m, R_m, L_m$ .  $C_m$  étant le nombre maximum de nœuds enfants qu'un routeur puisse avoir,  $R_m$  le nombre maximum de routeurs fils, et  $L_m$  la profondeur maximum du réseau.

Dans ZigBee les communications incast sont effectuées en utilisant le routage hiérarchique. Nous supposons donc qu'une communication en incast à lieu entre  $C_a$  capteurs ( $C_a \leq C_m$ ) et une seule destination en utilisant  $k$  routeurs qui constituent le chemin incast (cf. Figure 39). Nous désignons par  $R = \{R_1, \dots, R_k\}$  l'ensemble des nœuds intermédiaires qui constituent cette

route. Notons que parmi  $C_a$  sources,  $C_i$  est le nombre de nœuds terminaux actifs (avec  $0 \leq C_a - C_i \leq C_m - R_m$ ). Nous supposons aussi que les données transmises à partir d'un nœud  $c_j$  sont représentées dans un vecteur  $I_{c_j}$  de taille  $N$  avec  $I_{c_j} = [m_{c_j,1}, m_{c_j,2}, \dots, m_{c_j,N}]$ .

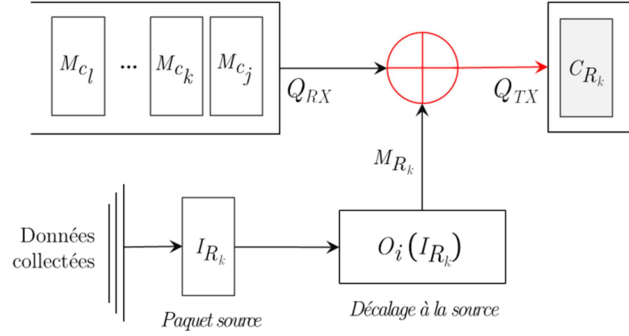


Figure 40. Codage par index au niveau du nœud intermédiaire  $R_k$ .

### 5.3.2. Présentation de ZInC

Le principe général de ZInC est illustré sur la Figure 40. Au lieu de relayer les paquets  $M_{c_i}$  reçus individuellement, le premier routeur  $R_k$  sur la route vers la destination met en tampon dans  $Q_{RX}$  ces paquets pendant une période prédéfinie. Ensuite, en utilisant le codage par index, il combine leurs charges utiles respectives avec ses données pré-codées (voir la section 5.3.2.1) en un seul paquet codé  $C_{R_k}$  sortant. Ce dernier est par la suite acheminé de proche-en-proche jusqu'à la destination pour le décodage. L'objectif ici est de réduire le nombre de transmissions, la latence de bout-en-bout et d'améliorer l'efficacité énergétique en prolongeant la durée d'inactivité (sommeil) des nœuds.

#### 5.3.2.1. Décalage à la source ZigBee

Similairement à InC, pour pouvoir appliquer le codage par index, les capteurs actifs  $c_j$  ( $\forall j = [1, a]$ ) appliquent une opération de décalage de bit au vecteur d'information  $I_{c_j}$  qui représente les données collectées par chacun d'eux. Cette opération est effectuée en calculant un offset en se basant sur l'adresse ZigBee du nœud en question comme décrit dans l'équation 19. Où  $o_i(I_{c_j})$  est la fonction qui applique un décalage de  $i$  bits à  $I_{c_j}$ .

$$\forall c_j \in C_a, I_{c_j} = [m_{c_j,1}, m_{c_j,2}, \dots, m_{c_j,N}] \Rightarrow O_i(I_{c_j}) = M_{c_j} \quad (19)$$

$$\text{Avec } i = \text{idx}(c_j) \text{ et } O_i(I_{c_j}) = M_{c_j} = \begin{bmatrix} 0 & 0 & \dots & m_{c_j,1} \\ 0 & 0 & \dots & m_{c_j,2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & m_{c_j,N} \end{bmatrix}$$

L'index  $i = \text{idx}(c_j)$  utilisé pour savoir de combien de bits  $I_{c_j}$  doit être décalé doit être unique pour chaque nœud enfant afin d'éviter des chevauchements dans les codes transmis. La valeur de  $i$  est calculée en utilisant l'adresse ZigBee  $A(c_j)$  du nœud  $c_j$ . Ce calcul est détaillé dans l'équation suivante :

$$idx(c_j) = \begin{cases} \frac{A(c_j) - A(\dot{c}_j) - 1 + C_{skip}(c_j)}{C_{skip}(\dot{c}_j)} & c_j \text{ est un router} \\ A(c_j) - A(\dot{c}_j) - [1 - C_{skip}(\dot{c}_j)] \cdot R_m & \text{sinon} \end{cases} \quad (20)$$

Sachant que d'après la norme ZigBee décrite dans [15], le calcul du  $C_{skip}$  qui permet l'allocation des blocs d'adresses hiérarchiques dans le réseau, se fait comme suit :

$$C_{skip}(x) = \frac{C_m \cdot R_m^{L_m - d(x) - 1} + R_m - C_m - 1}{R_m - 1}$$

$\dot{c}_j$  est la notation utilisée pour désigner le routeur parent du capteur  $c_j$  dans la topologie hiérarchique de ZigBee. La valeur de  $C_{skip}$  est calculée localement par chaque routeur afin de réserver l'intervalle d'adresses qu'il va allouer à ses nœuds enfants. Evidemment, tous les capteurs calculent  $idx$  localement et d'une manière complètement autonome, sans aucune connaissance centralisée du réseau ou de messages de contrôle.

### 5.3.2.2. Encodage

Afin de permettre à la destination de recouvrer les paquets sources, des codes distincts doivent être générés par les routeurs. Comme décrit sur la Figure 40 lorsqu'un routeur  $R_k$  reçoit des paquets à partir d'un sous ensemble de capteurs sources actifs  $C_i$  ( $\forall i, 1 \leq i \leq a$ ), il les met d'abord dans son buffer de réception  $Q_{RX}$  pendant une durée prédéfinie. Ensuite, il combine leurs charges utiles pré-décalées  $M_{C_i}$  avec son propre vecteur, dans un paquet codé  $C_{R_k}$  (cf. Equation 21), et le transmet vers la destination. Evidemment, dans le cas où le routeur n'est associé à aucune source, il transmet uniquement son vecteur d'information. Dans la mesure où les cycles de collecte de l'information dans un réseau de capteurs sont généralement périodiques et puisque les nœuds passent en mode inactif entre chaque transmission pour sauvegarder leur énergie, la période de mise en tampon peut facilement être optimisée pour attendre la réception des données collectées avant de les coder.

$$C_{R_k} = \text{XOR}(M_j)_{j \in Q_{RX}} \oplus M_{R_k} \quad (21)$$

De plus, dans la mesure où les opérations de décalage ne sont faites qu'à la source, il n'est pas nécessaire pour les routeurs intermédiaires de traiter les paquets entrants, ce qui minimise le coût en calcul processeur de ZInC. La consommation mémoire est aussi réduite puisque, dans le pire cas, un maximum de  $C_a$  paquets est mis en tampon dans  $Q_{RX}$ . Dans des applications scalaires des réseaux de capteurs, cette quantité de données représenterait une taille maximum de  $c_a \times (5 + \text{taille}(I_{c_j}))$ , ce qui reste raisonnable. Par exemple, si l'on considère un réseau très dense où un nœud peut avoir jusqu'à 50 voisins directs<sup>10</sup>, une opération de codage impliquerait la mise en tampon de  $50 \times (5 + \text{taille}(I_{c_j}))$  octets. Si la taille de la charge utile n'excède pas 4 octets, ce code nécessiterait dans le pire des cas 0.45 Ko de paquets mis en tampon. Dans le matériel dont nous disposons pour notre expérimentation, la mémoire RAM disponible est de 10 Ko, ce qui est largement suffisant pour le fonctionnement de ZInC.

<sup>10</sup> Supposer 50 voisins est un exemple délibérément surestimé puisque la couche MAC de ZigBee peut difficilement garantir une connectivité aussi dense.

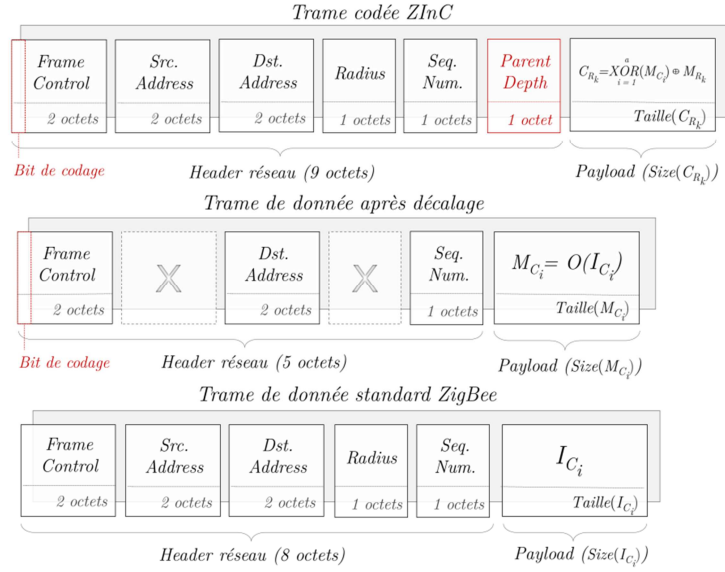


Figure 41. Comparaison des trames de données entre ZInC et ZigBee

### 5.3.2.3. Décodage

A chaque fois que la destination reçoit un paquet codé, elle applique une opération de codage par index inversée. I.e. elle extrait à partir du paquet codé  $C_{R_k}$  (cf. Equation 22) les vecteurs originaux  $v_i = [m_{c_i,1}, m_{c_i,2}, \dots, m_{c_i,N}]$  qui représentent les charges utiles de l'ensemble des paquets codés. Ensuite, en utilisant l'adresse source et l'index  $j$  de chaque vecteur, la destination procède au calcul des adresses ZigBee des sources de chaque vecteurs tel que décrit dans l'équation 23. Ainsi elle recouvre l'identité de l'ensemble des nœuds sources concernés par cette communication. Dans la mesure où la profondeur du père du nœud codeur est nécessaire pour calculer  $C_{skip}(\dot{c}_j)$ , sa valeur doit être incluse dans le paquet codé.

$$C_{R_k} = \begin{bmatrix} m_{c_1,1} & \dots & m_{c_i,1} & \dots & m_{c_a,1} & m_{R_k,1} \\ m_{c_1,2} & \dots & m_{c_i,2} & \dots & m_{c_a,2} & m_{R_k,2} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ m_{1,N} & \dots & m_{c_i,N} & \dots & m_{c_a,N} & m_{R_k,N} \end{bmatrix} \quad (22)$$

$$A(j) = \begin{cases} A(\dot{c}_j) + C_{skip}(\dot{c}_j) \cdot R_m + j - R_m & j > R_m \\ A(\dot{c}_j) + 1 + C_{skip}(\dot{c}_j) \cdot (j - 1) & j \leq R_m \end{cases} \quad (23)$$

Dans ce cas, la destination peut déduire le nombre maximum d'enfants  $C_a$  à partir des paramètres de configuration ZigBee. Cette valeur est nécessaire pour récupérer les  $C_a + 1$  paquets potentiellement codés à partir de  $C_{R_k}$ . Pour chaque  $j^{\text{me}}$  série de  $N$  symboles ( $j = [1, C_a + 1]$ ), la destination calcule  $A(j)$ , l'adresse ZigBee correspondante, comme décrit dans l'équation 23.

### 5.3.2.4. Surcharge par paquet

La Figure 41 montre une représentation des trames de données de ZigBee et de ZInC. Nous distinguons deux types de trames ZInC.

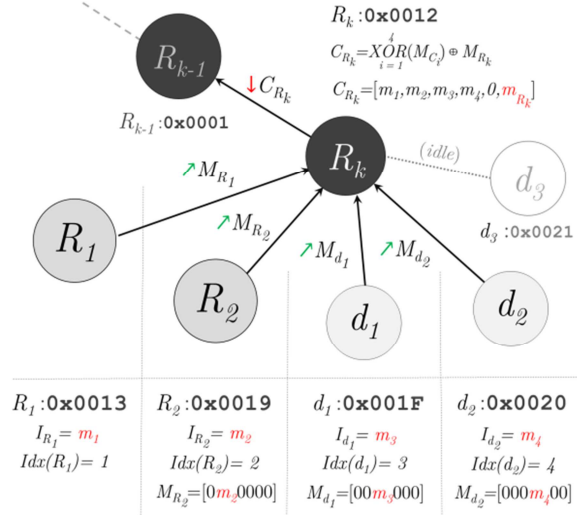


Figure 42. Exemple d'un codage ZInC

- a) *Paquets sources* : émis par les capteurs après la collecte de données, ces paquets sont le résultat du décalage à la source. La taille de leur entête est 3 octets plus courte que celle de ZigBee, car les capteurs sources dans ZInC n'ont pas besoin d'indiquer leur adresse ZigBee puisqu'elle est déjà signifiée via le codage par index. Le champ *radius* est aussi superflu puisque ces paquets ne seront par relayés individuellement.
- b) *Paquets codés* : délivrés par les routeurs ZigBee après une opération de codage, ils nécessitent un octet supplémentaire pour représenter la valeur de la profondeur du nœud père, puisque celle-ci est nécessaire lors du décodage. Ces paquets utilisent un *flag* de codage dans le champ *frame control*. Cet indicateur permet à la destination de distinguer les paquets codés des paquets ZigBee classiques.

On peut noter que les trames de données ZInC sont compatibles avec la norme IEEE 802.15.4 et donc lisibles par les capteurs qui n'embarquent pas notre couche de codage. Et contrairement à d'autres systèmes de codage présentés dans la littérature, la taille des entêtes utilisée par ZInC reste réduite et ne dépend pas du nombre de paquet encodés.

### 5.3.3. Exemple canonique

Considérons un exemple (cf. Figure 42) similaire à celui décrit dans la première partie de ce chapitre. Dans cette partie du réseau, les capteurs  $\{R_1, R_2, d_1, d_2\}$  transmettent un message  $M_{i,j}$  codé sur 2 bits à leur routeur parent  $R_k$ . Les paramètres de configuration ZigBee ici sont :  $C_m = 5, R_m = 2$  et  $L_m = 4$ . Dans ce cas précis, ZInC sélectionne pour chaque capteur source  $c_i$  son vecteur décalé correspondant en utilisant  $O_i(I_{c_j})$ . Il assigne les colonnes 1 et 2 au capteur  $R_1$ . Idem, à  $R_2$  sont assignées les colonnes 3 et 4 et ainsi de suite :

$$M_{R_1} = O_2(I_{R_1}) = [m_{R_1,1} \ m_{R_1,2}], \quad M_{R_2} = O_2(I_{R_2}) = [0 \ 0 \ m_{R_2,1} \ m_{R_2,2}], \quad M_{d_1} = O_2(I_{d_1}) = [0 \ 0 \ 0 \ 0 \ m_{d_1,1} \ m_{d_1,2}],$$

$$M_{d_2} = O_2(I_{d_2}) = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ m_{d_2,1} \ m_{d_2,2}], \quad M_{R_k} = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ m_{R_k,1} \ m_{R_k,2}]$$

Après la réception par le routeur  $R_k$  des versions décalées de chaque donnée source,  $R_k$  les combine (en incluant son propre message) et transmet le résultat  $C_{R_k}$  au saut suivant vers la destination.

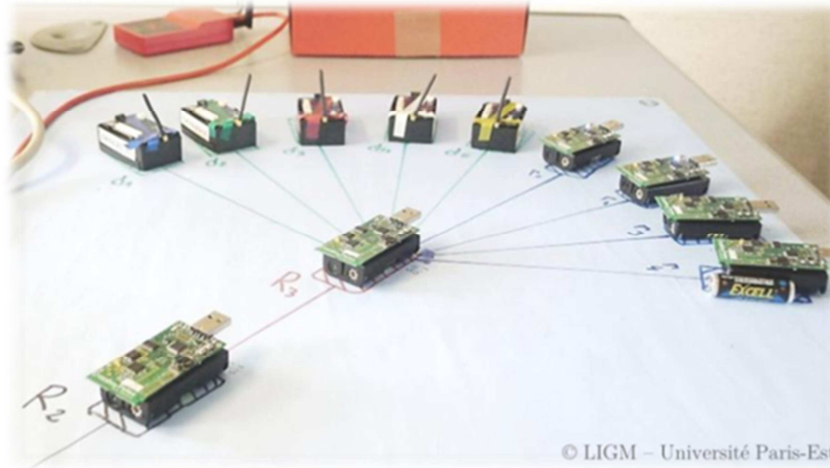


Figure 43. Banc d'essai expérimental ZInC

Une colonne nulle dans le paquet codé représente un capteur qui n'a pas transmis de données pendant ce cycle ( $d_3$  dans cet exemple).

$$C_{R_k} = M_{R_1} + M_{R_2} + M_{d_1} + M_{d_2} + M_{R_k} = \begin{bmatrix} m_{R_1,1} & m_{R_2,1} & m_{d_1,1} & m_{d_2,1} & 0 & m_{R_k,1} \\ m_{R_1,2} & m_{R_2,2} & m_{d_1,2} & m_{d_2,2} & 0 & m_{R_k,2} \end{bmatrix}$$

Lorsque  $C_{R_k}$  atteint sa destination, le décodage ZInC est appliqué pour extraire les adresses des paquets encodés. Par exemple, pour trouver l'adresse du 2<sup>ème</sup> et 3<sup>ème</sup> paquet codé  $v_2 = [m_{R_2,1}, m_{R_2,2}]$  et  $v_3 = [m_{d_1,1}, m_{d_1,2}]$ , la destination effectue l'application numérique décrite dans les équations suivantes :

$$A(v_2) = A(\check{v}_2) + 1 + C_{skip}(\check{v}_2) \cdot (2 - 1) = 19_{(16)}, \quad A(v_3) = A(\check{v}_3) + C_{skip}(\check{v}_3) \cdot R_m + 3 - R_m = 1F_{(16)}$$

Où  $A(v_i)$  est l'adresse du nœud ayant transmis le vecteur  $v_i$ .  $A(\check{v}_i)$  est l'adresse du codeur et  $C_{skip}$  le paramètre de configuration ZigBee décrit dans [15]. Ainsi, au lieu de relayer jusqu'à la destination les 4 paquets sources individuellement,  $R_k$  encode le tout en un seul paquet et transmet le résultat à son prochain saut. On peut aisément envisager les bénéfices potentiels d'un tel codage réseau à mesure que le nombre de sources et de sauts augmentent.

### 5.3.4. Evaluation de performance

Dans cette section nous montrons les performances de ZInC en comparaison avec celles du routage ZigBee dans un vrai réseau de capteurs sans fil. Nous arguons que les techniques de validation basées uniquement sur une modélisation analytique ne sont pas suffisantes pour évaluer des systèmes aussi dépendants de la technologie et aussi contraints que les réseaux de capteurs. En fait, ces méthodes théoriques négligent souvent des propriétés cruciales tel que la mémoire, la consommation d'énergie ou encore le coût processeur. Ainsi, pour démontrer que le codage par index est conçu pour fonctionner sur des capteurs réels et en compatibilité avec des protocoles standards, nous avons mis en place un banc d'essai expérimental (cf. Figure 43) qui permet de lancer des communications en incast au-dessus de la couche MAC IEEE 802.15.4 et en adaptant le routage ZigBee.

Ils existent de nombreux facteurs qui régissent le comportement des réseaux de capteurs sans fil : l'environnement ambiant, la topologie du réseau, le modèle de trafic et par extension, les

différents phénomènes physiques constatés lors d'une transmission. Il est difficile d'isoler ces phénomènes afin de pouvoir étudier exhaustivement l'impact direct de ces facteurs sur la fiabilité et les performances du réseau. Ainsi, nous entreprenons une étude expérimentale quelque peu mécanique en relevant systématiquement les différentes métriques de performance au niveau des couches MAC et réseau de la pile protocolaire. Mécanique, parce que nous appliquons certes un certain contrôle sur la topologie du réseau, la nature du trafic, le modèle de propagation des données et la durée de l'expérience, mais notre étude ne peut être entièrement sous contrôle, parce que nos mesures sont sujettes à l'imprévisibilité du médium radio.

Comme le montre la Figure 44, nous déployons 13 capteurs ZigBee dans une zone plate, parmi ces capteurs 5 sont des nœuds terminaux MicaZ et 8 des routeurs TelosB. Les 5 nœuds terminaux et 4 routeurs sont désignés comme sources potentielles qui communiquent les données collectées à différents débits constants (de 0,5 à 4 paquets/seconde). 4 routeurs établissent une route en multi-sauts qui permet l'acheminement de proche-en-proche des flux de paquets jusqu'à la destination. Les parents des nœuds actifs appliquent les opérations de codage par index et transmettent les paquets codés aux nœuds plus hauts dans la hiérarchie. L'ensemble des résultats constitue la moyenne de 20 expériences indépendantes. Le reste des paramètres de l'expérimentation est résumé dans le tableau 6.

**Tableau 6.** Paramètres de l'expérimentation

Nombre de noeuds	9 sources (5 TelosB + 4 MicaZ) 4 routeurs (TelosB)
Protocole de routage	Routage en cluster-tree ZigBee
Protocoles PHY/MAC	IEEE 802.15.4 (2.4 GHz) / CSMA/CA
Paramètres initiaux ZigBee	$Cm = 9$ $Rm = 4$ $Lm = 4$
Beacon Order	$BO = 15$
Super Frame Order (SO)	$SO = 15$
Modèle de trafic	Debit constant en incast
Nombre de paquets sources	200
Intervalle CBR	$I_1 = 5.0$ pkt/s $I_2 = 2.5$ pkt/s $I_3 = 1.0$ pkt/s $I_4 = 0.5$ pkt/s
Taille de la charge utile à la source	1 octet
Consommation d'énergie par nœud	Mode actif : 23 mA Mode inactif : 1 $\mu$ A
Débit théorique	250 kbps
Puissance de réception	0 dBm
Période de mise en tampon ZInC	30 ms
Nombre de nœuds actifs	1 – 10 nodes

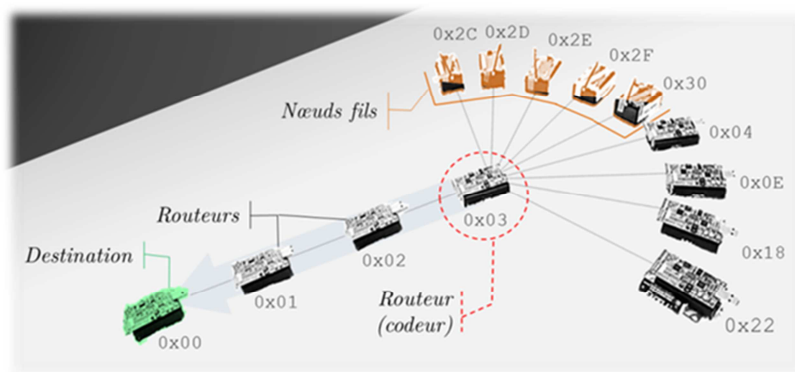


Figure 44. Configuration initiale du banc d'essai

#### 5.3.4.1. Usage mémoire

L'implémentation de ZInC dans TinyOs [96] nécessite entre 33 et 38 Ko de mémoire RAM à chaque capteur, ce qui représente une surcharge de moins de 2% comparé au coût effectif de la pile protocolaire ZigBee dans les deux plateformes matérielles utilisées. Cette surcharge est raisonnable si l'on considère le gain engendré par le codage par index. En outre, l'implémentation de ZInC démontre que les opérations de capture scalaire peuvent être effectuées normalement et que l'ensemble des données est délivré au destinataire sans qu'il y ait un impact notable sur les mécanismes d'ordonnancement des capteurs. Le tableau 7 illustre ce faible impact à la fois sur la ROM et la RAM, démontrant que l'ordonnancement ZInC implique moins de 0,4 % d'excédent mémoire comparé à ZigBee.

Tableau 7. Paramètres de simulation

	TelosB		MicaZ	
	ROM	RAM	ROM	RAM
ZigBee (bytes)	33958	3759	38706	3626
ZInC (bytes)	34693	3773	39171	3638
Surcharge (%)	2,16	0,37	1,20	0,33

#### 5.3.4.2. Nombre de transmissions

ZInC permet aux routeurs intermédiaires de combiner des messages provenant des nœuds actifs en un paquet codé. Un tel codage permet de réduire significativement le nombre de transmissions comme illustré sur la Figure 45. Nous comparons le nombre moyens de transmissions (y compris les retransmissions) pour chaque communication incast en utilisant ZigBee et ZInC. Nous calculons cette moyenne en utilisant différents débits CBR, en faisant varier le nombre de sources.

Les résultats montrent que ZInC permet de transmettre nettement moins de paquets que ZigBee pour un nombre de sources supérieur à 1 (ce qui est normal car nous considérons des communications incast à plusieurs sources). Alors que le nombre de transmissions dans le réseau ZigBee augmente de manière linéaire avec le nombre de sources actives, la courbe de ZInC suit une inclinaison moins forte. Evidemment, l'écart entre les deux protocoles se creuse à mesure que le nombre de sources augmente car les paquets codés embarquent plus de données.



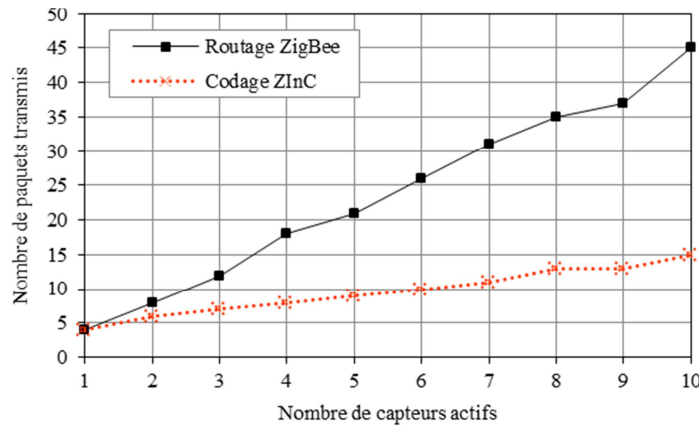


Figure 45. Nombre de transmissions par rapport au nombre de nœuds actifs

Par exemple, quand toutes les sources sont actives (9 nœuds fils plus le nœud codeur), le codage par index réduit le nombre de transmission de 66% par rapport à ZigBee. Cette diminution pourrait être encore plus importante si le nombre de sauts vers la destination était plus important. Dans notre cas, réduire le nombre de paquets transmis garantit deux avantages pour les réseaux de capteurs. Premièrement, une meilleure efficacité énergétique car les nœuds intermédiaires utilisent moins souvent leur module radio et peuvent ainsi passer plus de temps en mode inactif. Le deuxième avantage concerne la réduction de la probabilité de collision et du temps passé à essayer de transmettre un paquet en utilisant la méthode d'accéder au canal radio CSMA/CA<sup>11</sup>.

#### 5.3.4.3. Délai de bout-en-bout

Cette métrique importante dans les réseaux de capteurs, est fortement corrélée aux taux de pertes du réseau. Cela est dû à l'utilisation de la méthode d'accès CSMA/CA lorsque des nœuds essaient d'accéder simultanément au médium radio. En effet, quand le canal est occupé parce qu'il y a trop de paquets transmis dans la même zone, les nœuds calculent un délai aléatoire progressif, dit de *backoff*, qui sera utilisé pour savoir quand il est possible de refaire une tentative de transmission. Bien sûr, plus le canal est occupé, moins il y a de chance pour qu'un paquet soit transmis, ce qui augmente inévitablement le temps de transmission de proche-en-proche et indirectement la latence de bout-en-bout. Dans un réseau incast, nous définissons ce délai par le temps écoulé avant que tous les paquets ne soit correctement reçus/décodés par la destination. En utilisant ZInC nous réduisons l'impact du taux de perte sur la latence, puisque les nœuds transmettent beaucoup moins de paquets et donc réduisent la probabilité de pertes et les retards d'accès au canal.

Comme illustré sur la Figure 46, le délai de bout-en-bout en utilisant ZInC est légèrement plus élevé que celui de ZigBee tant que le nombre de sources actives est inférieur à 5. Ce résultat est la conséquence de la période de mise en tampon nécessaire pour encoder les paquets. Toutefois, au fur et à mesure que le nombre de nœuds actifs augmente, la latence assurée par ZInC diminue, allant jusqu'à atteindre 45 ms de moins que celle de ZigBee.

<sup>11</sup> Cette méthode est utilisée dans la plupart des protocoles MAC/PHY des réseaux de capteurs sans fil dont IEEE 802.15.4., principalement pour sa faible complexité et son efficacité énergétique.

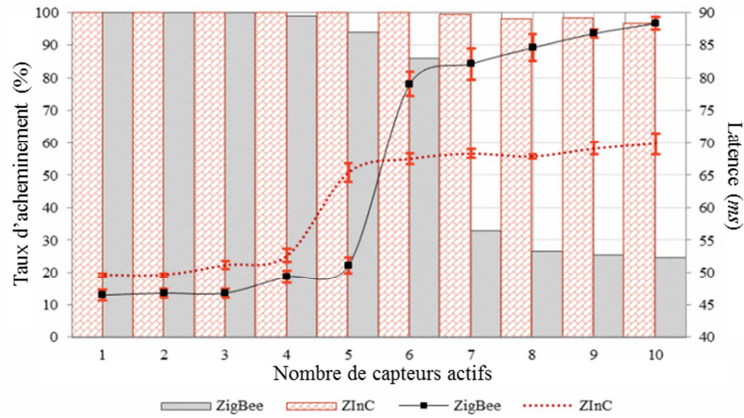


Figure 46. Latence et taux d'acheminement en fonction du nombre de nœuds actifs

Précisons que les paquets perdus ne sont pas pris en compte dans les calculs de latence (parce qu'ils impliquent un délai infini). Ainsi, les résultats obtenus en utilisant ZigBee représentent uniquement les valeurs des paquets qui ont pu arriver jusqu'à la destination.

#### 5.3.4.4. Fiabilité

Pour évaluer la fiabilité de ZInC et de ZigBee, nous générons à chaque nœud source un trafic CBR qui transmet 200 paquets à intervalles réguliers (0,5 paquet/seconde) pendant la durée de l'expérience et dénombrons les paquets effectivement reçus par le puits de données. En calculant le taux d'acheminement à la réception, nous avons pu constater que ZInC peut délivrer jusqu'à 3 fois plus de paquets que ZigBee et cela à mesure que le taux de pertes augmente (car plus de nœuds tentent d'accéder au canal). La Figure 47 montre les résultats de cette expérience et présente le nombre de paquets correctement délivrés à la destination ainsi que le nombre de retransmissions nécessaires pour y parvenir. Il est clair que ZInC devance ZigBee en termes de fiabilité. En effet, alors qu'une approche sans codage souffre très rapidement de l'augmentation du trafic, le codage par index assure des résultats positifs même lorsque tous les capteurs transmettent leurs paquets en même temps, fournissant toujours un taux d'acheminement qui frôle les 100%. Cela est dû à la réduction importante du nombre de transmissions nécessaires à l'acheminement des données collectées.

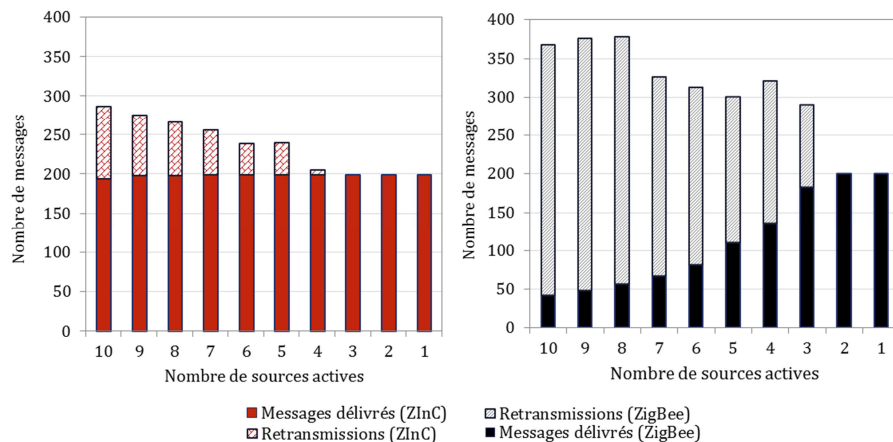


Figure 47. Nombre de paquets correctement reçus par rapport au nombre de nœuds actifs

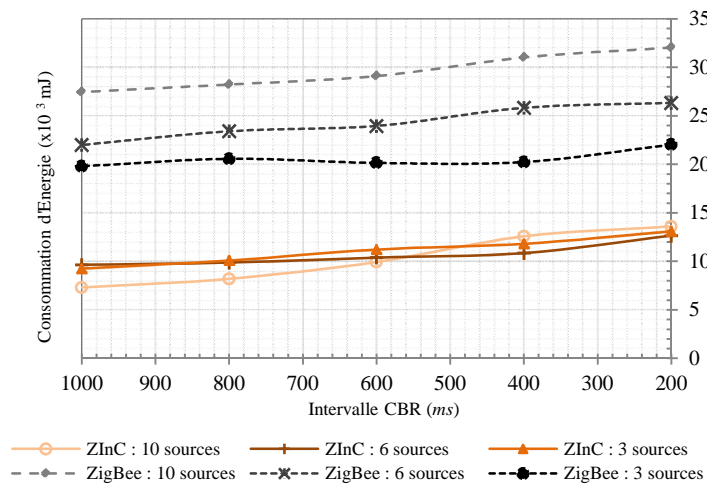


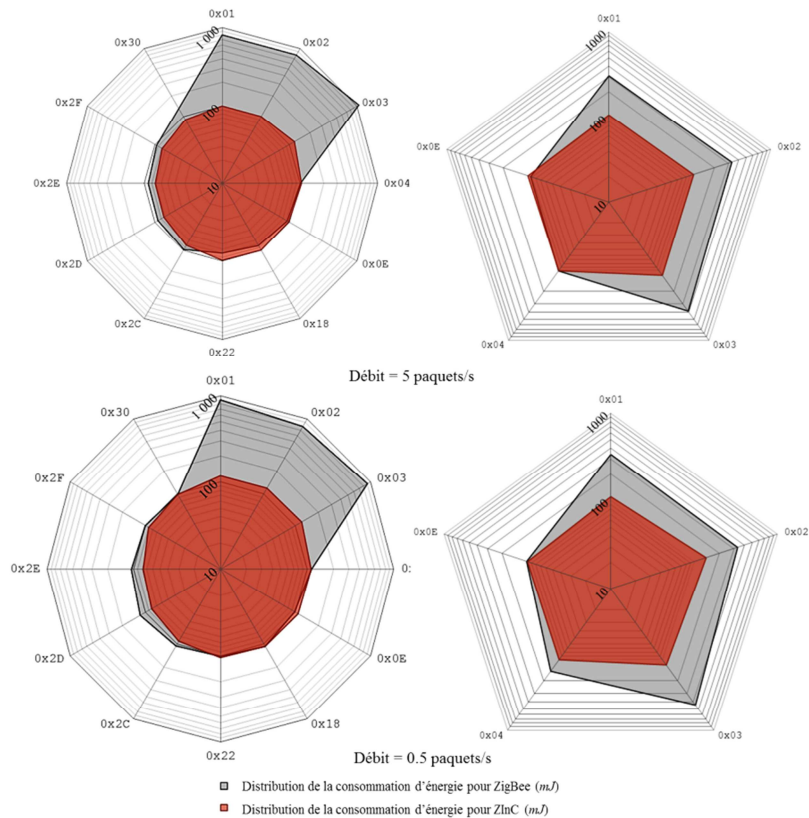
Figure 48. Consommation d'énergie par rapport à l'intervalle CBR

#### 5.3.4.5. Consommation d'énergie

Contrairement à ce que l'on pourrait penser, l'efficacité énergétique n'est pas seulement corrélée au nombre de transmissions, mais plutôt au temps pendant lequel le module de communication du nœud est activé. En effet, [65] affirme que les nœuds qui écoutent le canal radio consomment environ la même quantité d'énergie que ceux qui transmettent constamment des paquets. Ainsi, pour améliorer l'efficacité énergétique des nœuds, il est préférable de prolonger le délai d'inactivité du module radio plutôt que de chercher uniquement à réduire le nombre de transmissions.

ZInC permet aux routeurs impliqués dans des communications incast de passer en mode sommeil (inactif) plus souvent qu'avec ZigBee, puisque le codage par index ne nécessite pas qu'ils relayent l'ensemble des paquets sources. Au contraire, un routeur dans ZInC ne transmet qu'un seul paquet codé avant de passer en mode sommeil jusqu'au prochain cycle de capture. Dans notre expérience, nous quantifions la valeur de la consommation d'énergie qui est estimée par chaque nœud localement et transmise au puits de données à la fin de chaque expérience. La Figure 48 montre cette réduction de la consommation d'énergie lorsque l'intensité du trafic augmente, i.e. lorsque la fréquence des sources CBR augmente. ZInC garantit des communications incast moins gourmandes en énergie, surtout lorsque le nombre de nœuds actifs est élevé. Par exemple, on peut constater que ZInC permet d'acheminer la même quantité de données de 10 nœuds sources en consommant seulement 33% de l'énergie requise pour ZigBee pour acheminer la même quantité d'information.

De plus, nous montrons la distribution de la consommation d'énergie en fonction de la profondeur des nœuds dans le réseau. Cette métrique est très importante dans les réseaux contraints car elle permet de détecter la position des nœuds surexploités dans le réseau, i.e. ceux qui ont dissipé le plus d'énergie et qui sont donc plus sujets aux pannes. ZInC permet de garantir une distribution équitable de la consommation de ressources à travers le réseau. En fait, avec ZigBee, les routeurs les plus proches du puits de données sont plus sollicités et relayent plus de paquets que les autres nœuds, ce qui réduit leur temps d'inactivité. La Figure 49 montre à quel point ZInC permet une meilleure répartition de la consommation d'énergie avec deux débits différents et en variant le nombre de nœuds sources.



**Figure 49.** Distribution de la consommation d'énergie par rapport à l'intervalle CBR et au nombre de nœuds sources

On peut voir que lors de l'utilisation du codage par index, les routeurs (0x01, 0x02 et 0x03) consomment environ la même quantité d'énergie que les nœuds sources puisqu'ils relayent le même nombre de paquets et qu'ensuite ils passent en mode sommeil jusqu'au prochain cycle de capture. Par contre, dans le cas de ZigBee, ces mêmes routeurs consomment beaucoup plus d'énergie et sont donc plus sujets aux défaillances que les autres. Bien sûr, cet avantage est réduit avec la diminution du nombre de nœuds actifs, dans la mesure où moins de paquets doivent être relayés par les routeurs.

#### 5.4. Conclusion

Tout au long de ce chapitre, nous avons présenté le codage par index (InC), une technique innovante de codage réseau contraint conçue pour assurer des communications incast fiables et efficaces dans un réseau de capteurs sans fil. Ce nouveau système de codage qui exploite les propriétés naturelles des réseaux fortement contraints, a pour but de montrer l'adaptabilité des techniques de codage réseau sans fil aux limitations qu'impliquent les architectures sans fil inter-machines. Notre objectif est de montrer que le codage par index apporte une valeur ajoutée au codage réseau en améliorant l'utilisation de la bande passante tout en limitant sa consommation des ressources par nœud.

La simplicité et la nature distribuée du codage par index en fait une solution idéale pour différentes applications des réseaux incast fortement contraints telles que la supervision périodique ou les systèmes de *observateurs/observés*. Dans notre évaluation analytique, nous

définissons un cadre théorique qui permet de valider l'utilisation de InC. Elle prouve que cette technique de codage permet un décodage systématique à la destination tout en garantissant efficacité et passage à l'échelle.

L'étude empirique pour sa part, décrit ZInC, un protocole de communication basé sur le codage par index pour les réseaux de capteurs ZigBee, qui permet la collecte fiable et rapide de l'information dans un environnement réaliste. Ce cas d'étude pratique démontre que le codage par index améliore significativement les performances, la fiabilité et l'efficacité énergétique des communications incast. De plus, ZInC permet de montrer que le codage réseau contraint est une solution intéressante et réaliste pour l'amélioration des communications dans les réseaux inter-machines.

A notre connaissance, ZInC est la première application pratique des méthodes de codage réseau qui traite le problème de l'incast dans les réseaux de capteurs sans fil, et qui fournit des réponses concluantes tant en termes de performances et des besoins des réseaux contraints tels que l'efficacité énergétique, l'usage mémoire et les coûts processeur.





## *Chapitre 6*

# 6. Conclusion & perspectives

### *6.1. Conclusion*

Cette thèse s'articule autour d'une étude transversale qui a pour but la conception de nouveaux modèles d'acheminement de l'information dans les réseaux de capteurs sans fil. Le succès de ces réseaux dépend des efforts mis en œuvre pour permettre à la fois des communications rapides et fiables, tout en garantissant une consommation des ressources équitable et optimale des nœuds du réseau.

Jusqu'à maintenant les concepteurs des protocoles de communication pour ces réseaux ont d'avantage cherché à privilégier un des critères de performance, généralement au détriment des autres. Les différents développements et implémentations ont, quant à eux, concentré leurs efforts à tolérer les contraintes réseaux plutôt qu'à en améliorer la gestion. Dans ce travail, nous avons essayé d'avoir une approche différente et d'exploiter certaines de ces contraintes pour optimiser les communications de bout-en-bout. Que ce soit l'exploitation des réceptions multiples propres aux broadcast (pour la dissémination), les liens d'écoute dans les topologies hiérarchiques (pour les flux unicast), ou encore la taille réduite de la charge utile constatée dans les réseaux très contraints (pour les flux incast). Toutes ces caractéristiques nous ont permis de réfléchir à des solutions efficaces pour répondre à la sous-optimalité des modèles de communication en *store-and-forward* et de les revisiter en leur intégrant des mécanismes de codage réseau existants ou inédits.

De plus, notre étude va plus loin qu'une simple utilisation du codage réseau, nous y traitons son adaptabilité au médium sans fil (pertes, collisions, interférences) mais aussi aux contraintes des nœuds qui composent le réseau (mémoire, énergie, processeur). Ainsi, nous introduisons des formulations nouvelles qui permettent de mieux appréhender la fiabilité des communications codées et des systèmes qui s'appuient sur la nature limitée du réseau pour fournir des performances élevées dans un environnement réel.

Nous pensons qu'en vue des gains substantiels qu'elle promet, et ne serait-ce que par sa méthodologie, cette analyse transversale devrait être prise en compte lors de la conception de futurs systèmes de communication pour les réseaux inter-machines.



Nous avons étudié les apports prospectifs de l'utilisation du codage réseau dans les réseaux de capteurs sans fil. Cette étude basée sur l'appréciation à la fois du bénéfice, de la fiabilité et de la faisabilité des techniques du codage vise à répondre à trois questions importantes :

1. *Quel type de codage réseau pour quel réseau de capteurs ?*
2. *Qu'est-ce qu'un codage réseau fiable ? Et comment le garantir ?*
3. *Comment coder dans un environnement fortement contraint ?*

Dans ce qui suit, nous résumons les principales contributions de cette thèse, lesquelles ont pour objectif de répondre à ces trois questions.

**Codage réseau inter-machines.** Les premiers éléments de réponse à la première question sont donnés par l'évaluation des performances des techniques de codage linéaire et opportuniste simplement en termes de débit et de délai. Nous y montrons via WSC que le codage réseau linéaire constitue la solution la plus adéquate dans le cas des réseaux de capteurs maillés, car en plus d'améliorer le débit global lors de la dissémination de données, il permet aussi des temps de convergence réduits. D'autre part, nous présentons CNC et COZI, deux approches basées sur le codage réseau opportuniste qui permettent des communications optimisées dans les topologies hiérarchiques largement utilisées dans les réseaux inter-machines. Lors de l'évaluation de ces deux protocoles, nous avons pu mettre en exergue l'apport du codage à un saut pour différentes applications de supervision et d'orchestration inter-machines.

En proposant ces adaptations des techniques de codage aux différents types de réseaux de capteurs, nous avons pu identifier les méthodes les plus adaptées aux contraintes de chaque application. Les combinaisons linéaires sont idéales pour des communications à sources et destinations multiples sur des topologies maillées. En effet, de par sa nature aléatoire, le codage réseau linéaire peut assurer la propagation fiable et rapide des données pour des modèles de trafic asynchrones. Pour sa part, le codage opportuniste qui assure un décodage de proche-en-proche, convient plus aux applications à destination unique dans lesquelles des flux circulent selon un ensemble limité de chemins possibles (comme dans un réseau hiérarchique).

**Codage réseau et fiabilité.** Dans un deuxième temps, nous nous sommes intéressés à la question de la fiabilité et de la tolérance aux pertes du codage réseau sans fil. Grâce à l'analyse de l'impact du taux de pertes sur l'efficacité du codage, nous avons pu montrer que la fiabilité des communications codées ne peut pas être estimée de la même façon que dans un réseau en *store-and-forward*. Puis nous avons défini une série de métriques qui permettent une évaluation objective de ce que nous appelons un *codage réseau fiable*. Un code est dit fiable s'il permet à un maximum de ses destinataires de recouvrer de l'information utile même lorsque le réseau subit des pertes non négligeables. A partir de cette définition, nous avons proposé ReCoZi, un système de retransmission conçu pour assurer un codage inter-machines robuste qui garantit un traitement équitable des flux de données, un taux d'acheminement optimisé et un ordonnancement efficaces des paquets. Basé sur un codage opportuniste et des mécanismes d'acquittement à état de liens, ReCoZi permet de garder la valeur ajoutée du codage réseau sans fil lorsque le canal de

communication devient trop erratique. Cet effort a permis l'établissement de critères de fiabilisation précis qui font du codage réseau sans fil une solution plus réaliste au problème de la sous-optimalité des protocoles de communication en *store-and-forward*.

**Codage dans les réseaux contraints.** Cette étude aborde la problématique la plus importante à laquelle sont confrontés les protocoles de codage inter-machines existants, à savoir *l'inadaptabilité*. En effet, même si plusieurs techniques de codage offrent des résultats théoriques prometteurs, ces dernières sont souvent basées sur des hypothèses non réalistes qui ignorent les réalités pratiques des réseaux de capteurs par des simplifications analytiques ou des approximations qui sont parfois discutables. Nous avons montré que dans le cadre d'un réseau fortement contraint, ni le codage opportuniste, ni le codage linéaire ne peuvent opérer car les deux nécessitent des ressources (processeur et mémoire) qui ne sont pas disponibles. En conséquence, nous avons conçu un nouveau système de codage par index spécialement réfléchi pour des applications réelles de supervisions ou de surveillance qui utilisent l'incast comme modèle d'acheminement de l'information à partir des nœuds capteurs vers le puits de données. Il s'agit d'encoder des flux de paquets provenant de plusieurs sources et de transmettre de proche-en-proche le résultat à la destination. Afin d'évaluer notre système de codage par index, nous avons d'abord calculé ses limites théoriques grâce à un modèle analytique, puis nous avons implémenté ses concepts sur un vrai réseau de capteurs. Les résultats, qu'ils soient analytiques ou expérimentaux ont démontré une réduction significative du nombre de transmissions, des délais de bout-en-bout et de la consommation d'énergie. Tout cela en gardant des coûts mémoire et processeur très faible et une surcharge par paquet négligeable.

## 6.2. Perspectives

Une première extension possible et intéressante de nos travaux serait d'aborder l'évaluation du coût du codage réseau sous un angle plus pratique. Cela se traduit par l'étude de l'influence de plusieurs critères tels que la complexité, les méthodes de gestion des files d'attente ou encore les mécanismes d'économie d'énergie (activation désactivation des nœuds) et cela afin de constater leur impact sur l'efficacité du codage qu'il soit linéaire ou opportuniste.

Dans un effort d'amélioration du codage réseau sans fil d'une façon plus générale, nous voyons dans le domaine de la fiabilisation des réseaux codés une piste de recherche intéressante qui prélude de prochains travaux et cela au-delà même des réseaux de capteurs sans fil. En effet, alors que notre solution pour fiabiliser les codes inter-machines se base sur un système de retransmission qui permet uniquement de détecter la perte ou la corruption d'un paquet, il serait peut-être avantageux de considérer des systèmes plus bas niveau qui permettraient la correction d'erreurs à la volée dans des flux codés.

Enfin, nous avons constaté lors de son évaluation de performance que le codage par index peut, dans une certaine mesure, s'avérer inutile, voir pénalisant, lorsque la taille des paquets sources dépasse un certain seuil. Une amélioration dans cet axe pourrait étendre l'utilisation de ce nouveau système de codage à des applications plus diverses de

communications incast, comme dans les réseaux de capteurs multimédias où la taille des messages est plus conséquente. De plus, l'introduction d'un mécanisme plus sophistiqué de gestion des files d'attente pourrait rendre le codage par index beaucoup plus efficace en termes de délai d'acheminement.

# Publications

## *Articles soumis*

*On the Analysis of Index-Coding for Incast Communications in Sensor Networks,*  
Ismail Salhi, Yacine Ghamri-Doudane, Stéphane Lohier, Mario Gerla et Erwan Livolant.  
(Soumis à *Transaction on Mobile Computing*)

*Reliable Network Coding for ZigBee Sensor Networks,*  
Ismail Salhi, Yacine Ghamri-Doudane et Stéphane Lohier,  
(Soumis à *Elsevier Computer Networks 2012*)

## *Articles parus*

1. *ZInC: Index-Coding for Many-to-One Communications in ZigBee Sensor Networks,*  
Ismail Salhi, Erwan Livolant, Yacine Ghamri-Doudane et Stéphane Lohier  
IEEE/ICC'12, Ottawa, Canada, Juin 2012. (à paraître)
2. *Wireless Network Simulators Relevance Compared to a Real Testbed in Outdoor and Indoor Environments,* A. Rachedi, Stéphane Lohier, Sylvain Cherrier et Ismail Salhi. International Journal of Autonomous and Adaptive Communications Systems (IJAACS) Volume 5, N°1, 2012.
3. *When Network Coding improves the Performances of Clustered Wireless Networks,*  
Ismail Salhi, Yacine Ghamri-Doudane, Stéphane Lohier et Gilles Roussel  
IFIP/Wireless Days'11, Ontario, Canada, Octobre 2011.
4. *Reliable Network Coding for ZigBee Wireless Sensor Networks, (article court)*  
Ismail Salhi, Yacine Ghamri-Doudane, Stéphane Lohier et Gilles Roussel  
IEEE/MASS'11, Valence, Espagne, Octobre 2011.
5. *Wireless Sensor Network Simulators Relevance compared to a real IEEE 802.15.4 testbed,*  
Abderrezak Rachedi, Stéphane Lohier, Erwan Livolant et Ismail Salhi  
IEEE/IWCMC'11, Istanbul, Turquie, Juillet 2011.
6. *Architecting the Internet of Things,* dans le chapitre : *The DiY Smart Experiences Project* ,  
Marc Roelands, Johan Plomp, Diego C. Mansilla, Juan R. Velasco, Ismail Salhi, *et al.* Springer,  
ISBN 978-3-642-19156-5, Mars 2011.

7. *CoZi: basic Coding for better Bandwidth Utilization in ZigBee Sensor Networks*,  
Ismail Salhi, Yacine Ghamri-Doudane, Stéphane Lohier et Erwan Livolant  
IEEE/GLOBECOM'10 Miami, USA, Décembre 2010.
8. *Network Coding for Event-Centric Wireless Sensor Networks*,  
Ismail Salhi, Yacine Ghamri-Doudane, Stéphane Lohier et Gilles Roussel  
IEEE/ICC'10, Le Cape, Afrique du Sud, Mai 2010.
9. *Wireless Network Simulators Relevance Compared to a Real Testbed in Outdoor and Indoor Environments*,  
Abderrezak Rachedi, Stéphane Lohier, Ismail Salhi et Sylvain Cherrier,  
IEEE/IWCMC'10, Caen, France, Mars 2010.
10. *A New Architecture for Data Collection in Vehicular Networks*,  
Ismail Salhi, Sidi-Mohammed Senouci and Mohammed Cherif,  
IEEE/ICC'09, Dresde, Allemagne, juin, 2009.

# Références

- [1] I. Akyildiz, Wireless sensor networks: a survey, *Computer Networks*, vol. 38, no. 4, pp. 393-422, Mar. 2002..
- [2] A. Nayak and I. Stojmenovic. *Wireless Sensor and Actuator Networks: Algorithms and Protocols for Scalable Coordination and Data Communication*, Wiley, 2010.
- [3] I. Akyildiz, Wireless sensor and actor networks: research challenges, *Ad hoc networks*, vol. 2, pp. 351-367, 2004.
- [4] Dohler, Mischa, Thomas Watteyne, and Jesús Alonso-zárate. Machine-to-Machine: An Emerging Communication Paradigm. In *ICST Conference on Mobile Networks Management MONAMI2010*, 1-38.
- [5] J. Zhao and R. Govindan, Understanding packet delivery performance in dense wireless sensor networks, in *the international conference on Embedded networked sensor systems - SenSys'03*, 2003, p. 1.
- [6] K. Akkaya and M. Younis, A survey on routing protocols for wireless sensor networks, *Ad Hoc Networks*, vol. 3, pp. 325-349, 2005.
- [7] I. Demirkol, C. Ersoy and F. Alagoz, MAC protocols for wireless sensor networks: a survey. *IEEE Communications Magazine*, 115-121, 2006.
- [8] D. Bein, V. Jolly, B. Kumar and S. Latifi, Reliability Modeling in Wireless Sensor Networks. *International Journal of Information Technology*, 1-9, 2005.
- [9] Jie Wu, Ad Hoc Networks. *Ad Hoc Networks*, 29-31, 2004.
- [10] A. Awan, S. Jagannathan and A. Grama, Scalable Data Collection in Sensor Networks. *15th Annual IEEE International Conference on High Performance Computing*, 2008.
- [11] D. Uckelmann, M. Harrison, F. Michahelles, D. Uckelmann, M. Harrison, F. Michahelles, Architecting the Internet of Things. *Architecting the Internet of Things* 1-25, 2011.
- [12] I. Amundson, X. D. Koutsoukos, R. Fuller and X. D. Koutsoukos, A Survey on Localization for Mobile Wireless Sensor Networks. *Networks*, 235-254, 2009.
- [13] C. Schurgers, V. Tsiatsis, S. Ganeriwal and M. Srivastava, Topology Management for Sensor Networks : Exploiting Latency and Density. *Topology* 135-145, 2002.
- [14] C. E. Perkins and E. M. Royer, Ad-hoc on-demand distance vector routing. *ACM SIGMOBILE Mobile Computing and Communications Review*, 90-100, 1999.
- [15] ZigBee Alliance, ZigBee Specifications. December, 2006.
- [16] I. Stojmenovic and S. Olariu, Data-Centric Protocols for Wireless Sensor Networks. *Communications* 417-456, 2005.
- [17] G. Mulligan, The 6LoWPAN architecture. *Energy* 78-82, 2007.

- [18] O. Younis, M. Krunz, and S. Ramasubramanian, Node clustering in wireless sensor networks: recent developments and deployment challenges, *IEEE Network*, vol. 20, no. 3, pp. 20-25, May 2006.
- [19] M. Younis, P. Munshi, G. Gupta, and S. M. Elsharkawy, On Efficient Clustering of Wireless Sensor Networks, *System*, pp. 78-91, 2006.
- [20] IEEE Standard for Information Technology, Part 15.4: wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs). 2003.
- [21] J. Polastre, J. Hill and D. Culler, Versatile low power media access for wireless sensor networks. *SenSys'04*, (ACM Press), pp. 95-107, 2004.
- [22] S. Lee, Y. Yu, S. Nelakuditi, Z.-L. Zhang and C.-N. Chuah, Proactive vs reactive approaches to failure resilient routing. *INFOCOM*, 176-186, 2004.
- [23] I. Stojmenovic, Position-based routing in ad hoc networks. *IEEE Communications Magazine*, 128-134, 2002.
- [24] F. Cuomo, S. D. Luna, U. Monaco and T. Melodia, Routing in ZigBee: Benefits from Exploiting the IEEE 802.15.4 Association Tree. *IEEE International Conference on Communications*, 3271-3276, 2007.
- [25] M. Chatterjee, S. K. Das, and D. Turgut, WCA: A weighted clustering algorithm for mobile ad hoc networks, *Cluster Computing*, vol. 5, no. 2, pp. 193-204, 2002.
- [26] J.-Z. Sun, Dissemination Protocols for Reprogramming Wireless Sensor Networks: A Literature Survey. *International Conference on Sensor Technologies and Applications* 151-156, 2010.
- [27] A. N. Le, D.-W. Kum, S.-H. Lee, Y.-Z. Cho and I.-S. Lee, Directional AODV Routing Protocol for Wireless Mesh Networks. *IEEE International Symposium on Personal Indoor and Mobile Radio Communications*, 2007.
- [28] R. Liscano, Service Discovery in Sensor Networks: An Overview. *Networks*, 2003.
- [29] O. Younis and S. Fahmy, Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach. *IEEE Infocom*, 629-640, 2004.
- [30] W. R. Heinzelman, J. Kulik and H. Balakrishnan, Adaptive protocols for information dissemination in wireless sensor networks. *ACM/IEEE MobiCom*, 174-185, 1999.
- [31] R. C. Shah, S. Roy, S. Jain and W. Brunette, Data MULEs: modeling and analysis of a three-tier architecture for sparse sensor networks. *Ad Hoc Networks*, 215-233, 2003.
- [32] J. Barros, C. Peraki and S. D. Servetto, Efficient network architectures for sensor reachback, in *International Zurich Seminar on Communications, 2004*, pp. 184-187.
- [33] A. Toledo and X. Wang, Efficient Multipath in Sensor Networks using Diffusion and Network Coding, *Conference on Information Sciences and Systems*, pp. 87-92, 2006.
- [34] I. Solis and K. Obraczka, In-network aggregation trade-offs for data collection in wireless sensor networks. *International Journal of Sensor Networks*, 200-212, 2006.
- [35] J.-M. Verbree, M. De Graaf and J. Hurink, An analysis of the lifetime of OLSR networks. *Ad Hoc Networks*, 391-399 (2010).

- [36] Z. Rosberg, R. Liu, A. Dong, T. Le Dinh and S. Jha, ARQ with Implicit and Explicit ACKs in Sensor Networks. *IEEE Globecom*, 2008.
- [37] C. Fragouli, J. Widmer and J. Y. Le Boudec, On the Benefits of Network Coding for Wireless Applications, *International Symposium on Modeling and Optimization in Mobile Ad Hoc and Wireless Networks*, 1-6, 2006.
- [38] C. Wang and H. Wei, IEEE 802.11, IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specification. *Mobile Networks and Applications*, 2009.
- [39] ZigBee Alliance : <http://www.ZigBee.org>.
- [40] IPSO Alliance : <http://www.ipso-alliance.org/>
- [41] C. E. Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, ISSN 1559-1662. 2001.
- [42] R. Ahlswede, N. Cai, S.-yen R. Li, and R. W. Yeung, Network information flow, *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204-1216, 2000.
- [43] C. Fragouli and E. Soljanin, Network Coding Applications. *Foundations and Trends in Networking*, 135-269, 2007.
- [44] C. Fragouli and E. Soljanin, Network Coding Fundamentals. *Foundations and Trends in Networking*, 1-133, 2006.
- [45] T. Matsuda, T. Noguchi and T. Takine, Survey of Network Coding and Its Applications. *IEICE Transactions on Communications*, 698-717, 2011.
- [46] T. Ho et al., A Random Linear Network Coding Approach to Multicast, *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413-4430, 2006.
- [47] S. Katti, H. Rahul, D. Katabi, M. Medard and J. Crowcroft, XORs in the Air: Practical Wireless Network Coding. *IEEE/ACM Transactions on Networking*, 497-510, 2008.
- [48] J.-sang Park, D. S. Lung, F. Soldo, M. Gerlat and M. Medard, Performance of network coding in ad hoc networks, *Ad Hoc Networks*, 2006.
- [49] K. Han, T. Ho, R. Koetter, M. Medard, and F. Zhao, On network coding for security, *IEEE Military Communications Conference*, 2007.
- [50] C. Fragouli, J.-Y. L. Boudec and J. Widmer, Network coding: an instant primer. *ACM SIGCOMM Computer Communication Review*, 2006.
- [51] H. Seferoglu and A. Markopoulou, I2NC: Intra-and inter-session network coding for unicast flows in wireless networks, *INFOCOM*, 2010.
- [52] S. Katti, D. Katabi, W. Hu, H. Rahul, and M. Medard, The importance of being opportunistic: Practical network coding for wireless environments, in *Allerton Conference on Communication, Control, and Computing*, 2005.
- [53] P. Sadeghi, D. Traskov and R. Koetter, Adaptive Network Coding for Broadcast Channels. *Network*, 80-85, 2009.
- [54] C. Fragouli, D. Katabi, A. Markopoulou, M. Medard, and H. Rahul, Wireless Network Coding: Opportunities & Challenges, in *IEEE Military Communications Conference*, pp. 1-8, 2007.



- [55] T. Ho, R. Koetter, M. Medard, D. R. Karger and M. Effros, The benefits of coding over routing in a randomized setting. *IEEE International Symposium on Information Theory*, p 442-448, 2003.
- [56] C. Gkantsidis and P. Rodriguez, Network coding for large scale content distribution, *in. IEEE INFOCOM*, 2005.
- [57] A. Al Hamra, C. Barakat, and T. Turletti, Network coding for wireless mesh networks: A case study. In *IEEE WOWMOM*. 2006.
- [58] L. Lima, J. P. Vilela, P. F. Oliveira, and J. Barros, Network coding security: Attacks and countermeasures, *Computing Research Repository*, Sep. 2008. <http://arxiv.org/abs/0809>.
- [59] S. Jaggi et al., Resilient Network Coding in the Presence of Byzantine Adversaries, *IEEE Transactions on Information Theory*, vol. 54, no. 6, pp. 2596-2603, 2008.
- [60] D. Silva and F. R. Kschischang, On Metrics for Error Correction in Network Coding. *IEEE Transactions on Information Theory*, p28, 2008.
- [61] J. Le, J. C. S. Lui and D.-M. Chiu, On the Performance Bounds of Practical Wireless Network Coding. *IEEE Transactions on Mobile Computing* **9**, 1134-1146, 2010.
- [62] M. Ghaderi, D. Towsley and J. Kurose, Reliability Gain of Network Coding in Lossy Wireless Networks. *IEEE INFOCOM*, 2008.
- [63] J.-sang Park, M. Gerla, D. Lun, Y. Yi, and M. Medard, Codecast: a network-coding-based ad hoc multicast protocol, *IEEE Wireless Communications*, vol. 13, no. 5, pp. 76-81, 2006.
- [64] D. Katabi, S. R. Katti, and others, Network coded wireless architecture, Thesis, Massachusetts Institute of Technology, 2008.
- [65] A. Dunkels, F. Osterlind, N. Tsiftes, and Z. He, Software-based on-line energy estimation for sensor nodes, in *EmNet*, p. 28, 2007.
- [66] C. Fragouli, J. Widmer and J. Y. Le Boudec, Efficient Broadcasting Using Network Coding. *IEEE/ACM Transactions on Networking*, 450-463, 2008.
- [67] Y. Sasson, D. Cavin and A. Schiper, Probabilistic broadcast for flooding in wireless mobile ad hoc networks. *IEEE Wireless Communications and Networking*, 1124-1130, 2002.
- [68] D. Platz, D. H. Woldegebreal and H. Karl, Random Network Coding in Wireless Sensor Networks: Energy Efficiency via Cross-Layer Approach. *IEEE International Symposium on Spread Spectrum Techniques and Applications*, 654-660, 2008.
- [69] C. Steger and R. Weiss, Opportunistic Network Coding for Energy Conservation in Wireless Sensor Networks. *Energy*, 2011.
- [70] T. Cui, L. Chen, T. Ho, S. H. Low and L. H. Andrew, Opportunistic Source Coding for Data Gathering in Wireless Sensor Networks. *IEEE International Conference on Mobile Adhoc and Sensor Systems*, 1-11, 2007.
- [71] I.-H. Hou, Y.-E. Tsai, T. F. Abdelzaher, and I. Gupta, AdapCode: Adaptive Network Coding for Code Updates in Wireless Sensor Networks, *IEEE INFOCOM*, pp. 1517-1525, 2008..
- [72] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 4, pp. 56-67, 1998.

- [73] A. Fujimura, S. Y. Oh and M. Gerla, Network Coding vs. Erasure Coding : Reliable Multicast in Ad hoc Networks. *Ad Hoc Networks*, 1-7, 2008.
- [74] L. Rizzo. Effective erasure codes for reliable computer communication protocols. *ACM SIGCOMM Computer Communication Review*, 1997.
- [75] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege. A digital fountain approach to reliable distribution of bulk data. *ACM SIGCOMM Computer Communication Review*, 28(4):56–67, 1998.
- [76] E. Soljanin, N. Varnica, and P. Whiting. Incremental redundancy hybrid ARQ with ldpc and raptor codes. *IEEE Transactions on Information Theory*, 2005.
- [77] Y. Sankarasubramaniam, O. B. Akan and I. F. Akyildiz, ESRT: event-to-sink reliable transport in wireless sensor networks. *ACM international symposium on Mobile ad hoc networking computing* , 177-188, 2003.
- [78] V. Naik, A. Arora, P. Sinha and H. Zhang, Sprinkler: A reliable and scalable data dissemination service for wireless embedded devices. *IEEE International RealTime Systems*, 1-15, 2005.
- [79] P. Dutta, M. Grimmer, A. Arora, S. Bibyk and D. Culler, Design of a wireless sensor network platform for detecting rare, random, and ephemeral events. *International Symposium on Information Processing in Sensor Networks*, 497-502, 2005.
- [80] T. Ho, M. Médard, J. Shi, M. Effros, and D. Karger, On randomized network coding, in the *Allerton Conference on Communication Control and Computing*, vol.41, pp. 11–20, 2003.
- [81] R. Koetter and M. Medard, An algebraic approach to network coding. *IEEE/ACM Transactions on Networking*, 782-795, 2003.
- [82] W. R. Heinzelman, J. Kulik and H. Balakrishnan, Adaptive protocols for information dissemination in wireless sensor networks. *ACM/IEEE international conference on Mobile computing and networking MobiCom*, 174-185, 1999.
- [83] S. M. Hedetniemi, S. T. Hedetniemi and A. L. Liestman, A survey of gossiping and broadcasting in communication networks. *Networks*, 319-349, 2006.
- [84] Scalable Networks – Qualnet 4.5 : <http://www.scalable-networks.com>.
- [85] P. Rodrigues, J. Gama, and M. Gaber, Clustering Techniques in Sensor Networks. *Networks*, 125-142, 2007.
- [86] A. Abbasi and M. Younis, A survey on clustering algorithms for wireless sensor networks, *Computer Communications*, vol. 30, no. 14-15, pp. 2826-2841, 2007.
- [87] J. C. Hou, Dynamic clustering for acoustic target tracking in wireless sensor networks. *IEEE Transactions on Mobile Computing* **3**, 258-271, 2004.
- [88] H. Lu, J. Li and H. Kameda, A Secure Routing Protocol for Cluster-based Wireless Sensor Networks Using ID-based Digital Signature. *Communications Society*, 2010.
- [89] Z. Fan and H. Zhou, A Distributed Weight-Based Clustering Algorithm for WSNs, *International Conference on Wireless Communications Networking and Mobile Computing* 1-5, 2006.
- [90] C. Shen, C. Srisathapornphat and C. Jaikaeo, Sensor information networking architecture and applications. *IEEE Personal Communications*, 52-59, 2001.

- [91] D. Christin, A. Reinhardt, P. Mogre, R. Steinmetz, J. Silva, B. Krishnamachari and F. Boavida, Wireless Sensor Networks and the Internet of Things: Selected Challenges. *Structural Health Monitoring*, 31-33, 2009.
- [92] E. J. Coyle, An energy efficient hierarchical clustering algorithm for wireless sensor networks, *IEEE INFOCOM*, pp.1713-1723, 2003.
- [93] D. Bein, V. Jolly, B. Kumar and S. Latifi, Reliability Modeling in Wireless Sensor Networks. *International Journal of Information Technology*, 1-9, 2005.
- [94] J. P. Wagner and P. Frossard, Joint Playback Delay and Buffer Optimization in Scalable Video Streaming. *IEEE Transactions on Circuits and Systems for Video Technology*, 2008.
- [95] H. Seferoglu and A. Markopoulou, Distributed Rate Control for Video Streaming over Wireless Networks with Intersession Network Coding. *In Packet Video Workshop*, 2009.
- [96] J. Hill, P. Bounadonna and D. Culler, Active Message Communication for Tiny Network Sensors. *In Joint Conference of the IEEE Computer and Communications Societies*, 2001.
- [97] F. A. Tobagi, S. E. Lieberston and L. Kleinrock. On measurement facilities in packet radio systems. *In ACM AFIPS'76*. 1976.
- [98] J. Barros and S. D. Servetto, The sensor reachback problem, in *IEEE Transactions on Information Theory*, vol. 238271, pp. 1-49, 2003.
- [99] J. Barros and S. D. Servetto, The sensor reachback problem, in *IEEE Transactions Information Theory*, pp. 1-49, 2003.
- [100] J. Barros and S. D. Servetto, "Network information flow with correlated sources," *IEEE Transactions on Information Theory*, vol. 52, pp. 155-170, 2006.
- [101] R. Koetter and F. R. Kschischang, Coding for Errors and Erasures in Random Network Coding, in *IEEE Transactions on Information Theory*, vol. 54, pp. 3579-3591, 2008.
- [102] J F. J. MacWilliams and N. J. A. Sloane, The Theory of Error Correcting Codes. In *New York: North-Holland*, 1977.
- [103] M. Jafari, L. Keller, C. Fragouli, and K. Argyraki, Compressed network coding vectors, in *IEEE International Symposium on Information Theory*, pp. 109-113, 2009.
- [104] P. A. Levis, TinyOS: An Open Operating System for Wireless Sensor Networks, in *7th International Conference on Mobile Data Management*, pp. 63-63, 2006.

# Tables des figures

Figure 1. Classes d'architectures des réseaux de capteurs sans fil .....	9
Figure 2. Exemple de topologies des réseaux de capteurs sans fil .....	10
Figure 3. Exemple de codage linéaire.....	16
Figure 4. Exemple de codage opportuniste .....	17
Figure 5. Vue d'ensemble d'un réseau de capteurs basé sur des évènements.....	25
Figure 6. Système de codage WSC.....	27
Figure 7. Structure des messages codés de WSC.....	28
Figure 8. Problème de dépendance linéaire.....	29
Figure 9. Exemple de surcodage.....	30
Figure 10. Temps de convergence et fréquence d'apparition des évènements.....	31
Figure 11. Temps de convergence et fréquence d'apparition des évènements.....	32
Figure 12. Nombre de messages vs. Fréquence d'apparition des évènements .....	32
Figure 13. Liens de chevauchement dans un réseau hiérarchique .....	33
Figure 14. Vue schématique du fonctionnement de CNC dans le temps.....	35
Figure 15. Exemple d'un codage opportuniste.....	36
Figure 16. Evolution du taux d'acheminement des paquets dans le temps.....	38
Figure 17. Nombre de messages transmis en fonction du nombre de sources.....	38
Figure 18. Vue d'ensemble d'un réseau de capteurs basé orchestration.....	39
Figure 19. Mécanisme d'ordonnancement de CoZi .....	41
Figure 20. Structure d'une trame 802.15.4.....	42
Figure 21. Structure d'une trame 802.15.4 codée.....	42
Figure 22. Débit par rapport à la charge de trafic.....	43
Figure 23. Latence en fonction de la longueur de la route.....	44
Figure 24. Illustration de l'impact du taux de pertes sur l'efficacité du codage réseau .....	50
Figure 25. Illustration de l'impact du taux de pertes sur la fiabilité d'un code .....	51
Figure 26. Taux d'acheminement vs. taux de pertes.....	54
Figure 27. Champs supplémentaires nécessaires à l'acquittement par écho .....	56

Figure 28. Valeur de l'optimalité et taux de pertes de paquets .....	59
Figure 29. Taux d'acheminement et taux de pertes de paquets.....	59
Figure 30. Latence et taux de pertes .....	60
Figure 31. Débit et taux de pertes .....	60
Figure 32. Surcharge réseau et taux de pertes.....	61
Figure 33. Les nœuds sources transmettent leurs données vers S via M routes .....	68
Figure 34. Exemple de codage par index .....	71
Figure 35. Nombre de transmissions par rapport au nombre de nœuds sources.....	72
Figure 36. Délai de bout-en-bout par rapport au taux de pertes de messages .....	73
Figure 37. Consommation d'énergie par rapport à la profondeur du réseau. ....	74
Figure 38. Bénéfice du codage par index en fonction de la taille des données.....	75
Figure 39. Les nœuds $C_1, \dots, C_a$ envoient leur données à S via une route de $k$ sauts.....	76
Figure 40. Codage par index au niveau du nœud intermédiaire $R_k$ . ....	77
Figure 41. Comparaison des trames de données entre ZInC et ZigBee.....	79
Figure 42. Exemple d'un codage ZInC .....	80
Figure 43. Banc d'essai expérimental ZInC .....	81
Figure 44. Configuration initiale du banc d'essai .....	83
Figure 45. Nombre de transmissions par rapport au nombre de nœuds actifs.....	84
Figure 46. Latence et taux d'acheminement en fonction du nombre de nœuds actifs.....	85
Figure 47. Nombre de paquets reçus par rapport au nombre de nœuds actifs .....	85
Figure 48. Consommation d'énergie par rapport à l'intervalle CBR.....	86
Figure 49. Distribution de la consommation d'énergie par rapport à l'intervalle CBR.....	87