



**HAL**  
open science

# Approches cross-layer pour l'optimisation de la latence des communications dans les réseaux de capteurs sans fil

Lemia Louail

► **To cite this version:**

Lemia Louail. Approches cross-layer pour l'optimisation de la latence des communications dans les réseaux de capteurs sans fil. Réseaux et télécommunications [cs.NI]. Université de Franche-Comté, 2016. Français. NNT : 2016BESA2063 . tel-01661567

**HAL Id: tel-01661567**

**<https://theses.hal.science/tel-01661567>**

Submitted on 12 Dec 2017

**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.



# SPIM

## Thèse de Doctorat



UFC

école doctorale **sciences pour l'ingénieur et microtechniques**  
UNIVERSITÉ DE FRANCHE-COMTÉ

Approches cross-layer pour  
l'optimisation de la latence des  
communications dans les réseaux  
de capteurs sans fil

■ LEMIA LOUAIL



# SPIM

## Thèse de Doctorat

**UFC**

école doctorale **sciences pour l'ingénieur et microtechniques**  
**UNIVERSITÉ DE FRANCHE-COMTÉ**

N° | X | X | X |

THÈSE présentée par

**LEMIA LOUAIL**

pour obtenir le

Grade de Docteur de  
l'Université de Franche-Comté

Spécialité : **Informatique**

## Approches cross-layer pour l'optimisation de la latence des communications dans les réseaux de capteurs sans fil

Soutenue publiquement le 09 décembre 2016 devant le Jury composé de :

ZOUBIR MAMMERI	Rapporteur	Professeur à l'Université Paul Sabatier Toulouse
YE-QIONG SONG	Rapporteur	Professeur à l'Université de Lorraine Nancy
OLIVIER FLAUZAC	Examineur	Professeur à l'Université de Reims- Champagne-Ardenne Reims
LAURENT PHILIPPE	Examineur	Professeur à l'Université de Franche- Comté Besançon
VIOLETA FELEA	Co-directeur de thèse	Maître de conférences à l'Université de Franche-Comté Besançon
HERVÉ GUYENNET	Directeur de thèse	Professeur à l'Université de Franche- Comté Besançon



# REMERCIEMENTS

Je tiens à remercier en premier lieu **Mr. Hervé Guyennet**, mon directeur de thèse, de m'avoir accueillie dans son équipe, pour sa sympathie, ses conseils et ses encouragements qui m'ont permis de mener à bien cette thèse.

J'adresse mes très sincères remerciements à **Mme. Violeta Felea**. Ses conseils précieux, ses remarques pertinentes et son suivi quotidien ont été d'une importance cruciale dans l'aboutissement de ce projet, tant au niveau professionnel qu'au niveau personnel.

Je tiens aussi à remercier les professeurs **Ye-Qiong Song** et **Zoubir Mammeri** qui se sont intéressés à mon travail et qui ont accepté de l'évaluer en tant que rapporteurs. Leurs conseils et suggestions ont contribué à la clarification du manuscrit de la thèse. Je remercie également les professeurs **Olivier Flauzac** et **Laurent Phillipe** qui ont accepté d'être membres du jury de cette thèse.

Je remercie **Julien Bernard** pour les réunions et les discussions de recherche dans le domaine de réseau de capteurs.

Merci à tous les membres du **département DISC** qui m'ont offert un excellent cadre de travail ainsi qu'un séjour extrêmement agréable.

Sur un plan plus personnel, je remercie **ma mère, mon père** et mes frères **Samy** et **Anis** pour leur soutien affectif et moral durant ces trois années.

Un grand merci à mes très chers amis **Karla, Lucas, Kitsiri, Hana, Hamida, Abbas, Huda** et **Bassam**, avec qui j'ai partagé mon quotidien et qui étaient toujours présents à mes côtés.



# SOMMAIRE

<b>Introduction générale</b>	<b>1</b>
<b>I Les réseaux de capteurs sans fil et le modèle en couches</b>	<b>5</b>
<b>1 Les réseaux de capteurs sans fil</b>	<b>7</b>
1.1 Capteurs intelligents . . . . .	7
1.2 Réseaux de capteurs sans fil . . . . .	8
1.3 Domaines d'application des réseaux de capteurs sans fil . . . . .	9
1.4 Modèles de communication dans les réseaux de capteurs sans fil . . . . .	10
1.5 Caractéristiques et besoins des réseaux de capteurs sans fil . . . . .	12
1.6 Conclusion . . . . .	12
<b>2 Le modèle en couches utilisé dans les réseaux de capteurs sans fil</b>	<b>13</b>
2.1 Le modèle en couches . . . . .	13
2.2 Classification des protocoles MAC et routage . . . . .	15
2.2.1 Protocoles MAC . . . . .	15
2.2.2 Protocoles de routage . . . . .	17
2.3 Limites du modèle en couches . . . . .	20
2.4 Conclusion . . . . .	20
<b>II État de l'art sur les protocoles cross-layer</b>	<b>21</b>
<b>3 Les protocoles cross-layer</b>	<b>23</b>
3.1 Définition des protocoles cross-layer . . . . .	23
3.2 Classification des approches cross-layer . . . . .	24
3.3 Approches inter-couches Réseau et Liaison de données . . . . .	25
3.3.1 Approches de routage utilisant des informations du protocole MAC (MAC-aware routing) . . . . .	25
3.3.1.1 ORW (Opportunistic Routing for Wireless sensor networks)	25

3.3.1.2	Limites des approches de routage utilisant des informations du protocole MAC . . . . .	26
3.3.2	Approches MAC utilisant des informations du protocole de routage (routing-aware MAC) . . . . .	27
3.3.2.1	MAC-CROSS . . . . .	27
3.3.2.2	RMAC (Routing enhanced MAC protocol) . . . . .	28
3.3.2.3	CL-MAC (Cross-Layer MAC) . . . . .	29
3.3.2.4	AreaCast . . . . .	30
3.3.2.5	Approches de collection rapide de données . . . . .	31
3.3.2.6	CoLaNet . . . . .	31
3.3.2.7	Autres approches MAC utilisant des informations du routage	32
3.3.3	Limites des approches MAC utilisant des informations du routage .	32
3.4	Approches de fusion Réseau et Liaison de données . . . . .	33
3.4.1	AIMRP (Address-light Integrated MAC and Routing Protocol) . . . . .	33
3.4.2	RBF (RSSI-Based Forwarding) . . . . .	34
3.4.3	SIF (State-free Implicit Forwarding) . . . . .	35
3.4.4	Limites des approches combinant le MAC et le routage . . . . .	35
3.5	Bilan . . . . .	36
3.6	Problématique . . . . .	37
<b>III</b>	<b>Contributions</b>	<b>39</b>
<b>4</b>	<b>Modélisation et paramètres de simulation</b>	<b>43</b>
4.1	Topologie des réseaux et flux des communications . . . . .	43
4.2	Modélisation des TDMA et du routage . . . . .	44
4.2.1	Modélisation des TDMA . . . . .	44
4.2.2	Modélisation du routage . . . . .	46
4.3	Métriques évaluées . . . . .	47
4.3.1	La latence . . . . .	48
4.3.2	La latence moyenne . . . . .	51
4.3.3	La latence normalisée moyenne . . . . .	51
4.3.4	La consommation énergétique . . . . .	52
4.3.5	Le duty cycle . . . . .	52
4.4	Configuration des simulations . . . . .	53
<b>5</b>	<b>Approche de routage utilisant des informations du MAC</b>	<b>57</b>

5.1	Motivation . . . . .	57
5.2	MAR-WSN (MAC-aware Routing in Wireless Sensor Networks) . . . . .	58
5.3	Évaluation des performances de MAR-WSN . . . . .	60
5.3.1	Variation du paramètre $\alpha$ dans <i>WDD</i> . . . . .	60
5.3.2	MAR-WSN vs routage géographique . . . . .	67
5.3.3	MAR-WSN vs ORW . . . . .	69
5.4	Conclusion . . . . .	72
<b>6</b>	<b>Approche MAC utilisant des informations du routage</b>	<b>73</b>
6.1	Motivation . . . . .	74
6.2	Méthodes d'allocation de slots . . . . .	75
6.2.1	Méthode ascendante d'allocation de slots . . . . .	75
6.2.2	Méthode descendante d'allocation de slots . . . . .	76
6.2.3	La recherche circulaire dans les méthodes d'allocation de slots . . . . .	76
6.3	Approches basées sur des informations de l'arbre de routage . . . . .	78
6.3.1	I-CoLaNet (Improved CoLaNet) . . . . .	78
6.3.2	Rand-LO (Random Leaves Ordering) . . . . .	79
6.3.3	Depth-LO (Depth Leaves Ordering) . . . . .	80
6.3.4	Depth-ReLO (Depth Remaining Leaves Ordering) . . . . .	80
6.3.5	BFS . . . . .	81
6.4	Approches basées sur des informations de l'arbre de routage et du graphe du réseau . . . . .	82
6.4.1	IDeg-LO (Interference Degree Leaves Ordering) . . . . .	82
6.4.2	IDeg-ReLO (Interference Degree Remaining Leaves Ordering) . . . . .	83
6.4.3	IDeg-BFS (Interference Degree BFS) . . . . .	84
6.5	Ordonnancement Optimal . . . . .	84
6.5.1	Principe et algorithme . . . . .	85
6.5.2	Exemple . . . . .	87
6.5.3	Remarques sur la complexité . . . . .	89
6.6	Évaluation des performances des approches TDMA utilisant des informa- tions du routage . . . . .	90
6.6.1	Approches basées sur des informations de l'arbre de routage . . . . .	91
6.6.2	Heuristiques basées sur des informations de l'arbre de routage et du graphe du réseau . . . . .	93
6.6.3	Méthode ascendante d'allocation de slots vs méthode descendante d'allocation de slots . . . . .	94

6.6.4	Synthèse des simulations . . . . .	96
6.6.5	Évaluation des performances des approches par rapport à l'Ordonnancement Optimal . . . . .	99
6.7	Conclusion . . . . .	100
<b>7</b>	<b>Approche combinée routage et MAC</b>	<b>103</b>
7.1	Motivation . . . . .	103
7.2	IDeg-Routing&MAC . . . . .	104
7.3	Évaluation des performances de IDeg-Routing&MAC . . . . .	106
7.4	Longueur théorique de l'ordonnancement TDMA . . . . .	108
7.5	Performances de IDeg-Routing&MAC vs MAR-WSN, IDeg-ReLO et Depth-ReLO . . . . .	113
7.5.1	IDeg-Routing&MAC vs MAR-WSN . . . . .	114
7.5.2	IDeg-Routing&MAC vs IDeg-ReLO et Depth-ReLO . . . . .	114
7.6	Conclusion . . . . .	115
	<b>Conclusion générale</b>	<b>117</b>
	<b>Liste des publications personnelles</b>	<b>121</b>
	<b>Bibliographie</b>	<b>123</b>
	<b>Table des figures</b>	<b>129</b>
	<b>Liste des tables</b>	<b>133</b>
<b>IV</b>	<b>Annexes</b>	<b>135</b>
<b>A</b>	<b>Le test du <math>\chi^2</math> d'ajustement</b>	<b>137</b>
<b>B</b>	<b>Performances des approches TDMA utilisant des informations du routage avec l'arbre Hop Count</b>	<b>139</b>
B.1	Performances des approches TDMA utilisant des informations de l'arbre de routage . . . . .	139
B.2	Performances des approches TDMA utilisant des informations du graphe et de l'arbre de routage . . . . .	140
B.3	Synthèse des performances des approches TDMA utilisant des informations du routage . . . . .	142
B.4	Performances des approches TDMA utilisant des informations de l'arbre de routage par rapport à l'Ordonnancement Optimal . . . . .	143

# INTRODUCTION GÉNÉRALE

## CONTEXTE

Les réseaux de capteurs sans fil sont une technologie innovante qui a envahi plusieurs domaines (applications militaires, santé, environnement, ...). Cette technologie assure la connexion entre plusieurs petits dispositifs, appelés capteurs ou nœuds, équipés de capacités de traitement et de transmission de données. Les nœuds d'un réseau sont déployés dans un environnement précis afin de collecter les informations nécessaires et de coopérer pour les transmettre à une station de base (sink).

Les communications au sein d'un réseau de capteurs sans fil sont assurées grâce à un modèle en couches utilisé par tous les éléments du réseau. Dans ce modèle en couches, inspiré du modèle OSI (Open Systems Interconnection), les couches sont implémentées indépendamment les unes des autres, chaque couche étant responsable d'assurer des fonctionnalités particulières et d'optimiser certaines métriques.

Dans ce travail, nous visons des applications exploitant le déploiement d'un réseau de capteurs à des fins de surveillance environnementale. Dans ce contexte, les capteurs collectent les données de manière périodique, qui sont acheminées vers la station de base. Dans un bon nombre d'applications s'appuyant sur les réseaux de capteurs, la surveillance œuvre dans un but d'assistance réactive, grâce aux actionneurs. Selon le contexte, les capteurs et les actionneurs agissent en collaboration et exigent différents degrés de latence. La surveillance des incendies dans les forêts ou encore la surveillance des niveaux des eaux dans les rivières sont deux exemples concrets de l'utilisation des capteurs pour surveiller une zone de déploiement et qui nécessitent l'acheminement rapide des données vers la station de base afin d'éviter les dégâts. Sans traiter les problèmes de temps réel, nous abordons le problème de réduction de la latence dans les communications à plusieurs sauts entre les capteurs et la station de base. Plus particulièrement, nous nous intéressons à deux couches qui sont directement impliquées dans les décisions de communication : la couche Réseau qui sert à trouver un chemin entre un émetteur et un récepteur grâce au protocole de routage ; et la couche Liaison de données qui permet, entre autres, d'organiser l'accès au canal de communication grâce au protocole MAC (Medium Access Control).

## PROBLÉMATIQUE

Le protocole de routage et le protocole MAC sont généralement implémentés indépendamment ce qui présente de bonnes performances en termes de métriques liées à la couche Réseau ou à la couche Liaison de données. Cependant, l'indépendance de ces deux protocoles ne permet pas l'optimisation de la performance globale du réseau. Par exemple, même si le protocole MAC est conçu pour minimiser la latence des communi-

cations, le protocole de routage peut construire des routes avec une grande latence, car le critère temporaire donné par le protocole MAC n'est pas transparent pour la couche réseau. De même, les chemins choisis par le protocole de routage visant à minimiser la latence peuvent être perturbés par l'ordonnement des communications dans le voisinage direct d'un nœud, contrôlé par le protocole MAC. La définition du prochain saut pour un paquet qui suit le chemin de routage, ce dernier étant défini au niveau de la couche réseau, est une décision que nous appelons spatiale. La couche MAC permet d'arbitrer entre les nœuds voisins qui souhaitent transmettre leurs paquets, décision que nous appelons temporelle, car indépendamment du mécanisme d'accès au canal (à base de contention ou sans contention), le moment de transmission effective est décidé à ce niveau.

Les décisions spatiales prises par le protocole de routage et les décisions temporelles prises par le protocole MAC doivent être corrélées afin d'assurer une minimisation de la latence des communications et ceci en adoptant le cross-layering.

Le cross-layering permet d'exploiter la dépendance entre deux ou plusieurs protocoles de couches différentes afin d'achever un gain en performance. Cette exploitation peut être faite de deux manières : dans la première, un protocole utilise des informations d'un autre protocole afin de réaliser ses fonctionnalités ; dans la deuxième, les deux protocoles sont combinés en un seul nouveau protocole en assurant les fonctionnalités des deux protocoles confondus.

Cette thèse s'intéresse essentiellement aux approches cross-layer entre les protocoles de routage et les protocoles MAC visant à optimiser la latence des communications. Au niveau Réseau, nous considérons le routage multi-sauts qui permet de relayer les données à partir d'un nœud émetteur jusqu'au sink. Au niveau Liaison de données, nous considérons des protocoles MAC sans contention de type TDMA (Time Division Multiple Access).

## CONTRIBUTIONS ET ORGANISATION

Le but de notre travail est d'aborder les différents types de cross-layering entre routage et MAC en visant la minimisation de la latence des communications car c'est l'une des métriques la plus influencée par les décisions non corrélées provenant des algorithmes sur lesquels sont basés les protocoles de routage et les protocoles MAC.

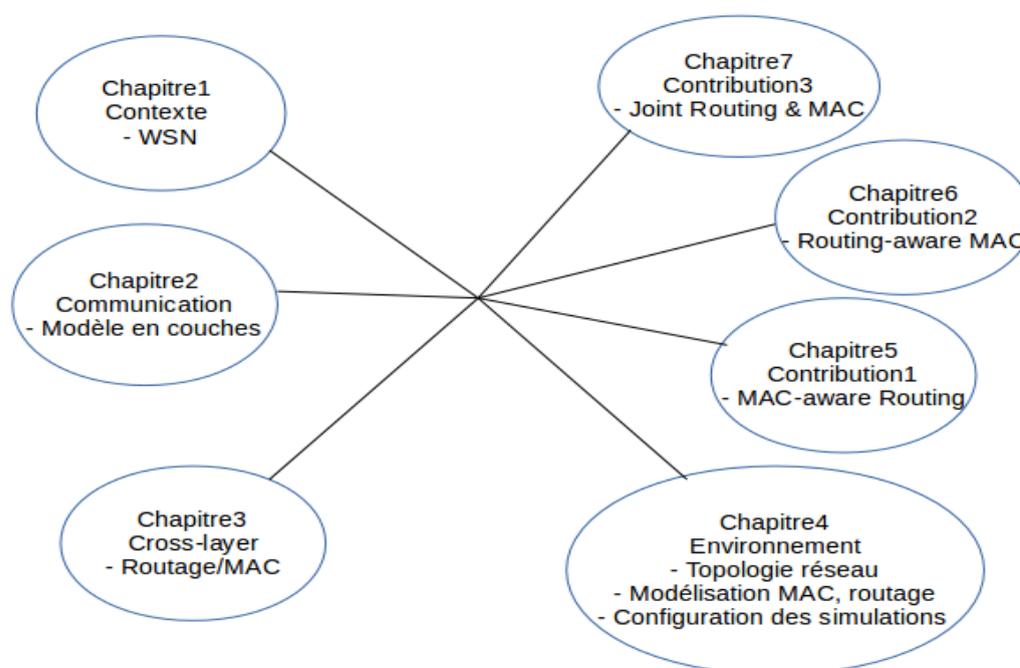
Pour arriver à ce but, nous avons commencé par proposer une approche de routage qui utilise les informations temporelles du TDMA et qui profite de la connaissance du voisinage à deux sauts. Grâce au délai de communication dans le voisinage à deux sauts issu de MAC, les routes sont construites avec l'objectif de minimiser la latence.

Ensuite, nous avons proposé deux classes d'algorithme d'ordonnement TDMA utilisant des informations du protocole de routage. Dans la première classe, les approches utilisent des informations de l'arbre de routage. Dans la deuxième classe, les approches d'ordonnement TDMA utilisent des informations de l'arbre de routage et du graphe modélisant le réseau de capteurs.

Enfin, nous avons proposé un algorithme qui construit, simultanément, un ordonnancement TDMA et l'arbre de routage. Cette approche se place dans un contexte de cross-layering de type fusion entre les protocoles des couches Liaison de données et Réseau.

La thèse est organisée comme suit :

- Le premier chapitre introduit des généralités sur les réseaux de capteurs sans fil, leurs domaines d'applications, leurs modèles de communication ainsi que les défis rencontrés au sein de ces réseaux.
- Le deuxième chapitre présente le modèle en couches assurant les communications dans les réseaux de capteurs sans fil, des classifications des protocoles MAC et des protocoles de routage sur lesquels notre travail est basé et les limites de ce modèle en couches.
- Dans le troisième chapitre nous présentons l'aspect du cross-layering et nous proposons une classification des approches cross-layer concernant les protocoles de routage et les protocoles MAC. Les approches existantes de type cross-layer concernant les protocoles de ces deux couches sont résumées et analysées.
- Le quatrième chapitre introduit les hypothèses de travail que nous considérons afin de réaliser nos contributions telles que la topologie des réseaux utilisés, la modélisation du routage et du TDMA, les métriques évaluées ainsi que les paramètres de configuration des simulations.
- Dans le cinquième chapitre est présentée la première contribution qui consiste en un algorithme de routage utilisant des informations du TDMA.
- Dans le sixième chapitre nous présentons différentes approches de construction d'un ordonnancement TDMA utilisant des informations de l'arbre de routage et minimisant la latence des communications.
- Le septième chapitre présente une approche fusionnant l'approche de routage et l'approche d'ordonnancement TDMA en un seul algorithme.
- Nous terminons avec une conclusion générale résumant nos contributions présentées dans cette thèse et introduisant des perspectives de nos travaux.







# LES RÉSEAUX DE CAPTEURS SANS FIL ET LE MODÈLE EN COUCHES



# LES RÉSEAUX DE CAPTEURS SANS FIL

## Sommaire

<b>1.1 Capteurs intelligents . . . . .</b>	<b>7</b>
<b>1.2 Réseaux de capteurs sans fil . . . . .</b>	<b>8</b>
<b>1.3 Domaines d'application des réseaux de capteurs sans fil . . . . .</b>	<b>9</b>
<b>1.4 Modèles de communication dans les réseaux de capteurs sans fil . . . . .</b>	<b>10</b>
<b>1.5 Caractéristiques et besoins des réseaux de capteurs sans fil . . . . .</b>	<b>12</b>
<b>1.6 Conclusion . . . . .</b>	<b>12</b>

Au cours de ces dernières années, la technologie des réseaux de capteurs sans fil n'a cessé de croître et plusieurs domaines ont profité de cette technologie pour répondre à leurs besoins, comme le domaine militaire, les applications environnementales, la santé ou encore la domotique.

Dans ce chapitre nous présentons ce qu'est un capteur, appelé aussi dans le domaine de l'informatique capteur intelligent ou nœud, nous définissons les réseaux de capteurs sans fil et nous donnons leurs domaines d'application.

## 1.1/ CAPTEURS INTELLIGENTS

Un capteur intelligent est un dispositif électronique de petite taille dont les fonctions principales sont la mesure des grandeurs physiques (température, humidité, ...), le traitement des données et leur transmission vers un utilisateur final qui peut être un simple ordinateur, un centre de calcul ou autre.

Comme le montre la figure 1.1, un capteur intelligent est composé essentiellement de quatre unités :

- Une unité d'acquisition qui comporte deux éléments, un capteur et un convertisseur. Le premier capte et recueille les événements et fournit des signaux analogiques, et le deuxième, le convertisseur analogique numérique (ADC), sert à transformer ces signaux analogiques en signaux numériques.
- Une unité de traitement qui reçoit les valeurs numériques de l'unité d'acquisition et les traite (calcul, stockage, ...).
- Une unité de communication qui est responsable de la communication d'un capteur

intelligent avec d'autres dispositifs. Cette communication comprend la transmission et la réception.

- La batterie qui sert à fournir de l'énergie permettant le fonctionnement des autres unités.

Notons qu'il existe des capteurs intelligents qui comportent deux unités supplémentaires : un système de localisation, permettant de localiser le capteur intelligent et/ou un mobilisateur, assurant le mouvement du capteur intelligent.

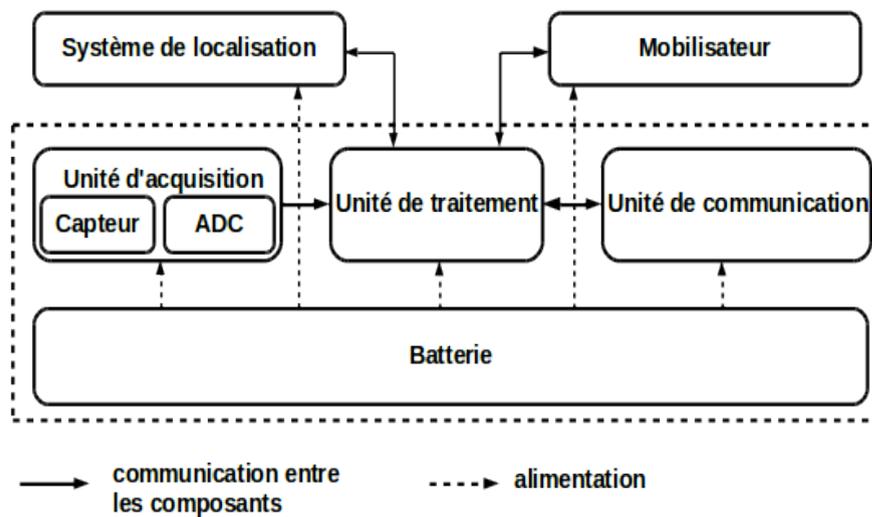


FIGURE 1.1 – Composants d'un capteur intelligent

En informatique, un capteur intelligent est appelé aussi *capteur* ou *nœud* (en anglais *sensor* ou *node/mote*), contrairement à l'électronique où le terme capteur indique seulement l'élément qui capte la grandeur physique. Dans la suite du document, les deux termes *capteur* et *nœud* feront référence à un capteur intelligent.

## 1.2/ RÉSEAUX DE CAPTEURS SANS FIL

Un réseau de capteurs sans fil ou WSN (Wireless Sensor Network en anglais), comme le montre la figure 1.2, est un réseau comportant un nombre de capteurs déployés dans une zone ou un environnement d'intérêt, communiquant entre eux par ondes radio afin de relayer l'information à une ou plusieurs stations de base généralement appelée *sink*, qui peut être un capteur ou un ordinateur. Le *sink* reçoit les données des capteurs et les transmet à l'utilisateur final via Internet ou le satellite. L'utilisateur final interroge les capteurs pour recevoir les données. Nous abordons dans ce travail les réseaux disposant d'un seul *sink*.

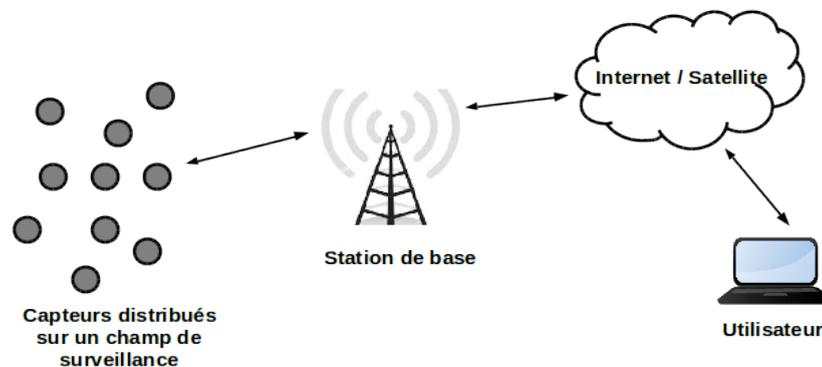


FIGURE 1.2 – Exemple d'un réseau de capteurs sans fil

### 1.3/ DOMAINES D'APPLICATION DES RÉSEAUX DE CAPTEURS SANS FIL

Les réseaux de capteurs sans fil ont envahi de nombreux domaines scientifiques et technologiques en participant à leur développement [1]. Parmi ces domaines nous citons :

#### 1. Les applications militaires

En effet, grâce à leur déploiement facile, leur tolérance aux pannes et leur auto-organisation, les réseaux de capteurs sans fil sont très utilisés dans le domaine militaire pour assurer la surveillance des champs de bataille, la reconnaissance des forces opposées et le ciblage.

#### 2. L'environnement

L'environnement est le grand bénéficiaire des applications des réseaux de capteurs sans fil, notamment pour la détection des incendies, des inondations, des fuites de produits toxiques (gaz, produits chimiques,...) ainsi que dans le contrôle de parcs naturels dans le but de suivre et contrôler les comportements de la faune et de la flore.

#### 3. La santé

Dans ce domaine, les chercheurs travaillent sur des micro-capteurs qui seront introduits par voie orale ou implantés sous la peau du patient, ce qui permet de télé-surveiller les données physiologiques humaines telles que la glycémie, les organes vitaux et même d'aider à la détection précoce de cancers. Les données obtenues seront exploitées à des fins de suivi et de surveillance des patients.

#### 4. La domotique

Les capteurs peuvent être intégrés dans les dispositifs domestiques (aspirateurs, réfrigérateurs, fours à micro-ondes, ...) afin de les gérer de façon optimale localement ou à distance.

#### 5. Les applications commerciales

C'est l'une des applications les plus fréquentes des capteurs. Des commerces, des grands magasins, des musées, des salles de cinéma et de concerts sont équipés de tels dispositifs nécessaires pour la surveillance ou pour l'échange des informations.

## 1.4/ MODÈLES DE COMMUNICATION DANS LES RÉSEAUX DE CAPTEURS SANS FIL

Au sein d'un réseau de capteurs sans fil, les nœuds communiquent entre eux en utilisant des antennes radio qui émettent et captent les signaux. Chaque nœud possède un champ de communication, appelé aussi *portée* de communication, déterminé par la puissance de son antenne radio. Cette portée étant généralement faible, un nœud passe par d'autres nœuds intermédiaires afin de relayer les informations jusqu'au sink. Cette procédure est appelée *routage multi-sauts*. Les nœuds qui se trouvent dans la portée de communication d'un capteur sont les nœuds avec lesquels il peut communiquer directement et sont considérés comme ses voisins ; en dehors de cette portée, le capteur doit utiliser le routage multi-sauts pour relayer ses informations.

Il existe plusieurs modèles de communication pour les réseaux de capteurs sans fil. Les plus utilisés, comme présenté dans [2], sont les suivants :

### 1. Le modèle de disque unitaire UDG (Unit Disk Graph) [3]

Grâce à une antenne omnidirectionnelle, l'émission d'un nœud peut être reçue par tous les nœuds dans sa proximité, qui se présente comme un disque centré sur le nœud émetteur. Le rayon de ce disque est la portée de communication. Deux nœuds sont considérés voisins si et seulement si chacun d'eux est dans la portée de l'autre. Le lien reliant deux nœuds voisins est donc bidirectionnel (le graphe modélisant le réseau est non-orienté). Pour une portée de communication  $r$  (voir la figure 1.3) le nœud A communique directement avec les nœuds B et D, par contre il ne peut communiquer avec le nœud C qu'en passant par le nœud B. En normalisant la portée de communication, nous obtenons une modélisation à base de disque unitaire UDG, où un lien existe entre deux nœuds si la distance Euclidienne qui les sépare est inférieure à 1.

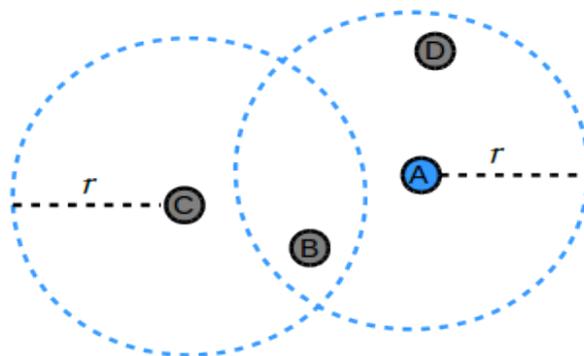


FIGURE 1.3 – Modèle de communication UDG

### 2. Le modèle de graphe général GG (General Graph) [4]

Quand le réseau est constitué de nœuds hétérogènes, de portées différentes, le graphe modélisant le réseau contient des liens asymétriques. Dans la figure 1.4, la portée du nœud A ( $r_1$ ) est supérieure à celle du nœud B ( $r_2$ ). Par conséquent, le nœud A peut communiquer avec le nœud B, mais le nœud B ne peut pas

communiquer avec le nœud A.

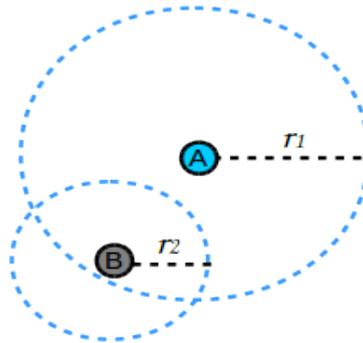


FIGURE 1.4 – Modèle de communication GG

### 3. Le modèle de disque quasi unitaire QUDG (Quasi Unit Disk Graph) [4]

Alors que certains trouvent que le modèle de disque unitaire est idéaliste, d'autres trouvent que le modèle de graphe général est très pessimiste. Le modèle de disque quasi unitaire propose une solution de compromis entre les deux solutions précédentes.

Dans ce modèle, deux nœuds sont connectés si la distance entre les deux est inférieure à  $d$ ,  $d$  étant entre 0 et 1. Si la distance est supérieure à 1, il n'y a pas de communication directe entre les deux nœuds. Entre  $d$  et 1, l'existence d'un lien de communication dépend de la propagation des ondes radio et donc n'est pas connue à l'avance. Les auteurs de [5] proposent deux méthodes pour déterminer si un lien existe entre deux nœuds séparés par une distance entre  $d$  et 1. Dans la première méthode, le lien entre ces deux nœuds existe avec une probabilité fixe. Dans la deuxième méthode, plus la distance entre les deux nœuds est proche de  $d$ , plus forte est la probabilité d'existence du lien entre ces deux nœuds.

Dans la figure 1.5, le nœud B est un voisin du nœud A parce que la distance qui les sépare est inférieure à  $d$ ; par contre le nœud C n'est pas un voisin du nœud A car la distance entre les deux est supérieure à 1. Pour les nœuds D et E, les distances qui les séparent du nœud A sont entre  $d$  et 1; dans cet exemple, le nœud E est un voisin du nœud A mais le nœud D ne l'est pas.

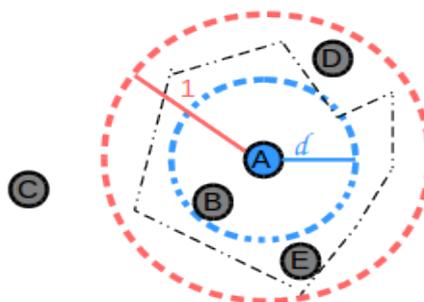


FIGURE 1.5 – Modèle de communication QUDG

## 1.5/ CARACTÉRISTIQUES ET BESOINS DES RÉSEAUX DE CAPTEURS SANS FIL

Les réseaux de capteurs se distinguent des réseaux en général par des caractéristiques intrinsèques telles que :

- La limitation des ressources  
Comme les capteurs sont des petits dispositifs électroniques, ils sont dotés de batteries limitées en quantité d'énergie, c'est pourquoi la consommation énergétique est l'une des métriques de performance des réseaux de capteurs sans fil notamment dans les applications où les capteurs sont déployés dans des zones inaccessibles. Les capteurs sont aussi limités en espace de stockage et en capacité de traitement.
- Le changement de topologie  
Les nœuds du réseau et/ou la station de base peuvent être mobiles ou peuvent ne plus fonctionner, engendrant des changements dans la topologie du réseau.

Selon l'application pour laquelle le réseau de capteurs est utilisé, des besoins et des exigences peuvent apparaître :

- Le passage à l'échelle  
Le nombre de capteurs peut varier de quelques nœuds à des centaines de nœuds. Quelque soit le nombre, les nœuds doivent pouvoir s'auto-organiser et assurer l'efficacité du réseau.
- La tolérance aux défaillances  
Comme dans les réseaux en général, dans un réseau de capteurs, un ou plusieurs nœuds peuvent ne plus fonctionner correctement et ne plus assurer leurs fonctionnalités. Dans ce cas, les autres capteurs du réseau doivent détecter ce problème et continuer à assurer l'acheminement des données vers la station de base.
- L'acheminement des paquets en temps réel  
Les applications temps réel, comme les applications d'urgences, nécessitent l'acheminement des paquets en un temps spécifié, autrement les informations deviennent obsolètes. C'est pourquoi, contrôler ou minimiser la latence des communications est un point essentiel dans les réseaux de capteurs sans fil.

## 1.6/ CONCLUSION

Les réseaux de capteurs sans fil sont une technologie innovante qui a facilité l'avancement et le progrès de plusieurs autres domaines et applications. Garantir l'acheminement des données au sein de ces réseaux dans les meilleurs délais est l'une des tâches les plus importantes.

Le modèle en couches utilisé pour assurer les communications au sein des réseaux de capteurs sans fil présente quelques limites de performance qui peuvent affecter la latence des communications.

Le prochain chapitre présente ce modèle en couches en général, avec une présentation détaillée des deux couches Réseau et Liaison de données qui sont directement impliquées dans les communications au sein de ces réseaux. La suite du document portera sur les solutions existantes ou que nous proposons afin de minimiser la latence des communications dans ces réseaux.

# LE MODÈLE EN COUCHES UTILISÉ DANS LES RÉSEAUX DE CAPTEURS SANS FIL

## Sommaire

<b>2.1 Le modèle en couches</b> . . . . .	<b>13</b>
<b>2.2 Classification des protocoles MAC et routage</b> . . . . .	<b>15</b>
<b>2.3 Limites du modèle en couches</b> . . . . .	<b>20</b>
<b>2.4 Conclusion</b> . . . . .	<b>20</b>

Pour assurer l'interconnexion des éléments d'un réseau, l'organisme ISO (International Organisation for Standardization) a proposé un découpage fonctionnel en 7 couches de l'ensemble des processus communicants d'un réseau, appelé **modèle en couches OSI** [6]. Pour les réseaux de capteurs, ce modèle OSI (Open Systems Interconnection) apparaît trop complexe et certaines couches ne sont pas utiles. Cinq couches étant suffisantes pour modéliser le découpage fonctionnel des réseaux de capteurs sans fil.

Dans ce chapitre nous présentons le modèle en couches utilisé pour assurer les communications dans les réseaux de capteurs sans fil, ainsi qu'une synthèse des approches d'accès au canal de communication proposées dans la couche MAC et des approches de routage proposées dans la couche réseau. Nous dressons un bilan soulignant les limites du modèle en couches.

## 2.1/ LE MODÈLE EN COUCHES

Dans cette modélisation (Figure 2.1), une couche correspond à un ensemble de fonctions ou de processus cohérents entre eux et assurant des fonctionnalités précises.

### 1. Couche Physique

La couche Physique est responsable de garantir, en particulier, la parfaite transmission et réception des séquences de bits de façon brute, la sélection de fréquence et la détection du signal.

### 2. Couche Liaison de données

La couche Liaison de données est responsable de la détection des trames de données et leur transfert sans erreurs, le contrôle de l'accès au canal de communication

et la détection de collisions.

Elle est composée de deux sous-couches :

- **La sous-couche de contrôle d'accès au canal (Media Access Control MAC)** : C'est la sous-couche qui détermine, pour un nœud, la possibilité et le moment pour accéder au canal de communication. Les protocoles permettant la gestion de cette sous-couche sont appelés protocoles MAC.
- **La sous-couche de contrôle de la liaison logique (Logical Link Control LLC)** : c'est la sous-couche qui contrôle le flux de données.

### 3. Couche Réseau

La couche Réseau est responsable principalement de l'acheminement et du routage des paquets dans le réseau ; i.e. elle doit trouver un chemin entre tout émetteur et destinataire.

### 4. Couche Transport

La couche Transport garantit le transfert de paquets ou d'octets bruts sans corruption, perte, réordonnancement ou duplication. Elle est responsable de gérer les communications de bout en bout.

### 5. Couche Application

La couche Application rend les couches basses transparentes à l'utilisateur en lui offrant des services permettant son interaction avec le réseau de capteurs de façon transparente (ex : allumer et éteindre les capteurs, interroger les capteurs, ...).

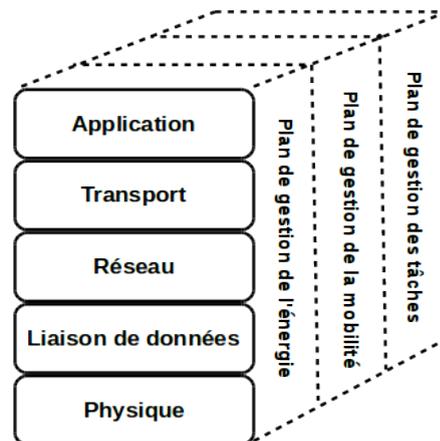


FIGURE 2.1 – Modèle en couches utilisé dans les réseaux de capteurs sans fil [1]

En plus des cinq couches (inspirées des couches du modèle OSI), trois plans de gestion sont implémentés pour assurer davantage de fonctionnalités et qui servent parfois à lier les couches :

#### 1. Plan de Gestion d'énergie

Ce plan est responsable de gérer la consommation énergétique du capteur. Par exemple, grâce à ce plan, si le niveau énergétique d'un capteur est faible, il peut informer ses voisins qu'il ne peut plus participer au routage des informations.

#### 2. Plan de Gestion de la mobilité

Dans le cas des capteurs mobiles, ce plan est responsable d'enregistrer les mouvements d'un capteur et de connaître sa localisation.

### 3. Plan de gestion des tâches

Ce plan est responsable de la gestion de l'ordonnancement des tâches que doit effectuer un capteur.

Le but de notre étude est d'assurer l'acheminement des données de n'importe quel capteur vers la station de base au plus tôt. C'est pourquoi nous nous intéressons principalement à deux couches, la couche Réseau pour trouver un chemin entre le nœud émetteur et la station de base, et la couche Liaison de données, plus précisément la sous-couche MAC, pour gérer l'accès au canal de communication.

Afin de mieux comprendre ces deux couches, nous présentons une classification des protocoles de chacune d'elles.

## 2.2/ CLASSIFICATION DES PROTOCOLES MAC ET DES PROTOCOLES DE ROUTAGE

Plusieurs classifications des protocoles MAC et des protocoles de routage existent dans la littérature. Nous présentons les deux classifications les plus répandues dans les réseaux de capteurs sans fil.

### 2.2.1/ PROTOCOLES MAC

Dans la figure 2.2 nous présentons les différents protocoles MAC présents dans [7] et [8]. Comme le montre cette figure, les protocoles MAC peuvent être classés en 2 groupes : les protocoles avec contention et les protocoles sans contention.

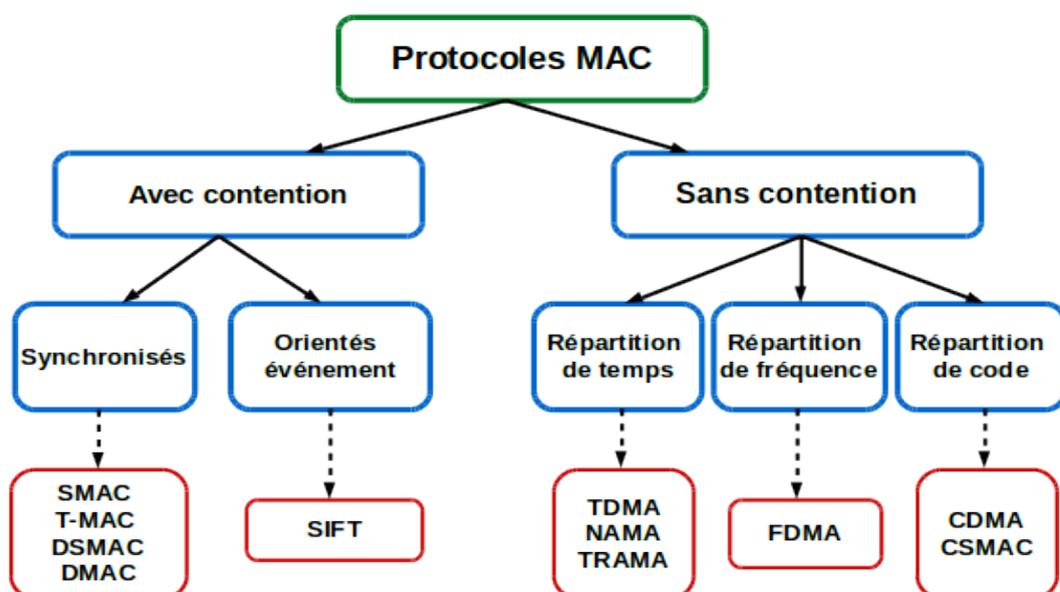


FIGURE 2.2 – Classification des protocoles MAC

### 1. Protocoles avec contention (contention-based)

Dans les protocoles avec contention il n'existe pas de slots de temps prédéterminés pour émettre ou recevoir des données. Le mécanisme de contention le plus utilisé est basé sur les paquets de contrôle RTS/CTS (Request To Send/Clear To Send). Le nœud émetteur envoie un message RTS à ses voisins, si un voisin est disponible pour recevoir les données, il renvoie un CTS afin d'informer l'émetteur de sa disponibilité.

Les protocoles MAC avec contention peuvent, eux aussi, être classés en 2 catégories :

- (a) **Protocoles synchronisés** : Les capteurs ont des périodes réveil/sommeil (wakeup/sleep) identiques, i.e ils se réveillent en même temps et dorment en même temps. Les protocoles synchronisés les plus connus sont : SMAC [9], T-MAC [10], DSMAC [11], DMAC [12], ZMAC [13] et SCP-MAC [14]. SMAC est l'un des protocoles MAC synchronisés les plus utilisés car il minimise la consommation énergétique dans le réseau de capteurs en utilisant des états de sommeil périodiques.
- (b) **Protocoles orientés événement** : Le but de ces protocoles est d'avoir des informations sur les événements surveillés avec le plus bref délai une fois qu'un événement critique se produit. SIFT [15] est le premier protocole dans cette catégorie assurant la minimisation de la latence des communications.

### 2. Protocoles sans contention (contention-free)

Dans les protocoles sans contention, il n'y a pas de concurrence pour accéder au canal de transmission. Chaque nœud a son propre temps/fréquence/code qui lui permet d'accéder au canal et aucun nœud dans son voisinage ne pourra le faire en même temps que lui. Trois catégories sont identifiées :

- (a) **Protocoles avec répartition de temps** : Chaque nœud a un intervalle de temps pendant lequel il peut accéder au canal, en dehors de ce temps l'accès au canal ne lui est pas permis. Exemples : TDMA [16], NAMA [17] et TRAMA [18].  
TDMA (Time Division Multiple Access) permet la répartition de l'intervalle de temps en un nombre de slots d'une même taille. Chaque nœud utilise un slot pour transmettre les données à ses voisins, les autres slots sont utilisés soit pour recevoir les données des voisins soit pour passer en mode sommeil ce qui permet de conserver l'énergie des nœuds.  
Plusieurs travaux proposent des variantes de TDMA dans le but de minimiser la consommation énergétique des nœuds du réseau, tels que : TDMA-ASAP [19], self-stabilizing TDMA [20] ou encore ETDMA et OTAG proposés dans [21].
- (b) **Protocoles avec répartition de fréquence** : Chaque nœud possède plusieurs bandes de fréquence et connaît la fréquence de chacun de ses voisins. Quand il veut communiquer avec un voisin, il utilise la fréquence de ce dernier pour transmettre les données. Le protocole le plus utilisé est FDMA.
- (c) **Protocoles avec répartition de code** : Chaque nœud a une seule bande de fréquence pour effectuer les transmissions. Le nœud code ses données avec son propre code et utilise la bande de fréquence pour les transmettre. De l'autre côté, le récepteur utilise le code de l'émetteur pour recevoir les données. Exemples : CDMA [22] et CSMAC [23].

CDMA permet à plusieurs utilisateurs de partager la même fréquence au même temps, ce qui engendre un gain important en temps de communication.

### 2.2.2/ PROTOCOLES DE ROUTAGE

Dans la figure 2.3 nous reprenons la classification des protocoles de routage proposée dans [24] et qui se présente comme suit.

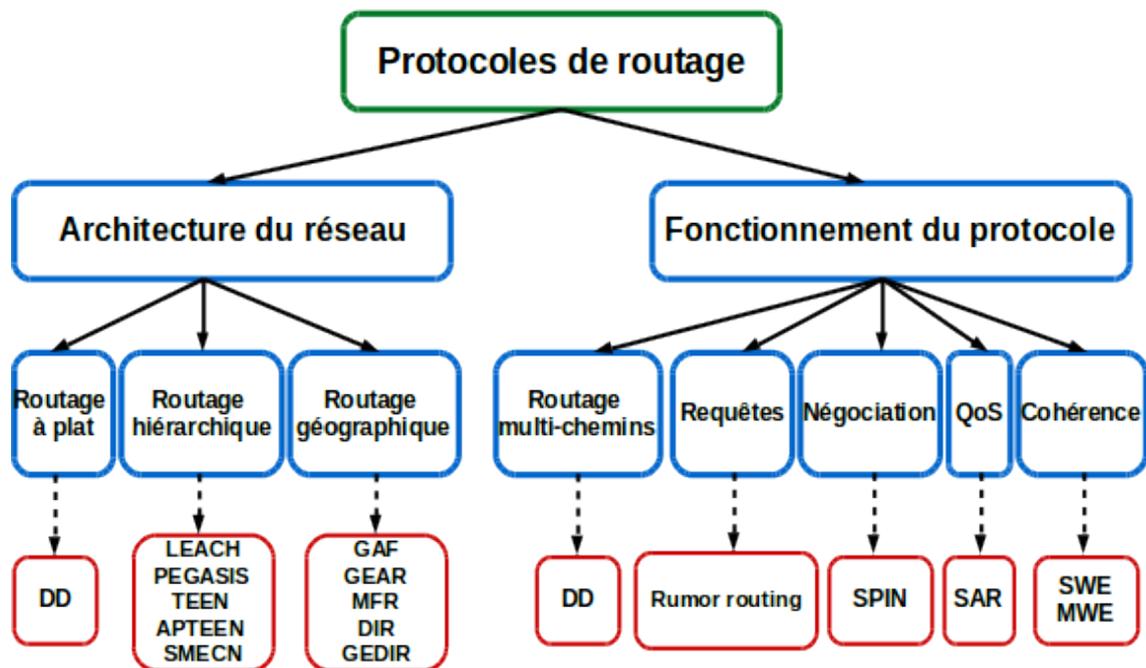


FIGURE 2.3 – Classification des protocoles de routage [24]

## 1. Protocoles basés sur l'architecture du réseau

### (a) Protocoles de routage des réseaux plats

Un réseau est dit plat lorsque tous ses éléments interconnectés ont les mêmes fonctionnalités et le réseau n'est pas divisé en parties ou en groupes. Dans la plupart des applications des réseaux de capteurs, les capteurs ont les mêmes fonctionnalités et ils collaborent pour réaliser une tâche.

Un exemple de routage dans ces réseaux est la diffusion dirigée (Directed Diffusion ou DD [25]), où un nœud combine toutes les données qu'il reçoit de différentes sources, il élimine les données redondantes et crée des gradients d'informations de son voisinage. La station de base diffuse ses intérêts dans le réseau ; pendant la diffusion, les gradients sont utilisés pour trouver les données satisfaisant les intérêts de la station de base. De cette façon, le nombre de retransmissions est réduit.

Il existe d'autres protocoles de cette catégorie comme : SPIN (Sensor Proto-

cols for Information via Navigation) [26, 27], Rumor Routing [28], MCFA (Minimum Cost Forwarding Algorithm) [29].

### (b) Protocoles de routage des réseaux hiérarchiques

Dans cette catégorie, les nœuds avec une grande énergie sont utilisés pour traiter et envoyer les informations, alors que les nœuds avec moins d'énergie sont utilisés pour la détection des événements.

C'est pourquoi le réseau est généralement structuré en clusters ; chaque cluster contient un cluster-head (un chef de groupe) et un nombre de capteurs. Les capteurs détectent les événements et les envoient au cluster-head ; ce dernier traite les données qu'il reçoit et les envoie à la station de base.

L'un des premiers protocoles de cette catégorie est LEACH (Low Energy Adaptive Clustering Hierarchy [30]). Au départ, LEACH choisit aléatoirement quelques nœuds comme cluster-head, ensuite il accorde ce rôle aux nœuds à tour de rôle pour distribuer équitablement le travail entre les capteurs.

Une valeur  $T$  représentant le souhait des nœuds de devenir cluster-head est calculée :  $T = \frac{P}{1 - P(\text{mod}(\frac{1}{P}))}$  où  $P$  est le nombre de nœuds qui veulent devenir cluster-heads. Lorsqu'un nœud  $n$  veut devenir cluster-head il choisit un nombre aléatoire  $r$  entre 0 et 1 et il compare ce nombre avec  $T$  ; si  $r$  est inférieur à  $T$  alors le nœud  $n$  devient cluster-head. Lorsqu'un nœud devient cluster-head, il diffuse un paquet dans le réseau pour informer les autres nœuds qu'il est cluster-head. Les autres nœuds choisissent à quel cluster ils appartiennent selon la puissance du signal du paquet reçu et en informent le cluster-head. Chaque cluster-head effectue un calendrier TDMA (Time Division Multiple Access) et le diffuse dans son cluster pour que les nœuds de son cluster puissent lui envoyer les données sans collisions.

D'autres protocoles comme PEGASIS [31] (Power Efficient GATHERing in Sensor Information Systems), TEEN [32], APTEEN [33] (Adaptive Periodic/Threshold sensitive Energy Efficient sensor Network) et SMECN [34] (Small Minimum Energy Communication Network) assurent aussi le routage dans les WSNs hiérarchiques.

### (c) Protocoles de routage géographique (basés sur la localisation)

Dans ce type de protocoles, les nœuds sont adressés suivant leur localisation géographique. L'approche de localisation la plus classique est le GPS [35] qui utilise 24 satellites en orbite autour de la terre. Les capteurs doivent être dotés d'un récepteur GPS qui captent les signaux des satellites et leur permettant ainsi de calculer leurs positions. Aussi, la technique des centroides à un saut [36] suppose l'existence de nœuds appelés ancres dans le réseau, ces ancres diffusant régulièrement leurs localisations. Un nœud estime son emplacement comme le centroïde des emplacements des ancres qu'il peut entendre. Un exemple de protocole de routage géographique est le protocole Greedy Routing [37] dans lequel un nœud connaît sa position géographique et celles de ses voisins. Les paquets transmis contiennent l'adresse de la destination, ce qui permet de choisir comme prochain saut le voisin le plus proche de la destination en terme de distance géographique.

D'autres protocoles existent dans cette catégorie : GPSR [38] (Greedy Perimeter Stateless Routing), GAF [39] (Geographic Adaptive Fidelity) et GEAR [40] (Geographic and Energy Aware Routing).

## 2. Protocoles basés sur le fonctionnement du protocole

**(a) Protocoles de routage multi-chemins (Multipath routing protocols)**

Dans ce type de protocoles, les nœuds maintiennent plusieurs chemins entre la source et la destination, l'acheminement des données étant réalisé par le chemin dont les nœuds ont le plus d'énergie ; ce chemin est utilisé jusqu'à ce qu'un meilleur chemin soit découvert.

La diffusion dirigée (Directed diffusion ou DD [25]) est un exemple de protocole qui utilise le concept de multi-chemins pour éviter les routes défectueuses. Chaque nœud sauvegarde plusieurs chemins qui le relient au sink et selon la qualité de réception par chaque chemin, le nœud peut choisir le meilleur.

**(b) Protocoles basés sur les requêtes (Query based Routing)**

La station de base diffuse une requête, les nœuds continuent la rediffusion de cette requête jusqu'à ce qu'un nœud ait la réponse et la renvoie alors vers le demandeur.

Le routage avec Rumeur (Rumor Routing) [28] est un protocole de routage qui permet d'acheminer des requêtes dans le réseau. Il est utilisé dans les cas où le routage géographique n'est pas applicable.

**(c) Protocoles de routages basés sur la négociation**

Ces protocoles utilisent des descriptions de données pour éliminer les transmissions redondantes.

Le protocole SPIN [26, 27] (Sensor Protocols for Information Negotiation) est un exemple de ce type. Un nœud diffuse un paquet contenant la description des données qu'il possède. Les nœuds voisins intéressés par ces données lui envoient une réponse. Enfin, le nœud à l'origine de la diffusion leur envoie les données.

Ce protocole conserve de l'énergie car les nœuds diffusent les descriptions et non pas toutes les données.

**(d) Protocoles basés sur la qualité de service (QoS)**

Dans cette catégorie, le protocole doit satisfaire certains critères qui assurent la qualité de service : délai de réponse, conservation d'énergie, utilisation de la bande passante, etc. Le protocole SAR (Sequential Assignment Routing) [41] crée des arbres pour l'acheminement des données. La racine de chaque arbre est un voisin (à un saut) de la station de base. Chaque nœud choisit ses fils dans l'arbre en évitant les nœuds avec une faible qualité de service. La qualité de service considérée dans SAR est la disponibilité des nœuds qui peut être évaluée en considérant le niveau énergétique de chaque nœud. Si un nœud appartient à plusieurs arbres, il choisit un seul arbre pour relayer ses informations.

**(e) Protocoles de routage cohérent et non cohérent**

Dans les protocoles de routage non cohérent, chaque nœud traite localement les données avant de les envoyer à d'autres nœuds.

SWE (Single Winner Election) [41] est un protocole de routage non cohérent où un seul nœud agrégateur est élu pour assurer les traitements complexes. Par contre, dans les protocoles de routage cohérent, comme MWE (Multiple Winner Election) [41], les nœuds effectuent un traitement minimal sur les données (suppression de duplications, ajout de la date, etc.) et les passent ensuite aux nœuds qui font plus de traitement, ces derniers étant appelés agrégateurs.

### 2.3/ LIMITES DU MODÈLE EN COUCHES

Bien que le modèle en couches soit très répandu dans le domaine des réseaux filaires grâce à sa modularité, le principe de séparation des couches n'est pas très efficace pour les réseaux de capteurs sans fil.

En effet, ce modèle ne garantit pas des performances optimales pour les réseaux de capteurs sans fil car ces derniers présentent certaines caractéristiques différentes des autres types de réseaux telles que les ressources limitées (énergie, mémoire, calcul) et la présence d'interférences radio. Ces caractéristiques génèrent de nouvelles contraintes lors de la conception des protocoles dans le modèle en couches pour les réseaux de capteurs sans fil qui doivent être adaptatifs.

Parmi les limites du modèle en couches, nous citons :

- Les frontières entre les couches sont strictes.
- Chaque couche doit assurer des fonctionnalités très particulières et ne peut ajouter des fonctionnalités d'autres couches.
- Combiner des protocoles de couches différentes et les utiliser ensemble ne garantit pas de meilleures performances du réseau.
- Les décisions sur le transfert de données d'un nœud vers un autre sont prises indépendamment dans les couches (d'un point de vue temporel pour la couche Liaison de données, et d'un point de vue spatial pour la couche Réseau). Ces décisions ne sont pas corrélées dans une approche en couches, ce qui ne permet pas une optimisation simultanée de la même métrique.
- Les trois plans transversaux de gestion (figure 2.1) permettent une certaine communication entre les couches ; par exemple, les protocoles de routage qui s'appuient sur l'énergie extraient cette information depuis la couche physique. Mais cette communication reste très limitée et ne permet pas de surmonter les limites de chaque couche.

### 2.4/ CONCLUSION

Ce chapitre a présenté le modèle en couches utilisé dans les réseaux de capteurs sans fil pour assurer leur communication. Plus précisément, nous avons présenté les deux couches Réseau et Liaison de données qui sont directement impliquées dans l'acheminement des données dans un réseau de capteurs sans fil. Ce modèle en couches reste très limité à cause des indépendances des couches.

La partie suivante de cette thèse présente le cross-layering, une approche proposant la possibilité d'exploiter les dépendances entre les protocoles de couches différentes afin d'améliorer les performances des réseaux de capteurs sans fil.



# ÉTAT DE L'ART SUR LES PROTOCOLES CROSS-LAYER



## LES PROTOCOLES CROSS-LAYER

### Sommaire

3.1	Définition des protocoles cross-layer . . . . .	23
3.2	Classification des approches cross-layer . . . . .	24
3.3	Approches inter-couches Réseau et Liaison de données . . . . .	25
3.4	Approches de fusion Réseau et Liaison de données . . . . .	33
3.5	Bilan . . . . .	36
3.6	Problématique . . . . .	37

Le modèle en couches, inspiré du modèle OSI, utilisé pour assurer les communications dans les réseaux de capteurs sans fil, fonctionne sur le principe de séparation des couches. Bien que ce principe soit efficace pour les réseaux filaires, il n'est pas une solution optimale pour les réseaux de capteurs sans fil où d'autres caractéristiques doivent être prises en compte, telles que les ressources limitées, les interférences radio ou encore la mobilité et les changements de topologie. Par conséquent, l'exploitation de nouvelles techniques et approches dans le modèle en couches s'est avérée nécessaire. Parmi les solutions proposées dans la littérature, nous trouvons *les protocoles cross-layer* qui permettent plus de flexibilité entre les couches.

### 3.1/ DÉFINITION DES PROTOCOLES CROSS-LAYER

En général, un protocole est dit *protocole cross-layer* si les dépendances entre ce protocole et des protocoles d'autres couches sont exploitées afin de surmonter les limites de chaque protocole et atteindre un gain en performance.

D'après [42], il existe deux types d'approches cross-layer :

1. *Approches inter-couches* Ces approches définissent des actions d'une couche qui dépendent des informations d'une autre couche. Plus précisément, nous considérons un protocole comme cross-layer s'il utilise des informations issues d'un protocole d'une autre couche. Par exemple, dans la partie gauche de la figure 3.1, nous représentons les actions entre les couches par des flèches. Un protocole dans la couche 3 utilise des informations d'un protocole dans la couche 1. De même, un protocole dans la couche 2 utilise des informations d'un protocole dans la couche 1.

Par contre, nous ne considérons pas les protocoles qui utilisent l'information de l'énergie comme protocole cross-layer ; même si l'énergie est une information de la couche Physique, elle n'est pas issue d'un protocole de cette couche. De même, nous ne considérons pas les protocoles qui utilisent des informations des couches transversales du modèle en couches comme protocoles cross-layer.

2. *Approches de fusion de couches* Ces approches combinent des informations et des fonctionnalités de protocoles de couches différentes en une seule nouvelle couche. Par exemple, dans la partie droite de la figure 3.1, les couches 1 et 2 sont combinées en une seule nouvelle couche regroupant les fonctionnalités des deux.

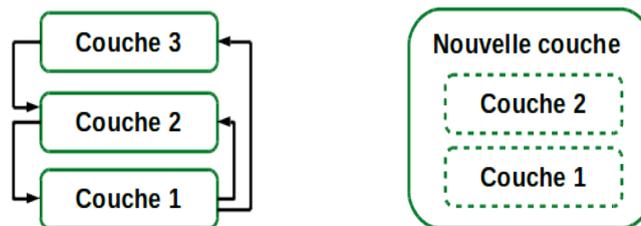


FIGURE 3.1 – Les deux types de Cross-Layering

Le cross-layering peut être réalisé entre deux ou plusieurs couches et il peut être basé sur une combinaison quelconque de couches [43] [44].

L'ITU (International Telecommunication Union) considère que le cross-layering consiste à permettre des interactions entre différentes couches (incluant les couches non adjacentes) dans le but d'atteindre des gains dans les performances du réseau [45].

Comme notre objectif principal est de minimiser la latence des communications dans les réseaux de capteurs sans fil en adoptant des approches cross-layer, nous nous intéressons aux approches cross-layer concernant la couche Réseau et la couche Liaison de données car leurs décisions ont un impact important sur la latence des communications. En partant du principe général des approches cross-layer, nous définissons deux types de protocoles cross-layer entre les couches Réseaux et Liaison de données :

- Le premier type assure des communications entre le protocole de routage et le protocole MAC en connectant la sortie d'un des deux protocoles en entrée au deuxième protocole. Par exemple, le protocole de routage fournit son résultat (la structure de routage) au protocole MAC pour qu'il l'utilise afin d'améliorer ses décisions temporelles ou vice versa.
- Le deuxième type regroupe les informations et les fonctionnalités du protocole MAC et du protocole de routage en un seul protocole qui se charge de prendre les décisions spatiales et temporelles.

### 3.2/ CLASSIFICATION DES APPROCHES CROSS-LAYER

Un nombre important d'approches cross-layer existe dans la littérature. Nous nous intéressons dans notre étude aux approches traitant la couche Liaison de données (plus

particulièrement la sous-couche MAC) et la couche Réseau. Nous les avons classées comme suit (Figure 3.2) :

1. Approches assurant des actions entre les couches Réseau et Liaison de données. Cette classe comporte deux catégories :
  - a. les protocoles de routage utilisant des informations du protocole MAC,
  - b. les protocoles MAC utilisant des informations du protocole de routage.
2. Approches combinant les couches Réseau et Liaison de données en une seule couche en utilisant un seul protocole assurant les fonctionnalités d'un protocole MAC et d'un protocole de routage en même temps.

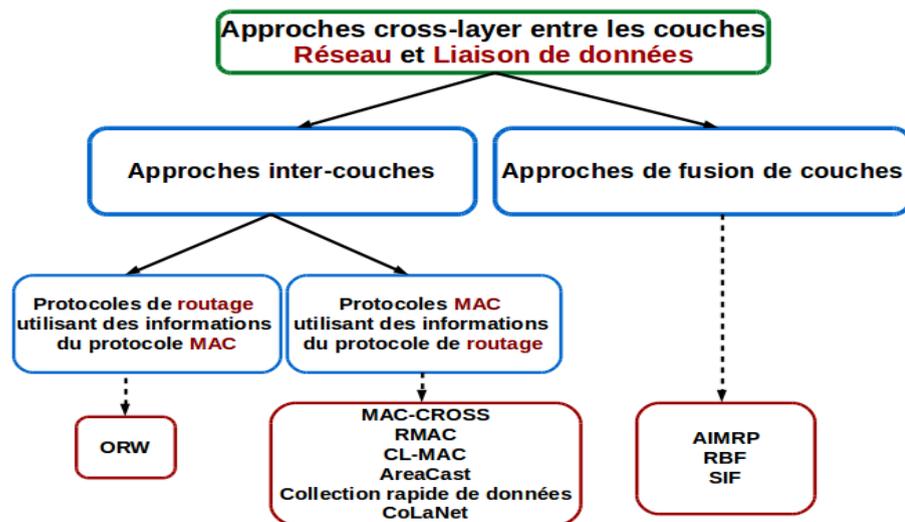


FIGURE 3.2 – Classification des approches cross-layer concernant les couches Réseau et Liaison de données

### 3.3/ APPROCHES INTER-COUCHE RÉSEAU ET LIAISON DE DONNÉES

#### 3.3.1/ APPROCHES DE ROUTAGE UTILISANT DES INFORMATIONS DU PROTOCOLE MAC (MAC-AWARE ROUTING)

Dans cette catégorie, le protocole MAC, une fois déterminé, doit envoyer des informations temporelles au protocole de routage pour que ce dernier puisse trouver le prochain saut pour chaque nœud.

Cette sous-classe est peu exploitée et le seul protocole en faisant partie, ORW [46], est présenté ci-dessous.

##### 3.3.1.1/ ORW (OPPORTUNISTIC ROUTING FOR WIRELESS SENSOR NETWORKS)

ORW [46] est un protocole de routage basé sur des informations du protocole MAC et qui vise à diminuer le délai et la consommation d'énergie en utilisant tous les voisins comme nœuds relais potentiels. Il propose un routage opportuniste qui se base sur un protocole

MAC utilisant le concept de cycles sommeil/réveil pour chaque nœud. Avec ce type de protocole MAC, les données dans ORW sont envoyées au premier voisin réveillé.

Les concepts de ORW étant génériques, le type du protocole MAC utilisé n'est pas essentiel (avec ou sans contention) ; la transmission se fait toujours vers le premier voisin qui se réveille et qui assure un rapprochement vers le sink.

Le principe de fonctionnement de ORW est le suivant :

- ORW utilise une métrique de routage appelée EDC (Expected Duty Cycle) qui reflète, pour un nœud donné  $i$ , le nombre de réveils prévus jusqu'à ce qu'un paquet arrive à destination.

$$EDC_i(S_i) = \frac{1}{\sum_{j \in S_i} p_{ij}} + \frac{\sum_{j \in S_i} p_{ij} EDC_j}{\sum_{j \in S_i} p_{ij}} + w \quad (3.1)$$

où :

$S_i$  est un sous ensemble des voisins de  $i$ ,  $p_{ij}$  est la qualité de lien entre les deux nœuds voisins  $i$  et  $j$  qui reflète la probabilité du succès de la transmission des données, et  $w$  est une constante.

Dans cette formule, le premier terme désigne le délai pour livrer un paquet à l'un des voisins dans  $S_i$ . Le deuxième terme décrit le progrès de routage offert par  $S_i$ . Le troisième terme est le coût de transmission.

La métrique EDC est calculée de manière itérative à partir du sink et jusqu'à ce qu'elle devienne stable, par des échanges de paquets dans le réseau.

- Quand un nœud a des données à transmettre au sink, il choisit comme prochain saut le premier voisin qui se réveille et dont l'EDC est inférieur au sien.
- La procédure est répétée chez chaque nœud, jusqu'à ce que les données arrivent au sink.

La métrique EDC permet de garantir la transmission des données dans la direction du sink, puisque son calcul commence au sink et se propage dans le réseau tout en s'incrémentant, d'où le choix d'un voisin avec un EDC inférieur à celui de l'émetteur.

ORW est un protocole cross-layer car, pour chaque émetteur, il choisit comme prochain saut le voisin qui se réveille le premier et cette information est extraite du protocole MAC.

### 3.3.1.2/ LIMITES DES APPROCHES DE ROUTAGE UTILISANT DES INFORMATIONS DU PROTOCOLE MAC

La version distribuée de l'implémentation de ORW nécessiterait un processus itératif pour stabiliser le calcul de la métrique EDC. Chaque nœud doit calculer son EDC, en informer ses voisins et mettre à jour son EDC lors de la réception des EDCs de son voisinage. Cette procédure devient de plus en plus difficile avec les réseaux denses et de tailles importantes.

Dans la version centralisée, le sink a besoin d'une vue globale du réseau pour effectuer le calcul de la métrique par tous les nœuds du réseau et enfin leur diffuser les résultats pour pouvoir effectuer le routage.

### 3.3.2/ APPROCHES MAC UTILISANT DES INFORMATIONS DU PROTOCOLE DE ROUTAGE (ROUTING-AWARE MAC)

Dans cette catégorie, la couche Liaison de données, et plus particulièrement le protocole MAC, attend des informations du protocole de routage afin d'établir un ordonnancement des communications pour les nœuds du réseau.

Comme pour les protocoles MAC classiques, les approches MAC cross-layer présentent deux formes : les approches basées sur la contention et les approches sans contention. Les approches MAC cross-layer basées sur la contention, trouvées dans la littérature, sont MAC-CROSS [47], RMAC [48], CL-MAC [49] et AreaCast [50]. Ces quatre approches sont distribuées. En revanche, une seule approche centralisée MAC cross-layer sans contention a été trouvée dans la littérature qui est CoLaNet [51].

#### 3.3.2.1/ MAC-CROSS

L'idée de MAC-CROSS [47] est de minimiser la consommation d'énergie en diminuant le nombre de nœuds qui doivent se réveiller pour participer à la transmission de données. Pour cela, les paquets de contrôle RTS (Request To Send) et CTS (Clear To Send) ont été légèrement modifiés en ajoutant à chacun un nouveau champ :

- RTS : le champ ajouté est "Adresse de la destination finale", grâce auquel un récepteur peut chercher un prochain saut vers cette destination dans sa table de routage.
- CTS : le champ ajouté est "Prochaine Adresse" par lequel un nœud informe son voisin qu'il doit se réveiller pour participer à la transmission de données.

#### Fonctionnement :

- Quand un nœud A capte un événement, il cherche dans sa table de routage le voisin qui lui permet d'atteindre la destination finale, par exemple le nœud B, et il lui envoie un RTS.
- Le nœud B reçoit le RTS, lit le champ "Adresse de la destination finale", cherche dans sa table de routage le prochain saut, par exemple le nœud C, et ajoute son adresse dans le CTS.
- Le nœud B diffuse le CTS.
- Quand le nœud A reçoit le CTS, il commence la transmission de données vers le nœud B.
- En même temps, le nœud C reçoit aussi le CTS et saura qu'il est le prochain saut dans la transmission de données. Il doit donc se réveiller à la fin de la transmission entre les nœuds A et B.
- Pendant la transmission de données entre les nœuds A et B, leurs voisins passent en mode sommeil, parce qu'ils ont reçu le CTS et savent qu'ils ne participent pas à la transmission de données. Seul le nœud C se réveillera.

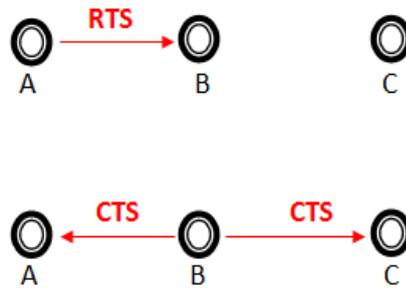


FIGURE 3.3 – Transmission des paquets RTS/CTS avec MAC-CROSS

### 3.3.2.2/ RMAC (ROUTING ENHANCED MAC PROTOCOL)

RMAC [48] est aussi un protocole MAC cross-layer avec contention. Il vise à réduire la latence de communication de bout à bout, et ce, en utilisant un seul paquet de contrôle appelé *Pion* et qui joue le rôle de RTS et CTS en même temps. En plus des champs trouvés dans un RTS classique (adresse de l'émetteur, adresse du prochain saut, durée de la transmission, ...), le Pion contient l'adresse de la destination finale et le nombre de sauts effectués.

#### Fonctionnement :

Dans RMAC, deux périodes sont utilisées, la période DATA et la période SLEEP. La première sert à trouver les nœuds qui participeront dans la transmission des données. Dans la deuxième période se fait la transmission effective des données.

#### 1. Période DATA :

- Quand un nœud A a des données à envoyer, il choisit une période aléatoire et attend que le canal de communication soit libre.
- Dès que le canal est libre, le nœud A envoie un Pion à son prochain saut (par exemple le nœud B) trouvé dans sa table de routage.
- Le nœud B vérifie s'il est la destination finale. S'il ne l'est pas, il cherche dans sa table de routage le prochain saut qui permet d'atteindre la destination finale.
- Le nœud B attend une période (SIFS) avant de transmettre son Pion :
  - Le nouveau Pion contient : l'adresse du saut précédent A, l'adresse de l'émetteur B, l'adresse du prochain saut C et l'adresse de la destination finale.
  - Le compteur de sauts est incrémenté dans ce Pion.
  - Le Pion servira aussi comme CTS pour le nœud A et comme RTS pour le nœud C.
- Le nœud A reçoit le Pion du nœud B et attend que la période SLEEP commence pour démarrer la transmission.
- Le nœud C reçoit le Pion du nœud B et refait la même procédure.

- La procédure continue jusqu'à ce qu'un Pion arrive à la destination finale, ou la période de transmission de données se termine.

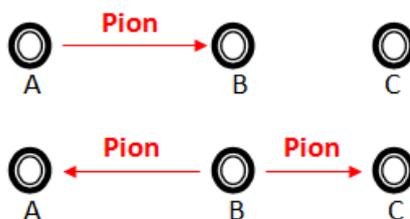


FIGURE 3.4 – Période DATA de RMAC

2. Période SLEEP : Après la fin de la période DATA qui détermine les nœuds participants dans l'acheminement des données, la période SLEEP commence et s'effectue comme suit :
  - Quand la période SLEEP commence, le nœud A envoie les données au nœud B.
  - Le nœud B renvoie un accusé de réception ACK à l'émetteur A.
  - Dès réception de l'accusé ACK, le nœud A passe à l'état de sommeil.
  - Le nœud B transmet les données au nœud C, attend l'accusé ACK puis passe à l'état sommeil.
  - La procédure continue jusqu'à ce que les données arrivent à la destination finale.

Pendant la transmission entre les nœuds A et B, les autres nœuds (comme le nœud C) peuvent s'endormir et se réveiller à la fin de cette transmission. La durée de sommeil est égale à la somme de la durée de transmission + SIFS + durée de l'ACK.

À chaque fois qu'un nœud a des données à transmettre à la station de base, les deux périodes, DATA et SLEEP, sont utilisées successivement.

### 3.3.2.3/ CL-MAC (CROSS-LAYER MAC)

CL-MAC [49] propose de gérer efficacement les trafics multi-flux et de réduire la latence des communications. La communication dans CL-MAC est faite à l'aide du paquet de contrôle FSP (Flow Setup Packet) contenant l'adresse de l'émetteur, les adresses des destinations finales et les adresses des prochains sauts trouvés dans les tables de routage. Le FSP sert comme un message RTS pour le nœud récepteur et en même temps comme un message CTS pour le nœud émetteur de données. Il a aussi la particularité de pouvoir s'adresser à plusieurs destinataires en même temps.

#### Fonctionnement :

- Un nœud A prépare le FSP<sub>i</sub> en incluant les adresses des prochains sauts (de sa table de routage) et les adresses des destinations finales, *i* étant le nombre de destinations finales.

- L'ordre dans lequel les destinations finales apparaissent dans le FSP<sub>i</sub> est important car il reflète la priorité dans l'ordonnancement.
- Selon sa position dans le FSP<sub>i</sub>, le voisin qui reçoit le FSP saura s'il participe à la communication ou non ; dans l'affirmative, il connaîtra la durée pour laquelle il peut s'endormir.
- Après la lecture du FSP<sub>i</sub>, le voisin cherche ses prochains sauts (dans sa table de routage), met à jour le FSP<sub>i</sub> et le diffuse.
- À la fin de la transmission des FSPs, chaque nœud connaît l'émetteur des données et le moment d'envoi.
- La transmission des données commence.

#### 3.3.2.4/ AREACAST

AreaCast [50] vise à améliorer la robustesse des protocoles de routage et le taux de livraison des paquets, tout en assurant une bonne consommation énergétique. Bien qu'il fonctionne de la même façon que les autres approches distribuées de sa catégorie, il corrige en plus le problème de nœuds défaillants i.e. dans le cas où le prochain saut est défaillant, un autre nœud le remplacera.

#### Fonctionnement :

- Les nœuds s'échangent des messages Hello entre eux pour connaître leurs voisins.
- Étape de préparation :
  - Un nœud émetteur diffuse un message RTS contenant l'adresse du prochain saut.
  - Les voisins de ce nœud reçoivent le message RTS et chaque voisin vérifie :
    - S'il est le prochain saut, alors il devient Relai Explicite.
    - Sinon, il applique un algorithme d'auto-élection qui permet d'élire les nœuds qui sont voisins à la fois avec l'émetteur, avec le Relai Explicite et avec le prochain saut :
      - Si les conditions sont vérifiées, il devient Relai Implicite.
      - Sinon, il s'endort le temps de la transmission.
  - Le Relai Explicite renvoie un message CTS ; s'il ne le fait pas, le premier Relai Implicite le remplace et ainsi de suite.
- Transmission de données : dès la réception du CTS, l'émetteur commence la transmission des données.
- Après réception des données, le Relai renvoie un accusé de réception ACK à l'émetteur et refait la même procédure.

### 3.3.2.5/ APPROCHES DE COLLECTION RAPIDE DE DONNÉES

Les auteurs de [52] proposent des approches de collection de données dans les réseaux de capteurs sans fil en essayant de construire des ordonnancements TDMA les plus courts possible (en terme de longueur de l'ordonnancement TDMA) à partir d'un arbre de routage.

- *BFS-TimeSlotAssignment* : cette approche permet de faire un parcours BFS sur l'arbre de routage et d'allouer un slot pour chaque nœud rencontré pendant le parcours. L'approche considère que les liens présents dans l'arbre sont les seules interférences présentes (le graphe modélisant le réseau est un arbre non orienté). Les nœuds qui ne sont pas voisins à un saut ou à deux sauts peuvent utiliser le même slot de transmission.
- *Local-TimeSlotAssignment* : dans cette deuxième approche, l'arbre est différent du réseau et tous les liens d'interférences entre les nœuds doivent être considérés. L'idée de l'approche est de considérer la racine de l'arbre et de lui allouer un slot, cette racine est ensuite supprimée et deux sous-arbres sont obtenus. Ces deux sous-arbres sont triés selon leur taille (le sous-arbre qui contient plus de nœuds est privilégié) et la même procédure se répète avec chacun des sous-arbres en le considérant comme un nouvel arbre jusqu'à ce que tous les nœuds aient des slots dans le TDMA. Les liens d'interférence présents dans le réseau sont respectés durant l'allocation des slots. De même que pour la première approche, les nœuds qui ne sont pas voisins à un saut ou à deux sauts peuvent utiliser le même slot de transmission.

### 3.3.2.6/ COLANET

Contrairement aux autres protocoles de cette catégorie, présentés précédemment, qui fonctionnent presque de la même manière (basée sur l'échange de messages RTS/CTS), CoLaNet [51], étant une approche centralisée, vise à économiser l'énergie des capteurs en proposant une technique différente. À partir du réseau, CoLaNet construit un arbre de routage appelé MinDegree ; ensuite, il applique un algorithme de coloration sur cet arbre ce qui permet d'établir un TDMA pour chaque nœud.

#### Fonctionnement :

- À partir du graphe, CoLaNet construit un arbre de routage MinDegree en choisissant au départ le sink comme racine de l'arbre, ensuite chaque nœud choisit comme père, le voisin, plus proche du sink que lui-même, qui a le moins de fils dans l'arbre.
- Un algorithme de coloration [53] est appliqué sur l'arbre MinDegree en commençant par colorer le nœud qui a le plus de voisins (fils et père) dans l'arbre de routage. Ensuite, la coloration s'applique sur les nœuds ayant un voisin déjà coloré. L'algorithme de coloration proposé dans [53] essaye de réutiliser les couleurs déjà attribuées, si ce n'est pas possible, une nouvelle couleur est ajoutée.
- À la fin de la coloration, le nombre de couleurs obtenu représente le nombre de slots du TDMA et chaque couleur représente le slot de transmission du nœud auquel la couleur est affectée.

Notons que deux autres arbres de routage ont été considérés par CoLaNet : nearest-first tree, dans lequel un nœud choisit son plus proche voisin comme père dans l'arbre, et le random tree, dans lequel un nœud choisit, aléatoirement parmi ses voisins, son père dans l'arbre. Selon l'évaluation faite par les chercheurs, l'arbre MinDegree présente de meilleures performances que les deux autres arbres en termes de temps de transmission de bout à bout et de consommation d'énergie.

### 3.3.2.7/ AUTRES APPROCHES MAC UTILISANT DES INFORMATIONS DU ROUTAGE

Il existe d'autres travaux proposant des approches MAC utilisant des informations du routage que nous n'incluons pas dans notre classification des approches cross-layer.

Par exemple, dans [54] et [55], les auteurs traitent la dérivation des exigences de la qualité de services (QoS) dans les réseaux de capteurs sans fil en se basant sur un ordonnancement TDMA. La longueur du TDMA, le délai et la bande passante sont les métriques que les auteurs ont considérées afin d'évaluer les performances de leur proposition.

L'ordonnancement TDMA est construit à partir de graphes ayant une architecture à base d'arbre clusterisé. Chaque cluster dans l'arbre comporte un nœud père qui est le cluster-head et les fils de ce nœud qui sont les membres du cluster.

La principale caractéristique de ce travail est le fait que le TDMA obtenu ne soit pas une simple trame TDMA mais plutôt une supertrame. L'approche alloue un slot dans le TDMA à chaque feuille de l'arbre et pour les cluster-heads elle leur alloue un slot pour chaque flux de données ; i.e. si un nœud est le père de deux feuilles dans l'arbre, il obtient deux slots dans la supertrame TDMA.

Cette caractéristique rend ce travail différent des autres et sa comparaison avec nos approches n'est pas faisable car la structure de l'ordonnancement TDMA n'est pas la même.

### 3.3.3/ LIMITES DES APPROCHES MAC UTILISANT DES INFORMATIONS DU ROUTAGE

Pour les approches cross-layer distribuées proposant un protocole MAC avec contention, nous avons remarqué les problèmes suivants :

- Le protocole de routage n'est pas spécifié, nous supposons que les approches fonctionnent avec tout protocole de routage.
- Les périodes de sommeil/réveil concernent les voisins à un saut, pas plus. Pour les autres voisins (à deux sauts et plus) aucune information concernant leurs périodes de sommeil n'est donnée.
- Les transmissions simultanées ne sont pas traitées.

Ces limites ne s'appliquent pas sur CoLaNet car il propose une approche centralisée, qui est un TDMA sans contention. Il fonctionne donc différemment des autres protocoles. L'approche de CoLaNet présente deux limites :

- Le nœud ayant le plus grand nombre de voisins dans l'arbre de routage MinDegree, et par lequel la procédure de coloration débute, peut avoir une position quelconque dans cet arbre.
- Continuer à colorer tous les voisins d'un nœud déjà coloré ne reflète pas la direction des communications.

### 3.4/ APPROCHES DE FUSION DES COUCHES RÉSEAU ET LIAISON DE DONNÉES (JOINT MAC & ROUTING)

Dans cette classe, les informations et les fonctionnalités des couches Réseau et Liaison de données sont regroupées et combinées en une seule couche. Un seul protocole est utilisé pour assurer le routage et l'accès au canal de communication en même temps.

#### 3.4.1/ AIMRP (ADDRESS-LIGHT INTEGRATED MAC AND ROUTING PROTOCOL)

AIMRP [56] organise les nœuds en tiers autour du sink et les données sont transmises d'un tiers à l'autre. Durant cette transmission se fait le choix du prochain saut en corrélation avec le cycle sommeil/réveil des nœuds. Ceci permet de diminuer la consommation d'énergie tout en satisfaisant la latence des communications.

#### Fonctionnement :

1. **Phase de Configuration** : Cette phase permet d'organiser les nœuds du réseau autour du sink.
  - Le sink envoie un paquet TIER1 avec un niveau de puissance.
  - Les nœuds qui reçoivent ce paquet se déclarent appartenant au TIER1.
  - Le sink envoie un deuxième paquet TIER2 avec une puissance un peu plus grande.
  - Les nœuds qui reçoivent ce paquet se déclarent appartenant au TIER2.
  - Cette procédure est répétée  $n$  fois ( $n$  étant choisi au départ).
  - À la fin de cette procédure, les nœuds sont organisés en tiers autour du sink.

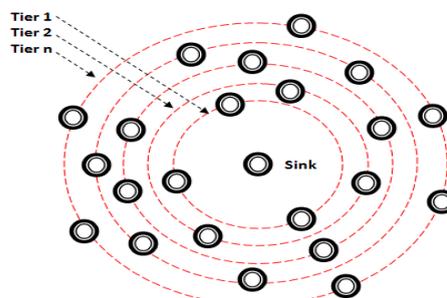


FIGURE 3.5 – Phase de configuration de AIMRP

**2. Phase Active** : Après la phase de configuration vient la phase active durant laquelle est réalisée la transmission des données.

- Un nœud est dans l'état "Listener" s'il détecte un événement ou s'il doit relayer des informations à d'autres nœuds, sinon il est en état "Sleep".
- Quand un nœud A détient des données à envoyer, il diffuse un RTR (Request to Relay) contenant son TIER\_ID.
- Les nœuds dans le voisinage du nœud A reçoivent le RTR et comparent le TIER\_ID reçu avec le leur ; si le TIER\_ID reçu est inférieur, le voisin comprend qu'il est plus loin du sink que le nœud A et donc ne répond pas au RTR et passe à l'état Sleep. Si le TIER\_ID reçu est supérieur, le voisin comprend qu'il est plus proche du sink que le nœud A et donc qu'il peut relayer les données du nœud A ; dans ce cas le voisin répond par un CTR (Clear To Relay).
- Le premier nœud qui répond au RTR par un CTR est choisi comme prochain saut ; il reçoit les données et refait la même procédure avec son propre voisinage pour que les données arrivent au sink.

### 3.4.2/ RBF (RSSI-BASED FORWARDING)

RBF [57] vise aussi à diminuer les délais d'acheminement des paquets vers le sink en utilisant la force du signal du sink pour organiser les nœuds autour du sink et pour assurer la transmission des données dans la bonne direction. RBF n'utilise aucune information sur les positions géographiques des nœuds.

#### Fonctionnement :

- Le sink diffuse un message Beacon dans le réseau.
- Chaque nœud mesure le RSSI (Received Signal Strength Indicator) qui représente la force du signal du Beacon reçu.
- Quand un nœud A a des données à envoyer, il diffuse un RTS (Request To Send) contenant son RSSI.
- Lorsqu'un voisin du nœud A reçoit le RTS, il compare le RSSI reçu avec le sien. Si le RSSI reçu est plus grand, le voisin comprend qu'il est plus loin du sink que le nœud A et donc ne participe pas à la transmission et passe à l'état Sleep. Sinon, si le RSSI reçu est plus petit, le voisin comprend qu'il est plus proche du sink ; il peut relayer les données donc il répond avec un CTS (Clear To Send).
- Parmi les voisins qui ont reçu le RTS, le premier qui renvoie le CTS est choisi comme prochain saut.
- La transmission de données se fait entre le nœud A et le prochain saut choisi ; ce dernier refait la même procédure jusqu'à ce que les données arrivent au sink.

Les auteurs de cette approche proposent aussi une extension de RBF dans [58] dans laquelle les positions géographiques des nœuds sont utilisées afin de privilégier, parmi les nœuds ayant le même RSSI, celui qui est le plus proche du sink en termes de distance géographique afin d'assurer un meilleur rapprochement vers le sink.

### 3.4.3/ SIF (STATE-FREE IMPLICIT FORWARDING)

SIF [59] est un protocole qui organise les nœuds du réseau selon les distances qui les séparent du sink, et qui utilise ces distances pour assurer l'acheminement des données dans la bonne direction tout en diminuant le délai moyen des communications.

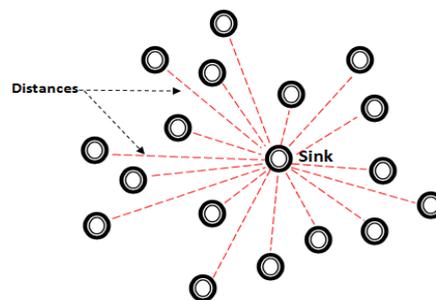


FIGURE 3.6 – Configuration de SIF

#### Fonctionnement :

- Chaque nœud du réseau connaît la distance qui le sépare du sink.
- Quand un nœud a des données à envoyer, il diffuse un BRTS (Broadcast Request To Send) qui contient, en plus des informations du RTS classique, la distance qui le sépare du sink.
- Chaque voisin reçoit le BRTS et compare la distance reçue avec la sienne. Si la distance reçue est plus petite, le voisin est plus loin du sink que l'émetteur, donc il ne participe pas à la transmission et passe à l'état Sleep. Dans le cas contraire, si la distance reçue est plus grande, alors le voisin est plus proche du sink et peut participer à la transmission des données ; il répond dans ce cas avec un CTS (Clear To Send).
- Le premier voisin qui répond avec un CTS est choisi comme prochain saut ; il reçoit les données et refait la même procédure jusqu'à ce que les données arrivent au sink.

### 3.4.4/ LIMITES DES APPROCHES COMBINANT LE MAC ET LE ROUTAGE

Les trois protocoles de cette classe se ressemblent beaucoup et utilisent le même principe en organisant les nœuds en tiers autour du sink. L'acheminement des données vers le sink se fait aussi de la même manière (de tiers en tiers), même si les noms donnés aux paquets et messages sont différents.

En revanche, aucune approche proposant de combiner un protocole MAC sans contention avec du routage n'a été trouvée dans la littérature.

## 3.5/ BILAN

		MAC		
		Sans contention	Avec contention	
routage	Routage opportuniste		ORW	MAC-aware routing
	Routage quelconque	Collection rapide de données	MAC-CROSS RMAC CL-MAC AreaCast	Routing-aware MAC
	MinDegree, nearest-first, random	CoLaNet		
	Geographic-based		AIMRP RBF SIF	Joint MAC & routing

TABLE 3.1 – Bilan des approches cross-layer

Dans le tableau 3.1, nous reprenons les différentes approches cross-layer existants dans la littérature et concernant les couches Réseau et Liaison de données en précisant le type des protocoles MAC et des protocoles de routage utilisés dans chaque approche et en indiquant le type de cross-layering.

Le routage opportuniste utilisé par ORW peut être classé avec les protocoles de routage basés sur la qualité de service, où la qualité requise est exprimée à travers la métrique EDC reflétant la probabilité du succès de la transmission.

Les deux approches routing-aware MAC sans contention, CoLaNet et Collection rapide de données, construisent des ordonnancements TDMA à partir d'un arbre de routage mais ne visent pas à minimiser la latence. La première s'intéresse à l'énergie et la deuxième vise à réduire la longueur du TDMA. Notons qu'un ordonnancement plus court n'assure pas moins de latence si les nœuds ne sont pas judicieusement placés dans le TDMA.

Dans les approches cross-layer routing-aware MAC avec contention (MAC-CROSS, RMAC, CL-MAC et AreaCast), le protocole de routage n'est pas spécifié, pour cela nous supposons que ces protocoles fonctionnent avec tout type de routage.

Nous considérons le routage utilisé dans les approches de fusion de MAC et de routage (joint MAC & routing) comme routage géographique car l'organisation du réseau en tiers et le passage d'un tiers à l'autre dans la direction du sink revient à un rapprochement géographique du sink.

### 3.6/ PROBLÉMATIQUE

Les approches cross-layer que nous avons présentées dans ce chapitre ne sont pas toutes destinées à optimiser la latence et celles qui le font présentent quelques inconvénients. Nous résumons ces limites dans la table 3.2 (page 38).

À partir de ces approches, nous avons remarqué que la latence des communications est calculée de deux façons. La première est la façon implicite dans laquelle la latence est estimée en utilisant par exemple le rapprochement du sink (comme dans AIMRP) ou encore en choisissant le premier voisin qui se réveille comme père dans l'arbre de routage (comme dans ORW). La deuxième façon est explicite et le calcul de la latence se fait par une approximation (comme dans RMAC et CL-MAC) .

À partir de la table 3.2, nous soulevons quelques problèmes existants dans les approches cross-layer :

- peu d'approches cross-layer considèrent les protocoles MAC sans contention, et en particulier le TDMA, en combinaison avec les protocoles de routage ;
- la plupart de ces approches n'optimisent pas essentiellement la latence ou la négligent complètement en visant d'autres métriques ;
- certaines approches nécessitent plus d'informations (par exemple, une vue globale du réseau pour ORW) afin d'atteindre leurs objectifs.

L'objectif du travail mené dans cette thèse est de considérer les différentes catégories d'approches cross-layer concernant les deux couches Réseau et Liaison de données décrites précédemment, et de proposer, pour chaque catégorie, de nouvelles approches cross-layer visant à minimiser la latence des communications dans les réseaux de capteurs sans fil. Nous considérons la latence des communications car elle est l'une des métriques les plus affectées par les limites du modèle en couches et par la non corrélation des décisions spatiales prises dans le protocole de routage avec les décisions temporelles prises dans le protocole MAC.

Même si notre objectif principal est d'optimiser la latence des communications, nous nous assurons que d'autres métriques, telles que la consommation énergétique ou encore le duty cycle, ne sont pas dégradées.

Nous nous intéressons aux approches MAC sans contention et plus particulièrement au TDMA car il permet d'avoir une estimation explicite du temps d'acheminement des données de bout en bout grâce au calcul précis du délai d'un paquet envoyé par un nœud et transféré par son père dans l'arbre de routage.

La partie suivante du document présente nos différents algorithmes assurant le cross-layering entre un protocole de routage et un protocole d'ordonnancement TDMA dans le but de minimiser la latence des communications.

<i>approche</i>	<i>distribuée/ centralisée</i>	<i>MAC</i>	<i>roulage</i>	<i>objectifs</i>	<i>obtention de la latence</i>	<i>limites</i>
ORW	distribuée	cycle sommeil/ réveil	roulage opportuniste	latence, énergie	1er voisin réveillé dans le cycle sommeil/réveil	processus de calcul itératif / vue globale du réseau
MAC-CROSS	distribuée	avec contention	quelconque	énergie		non orientée latence
RMAC	distribuée	avec contention	quelconque	latence	approximation	cycle sommeil/réveil seulement pour les voisins à un saut et absence des transmissions simultanées
CL-MAC	distribuée	avec contention	quelconque	latence	approximation	
AreaCast	distribuée	avec contention	quelconque	énergie, taux de livraison des paquets		
Collection rapide de données	distribuée /centralisée	sans contention	arbre	longueur de TDMA		non orientée latence
CoLaNet	centralisée	sans contention	arbre	énergie		
AIMRP	distribuée	avec contention	basé sur le roulage géographique	latence, énergie	rapprochement du sink	
RBF	distribuée	avec contention	basé sur le roulage géographique	latence	rapprochement du sink	les chemins obtenus ne sont pas forcément les plus rapides
SIF	distribuée	avec contention	basé sur le roulage géographique	latence	rapprochement du sink	

TABLE 3.2 – Table résumant les caractéristiques des approches cross-layer existantes ainsi que leurs inconvénients



## CONTRIBUTIONS



L'objectif de notre travail est de proposer des mécanismes de communications qui minimisent la latence des communications dans les réseaux de capteurs sans fil en adoptant les approches cross-layer entre les protocoles de routage et les protocoles TDMA.

D'un côté, le protocole de routage assure ses fonctionnalités grâce à des algorithmes permettant de choisir pour chaque nœud émetteur, un nœud relais qui achemine à son tour les données vers le sink. D'un autre côté, le protocole TDMA assure le réveil des nœuds durant leurs slots de transmission ou de réception grâce à des algorithmes d'ordonnement qui permettent l'allocation des slots aux nœuds du réseau.

Pour réaliser le cross-layering entre ces deux protocoles, nous proposons :

- un algorithme distribué permettant de construire un arbre de routage en utilisant des informations de l'ordonnement TDMA,
- des algorithmes centralisés permettant d'établir des ordonnements TDMA en utilisant des informations de l'arbre de routage,
- un algorithme distribué permettant de construire un arbre de routage et d'établir un ordonnancement TDMA en même temps.

Nous notons que les approches centralisées d'ordonnement TDMA que nous proposons peuvent avoir une implémentation distribuée mais cela nécessite une synchronisation entre les nœuds d'un réseau. Ces solutions distribuées (par exemple [60, 61]) consomment, généralement, beaucoup d'énergie tenant compte du nombre de messages de contrôle échangés entre les nœuds. En effet, chaque nœud doit informer ses voisins directs de son slot de communication et ces derniers doivent mettre à jour leurs ordonnements. Cette procédure est répétée jusqu'à la convergence (c'est à dire tant qu'il y a encore des messages de mise à jour qui circulent dans le réseau).

Comme notre but principal est de minimiser la latence des communications, nous évaluons les différentes approches proposées en les comparant avec des approches existantes par rapport à cette métrique. Nous définissons et évaluons d'autres métriques, telles que la consommation énergétique, pour s'assurer qu'elles ne soient pas altérées par les décisions prises par nos différentes approches.

Nos contributions ont toutes des points en communs. D'une part, l'ordonnement TDMA i.e. l'allocation des slots pour chaque nœud du réseau et la connaissance du prochain saut dans le routage, constituant l'arbre de routage. D'autre part, l'ensemble de nos propositions est évalué par simulation selon les mêmes configurations du réseau (paramètres, topologie) et les mêmes métriques. Ces éléments communs font l'objet du chapitre 4.

Les chapitres suivants présentent les différentes contributions, en commençant par une approche de routage utilisant des informations du TDMA, des approches d'ordonnement TDMA utilisant des informations de l'arbre de routage et enfin une approche permettant la construction d'un arbre de routage et d'un ordonnancement TDMA en même temps.



# MODÉLISATION ET PARAMÈTRES DE SIMULATION

## Sommaire

4.1	Topologie des réseaux et flux des communications . . . . .	43
4.2	Modélisation des TDMA et du routage . . . . .	44
4.3	Métriques évaluées . . . . .	47
4.4	Configuration des simulations . . . . .	53

Ce chapitre se constitue en une présentation introductive de nos différentes contributions. Il décrit les hypothèses essentielles sur lesquelles sont basés nos travaux avec des explications et des argumentations et il regroupe les éléments communs aux trois chapitres suivants.

## 4.1/ TOPOLOGIE DES RÉSEAUX ET FLUX DES COMMUNICATIONS

Dans ce travail, les réseaux de capteurs sont modélisés par des graphes non orientés ayant un ensemble de sommets représentant les nœuds du réseau et un ensemble d'arcs représentant les communications directes entre les capteurs. Nous ne considérons que des réseaux connexes c'est-à-dire qu'aucun nœud ne peut être isolé.

Chaque nœud est identifié par un nombre entier  $i$ ,  $1 \leq i \leq n$ , où  $n$  est la taille du réseau (le nombre total de nœuds dans le réseau) et le nœud 1 désigne le sink.

Les nœuds d'un réseau ont la même portée de communication  $r$  selon le modèle de communication UDG présenté dans la section 1.4 (page 10) et sont déployés de manière fixe dans une zone carrée de côté  $a$ . Dans notre travail, les réseaux considérés comportent un seul sink qui est toujours positionné dans le coin supérieur droit de la zone de déploiement.

La densité des réseaux est alors exprimée comme suit :  $\delta = \pi \times r^2 \times n/a^2$ . Elle représente, le nombre moyen de nœuds se trouvant dans la portée de communication d'un nœud donné, incrémenté. En fonction des besoins applicatifs ou tenant compte des contraintes de l'environnement de déploiement, les réseaux peuvent présenter des densités variées. Une grande densité est généralement proposée afin d'assurer la redondance des informations et la tolérance aux pannes. En effet, d'une part, la redondance assure une meilleure fiabilité des données à récolter, car une même caractéristique peut être me-

surée de manière différente de la part de plusieurs capteurs (mesures altérées par la présence d'obstacles ou à cause des conditions environnementales). D'autre part, les capteurs sont souvent alimentés en énergie par les batteries de durée de vie limitée, leur remplacement étant parfois fastidieux. Prévoir une densité forte assure une couverture à plus long terme de l'environnement surveillé. L'inconvénient d'une densité forte est d'une part, le taux élevé d'interférences (avec un accès médium à base de contention) et d'autre part, l'augmentation de la consommation énergétique.

Nous avons considéré différentes densités  $\delta \in [4, 20]$  (le même intervalle utilisé dans [62]), en faisant varier la taille de la zone de déploiement.

La figure 4.1 présente un graphe qui modélise un réseau comportant 8 capteurs (de 1 à 8), le nœud 1 étant le sink.

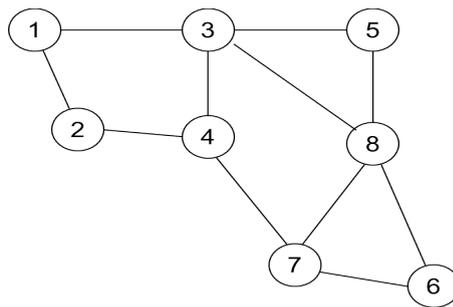


FIGURE 4.1 – Un graphe modélisant un réseau comportant 8 capteurs (dont un sink)

Dans cet exemple, les voisins directs (à un saut) du nœud 2 sont le sink 1 et le nœud 4. Les voisins à deux sauts du nœud 2 sont les nœuds 3 et 7.

Dans le reste du document, si le nombre de sauts n'est pas précisé pour un voisin, cela signifie qu'il est voisin direct (à un saut).

Il existe trois modes de communication pour les réseaux de capteurs sans fil [63] :

- le mode événementiel où les capteurs envoient les données vers le sink dès qu'un événement précis est détecté ;
- le mode à la demande où le sink envoie une requête à un ou plusieurs capteurs demandant de lui envoyer des données ;
- le mode périodique où les capteurs envoient périodiquement des informations vers le sink.

Le choix du mode de communication dépend du type de l'application dans laquelle les capteurs sont utilisés. Pour notre travail, nous optons pour le mode de communication périodique.

## 4.2/ MODÉLISATION DES TDMA ET DU ROUTAGE

### 4.2.1/ MODÉLISATION DES TDMA

Un frame (trame, ordonnancement ou schedule), appelé TDMA dans le reste du document, est un nombre de slots de temps, numérotés, d'une même durée fixe. Durant un

slot de temps dans un voisinage à un saut, un seul nœud peut émettre. Les TDMA diffèrent d'un nœud à un autre ; cependant, ils ont la même taille (appelée aussi longueur du TDMA) afin de permettre la synchronisation des communications car le slot d'envoi d'un nœud  $i$  correspond à un slot de réception par tous ses voisins.

Par conséquent, nous modélisons les TDMA dans un réseau de capteurs par une matrice d'allocation  $Schedule_{n \times l}$ . Le nombre de lignes de la matrice correspond à la taille du réseau  $n$ . Le nombre de colonnes, représentant la longueur des TDMA, est  $l$ . Cette longueur  $l$  varie d'une matrice d'allocation à une autre mais sa valeur minimale est toujours égale au degré maximal dans le graphe, incrémenté (degré maximal +1), car le nœud ayant le degré maximal dans le graphe a besoin d'un nombre de slots égal au nombre de ses voisins pour recevoir les données de ces derniers et d'un slot pour la transmission de ses données. Chaque élément de la matrice est un entier ayant un indice (de 1 à  $l$ ) et qui est défini comme suit :

- $Schedule[i, j] = i$  si le nœud  $i$  transmet ses données durant le slot  $j$ ,
- $Schedule[i, j] = m$  si le nœud  $i$  reçoit des données de son voisin  $m$  durant le slot  $j$ ,
- $Schedule[i, j] = 0$  si le nœud  $i$  n'utilise le slot  $j$  ni pour la transmission ni pour la réception.

Un exemple d'ordonnancement TDMA pour le réseau de la figure 4.1 est présenté dans la table 4.1. Le nombre de colonnes de la matrice est égal à 5 (ce qui correspond à la longueur initiale de l'ordonnancement TDMA, qui est le degré maximal dans le réseau incrémenté).

	slot1	slot2	slot3	slot4	slot5	slot6
node 1	2	1	0	0	3	0
node 2	2	1	4	0	0	0
node 3	5	1	4	0	3	8
node 4	2	7	4	0	3	0
node 5	5	0	0	0	3	8
node 6	0	7	0	6	0	8
node 7	0	7	4	6	0	8
node 8	5	7	0	6	3	8

TABLE 4.1 – Exemple d'un ordonnancement TDMA pour le réseau de la figure 4.1

Un ordonnancement TDMA est obtenu en identifiant, pour chaque nœud, son slot de transmission. Ayant connaissance de ses voisins, un nœud peut déduire ensuite les slots de réception à utiliser, un slot par voisin, grâce à la synchronisation TDMA entre voisins.

Nous définissons un algorithme générique pour construire un ordonnancement TDMA pour un réseau donné, appelé TDMA aléatoire (Random TDMA). Le terme aléatoire désigne le fait que l'allocation des slots respecte les contraintes de voisinage des nœuds mais qu'aucune autre information n'est exploitée. Le TDMA aléatoire est utile pour la première contribution où le routage est défini en fonction des slots de communication des nœuds.

La procédure pour obtenir un TDMA aléatoire est la suivante. Pour chaque nœud, sélectionné parmi les nœuds non encore ordonnancés, un slot compatible lui est affecté. L'ordre de vérification des slots compatibles est l'ordre donné par les indices des slots. Un slot est considéré *compatible* pour un nœud donné si une des deux conditions suivantes est vérifiée :

- le slot est vide i.e. le slot n'est alloué à aucun nœud,
- les nœuds utilisant le slot en transmission ne sont pas les voisins à un saut ou à deux sauts du nœud concerné par l'allocation. Cette condition permet d'éliminer les collisions causées par les voisins à un saut ou les collisions induites par les voisins à deux sauts. Ces derniers apparaissent quand les nœuds à 2 sauts, ayant un voisin commun, émettent simultanément (problème de la station cachée).

Si aucun slot du TDMA n'est compatible, un nouveau slot est ajouté à la fin du TDMA et est alloué au nœud.

Par exemple, dans la table 4.1, le nœud 4 utilise le troisième slot pour transmettre ses données, il utilise le premier slot pour recevoir les données du nœud 2, et il n'utilise le quatrième slot ni pour la transmission ni pour la réception. Les nœuds 2 et 5 dans le graphe de la figure 4.1 utilisent le même slot (le premier slot) dans la matrice d'allocation présentée dans la table 4.1, car ils ne sont voisins ni à un saut ni à deux sauts.

La maintenance des TDMA dépend de la topologie du réseau ; si un nœud tombe en panne ou quitte le réseau, une réorganisation du réseau est nécessaire avant la mise à jour des TDMA. En revanche, la facilité d'implémentation du TDMA, sa simplicité, la maîtrise du calcul de la latence et la possibilité des nœuds de faire des transmissions simultanées sans collisions, favorisent son utilisation.

#### 4.2.2/ MODÉLISATION DU ROUTAGE

Nous considérons, dans notre travail, des algorithmes de routage générant des chemins fixes à plusieurs sauts (routage multi-sauts) dans lesquels chaque nœud, sauf le sink, a un seul nœud relais, appelé aussi nœud père.

Nous modélisons les informations du routage sous forme d'arbre orienté ayant le sink comme racine car il est la destination de toutes les communications.

La figure 4.2 présente un exemple d'arbre de routage pour le réseau de la figure 4.1.

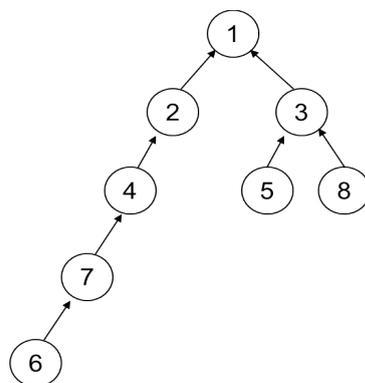


FIGURE 4.2 – Exemple d'un arbre de routage pour le réseau de la figure 4.1

Certaines de nos contributions utiliseront l'arbre de routage comme paramètre d'entrée ; dans ces cas trois arbres de routage sont utilisés :

1. l'arbre MinDegree, proposé par les auteurs de CoLaNet dans [51] et dans lequel le sink est la racine de l'arbre, ensuite chaque nœud choisit comme père, le voisin, plus proche du sink que lui-même, ayant le moins de fils dans l'arbre ;

2. l'arbre géographique où le sink est la racine de l'arbre et chaque nœud choisit comme père, son voisin le plus proche du sink en termes de distance géographique ;
3. l'arbre Hop Count où le sink est la racine de l'arbre et chaque nœud choisit comme père, son voisin le plus proche du sink en termes de nombre de sauts.

Dans les transmissions non-filaires, lorsqu'un nœud transmet des données, tous les nœuds en écoute se trouvant dans sa portée de communication (ses voisins) reçoivent ces données. Puisque l'adresse du père dans l'arbre de routage est incluse avec les données, seul le nœud concerné par la transmission continue l'acheminement des données vers le sink, les autres nœuds devant ignorer ces données reçues.

### 4.3/ MÉTRIQUES ÉVALUÉES

Nous définissons cinq métriques pour l'évaluation des différentes contributions : la latence, la latence moyenne, la latence normalisée moyenne, la consommation énergétique et le duty cycle. Notons que  $l$  représente la longueur du TDMA (nombre total de slots) et  $k_i$  le nombre de sauts du chemin de routage entre un nœud  $i$  et le sink.

Les métriques de latence font appel à un temps de référence ( $t_0$ ), le moment où un nœud souhaite envoyer un paquet de données. En considérant le mode périodique de communication (voir page 44), chaque capteur effectue trois actions liées au transfert de données : l'action de prises de mesure (capture), l'action de demande d'envoi d'un paquet de données et de l'action de communication (transmission/réception). Nous définissons, d'une part, deux types de cadence pour les données, *la cadence de prises de mesure* et *la cadence de demande de transmission de paquets de données* ; et d'autre part *une cadence pour les transmissions* d'un nœud, donnée par l'ordonnancement TDMA. Une illustration est donnée dans la figure 4.3.

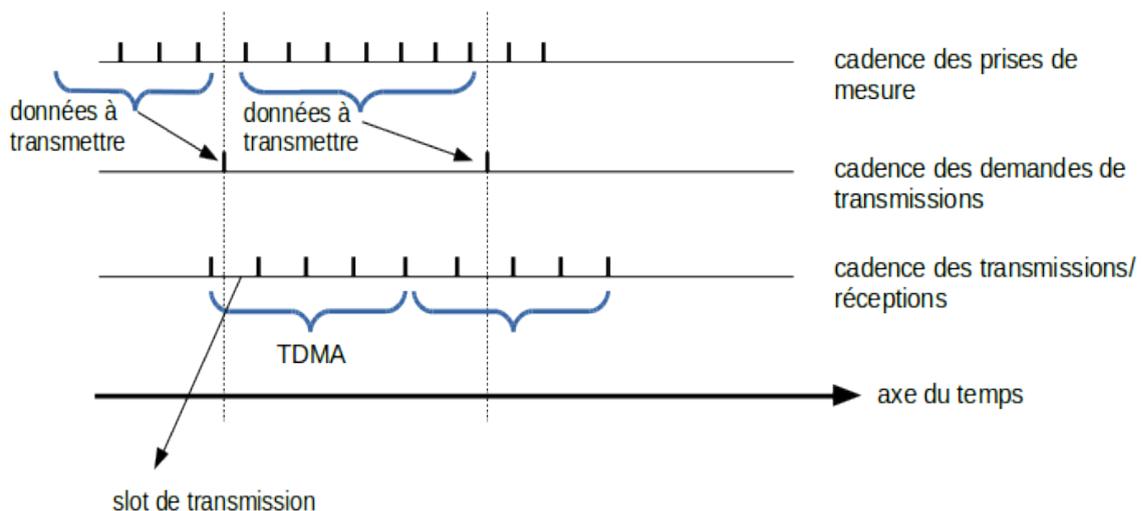


FIGURE 4.3 – Les cadences pour les données et pour les transmissions

Les différentes cadences peuvent ne pas être synchronisées. Le paquet de données à

envoyer contiendra, dans la limite de la taille du paquet, les données captées, non encore envoyées depuis la dernière transmission.

La demande de transmission de paquets peut être synchronisée avec le frame TDMA, chaque nœud réalisant une demande de transmission avant son slot d'émission. Cette cadence de transmission déphasée entre les nœuds génère des paquets de tailles inégales. Nous choisissons la même cadence des demandes de transmission pour tous les nœuds. De plus, pour faciliter le calcul de ces métriques, nous supposons que la cadence des demandes de transmissions est synchronisée avec la cadence des frames. Plus particulièrement, chaque nœud demande l'envoi d'un paquet de données en début du frame TDMA, avant le premier slot. Une analyse de ce choix est donnée dans la section suivante.

### 4.3.1/ LA LATENCE

#### Définition

La latence représente le temps nécessaire pour un nœud pour acheminer un paquet de données jusqu'au sink. Ce temps, exprimé en nombre de slots, comporte le temps d'attente nécessaire avant que le nœud émetteur puisse envoyer le paquet et le temps nécessaire pour que ce paquet arrive au sink. Comme tout nœud émetteur enclenche sa demande d'envoi du paquet de données vers le sink avant le premier slot du frame, le calcul de la latence considère ce slot comme moment de référence ( $t_0$ ).

Formellement, si un nœud  $i$  utilise le chemin  $i \rightarrow n_1 \rightarrow n_2 \rightarrow \dots \rightarrow n_{k_i-1} \rightarrow sink$ , avec  $k_i \geq 2$  le nombre de sauts dans ce chemin, sa latence est :

$$dt_i = slot_i + (slot_{n_1} - slot_i) \bmod l + (slot_{n_2} - slot_{n_1}) \bmod l + \dots + (slot_{k_i-1} - slot_{k_i-2}) \bmod l$$

sachant que  $slot_i$  représente l'indice du slot alloué en transmission au nœud  $i$  et que :

$$(x - y) \bmod l = \begin{cases} (x - y) \bmod l, & \text{si } x \geq y \\ (x - y + l) \bmod l, & \text{si } x < y \end{cases}$$

Si le sink doit diffuser des données dans le réseau alors  $dt_1 = slot_1$ , représentant le temps d'attente du sink avant qu'il puisse diffuser ses données.

#### Le temps de référence

Pour calculer la latence, il a fallu choisir un slot dans le TDMA à partir duquel nous commençons la transmission de données vers le sink. Le choix du premier slot du TDMA comme temps de référence ( $t_0$ ) semble le plus simple, mais la désignation d'un autre slot que le premier comme temps de référence aura-t-elle une incidence sur la latence ?

Nous supposons que le premier slot du TDMA est le temps de référence et nous notons  $\Delta lat_x$  le temps additionnel (en nombre de slots) si ce temps de référence est décalé vers la droite de  $x$  slots. Le temps additionnel (équation 4.1) inclut :

- le temps d'attente induit par les nœuds ayant des slots après le temps de référence (premier terme de l'équation). Chaque nœud ayant un slot après le temps de référence gagne les  $x$  premiers slots car ils ne font plus partie de son temps d'attente. Par conséquent, le gain, en temps d'attente, pour un nœud de ce type est égal à  $x$  slots.

- le temps d'attente induit par les nœuds ayant des slots avant le temps de référence (deuxième terme de l'équation). Chaque nœud ayant un slot avant le temps de référence perd en temps d'attente car en plus de son temps d'attente à partir du début du TDMA, il doit aussi ajouter le nombre de slots entre le slot  $x$  et la fin du TDMA ( $l$ ). La perte, en temps d'attente, pour un nœud de ce type est égale à  $l-x$  slots.

$$\Delta lat_x = - \sum_{i=x+1}^l m_i x + \sum_{i=1}^x m_i (l-x) \quad (4.1)$$

avec  $l$  la longueur du TDMA et  $m_i$  le nombre de nœuds affectés au slot  $i$  ( $i \in [1, l]$ ) sachant que  $\sum_{i=1}^l m_i = n$  (où  $n$  est le nombre de nœuds dans le réseau), car le nombre de nœuds dans la première situation est  $\sum_{i=x+1}^l m_i$  et le nombre de nœuds dans la deuxième situation est  $\sum_{i=1}^x m_i$ .

En simplifiant l'équation nous obtenons :

$$\Delta lat_x = l \sum_{i=1}^x m_i - xn$$

Si les nœuds sont distribués de manière uniforme sur les slots :  $m_i = \frac{n}{l}$  alors  $\Delta lat_x = 0$  i.e. aucune incidence sur la latence n'est constatée si le temps de référence est modifié.

Pour cela, nous examinons la distribution des nœuds sur les slots du TDMA aléatoire (présenté dans la section 4.2.1, page 44) dans lequel les slots sont alloués aux nœuds aléatoirement. Le but étant d'analyser la distribution des nœuds sur les slots des TDMA aléatoires ; si la distribution est uniforme, le changement de temps de référence  $t_0$  n'aura pas d'influence sur la latence calculée à partir de ces TDMA.

Pour vérifier si la distribution des nœuds dans les TDMA aléatoires est une distribution uniforme, nous utilisons un des tests de khi-deux  $\chi^2$  (chi-square test [64, 65]). Plus particulièrement, *le test du  $\chi^2$  d'ajustement* permet de comparer une distribution observée sur un échantillon à une distribution théorique (binomiale, Poisson, normale, etc.).

La loi théorique utilisée dans le test du  $\chi^2$  d'ajustement appliquée dans notre cas est la loi de distribution uniforme (voir l'annexe A, page 137 pour le rappel mathématique du test du  $\chi^2$ ).

Nous illustrons une application numérique pour un réseau de  $n = 100$  et  $\delta = 10$  et un TDMA obtenu aléatoirement par l'algorithme décrit à la page 45.

Nous définissons :

$H_0$  : la distribution observée suit la loi uniforme.

$H_1$  : la distribution observée ne suit pas la loi uniforme.

Le TDMA obtenu est d'une longueur  $l = 12$ , par conséquent, suivant la loi uniforme, les valeurs des effectifs théoriques sont toutes égales à  $n/l = 100/12 = 8.33$ . Par conséquent, la condition d'application du test  $\chi^2$  d'ajustement est remplie. La table des variables est la suivante :

variable	effectifs observés	effectifs théoriques
slot 1	9	8.33
slot 2	8	8.33
slot 3	8	8.33
slot 4	9	8.33
slot 5	12	8.33
slot 6	10	8.33
slot 7	7	8.33
slot 8	6	8.33
slot 9	6	8.33
slot 10	6	8.33
slot 11	9	8.33
slot 12	10	8.33
total	100	100

Notons que les effectifs observés représentent le nombre de nœuds utilisant chaque slot pour la transmission des données.

À partir de ces valeurs :

$$\chi_{calc}^2 = \sum_{i=1}^p \frac{(o_i - e_i)^2}{e_i} = 4.64$$

Pour un seuil de confiance  $\alpha = 0.05$  (cette valeur étant souvent utilisée en probabilité) et le degré de liberté  $\nu = l - 1 = 12 - 1 = 11$ , la valeur de  $\chi_{\alpha,\nu}^2$  (voir la table A.1) est  $\chi_{0.05,11}^2 = 19.675$ .

Puisque  $\chi_{calc}^2 < \chi_{\alpha,\nu}^2$ , cela signifie que nous conservons l'hypothèse nulle et que la distribution analysée suit une loi uniforme.

Nous avons appliqué le test du  $\chi^2$  d'ajustement sur des ordonnancements TDMA aléatoires obtenus à partir de réseaux de  $n = 100$  nœuds avec une variation de la densité  $\delta$  des réseaux. La table 4.2 présente le nombre de matrices d'allocation rejetées par le test du  $\chi^2$  d'ajustement pour une distribution uniforme.

$\delta$	nb de matrices		
	100	1000	5000
4	1	1	1
8	0	1	2
10	0	0	0
16	0	0	0

TABLE 4.2 – Nombre de TDMA rejetés par le test  $\chi^2$  d'ajustement à la loi de distribution uniforme

Quasiment la totalité des matrices d'allocation sont acceptées comme ayant des distributions uniformes. Par conséquent, le changement du temps de référence  $t_0$  n'a pas d'influence sur le calcul de la latence sur ces TDMA car le nombre moyen de nœuds dans chaque slot est  $m_i = \frac{n}{7}$  ce qui implique  $\Delta lat_x = 0$ . Ainsi, nous gardons notre choix du premier slot comme temps de référence dans nos simulations.

### 4.3.2/ LA LATENCE MOYENNE

La latence moyenne est exprimée en nombre de slots et représente la moyenne des latences des communications de chaque nœud vers le sink. Cette latence comporte le nombre de slots avant qu'un nœud envoie ses données (la référence étant le premier slot du TDMA), et le nombre de slots nécessaires durant la transmission des données, en plusieurs sauts, vers le sink.

Formellement, la latence moyenne dans un réseau de taille  $n$  est :

$$dt = \left( \sum_{i=1}^n dt_i \right) / n$$

où  $i$  représente les identifiants des nœuds.

Le choix du premier slot comme temps de référence est maintenu car la fonction de la latence moyenne a la même allure que celle de la latence. En effet, si nous notons la somme des latences, dépendant du moment de référence  $t_0$ , par  $f(t_0) = \sum_{i=1}^n dt_i$ , alors la fonction pour la latence moyenne est  $g(t_0) = f(t_0)/n$ .

### 4.3.3/ LA LATENCE NORMALISÉE MOYENNE

La latence moyenne est une mesure globale de la performance du réseau, son inconvénient étant l'effet cumulatif. Les nœuds cumulent les latences des nœuds prédécesseurs dans le chemin de routage vers le sink. Cette métrique est donc dépendante de la profondeur d'un nœud dans l'arbre de routage, ainsi que du délai d'attente entre le moment où un nœud reçoit un paquet et celui où il peut le transmettre à son père. Nous souhaitons distinguer les deux sources de délai grâce à une nouvelle métrique, la latence normalisée. En effet, soit un nœud  $n_1$  à deux sauts du sink de latence  $lat_1$  et un nœud  $n_2$  à cinq sauts du sink de latence  $lat_2$ . Si les écarts entre les slots des nœuds dans le chemin entre  $n_1$  et le sink sont petits, le nœud  $n_1$  aura une meilleure latence ( $lat_1 < lat_2$ ). En revanche, si les écarts entre les slots des nœuds dans le chemin entre  $n_2$  et le sink ne sont pas grands, le nœud  $n_2$  peut avoir une meilleure latence ( $lat_2 < lat_1$ ). À cause des délais d'acheminement par lien, un chemin le plus court en nombre de sauts n'implique pas systématiquement un chemin de meilleure latence.

La latence normalisée moyenne est exprimée en nombre de sauts et représente la moyenne des latences par lien de tous les nœuds du réseau. La latence par lien d'un nœud est la latence totale du nœud divisée par le nombre de sauts dans le chemin entre le nœud et le sink.

Formellement, la latence normalisée moyenne dans un réseau de taille  $n$  est :

$$dtn = \left( \sum_{i=1}^n dtn_i \right) / (n - 1)$$

avec  $dtn_i = dt_i/k_i, \forall i, 2 \leq i \leq n$ ,

où  $i$  est le nœud ayant l'identifiant  $i$  et en considérant que le nœud 1 est le sink,  $k_i$  est la longueur du chemin, en nombre de sauts, entre le nœud  $i$  et le sink.

Nous utiliserons cette métrique pour l'évaluation de nos approches car elle représente, en moyenne, le délai entre le slot d'émission d'un nœud et le slot d'émission de son père dans l'arbre de routage.

#### 4.3.4/ LA CONSOMMATION ÉNERGÉTIQUE

Notre objectif principal dans ce travail est de minimiser la latence des communications dans les réseaux de capteurs sans fil. La minimisation de cette métrique ne doit pas engendrer une consommation énergétique importante, sinon la durée de vie du réseau devient critique. Ainsi, nous évaluons la consommation énergétique pour s'assurer qu'elle ne soit pas altérée dans nos approches.

La consommation énergétique est exprimée en millijoules (mJ) et elle est calculée pour chaque nœud du réseau selon le modèle de consommation proposé par LEACH [30].

LEACH suppose que les nœuds consomment  $E_{elec} = 50$  nJ/bit pour démarrer le circuit émetteur ou récepteur et  $\epsilon_{amp} = 100$  pJ/bit/m<sup>2</sup> pour l'amplificateur de transmission. Il suppose aussi qu'une perte d'énergie  $r^2$  est causée par le canal de transmission.

Par conséquent, afin de transmettre un paquet de  $k$ -bits à une distance  $d$ , l'émetteur consomme :  $E_{Tx}(k, d) = E_{Tx-elec}(k) + E_{Tx-amp}(k, d)$  donc :  $E_{Tx}(k, d) = E_{elec} * k + \epsilon_{amp}$   
Et les récepteurs consomment :  $E_{Rx}(k) = E_{Rx-elec}(k)$  donc :  $E_{Rx}(k) = E_{elec} * k$

#### 4.3.5/ LE DUTY CYCLE

Le calcul de la consommation énergétique présenté précédemment dépend du chemin de routage utilisé par chaque nœud du réseau ; un émetteur consomme une quantité d'énergie lorsqu'il émet et tous les récepteurs perdent une quantité d'énergie lors de la réception. Or, cette méthode ne prend pas en compte les slots où les nœuds sont en mode sommeil (slot vide dans l'ordonnancement TDMA). Si l'arbre de routage utilisé est le même pour plusieurs approches MAC qui s'appuient sur le routage, le calcul précédent de l'énergie donne le même résultat pour toutes ces approches car les chemins de routage sont les mêmes. Le duty cycle permet d'affiner la métrique de consommation énergétique car il reflète l'état des nœuds dans le TDMA i.e. si les nœuds passent plus de temps en mode réveil ou en mode sommeil.

Le duty cycle d'un nœud représente le ratio de la période active de son TDMA (estimée en nombre de slots utilisés pour la transmission ou pour la réception) à la période totale (la longueur du TDMA).

$$DutyCycle_i = \frac{T_{awake}^i}{l} = \frac{T_{Tx}^i + T_{Rx}^i}{l}$$

où  $T_{Tx}^i$  est le nombre de slots durant lesquels le nœud  $i$  transmet ses données (toujours égal à 1 car chaque nœud a un seul slot de transmission) et  $T_{Rx}^i$  est le nombre de slots durant lesquels ce nœud écoute ses voisins. Par conséquent,

$$DutyCycle_i = \frac{1 + T_{Rx}^i}{l} = \frac{1 + |N_i|}{l}$$

où  $N_i$  désigne l'ensemble des voisins du nœud  $i$ .

Pour une topologie statique<sup>1</sup> où le nombre de voisins d'un nœud est fixe, la formule précédente montre que le duty cycle ne dépend que de la longueur du TDMA. cette dernière varie en fonction de l'approche d'allocation de slots.

Il est important d'obtenir des faibles duty cycles, car ceci permet d'économiser la consommation énergétique des nœuds du réseau [66].

1. La topologie statique n'intègre pas les cas de pannes de nœuds ou d'ajout de nouveaux nœuds.

## 4.4/ CONFIGURATION DES SIMULATIONS

L'évaluation des approches proposées est réalisée par simulation en utilisant un simulateur que nous avons développé, basé sur JUNG [67]. JUNG est un ensemble de bibliothèques JAVA fournissant un langage commun pour la modélisation, l'analyse et la visualisation de données qui peuvent être représentées sous forme de graphes ou de réseaux. JUNG est écrit en Java ce qui lui permet de profiter des vastes capacités de l'API Java ainsi que des bibliothèques tierces existantes en Java. L'architecture de JUNG est conçue pour supporter une variété de représentations d'entités et de leurs relations telles que les graphes orientés et non orientés, les graphes multi-modèles et les hypergraphes. Il fournit un simple mécanisme d'annotation des graphes et des entités et facilite aussi la création d'outils analytiques pour les données complexes.

Nous avons choisi JUNG principalement pour sa simplicité, sa modularité et sa flexibilité. Les simulateurs dédiés généralement aux réseaux de capteurs sans fil sont plus complexes et nécessitent la maîtrise et la gestion d'autres paramètres qui ne sont pas forcément nécessaires dans les simulations effectuées. JUNG nous permet de se focaliser sur les simulations des approches cross-layer entre les protocoles MAC sans contention (de type TDMA) et les protocoles de routage d'une manière simple.

Nous avons généré 5000 réseaux aléatoires comportant  $n = 100$  nœuds, la portée de communication des nœuds étant fixée à  $r = 25m$ . Le sink est toujours placé dans le coin supérieur à droite de la zone de déploiement car nous cherchons à maximiser le nombre de sauts entre le sink et les nœuds du réseau afin d'observer au mieux les performances des différentes approches évaluées. Ce placement du sink est aussi utilisé dans [46] et [68].

L'algorithme de génération des graphes est résumé ci-après :

---

**Algorithm** RandomGraph ( $n, r, a$ )

---

```

Input   :  $n$            # le nombre de noeuds
           :  $r$            # la portée des noeuds
           :  $a$            # le côté de la zone de déploiement
Output  : graph       # le graphe modélisant le réseau
Local   :  $i$            # compteur
           :  $n1, n2$      # noeuds courants
           # deployer le sink ayant l'identifiant 1 au coin superieur à droite de la zone
1: addNode(graph, 1, a, a)
2: for  $i=2$  to  $n$  do
   # deployer un nouveau noeud ayant l'identifiant  $n$  aléatoirement
3:   addNode(graph,  $i$ , random(0, a), random(0, a))
4: end for
5: for each deployed node  $n1$  do
6:   for each deployed node  $n2 \neq n1$  do
7:     if distance( $n1, n2$ )  $\leq r$  then
8:       addLink(graph,  $n1, n2$ )
9:     end if
10:  end for
11: end for

```

---

La fonction `addNode(graph,  $i$ , random(0, a))` permet de générer un nouveau nœud avec une position aléatoire (abscisse et ordonnée entre 0 et  $a$  le côté de la zone de déploie-

ment).

L'algorithme *RandomGraph* ne garantit pas que les réseaux obtenus soient connexes ; par conséquent, nous analysons les réseaux obtenus et nous rejetons ceux qui ne sont pas connexes pour ne pas les considérer dans l'évaluation des approches proposées.

La génération aléatoire de 5000 réseaux de faibles densités  $\delta \leq 8$  avec le modèle UDG n'était pas faisable en un temps limité car la plupart des réseaux obtenus - en moyenne 81% des réseaux générés - n'étaient pas connexes. Par conséquent, pour la première contribution décrite dans le chapitre suivant, avec  $6 \leq \delta \leq 8$ , seule une centaine de réseaux a été considérée pour la moyenne.

Certaines applications utilisant le déploiement des capteurs ont des besoins moindres en terme de taille moyenne du voisinage d'un nœud. Pour cette raison, nous nous sommes intéressés à la génération de réseaux de plus faible densité, tout en assurant la connexité du réseau et un temps raisonnable d'obtention.

L'idée de l'algorithme de génération de réseaux de faibles densités, appelé MAX-DPA et présenté dans [69], est de commencer par déployer le premier nœud du réseau dans un coin de la zone de déploiement ; ensuite, le nœud suivant est positionné en s'assurant qu'il a au moins un voisin (pour que le réseau soit connexe) et que le nombre moyen de voisins de chaque nœud déjà déployé ne dépasse pas  $\delta + 2$  pour une densité  $\delta$  choisie. Le déploiement des nœuds du réseau continue tout en appliquant les deux conditions précédentes. À la fin de cette procédure, le réseau généré est connexe avec une densité moyenne  $\delta$ .

---

#### Algorithm MAX-DPA ( $n, r, a$ )

---

```

Input  : n      # le nombre de noeuds
          r      # la portée des noeuds
          a      # le côté de la zone de déploiement
Output : graph  # le graphe modélisant le réseau
Local  : i      # compteur
          n1, n2 # noeuds courants
          newNode # nouveau noeud
          # deployer le sink ayant l'identifiant 1 à la position (a,a)
1: addNode(graph, 1, a, a)
2:  $\delta = \pi r^2 n / a^2$ 
3: for i=2 to n do
4:   newNode = generatePosition(random(0,a), random(0,a))
   # vérifier que le noeud aura au moins un voisin
5:   if hasNeighbor(newNode) then
6:     for each deployed node n1 do
   # le nombre de voisins de chaque noeud ne doit pas être loin de la densité
7:       if numberofNeighbor(n1)  $\leq \delta + 2$ 
8:         addNodeWithPosition(graph, i, newNode)
9:       end if
10:    end for
11:   end if
12: end for
13: for each deployed node n1 do
14:   for each deployed node n2  $\neq$  n1 do
15:     if distance(n1, n2)  $\leq r$  then
16:       addLink(graph, n1, n2)
17:     end if

```

```

18:   end for
19: end for

```

Nous notons aussi qu'avec les deux algorithmes de génération de graphes décrits ci-dessus, certains graphes, en moyenne 19.3% des graphes connexes, présentent le problème du vide (void ou holes), i.e. un nœud n'a pas de voisins plus proches du sink que lui-même, ce qui ne permet pas l'acheminement des données vers le sink avec certains algorithmes de routage comme le routage géographique. Des travaux comme [70] et [71] proposent des solutions pour ce problème en optant pour des heuristiques de découverte de chemins ou des techniques de détournement. Comme le but de notre travail n'est pas de résoudre le problème du vide, nous éliminons ces réseaux et nous ne les considérons pas dans les différentes simulations des chapitres suivants.

Pour résumer, le nombre de réseaux connexes générés avec l'algorithme *RandomGraph* est 5000 réseaux pour  $\delta \geq 11$ . Pour les densités  $8 \leq \delta \leq 10$ , en moyenne, 4500 réseaux sont connexes sur les 5000 réseaux générés. Concernant les faibles densités  $\delta \leq 7$ , l'algorithme *MAX-DPA* permet d'obtenir, en moyenne, 3750 réseaux connexes sur les 5000 réseaux générés.

La figure 4.4 présente des exemples de graphes générés avec les densités 4, 10 et 20 que nous avons utilisés dans les simulations.

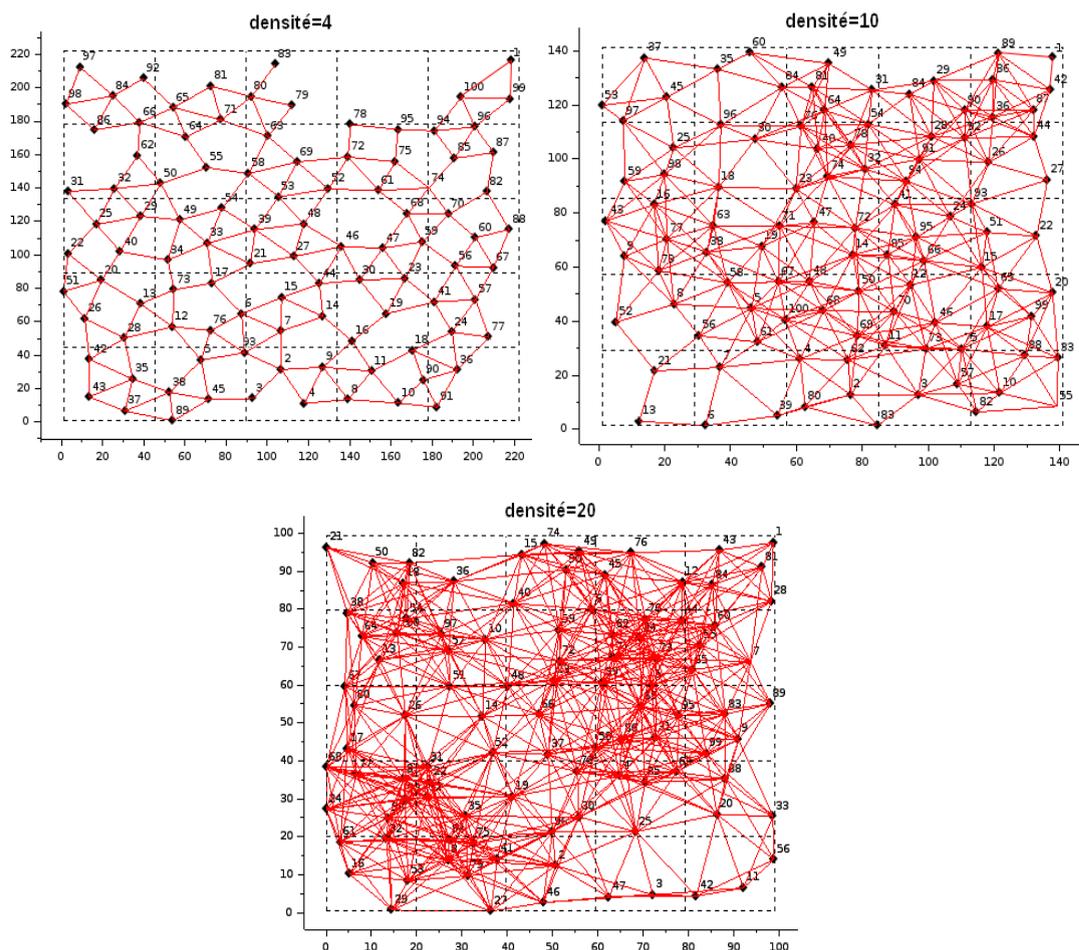


FIGURE 4.4 – Exemples de graphes générés avec les densités 4, 10 et 20

Puisque les valeurs des métriques que nous évaluons dans les chapitres suivants sont

des moyennes sur le nombre de réseaux générés, elles ne reflètent pas les données individuellement. Par conséquent, nous proposons d'analyser l'ampleur de la variabilité des valeurs par rapport à la moyenne correspondante et ce, en utilisant une des métriques statistiques de dispersion, le coefficient de variation (CV). Le coefficient de variation est exprimé en pourcentage de la moyenne :  $\frac{\sigma}{\mu}$ , où  $\sigma$  est l'écart type et  $\mu$  est la moyenne des valeurs analysées.

Plus le CV est petit, plus il y a uniformité des valeurs, car la variabilité est moindre. Il n'y a pas, cependant, un seuil unique déterminant un taux acceptable de dispersion. Ce seuil varie en fonction du domaine d'application : 5% [72] ou 10% [73] dans les techniques de sondage, 3% [74] dans le dosage de  $HbA_{1c}$  marqueur pour le diagnostic du diabète, 20% [75] dans le contexte de l'épidémiologie vétérinaire, ou encore 16.5% [76] pour la fiabilité des statistiques publiées sur la population active au Canada. Le coefficient de variation à 15% est estimé comme un seuil convenable [75], que nous prenons également comme référence dans notre analyse statistique.

## APPROCHE DE ROUTAGE UTILISANT DES INFORMATIONS DU MAC

### Sommaire

5.1 Motivation . . . . .	57
5.2 MAR-WSN (MAC-aware Routing in Wireless Sensor Networks) . . .	58
5.3 Évaluation des performances de MAR-WSN . . . . .	60
5.4 Conclusion . . . . .	72

Dans ce chapitre, nous présentons MAR-WSN [77] une approche proactive de routage qui utilise des informations temporelles provenant du TDMA pour trouver un chemin, entre le nœud qui capte les données et le sink, assurant un compromis entre la latence et la distance géographique. Par conséquent, MAR-WSN permet de choisir, pour chaque nœud, un saut suivant qui assure une progression non seulement spatiale mais aussi temporelle, pour relayer les informations au sink le plus tôt possible.

### 5.1/ MOTIVATION

La figure 5.1 présente un réseau de huit capteurs (de A à H) utilisant un ordonnancement TDMA. Pour chaque nœud, le slot en gras indique le slot de transmission et les slots en italique indiquent les slots de réception. Supposons que nous voulions envoyer des données du nœud A vers le nœud F.

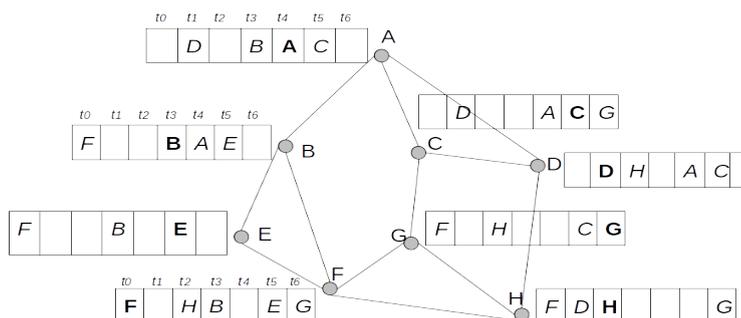


FIGURE 5.1 – Exemple d'un réseau de capteurs utilisant TDMA

Un protocole de routage par exemple permet de trouver un chemin court (A-B-F) com-

portant 3 sauts et engendrant une latence de 11 slots de temps. Ce type de routage n'est pas capable de trouver le chemin A-C-G-F comportant 4 sauts mais avec une latence inférieure, de 7 slots de temps. Ainsi, un chemin le plus court en terme de nombre de sauts ne garantit pas la meilleure latence. Ce comportement observé avec la métrique de nombre de sauts apparaît aussi avec la métrique de distance géographique.

Par conséquent, le protocole de routage doit profiter des informations temporelles venant du TDMA afin de minimiser la latence des communications.

## 5.2/ MAR-WSN (MAC-AWARE ROUTING IN WIRELESS SENSOR NETWORKS)

MAR-WSN est une approche de routage cross-layer utilisant les informations temporelles de l'ordonnancement TDMA et exploitant le voisinage à deux sauts afin d'assurer un acheminement de données adéquat spatialement et temporellement.

Un chemin plus court en terme de nombre de sauts ne garantit pas un chemin de latence minimale. D'autre part, un chemin plus rapide ne garantit pas un minimum de sauts. Or, plus le nombre de sauts est grand, plus importante est la consommation énergétique. Par conséquent, il faut trouver un compromis entre le nombre de sauts et la latence de communication, le premier critère étant une conséquence d'une décision spatiale et le deuxième, d'une décision temporelle.

MAR-WSN fonctionne avec un TDMA quelconque. Nous avons choisi un TDMA aléatoire (l'algorithme Random TDMA décrit en détail à la page 45), dans lequel, pour chaque nœud parmi les nœuds non encore ordonnancés, un slot compatible lui est affecté. L'ordre de vérification des slots est l'ordre donné par les indices des slots. Pour rappel, un slot est compatible pour un nœud s'il est vide ou si les nœuds utilisant ce slot en transmission ne sont pas les voisins à un ou à deux sauts de ce nœud.

Nous supposons que chaque nœud connaît sa position géographique, la position géographique du sink, et les positions géographiques des ses voisins à un saut et de ses voisins à deux sauts.

Le concept de MAR-WSN est le suivant :

1. Le TDMA aléatoire est déjà établi.
2. Quand un nœud  $i$  veut transmettre des données au sink  $s$ , il construit deux ensembles :
  - l'ensemble  $F_i^1 = \{j \in N_i, d_{j,s} < d_{i,s}\}$ , contenant les voisins à un saut du nœud  $i$  qui sont plus proches du sink que  $i$ ,
  - l'ensemble  $F_i^2 = \{k \in N_j, j \in F_i^1, d_{k,s} < d_{j,s}\}$ , contenant les voisins à deux sauts du nœud  $i$  qui sont plus proches du sink que les voisins à un saut dans  $F_i^1$ ,
 où :
  - $i, j$ , et  $k$  sont des nœuds,  $s$  est le sink,  $N_{n_1}$  est l'ensemble des voisins du nœud  $n_1$  et  $d_{n_1,n_2}$  est la distance entre les nœuds  $n_1$  et  $n_2$ .
3. Pour chaque nœud  $i$ , on définit une métrique  $WDD$  (Weighted Distance Delay), qui implique le nœud  $i$ , son voisin à un saut  $j$  et son voisin à deux sauts  $k$  :

$$WDD_i^{j,k} = \alpha * \frac{\Delta t_{i,j} + \Delta t_{j,k}}{2 * l} + (1 - \alpha) * \frac{d_{k,s}}{d_{i,s}} \quad (5.1)$$

où :

- $l$  est la longueur du TDMA (en nombre de slots de temps), qui est identique pour tous les nœuds ;
- $\Delta t_{n_1, n_2}$  est le nombre de slots mesurant l'écart entre le slot d'envoi du nœud  $n_1$  et le slot d'envoi du nœud  $n_2$  qui est un voisin de  $n_1$ .

La figure 5.2 présente un exemple d'un nœud  $i$ , son voisin à un saut  $j$  et son voisin à deux sauts  $k$  avec des TDMA particuliers. Dans cet exemple,  $\Delta t_{i,j} = 2$  et  $\Delta t_{j,k} = 2$ .

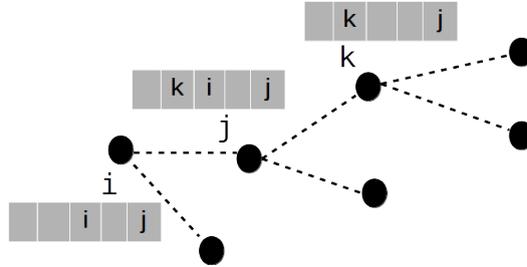


FIGURE 5.2 – Exemple d'un TDMA d'un nœud avec un voisin à un saut et un voisin à deux sauts

-  $\alpha$  est un coefficient réel  $\in ]0, 1]$  qui permet l'agrégation de deux critères à optimiser simultanément. Quand  $\alpha$  est grand, plus d'importance est donnée au délai ; quand  $\alpha$  est petit, plus d'importance est donnée à la distance.

Le premier terme  $\frac{\Delta t_{i,j} + \Delta t_{j,k}}{2 * l}$  reflète la latence entre le nœud  $i$  et son voisin à deux sauts  $k$  et le deuxième terme  $\frac{d_{k,s}}{d_{i,s}}$  reflète la latence entre le voisin à deux sauts  $k$  et le sink  $s$ .

Notons que lorsque le nœud  $j$  est le sink, la métrique  $WDD$  n'est pas utilisée car  $i$  envoie directement les données au sink. En revanche, lorsque le nœud  $k$  est le sink, le deuxième terme de  $WDD$  devient zéro car  $d_{k,s} = 0$  et  $\Delta t_{j,k}$  devient aussi zéro parce que le nombre de slots séparant le slot de  $j$  du slot du sink n'est pas comptabilisé puisque le sink reçoit les données lorsque  $j$  utilise son slot de transmission.

L'intérêt de la métrique  $WDD$  est d'associer deux critères contradictoires (latence et distance) à une même formule visant l'optimisation des deux critères par leur agrégation.

4. La métrique  $WDD$  est calculée pour chaque triplet  $\{i, j, k\}$ . Ensuite, le nœud  $i$  choisit comme père  $frwd_i$ , le nœud  $j$  de l'ensemble  $F_i^1$  ayant un voisin  $k$  dans l'ensemble  $F_i^2$  pour lesquels la métrique  $WDD_i^{j,k}$  a la valeur minimale :
 
$$frwd_i = \{j \mid j \in F_i^1, k \in F_i^2, \min_{j,k} WDD_i^{j,k}\}.$$
5. Chaque nœud choisi comme père répète la procédure à partir de l'étape 2 jusqu'à ce que les données arrivent au sink.

Les deux ensembles de voisinage  $F_i^1$  et  $F_i^2$  permettent d'assurer l'acheminement des données dans la direction du sink.

La métrique  $WDD$  permet de pondérer les deux métriques latence et distance et par conséquent elle assure un compromis entre la progression temporelle (délai) et la progression spatiale (distance) pour chaque nœud afin d'acheminer les données vers le sink.

### 5.3/ ÉVALUATION DES PERFORMANCES DE MAR-WSN

Nous procédons à l'évaluation des performances de MAR-WSN en les comparant avec celles de ORW [46], approche présentée dans la section 3.3.1.1. Pour rappel, ORW est aussi une approche de routage cross-layer utilisant des informations de l'ordonnancement TDMA. Nous comparons aussi notre approche avec un routage géographique associé au même TDMA aléatoire mais qui n'utilise aucune information de ce dernier ; il sert comme référence d'une approche en couches.

L'environnement de simulation est celui décrit dans la section 4.4 (page 53). Nous avons considéré un seul nœud émetteur qui est placé dans le coin opposé à celui du sink pour qu'il soit le plus loin possible de ce dernier.

Les métriques que nous évaluons dans ce chapitre sont la latence (en nombre de slots), la consommation énergétique (en mJ) et le nombre de sauts dont les définitions sont détaillées dans la section 4.3 (page 47).

#### 5.3.1/ VARIATION DU PARAMÈTRE $\alpha$ DANS $WDD$

Le paramètre  $\alpha$  est essentiel dans la métrique  $WDD$  (équation 5.1), puisqu'il permet de balancer entre la distance et la latence pour le choix du prochain saut de chaque nœud du réseau. Ainsi, sa valeur doit être bien choisie afin de permettre à MAR-WSN d'offrir les meilleures performances.

Dans un premier temps, nous fixons la densité des réseaux utilisés pour les expérimentations concernant le paramètre  $\alpha$  à  $\delta = 10$ .

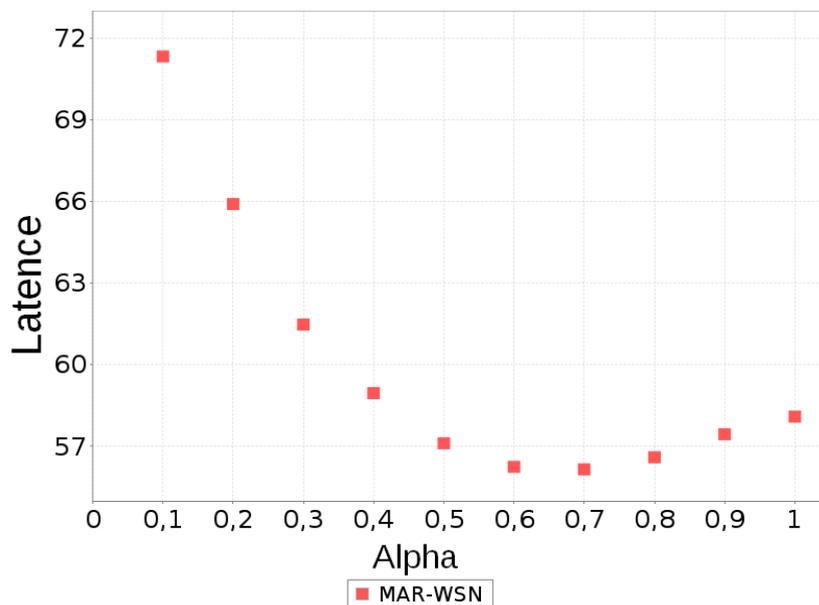


FIGURE 5.3 – MAR-WSN : latence des transmissions entre l'émetteur et le sink ( $\delta=10$ )

Comme le montre la figure 5.3, quand  $\alpha$  est petit la latence est importante parce que la métrique  $WDD$  donne plus d'importance à la distance et moins d'importance à la latence. Par conséquent, les nœuds du chemin choisi offrent un bon rapprochement vers le sink

mais génèrent une latence importante ce qui explique la consommation énergétique et le nombre de sauts faibles dans les figures 5.4 et 5.5 respectivement.

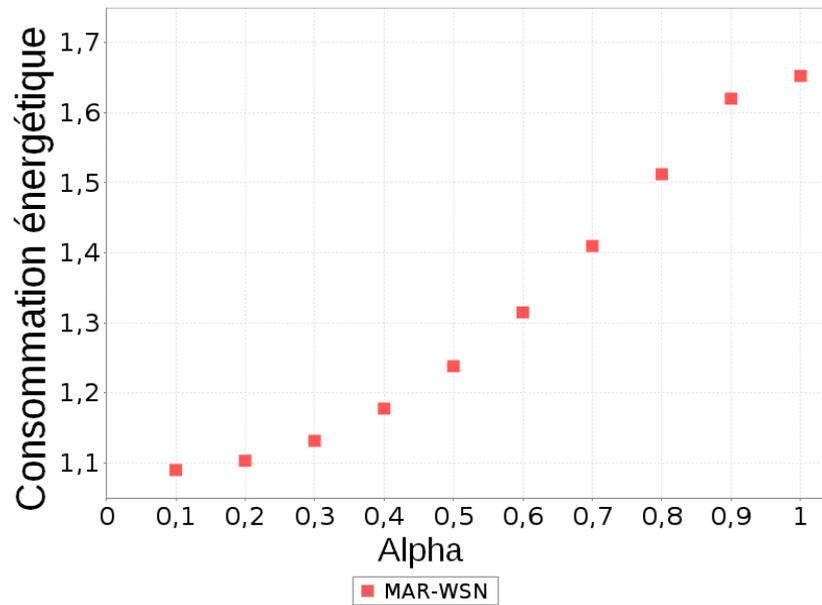


FIGURE 5.4 – MAR-WSN : consommation énergétique des nœuds du chemin entre l'émetteur et le sink ( $\delta=10$ )

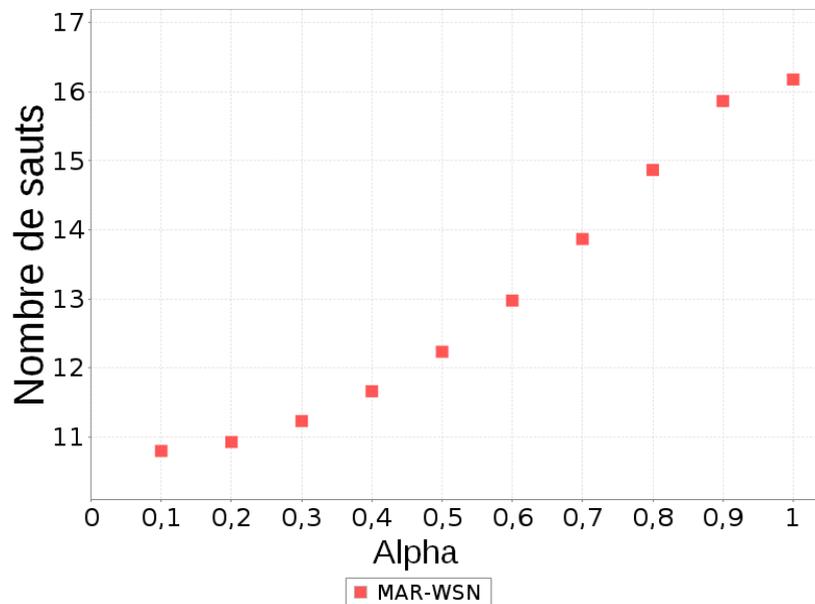


FIGURE 5.5 – MAR-WSN : nombre de sauts entre l'émetteur et le sink ( $\delta=10$ )

Lorsque  $\alpha$  est grand, la métrique *WDD* donne plus d'importance à la latence, ce qui retourne un chemin avec des nœuds qui ont une meilleure latence mais qui n'assurent pas un bon rapprochement vers le sink. Par conséquent, les chemins choisis ont plus de sauts et la consommation énergétique augmente comme montré dans les figures 5.4 et 5.5.

Afin de valider les résultats précédents, nous étendons les expérimentations en variant la densité des réseaux simulés  $\delta \in [6, 20]$  et ceci en changeant la taille du côté de la zone de déploiement ( $a$ ).

La figure 5.6 montre que les valeurs faibles de  $\alpha$  dans la métrique  $WDD$  génèrent des valeurs de latence importantes pour toutes densités confondues. Cela est la conséquence d'utiliser principalement la distance dans cette métrique. Nous remarquons aussi que la latence ne varie pas beaucoup pour les densités faibles parce que le nombre de voisins de chaque nœud du réseau est très faible, donc un nœud émetteur n'a pas assez de choix pour désigner le prochain saut.

Concernant les grandes valeurs de  $\alpha$ , la métrique  $WDD$  offre plus d'importance à la latence causant sa diminution (figure 5.6). Plus encore, quelque soit la densité, la latence diminue de 7% à 30% quand  $\alpha \leq 0.7$  pour les densités faibles et moyennes ( $\delta \leq 12$ ), et jusqu'à 50% quand  $\alpha \leq 0.8$  pour les grandes densités ( $\delta \geq 13$ ).

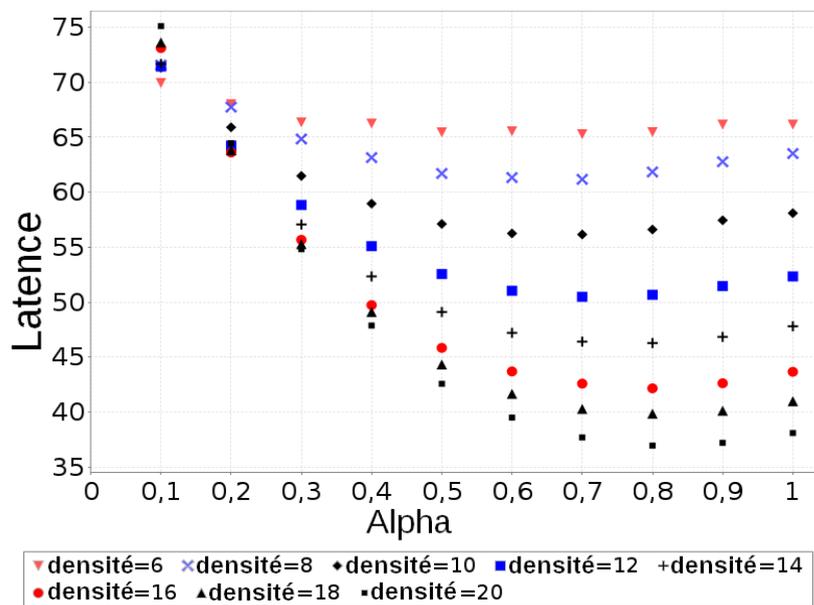


FIGURE 5.6 – MAR-WSN : latence des transmissions entre l'émetteur et le sink ( $\delta \in [6, 20]$ )

Les chemins de routage provenant des réseaux ayant une grande densité contiennent moins de sauts que les chemins de routage provenant des réseaux ayant une faible densité (figure 5.8) parce que les nœuds sont proches du sink. Ceci implique une consommation énergétique plus importante que celle générée par les nœuds des réseaux de faibles densités (figure 5.7) puisque le nombre de voisins des nœuds participant dans la transmission des données augmente et ces nœuds doivent donc consommer de l'énergie en réception.

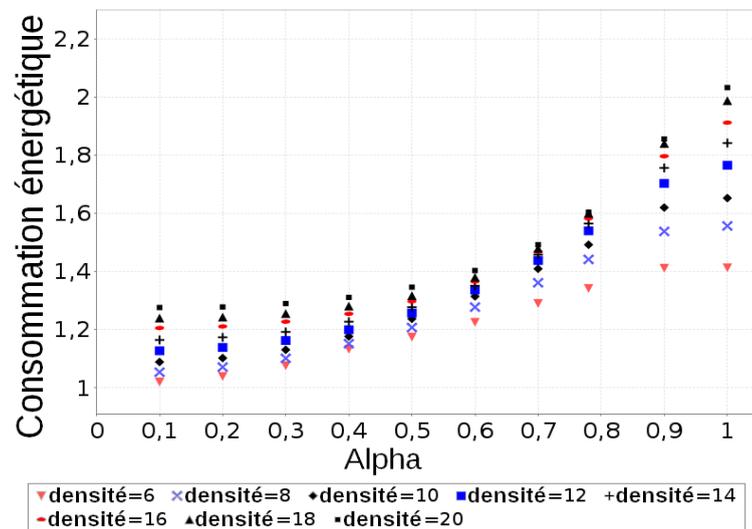


FIGURE 5.7 – MAR-WSN : consommation énergétique des nœuds du chemin entre l'émetteur et le sink ( $\delta \in [6, 20]$ )

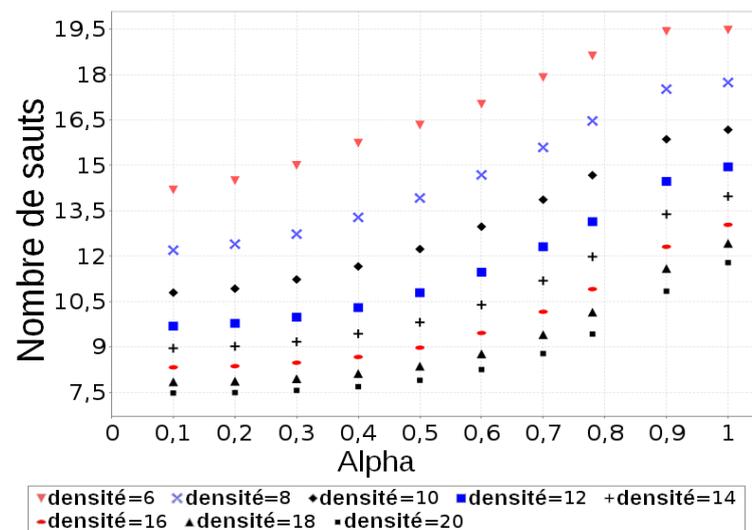


FIGURE 5.8 – MAR-WSN : nombre de sauts entre l'émetteur et le sink ( $\delta \in [6, 20]$ )

Nous souhaitons nous appuyer sur les résultats des simulations pour définir le coefficient  $\alpha$  à retenir. Ce coefficient devra permettre, d'un côté, d'obtenir les meilleures performances en terme de latence car c'est notre objectif principal et, d'un autre côté, de s'assurer que les autres métriques (consommation énergétique et nombre de sauts) ne soient pas altérées.

Idéalement, ce  $\alpha$  choisi devrait assurer deux conditions :

- d'une part, une valeur minimale de la métrique pour toute densité considérée, et ceci pour les trois métriques simultanément ;
- d'autre part, une moyenne minimale de la métrique sur toutes les densités considérées ; et celles-ci pour chaque métrique analysée.

Une telle valeur, selon la première condition, n'existe pas car les trois métriques ne sont

pas minimisées simultanément. Ce comportement est constaté indépendamment de la densité du réseau.

La deuxième condition impose la recherche d'un coefficient  $\alpha$  pour lequel les valeurs d'une même métrique sont les plus petites possible, indépendamment de la densité, et ceci, pour les trois métriques simultanément.

De même que pour la première condition, un tel  $\alpha$  n'existe pas (0.8 est le coefficient donnant la plus petite moyenne des latences sur toutes les densités, tandis que 0.1 est le coefficient de la plus petite moyenne en considérant l'énergie et le nombre de sauts sur toutes les densités).

Les coefficients sont :  $c_{\text{lat}}=0.6$ ,  $c_{\text{énergie}}=0.2$ ,  $c_{\text{sauts}}=0.2$ .

La latence étant l'objectif principal de l'algorithme, son coefficient de pondération sera le plus importants  $c_{\text{lat}}=0.6$ . L'énergie et le nombre de sauts sont des métriques qui ont une évolution similaire donc leur coefficient de pondération sera identique, 0.2 :  $c_{\text{énergie}}=0.2$ ,  $c_{\text{sauts}}=0.2$ .

Puisque nous ne pouvons pas additionner des opérandes ayant des unités différentes, nous utilisons les rangs à la place des valeurs. Pour que la première condition soit satisfaite, nous cherchons  $\alpha$  tel que  $rg_{\min}^{\alpha}$  soit minimal (voir équation 5.2).

$$rg_{\min}^{\alpha} = c_{\text{lat}} \times rg(\min_{\text{lat}}^{\forall \delta}) + c_{\text{énergie}} \times rg(\min_{\text{énergie}}^{\forall \delta}) + c_{\text{sauts}} \times rg(\min_{\text{sauts}}^{\forall \delta}) \quad (5.2)$$

où  $rg(\min_{\text{métrique}}^{\forall \delta})$  représente le rang de la valeur minimale de la métrique indépendamment de la densité. La métrique peut être la latence, la consommation énergétique ou le nombre de sauts.

Afin de calculer les rangs des valeurs minimales, indépendamment de la densité, nous identifions le  $\alpha$  pour lequel la métrique est minimale pour chaque densité (table 5.1).

métriques		$\alpha$ pour valeur minimale		
		latence	énergie	nombre de sauts
$\delta$				
	6	0.7	0.1	0.1
	8	0.7	0.1	0.1
	10	0.7	0.1	0.1
	12	0.7	0.1	0.1
	14	0.8	0.1	0.1
	16	0.8	0.1	0.1
	18	0.8	0.1	0.1
	20	0.8	0.1	0.1

TABLE 5.1 – La valeur de  $\alpha$  pour chaque  $\delta$ , pour laquelle la valeur minimale de chaque métrique est obtenue

Nous constatons que pour la latence, la valeur 0.7 apparaît quatre fois, la valeur 0.8 apparaît aussi quatre fois mais les autres valeurs n'apparaissent pas (zéro fois). En considérant la consommation énergétique et le nombre de sauts, la valeur 0.1 apparaît huit fois et les autres valeurs n'apparaissent pas. Nous présentons le nombre d'occurrences des valeurs de  $\alpha$  pour chaque métrique dans la table 5.2.

$\alpha$	nbr d'occurrences	latence	énergie	nombre de sauts
	0.1	0	8	8
	0.2	0	0	0
	0.3	0	0	0
	0.4	0	0	0
	0.5	0	0	0
	0.6	0	0	0
	0.7	4	0	0
	0.8	4	0	0
	0.9	0	0	0
	1	0	0	0

TABLE 5.2 – Nombre d'occurrences des valeurs de  $\alpha$  pour de chaque métrique

À partir de la table 5.2, nous attribuons un rang, allant de 1 à 10 (1 étant le meilleur), à chaque valeur de  $\alpha$ . Nous présentons ces rangs et les valeurs de  $rg_{min}^{\alpha}$  (formule 5.2) dans la table 5.3.

$\alpha$	rang	$rg(\min_{lat}^{\forall\delta})$	$rg(\min_{\text{énergie}}^{\forall\delta})$	$rg(\min_{\text{sauts}}^{\forall\delta})$	$rg_{min}^{\alpha}$
	0.1	3	1	1	2.2
	0.2	3	2	2	2.6
	0.3	3	2	2	2.6
	0.4	3	2	2	2.6
	0.5	3	2	2	2.6
	0.6	3	2	2	2.6
	0.7	1	2	2	1.4
	0.8	1	2	2	1.4
	0.9	3	2	2	2.6
	1	3	2	2	2.6

TABLE 5.3 – Les rangs des métriques minimales et les valeurs de  $rg_{min}^{\alpha}$ 

Nous constatons que les coefficients 0.7 et 0.8 offrent le meilleur compromis pour les valeurs minimales de la latence, la consommation énergétique et le nombre de sauts, indépendamment de la densité des réseaux.

Considérons maintenant la deuxième condition, i.e. la moyenne des métriques.

La table 5.4 indique la moyenne de chaque métrique pour les densités, par coefficient  $\alpha$ .

métriques		moyenne		
		latence	énergie	nombre de sauts
$\alpha$				
	0.1	72.21	1.15	9.94
	0.2	64.19	1.16	10.04
	0.3	59.29	1.18	10.26
	0.4	55.31	1.22	10.61
	0.5	52.34	1.26	11.04
	0.6	50.78	1.33	11.63
	0.7	50.01	1.42	12.40
	0.8	49.98	1.55	13.36
	0.9	50.58	1.69	14.43
	1	51.34	1.77	14.95

TABLE 5.4 – La moyenne de chaque métrique pour toutes les densités

De la même manière, nous utilisons la formule suivante pour les rangs des valeurs de la moyenne :

$$rg_{moy}^{\alpha} = C_{lat} \times rg(moy_{lat}^{\forall\delta}) + C_{\text{énergie}} \times rg(moy_{\text{énergie}}^{\forall\delta}) + C_{\text{sauts}} \times rg(moy_{\text{sauts}}^{\forall\delta}) \quad (5.3)$$

où  $rg(moy_{\text{métrique}}^{\forall\delta})$  représente le rang de la moyenne de la métrique indépendamment de la densité. La métrique peut être la latence, la consommation énergétique ou le nombre de sauts.

Les résultats obtenus sont présentés dans la table 5.5.

$\alpha$	rang	$rg(moy_{lat}^{\forall\delta})$	$rg(moy_{\text{énergie}}^{\forall\delta})$	$rg(moy_{\text{sauts}}^{\forall\delta})$	$rg_{moy}^{\alpha}$
	0.1	10	1	1	6.4
	0.2	9	2	2	6.2
	0.3	8	3	3	6
	0.4	7	4	4	5.8
	0.5	6	5	5	5.6
	0.6	4	6	6	4.8
	0.7	2	7	7	4
	0.8	1	8	8	3.8
	0.9	3	9	9	5.4
	1	5	10	10	7

TABLE 5.5 – Les rangs des  $\alpha$  selon la moyenne de chaque métrique

La dernière colonne de la table 5.5 indique que seul le coefficient 0.8 présente le meilleur compromis pour la moyenne des trois métriques.

Nous appliquons maintenant un principe d'agrégation, par pondération des deux conditions, ici sur les rangs  $rg_{min}^{\alpha}$  et  $rg_{moy}^{\alpha}$  obtenus précédemment. Les coefficients de pondération sont 0.5 pour la première condition et 0.5 pour la deuxième condition.

$$rg^{\alpha} = \frac{1}{2}rg_{min}^{\alpha} + \frac{1}{2}rg_{moy}^{\alpha} \quad (5.4)$$

Les résultats obtenus en appliquant la formule 5.4 sont présentés dans la table 5.6.

$\alpha$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
$rg^\alpha$	4.3	4.4	4.3	4.2	4.1	3.7	2.7	2.6	4	4.8

TABLE 5.6 – Les valeurs de  $rg^\alpha$

Nous constatons, à partir de la table 5.6, que les coefficients 0.8 et 0.7 sont classés premiers. La valeur 0.8 permet d'obtenir les meilleures performances, par compromis, en considérant les deux conditions de départ et les trois métriques évaluées.

Pour le reste de ce chapitre, nous utilisons cette valeur ( $\alpha=0.8$ ) dans la métrique  $WDD$ .

### 5.3.2/ MAR-WSN VS ROUTAGE GÉOGRAPHIQUE

Comme MAR-WSN est une approche de routage utilisant des informations du TDMA, nous évaluons dans un premier temps, ses performances en les comparant avec les performances d'un protocole de routage classique associé à un TDMA aléatoire mais qui n'utilise aucune information de ce dernier. Le protocole choisi est le routage géographique. Dans un routage géographique, chaque nœud choisit comme père son voisin le plus proche du sink en terme de distance.

La figure 5.9 montre clairement que MAR-WSN a une meilleure latence que le routage géographique grâce au cross-layering qui lui permet de profiter des informations de l'ordonnancement MAC afin d'optimiser la latence des communications.

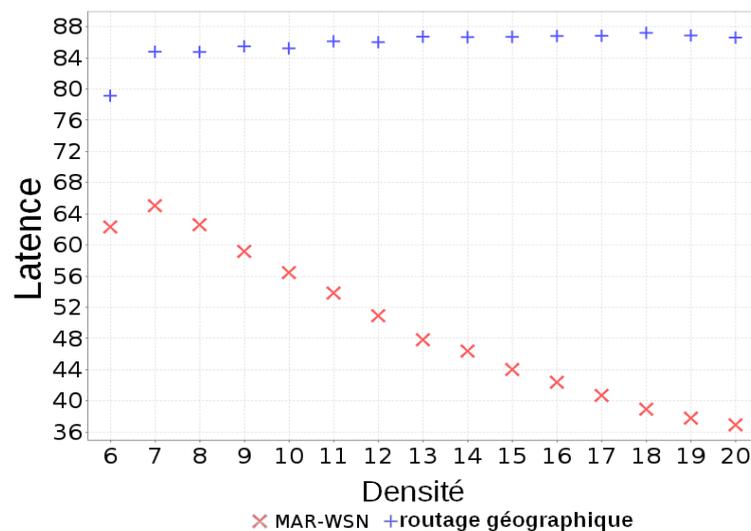


FIGURE 5.9 – Latence de MAR-WSN vs routage géographique

Plus la densité augmente (en réduisant la taille du côté  $a$  de la zone de déploiement), plus les nœuds sont proches du sink et la longueur de l'ordonnancement TDMA devient plus importante. Comme le routage géographique n'utilise pas les informations de l'ordonnancement TDMA, il essaye de gagner en latence en minimisant le nombre de sauts mais il en perd en même temps car il ne profite pas de l'ordonnancement TDMA. Ceci explique

ces performances stables en termes de latence indépendamment de la densité. En revanche, MAR-WSN, profite de l'augmentation de la densité et donc de la minimisation du nombre de sauts et en même temps il profite des informations de l'ordonnancement TDMA. Par conséquent, les latences de MAR-WSN décroissent avec l'augmentation de la densité.

Puisque le routage géographique permet de se rapprocher du sink en terme de distance, il fournit des chemins avec moins de sauts que MAR-WSN (figure 5.10) et donc moins de consommation énergétique (figure 5.11).

Notons que la consommation énergétique augmente avec l'augmentation de la densité car, pour chaque nœud, il y a plus plus de voisins et donc pour chaque transmission, il y a plus de nœuds qui perdent de l'énergie en mode réception.

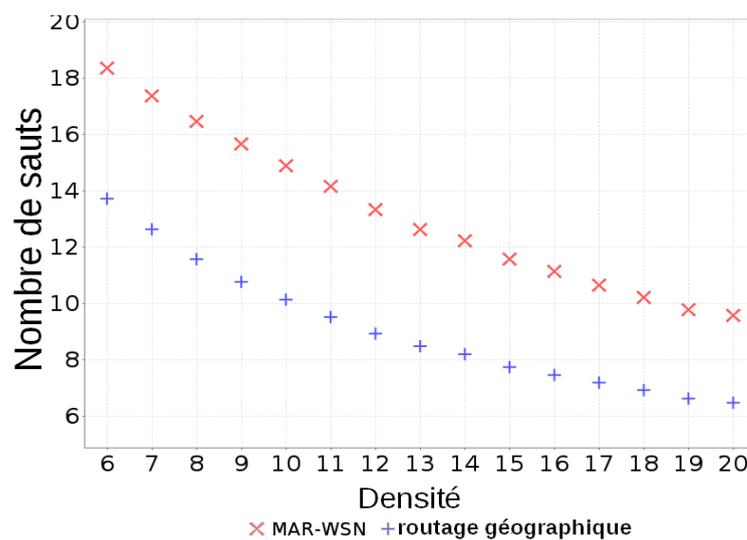


FIGURE 5.10 – Nombre de sauts de MAR-WSN vs routage géographique

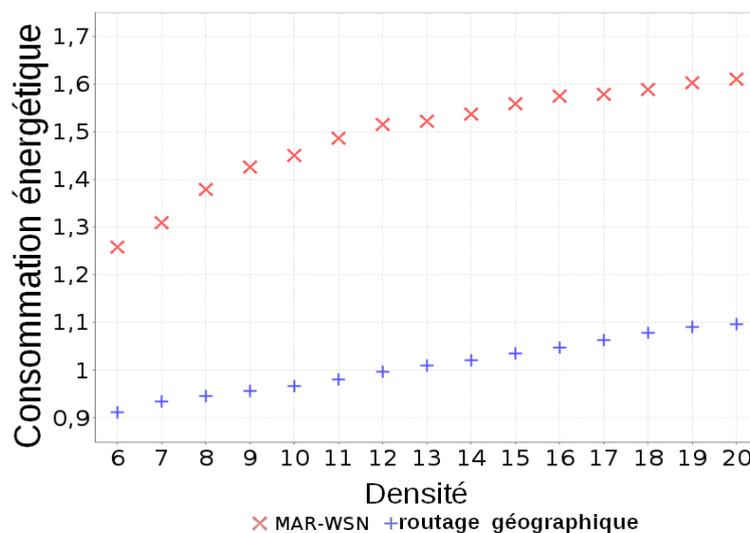


FIGURE 5.11 – Consommation énergétique de MAR-WSN vs routage géographique

La figure 5.12 résume les gains et pertes de MAR-WSN par rapport au routage géographique. Les valeurs positives représentent les gains de MAR-WSN tandis que les valeurs négatives représentent ses pertes. Comme montré par les résultats précédents, MAR-WSN permet l'optimisation de la latence des communications par rapport au routage géographique mais ce dernier reste en avance concernant le nombre de sauts et donc la consommation énergétique.

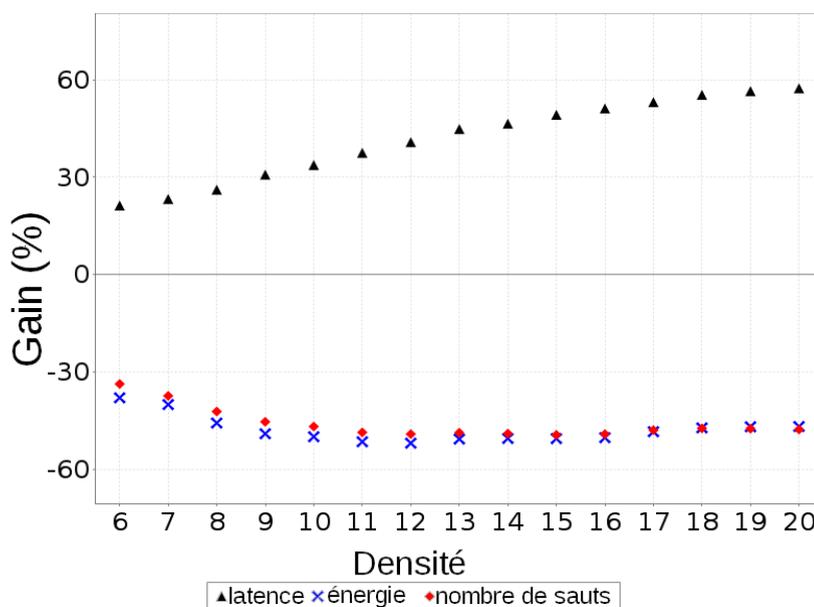


FIGURE 5.12 – Gains et pertes de MAR-WSN par rapport au routage géographique

### 5.3.3/ MAR-WSN vs ORW

Ensuite, nous procédons à la comparaison de MAR-WSN avec une approche cross-layer du même type. Comme expliqué dans la partie 3.3.1, ORW est la seule approche de routage basée sur des informations du TDMA trouvée dans la littérature. ORW est détaillé dans la section 3.3.1.1 (page 25).

Nous notons que pour avoir une comparaison équitable, les mêmes ordonnancements TDMA sont utilisés avec MAR-WSN et ORW.

D'un point de vue qualitatif, MAR-WSN nécessite une connaissance locale des informations du TDMA (dans un voisinage à 2 sauts). Par comparaison, ORW se crée une connaissance globale du réseau grâce à la propagation de l'information sur sa métrique EDC qui nécessite d'être stabilisée avant que la décision sur le routage puisse être prise.

D'un point de vue quantitatif, nous comparons les deux approches par rapport à la latence de bout en bout, la consommation énergétique et le nombre de sauts. De plus, pour un même réseau généré, le TDMA aléatoire (voir page 44) est calculé une seule fois et les deux approches de routage s'appuient sur le même ordonnancement.

La figure 5.13 montre que pour les réseaux de faibles densités, ORW est légèrement meilleur que MAR-WSN en terme de latence ; à partir de la densité  $\delta = 12$  MAR-WSN commence à compenser sa perte. Ceci s'explique par le fait qu'avec un réseau de faible densité, ORW dispose d'une vue globale du réseau et comme il y a peu de chemins vers

le sink, ORW arrive à choisir facilement le meilleur prochain saut pour chaque nœud. En revanche, avec des réseaux de fortes densités, beaucoup de chemins mènent vers le sink et une fois le choix de ORW fait sur un prochain saut, ce choix ne peut plus être changé ou amélioré ultérieurement. Avec MAR-WSN, la connaissance du voisinage à deux sauts permet d'améliorer voire corriger plus tard un chemin choisi à cause d'un prochain saut désigné au début mais qui ne génère pas la bonne latence. Donc, pour les réseaux de fortes densités, MAR-WSN est plus performant, en terme de latence, qu'ORW.

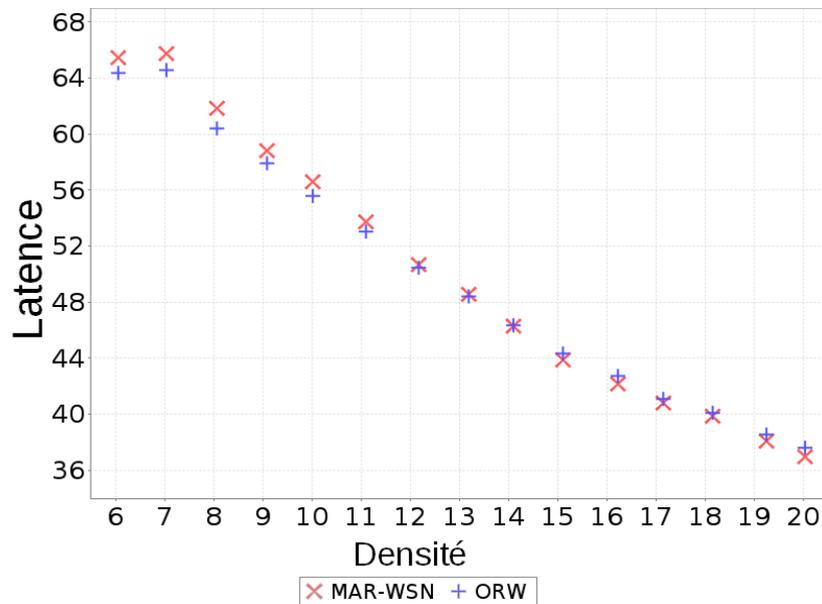


FIGURE 5.13 – Latence de MAR-WSN vs ORW

Les figures 5.14 et 5.15 confirment les résultats précédents ; MAR-WSN est légèrement derrière ORW, en termes de consommation énergétique et nombre de sauts, pour les faibles densités mais rattrape et compense cette perte dans les réseaux de fortes densités.

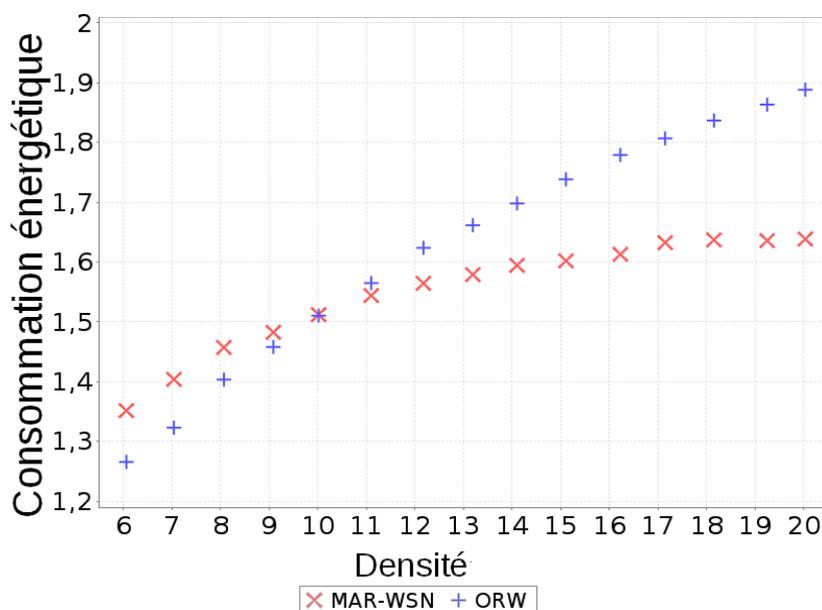


FIGURE 5.14 – Consommation énergétique de MAR-WSN vs ORW

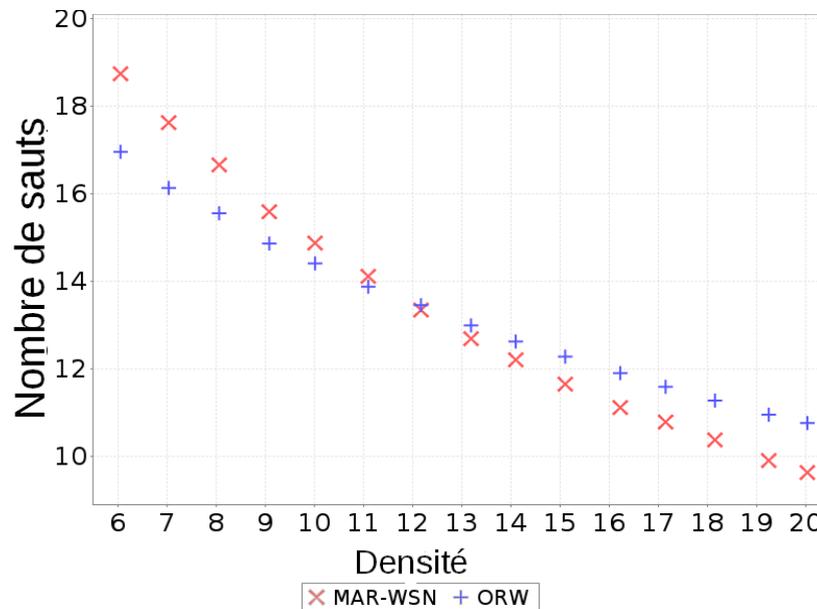


FIGURE 5.15 – Nombre de sauts de MAR-WSN vs ORW

Notons que, plus la densité augmente, plus le nombre de voisins augmente, par conséquent, même si le nombre de sauts diminue, il y a plus de nœuds qui consomment de l'énergie en mode réception.

La table 5.7 présente les coefficients de variation CV (voir description à la page 55) des trois métriques (latence, consommation énergétique et nombre de sauts), toutes densités confondues, pour MAR-WSN et ORW.

[CVmin, CVmax]	latence	consommation énergétique	nombre de sauts
MAR-WSN	[9.6 ; 10.3]%	[6.5 ; 8.9]%	[7.0 ; 10.6]%
ORW	[10.1 ; 12.4]%	[6.2 ; 9.0]%	[7.2 ; 9.5]%

TABLE 5.7 – Les intervalles des coefficients de variation de MAR-WSN et ORW pour toutes densités confondues

Les coefficients obtenus sont inférieurs à 12.4% ; par conséquent, les moyennes reflètent bien la tendance des valeurs.

Enfin, la figure 5.16 résume les gains et pertes de MAR-WSN par rapport à ORW concernant la latence des communications, la consommation énergétique et le nombre de sauts. Pour des réseaux de faibles densités, ORW offre de meilleurs résultats grâce à la vue globale qu'il peut avoir sur le réseau. Cette propriété ne le favorise plus avec les réseaux de fortes densités à cause de la présence de plus de chemins de routage, de ce fait un chemin une fois emprunté ne peut être changé plus tard dans ORW même si de meilleurs chemins existent. MAR-WSN devient meilleur grâce à la connaissance du voisinage à deux sauts.

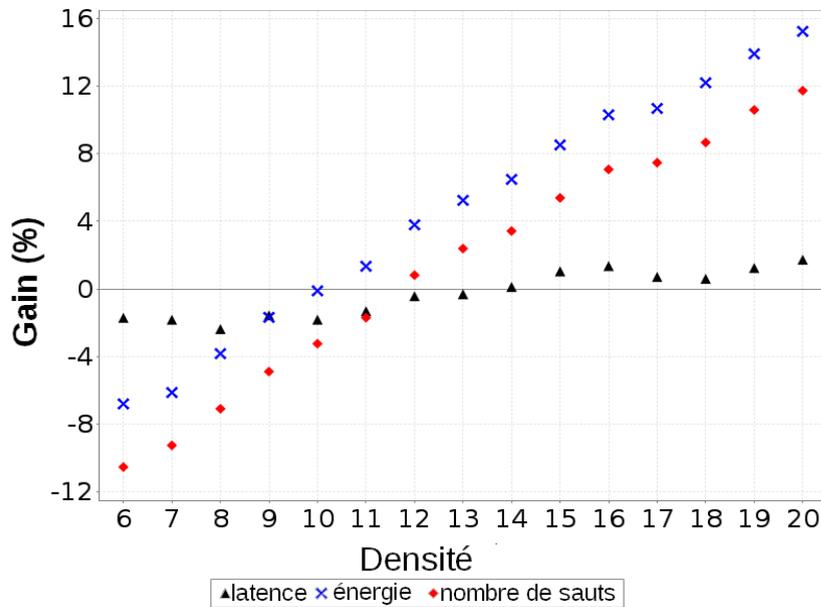


FIGURE 5.16 – Gains et pertes de MAR-WSN par rapport à ORW

## 5.4/ CONCLUSION

Dans ce chapitre, nous avons proposé une nouvelle approche cross-layer MAR-WSN (MAC-Aware Routing in Wireless Sensor Networks), qui est une approche de routage utilisant des informations du TDMA. MAR-WSN profite de la connaissance du voisinage à deux sauts et du TDMA pour le choix du chemin d'acheminement des données dans le réseau.

Les résultats de simulations Les résultats de simulations montrent que pour les réseaux de faibles densités, les performances de MAR-WSN sont proches de celles de ORW alors que MAR-WSN n'a pas besoin de propager sa métrique *WDD* dans le réseau contrairement à ORW. En revanche, considérant les réseaux de densités moyennes et fortes, MAR-WSN fournit de meilleures performances grâce à la connaissance du voisinage à deux sauts qui lui permet de construire les chemins de routage par parties.

# APPROCHE MAC UTILISANT DES INFORMATIONS DU ROUTAGE

## Sommaire

<b>6.1 Motivation</b>	<b>74</b>
<b>6.2 Méthodes d'allocation de slots</b>	<b>75</b>
<b>6.3 Approches basées sur des informations de l'arbre de routage</b>	<b>78</b>
<b>6.4 Approches basées sur des informations de l'arbre de routage et du graphe du réseau</b>	<b>82</b>
<b>6.5 Ordonnancement Optimal</b>	<b>84</b>
<b>6.6 Évaluation des performances des approches TDMA utilisant des informations du routage</b>	<b>90</b>
<b>6.7 Conclusion</b>	<b>100</b>

Dans ce chapitre, nous nous intéressons au problème inverse de celui présenté précédemment. Il s'agit de considérer l'hypothèse que le routage ait été construit, par un mécanisme indépendant (de type géographique par exemple). Notre objectif est de construire un ordonnancement TDMA pour tous les nœuds du réseau, afin que, en respectant le chemin de routage prévu, les latences de communication soient minimales. Les approches de construction des TDMA proposées dans ce chapitre s'inscrivent donc dans la classe cross-layer MAC utilisant des informations du routage (voir description à la section 3.3.2). Nous considérons disposer, pour un réseau, d'un arbre de routage dans lequel les nœuds transmettent les données vers le sink qui est la racine de l'arbre.

Nous proposons des approches qui utilisent différents parcours de l'arbre de routage avec deux méthodes d'allocation de slots dans le TDMA pour les nœuds du réseau. Les parcours de l'arbre de routage permettent de donner un ordre pour l'allocation des nœuds dans les slots du TDMA afin de privilégier certains nœuds qui sont plus critiques dans le réseau que d'autres nœuds.

Les deux méthodes d'allocation de slots permettent de rapprocher les slots de communications d'un nœud et de son père dans l'arbre de routage afin de minimiser le nombre de slots qui séparent les deux nœuds et par conséquent minimiser la latence des communications entre les deux et donc la latence de communication dans le réseau.

Nous identifions deux classes d'approches. La première classe contient les approches basées uniquement sur des informations de l'arbre de routage ; les approches proposées sont : I-CoLaNet [78] (sous-section 6.3.1), Rand-LO [78] (sous-section 6.3.2), Depth-LO [78] (sous-section 6.3.3), Depth-ReLO [78] (sous-section 6.3.4) et BFS [79] (sous-section 6.3.5). La deuxième classe contient les approches basées sur des informations du graphe

du réseau et des informations de l'arbre de routage ; ces approches sont IDeg-LO [80] (sous-section 6.4.1), IDeg-ReLO [80] (sous-section 6.4.2) et IDeg-BFS [79] (sous-section 6.4.3). Nous présentons aussi un algorithme d'Ordonnement Optimal par rapport à la latence [79] avec lequel nous comparons les performances des différentes approches d'ordonnement.

## 6.1/ MOTIVATION

La partie gauche de la figure 6.1, présente un réseau de capteurs composé de 6 nœuds (de A à F). Dans la partie droite de la figure, un arbre de routage est présenté où le nœud A est la racine de l'arbre et les autres nœuds transmettent, par leur père dans l'arbre, leurs données vers ce nœud.

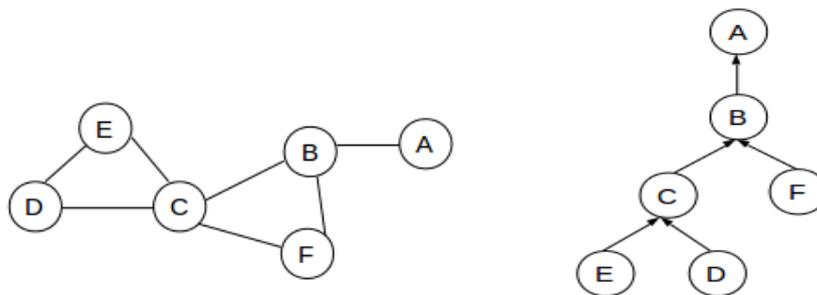


FIGURE 6.1 – Exemples d'un graphe représentant un réseau de capteurs (à gauche) et son arbre de routage (à droite)

La figure 6.2 présente deux ordonnancements TDMA différents pour le réseau de la figure 6.1. Dans le TDMA à gauche, aucune information du routage n'a été utilisée pour établir l'ordonnement. Pour ce cas, en moyenne, un nœud du réseau génère une latence de 6 slots de temps. Par exemple, si nous considérons le nœud D, il prend 9 slots de temps pour acheminer ses données au sink A : 2 slots pour envoyer les données au nœud C, 4 slots pour que le nœud C envoie les données au nœud B et 3 slots pour que le nœud B envoie les données au nœud A.

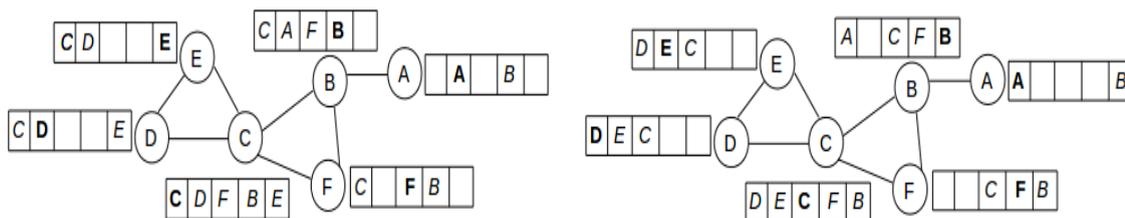


FIGURE 6.2 – Exemples de deux TDMA pour un réseau de capteurs

De l'autre côté, dans la partie droite de la figure 6.2, le TDMA est établi en se basant sur des informations de l'arbre de routage. Le nœud C a un slot après ses fils D et E dans l'arbre de routage. Le nœud B a son slot après ses fils C et F, et le nœud A a son slot

après le slot de son fils B.

Avec ce TDMA, un nœud génère, en moyenne, une latence de 5 slots de temps.

Le nœud D, par exemple, a besoin de 5 slots de temps pour transmettre ses informations au sink : 1 slot pour envoyer les données au nœud C, 2 slots pour que le nœud C envoie les données au nœud B et enfin 2 slots pour que le nœud B envoie les données au sink A.

Par conséquent, le TDMA peut profiter des informations spatiales venant de l'approche de routage afin de minimiser la latence des communications dans les réseaux de capteurs sans fil.

## 6.2/ MÉTHODES D'ALLOCATION DE SLOTS

Les méthodes d'allocation de slots permettent de trouver un slot de communication compatible pour chaque nœud du réseau. Un slot est considéré compatible s'il n'est alloué à aucun nœud ou si les nœuds alloués dans le slot ne sont pas les voisins à un saut ou à deux sauts du nœud considéré pour l'allocation. Ceci permet d'assurer l'absence des collisions.

Notons que, dans tous les cas, la longueur minimale du TDMA est égale au degré maximal dans le réseau, incrémenté, car le nœud ayant le degré maximal a besoin d'un slot pour transmettre ses données et d'un nombre de slots égal au nombre de ses voisins pour recevoir leurs données. Cette valeur est considérée comme valeur initiale de l'ordonnement TDMA.

Le but des deux méthodes d'allocation de slots suivantes est de minimiser le délai d'acheminement des données entre un nœud et son père dans l'arbre de routage tout en réduisant la longueur du TDMA et, par conséquent, le nombre de cas d'ajout de slots.

### 6.2.1/ MÉTHODE ASCENDANTE D'ALLOCATION DE SLOTS

Nous utilisons cette méthode avec les parcours ascendants d'arbre (de bas en haut) pour rapprocher le slot de communication d'un père dans l'arbre des slots de communications de ses fils. La méthode ascendante d'allocation de slots est décrite comme suit :

- Pour chaque nœud donné par le parcours de l'arbre de routage faire :
  - si le nœud est une feuille dans l'arbre alors trouver un slot compatible à partir du début du TDMA ;
  - si le nœud n'est pas une feuille alors trouver les slots de ses fils dans l'arbre et trouver un slot compatible dans le TDMA en commençant la recherche par le dernier slot de ses fils et en faisant une recherche circulaire (qui revient chercher un slot compatible depuis le début du TDMA) ;
  - dans les deux cas, si aucun slot compatible n'a été trouvé, ajouter un slot à la fin du TDMA et l'allouer au nœud.

### 6.2.2/ MÉTHODE DESCENDANTE D'ALLOCATION DE SLOTS

Contrairement à la méthode ascendante, la méthode descendante d'allocation de slots est utilisée avec les parcours descendants d'arbres (de haut en bas). Elle est décrite comme suit :

- Allouer le premier slot à la racine de l'arbre.
- Pour chaque nœud donné par le parcours de l'arbre de routage faire :
  - trouver un slot compatible en commençant la recherche par le slot du père dans l'arbre et en faisant une recherche circulaire ;
  - si aucun slot n'a été trouvé, ajouter un slot à la fin du TDMA et l'allouer au nœud.

### 6.2.3/ LA RECHERCHE CIRCULAIRE DANS LES MÉTHODES D'ALLOCATION DE SLOTS

Au lieu de rechercher un slot compatible à partir d'un slot  $i$  jusqu'à la fin du TDMA et d'en ajouter un à la fin du TDMA dans le cas d'absence de slots compatibles, nous avons opté pour une recherche circulaire. Celle-ci continue la recherche de slot compatible depuis le début du TDMA, si aucun slot compatible n'a été trouvé après le slot  $i$ . Si cette recherche reste infructueuse, l'ajout d'un nouveau slot se produit. Les deux méthodes d'allocation de slots présentées ci-dessus utilisent cette recherche circulaire qui permet d'obtenir un gain en délai d'attente des nœuds. Nous démontrons ce propos ci-après.

Nous appelons la première méthode qui n'utilise pas la recherche circulaire *RechSlot* et nous appelons notre méthode basée sur la recherche circulaire *RechSlotCirc*. Nous nous intéressons à comparer le délai engendré lorsque ces deux algorithmes différents sont appliqués pour un nœud  $n_j$ , père du nœud  $n_i$ . Nous considérons dans la suite que l'indice du slot du nœud  $n_i$  est  $i$  et que  $l$  est la longueur du TDMA.

#### 1. *RechSlot* :

- La recherche se fait entre les slots d'indices  $i + 1$  et  $l$ . Nous rappelons que l'indigage des slots commence à 1.
- Si aucun slot n'est compatible, alors un nouveau slot est ajouté à la fin du TDMA et est affecté au nœud  $n_j$ . La longueur du TDMA devient  $l+1$ . Le délai d'attente du nœud  $n_j$  est :

$$attente_{n_j} = 2l - i + 2$$

car  $n_j$  attend  $l$  slots avant de pouvoir émettre ses données, il prend un slot lors de l'émission et attend  $l+1-i$  slots pour pouvoir acheminer les données qu'il reçoit de son fils  $n_i$ .

Nous supposons que la position du slot  $i$  correspond à un événement équiprobable sur  $[1, l]$ , d'où, en moyenne,  $i = (l + 1)/2$ . Par conséquent, le délai d'attente de ce nœud est

$$attente_{n_j} = 2l - \frac{l + 1}{2} + 2 = \frac{3l + 3}{2}$$

- Sinon, le premier slot compatible est affecté au nœud  $n_j$ . Notons ce slot par  $j$ . Nous avons

$$attente2_{n_j} = 2j - i$$

car le nœud  $n_j$  attend  $j-1$  slots avant de pouvoir émettre ses données, il prend un slot lors de l'émission et attend  $j-i$  slots pour pouvoir acheminer les données qu'il reçoit de son fils  $n_i$ .

Nous supposons que la position du slot  $j$  correspond à un événement équiprobable sur  $[i+1, l]$ , d'où, en moyenne,  $j = (l+i+1)/2$ . Par conséquent, le délai d'attente de ce nœud est

$$attente2_{n_j} = l + i + 1 - i = l + 1$$

## 2. RechSlotCirc :

- Avec le principe circulaire, la recherche se fait entre les slots  $i+1$  et  $l$ , ensuite entre  $1$  et  $i-1$ .
- Si aucun slot n'est compatible, alors un nouveau slot est ajouté à la fin du TDMA et est affecté au nœud. La longueur du TDMA devient  $l+1$  et le délai d'attente de  $n_j$  est :

$$attente1_{n_j}^{circ} = 2l - i + 2$$

car  $n_j$  attend  $l$  slots avant de pouvoir émettre ses données, il prend un slot lors de l'émission et attend  $l+1-i$  slots pour pouvoir acheminer les données qu'il reçoit de son fils  $n_i$ .

Nous supposons que la position du slot  $i$  correspond à un événement équiprobable sur  $[1, l]$ , d'où, en moyenne,  $i = (l+1)/2$ . Par conséquent, le délai d'attente de ce nœud est

$$attente1_{n_j}^{circ} = 2l - \frac{l+1}{2} + 2 = \frac{3l+3}{2}$$

- Sinon, le premier slot compatible  $j$  est affecté au nœud  $n_j$ . Le délai d'attente de ce nœud est

$$attente2_{n_j}^{circ} = \frac{l-i+1}{l}(2j-i) + \frac{i-1}{l}(l-i+2j)$$

parce que dans ce cas nous avons deux possibilités.

- Si le slot  $j$  se trouve entre  $i+1$  et  $l$  le délai d'attente de  $n_j$  est  $2j-i$  car  $n_j$  attend  $j-1$  slots avant de pouvoir émettre, il prend un slot lors de l'émission et attend  $j-i$  slots pour pouvoir acheminer les données qu'il reçoit de son fils  $n_i$ .
- Par contre, si le slot  $j$  se trouve entre  $1$  et  $i-1$ , le délai d'attente de  $n_j$  est  $l-i+2j$ , car  $n_j$  attend  $j-1$  slots avant de pouvoir transmettre, il prend un slot lors de l'émission et attend  $l-i+j$  slots pour pouvoir acheminer les données qu'il reçoit de son fils  $n_j$ .

La probabilité de l'occurrence de la première possibilité est  $\frac{l-i+1}{l}$  et de la deuxième est  $\frac{i-1}{l}$ .

Nous supposons que la position du slot  $j$  correspond à un événement équiprobable sur  $[1, l]$ , d'où, en moyenne,  $j = \frac{l+1}{2} - \frac{i}{l}$ . Ainsi,

$$attente2_{n_j}^{circ} = l - \frac{l+1}{l}$$

Dans le cas d'ajout de slot, le délai d'attente est le même pour les deux algorithmes, mais pour le cas de non ajout de slot, l'algorithme circulaire présente un délai moindre :

$$attente_{2n_j}^{circ} < attente_{2n_j}$$

### 6.3/ APPROCHES BASÉES SUR DES INFORMATIONS DE L'ARBRE DE ROUTAGE

Dans cette classe d'approches, différents parcours sont appliqués sur l'arbre de routage, ce qui donne un ordre pour les nœuds du réseau. Ensuite, une méthode d'allocation de slots est appliquée sur les nœuds suivant l'ordre donné par les parcours. Pour toutes ces approches, l'entrée est un arbre de routage quelconque.

#### 6.3.1/ I-COLANET (IMPROVED COLANET)

I-CoLaNet est une amélioration de l'approche cross-layer CoLaNet [51]. Au lieu de colorer tous les voisins d'un nœud déjà coloré, la coloration se fait uniquement sur les nœuds ayant déjà un père coloré dans l'arbre de routage. Ceci permet de colorer un nœud, ensuite ses fils dans l'arbre et donc de refléter le sens des communications dans le processus d'ordonnement du TDMA, information que CoLaNet n'exploite pas dans le processus de coloriage.

Fonctionnement de I-CoLaNet :

1. Sélectionner le nœud ayant le nombre maximum de fils dans l'arbre de routage et qui n'est pas coloré.
2. Colorer ce nœud.
3. Sélectionner les nœuds ayant un père déjà coloré dans l'arbre de routage.
4. Colorer les nœuds sélectionnés.
5. Répéter à partir de l'étape 3 tant qu'il y a un nœud qui peut être coloré.
6. S'il y a encore des nœuds non colorés, répéter à partir de l'étape 1.
7. Transformer les couleurs en un ordonnancement tel que la couleur d'un nœud représente son slot de communication.

---

**Algorithm** I-CoLaNet(graph, routTree, colorTable)

---

```

Input : graph      # le graphe modélisant le réseau
          routTree   # l'arbre de routage
Output: colorTable # table associant chaque noeud à sa couleur
Local : maxDegreeN # noeud ayant le degré maximal dans le graphe
          nextToColor # liste des noeuds à colorer
          currentNode # noeud courant
          coloredNodes # liste des noeuds ayant des couleurs
          n            # le nombre de noeuds
1: while sizeOf(coloredNodes) < n do
2:   maxDegreeN = getMaxChildrenUncoloredNode(routTree)
3:   add(maxDegreeN, nextToColor)

```

```

4:   while nextToColor  $\neq$   $\emptyset$  do
5:     currentNode = removeFirst(nextToColor)
6:     color(currentNode, graph, colorTable)
7:     add(currentNode, coloredNodes)
8:     add(getUncoloredChildren(currentNode, routTree), nextToColor)
9:   end while
10: end while

```

---

### 6.3.2/ RAND-LO (RANDOM LEAVES ORDERING)

CoLaNet et I-CoLaNet commencent le coloriage des nœuds par le nœud ayant le degré maximal dans l'arbre de routage. Comme ce nœud peut se trouver dans un endroit quelconque dans l'arbre de routage, ce choix ne permet pas l'optimisation de la latence. Plutôt que de commencer l'ordonnement avec ce nœud nous privilégions les feuilles de l'arbre de routage. Ce sont les nœuds les plus éloignés du sink et par conséquent leurs latences sont les plus importantes puisqu'elles cumulent les latences des nœuds qui se trouvent dans leurs chemins vers le sink.

Rand-LO sélectionne les feuilles de l'arbre de routage dans un ordre aléatoire pour leur attribuer des slots dans le TDMA ; l'algorithme continue ensuite l'allocation des slots en remontant dans l'arbre.

Fonctionnement de Rand-LO :

1. Sélectionner aléatoirement les feuilles de l'arbre.
2. Pour chaque feuille sélectionnée, trouver un slot compatible en appliquant la méthode ascendante d'allocation de slots.
3. Sélectionner les pères des nœuds déjà alloués dans les slots en gardant le même ordre de sélection des fils.
4. Pour chaque père, trouver un slot en appliquant la méthode ascendante d'allocation de slots.
5. Répéter à partir de l'étape 3 jusqu'à ce que tous les nœuds du réseau aient des slots dans le TDMA.

---

**Algorithm** Rand-LO(graph, routTree, slotAlloc)

---

```

Input   : graph      # le graphe modélisant le réseau
           : routTree  # l'arbre de routage
Output : slotAlloc  # la matrice d'allocation (le TDMA)
Local   : currentNode # le noeud courant
           : listNodes  # liste des noeuds sélectionnés
1: add(getLeaves(routTree), listNodes)
2: while listNodes  $\neq$   $\emptyset$  do
3:   currentNode = removeFirst(listNodes)
4:   ascendingSlotAllocMethod(currentNode, graph, routTree, slotAlloc)
5: end while
6: for each node, currentNode, having a scheduled child do
   # suivant l'ordre des fils
7:   ascendingSlotAllocMethod(currentNode, graph, routTree, slotAlloc)
8: end for

```

---

### 6.3.3/ DEPTH-LO (DEPTH LEAVES ORDERING)

À partir de Rand-LO nous avons constaté que les feuilles de l'arbre de routage ne sont pas toutes sur des chemins vers le sink ayant la même longueur et donc les sélectionner aléatoirement n'est pas la meilleure solution. Dans Depth-LO, les feuilles de l'arbre de routage sont triées selon leur profondeur dans l'arbre pour privilégier les feuilles les plus éloignées du sink.

Fonctionnement de Depth-LO :

1. Sélectionner les feuilles de l'arbre.
2. Trier les feuilles, en ordre décroissant, selon leur profondeur dans l'arbre.
3. Pour chaque feuille, trouver un slot compatible en appliquant la méthode ascendante d'allocation de slots.
4. Sélectionner les pères des nœuds déjà alloués dans les slots en gardant le même ordre des fils.
5. Pour chaque père, trouver un slot en appliquant la méthode ascendante d'allocation de slots.
6. Répéter à partir de l'étape 4 jusqu'à ce que tous les nœuds du réseau aient des slots dans le TDMA.

---

**Algorithm** Depth-LO(graph, routTree, slotAlloc)

---

```

Input  : graph      # le graphe modélisant le réseau
           routTree   # l'arbre de routage
Output : slotAlloc  # la matrice d'allocation (le TDMA)
Local  : currentNode # le noeud courant
           listNodes  # liste des noeuds sélectionnés
1: add(getLeaves(routTree), listNodes)
2: descendingSortDepth(listNodes, routTree)
3: while listNodes ≠ ∅ do
4:   currentNode = removeFirst(listNodes)
5:   ascendingSlotAllocMethod(currentNode, graph, routTree, slotAlloc)
6: end while
7: for each node, currentNode, having a scheduled child do
   # suivant l'ordre des fils
8:   ascendingSlotAllocMethod(currentNode, graph, routTree, slotAlloc)
9: end for

```

---

### 6.3.4/ DEPTH-RELO (DEPTH REMAINING LEAVES ORDERING)

Si un arbre de routage contient des chemins de longueurs différentes, les nœuds internes d'un chemin peuvent avoir une profondeur plus importante que les feuilles d'un autre chemin. Dans Depth-ReLO, l'allocation des slots considère les nœuds dans l'ordre de leur profondeur dans l'arbre de routage.

Fonctionnement de Depth-ReLO :

1. Sélectionner la feuille ayant la plus grande profondeur dans l'arbre.

2. Trouver un slot compatible pour cette feuille en appliquant la méthode ascendante d'allocation de slots.
3. Supprimer la feuille de l'arbre.
4. Répéter à partir de l'étape 1 jusqu'à ce que tous les nœuds du réseau aient des slots dans le TDMA.

---

**Algorithm** Depth-Relo(graph, routTree, slotAlloc)

---

**Input** : graph # le graphe modélisant le réseau  
           routTree # l'arbre de routage  
**Output** : slotAlloc # la matrice d'allocation (le TDMA)  
**Local** : currentNode # le nœud courant  
 1: **while** routTree  $\neq \emptyset$  **do**  
 2:   currentNode = findHighestDepthNode(routTree)  
 3:   ascendingSlotAllocMethod(currentNode, graph, routTree, slotAlloc)  
 4:   remove(currentNode, routTree)  
 5: **end while**

---

### 6.3.5/ BFS

BFS fonctionne différemment des autres approches de cette classe, car il commence l'ordonnancement par les nœuds appartenant à la partie haute de l'arbre de routage et continue l'allocation des slots en descendant dans l'arbre. Ceci permet de commencer l'ordonnancement avec les nœuds les plus proches du sink car ils sont inclus dans tous les chemins de communication. Leur affecter des slots au plus tôt induit une latence minimale à ces nœuds très sollicités, étant relais pour les nœuds accédant au sink, et par conséquent aux autres nœuds. Le TDMA obtenu est ensuite inversé.

Fonctionnement de BFS :

1. Sélectionner la racine de l'arbre et appliquer la méthode descendante d'allocation de slots.
2. Sélectionner les nœuds ayant déjà un père alloué dans le TDMA en gardant le même ordre des pères.
3. Pour chaque nœud sélectionné dans l'étape 2 appliquer la méthode descendante d'allocation de slots.
4. Répéter à partir de l'étape 2 jusqu'à ce que tous les nœuds du réseau aient des slots dans le TDMA.
5. Inverser le TDMA obtenu.

---

**Algorithm** BFS(graph, routTree, slotAlloc)

---

**Input** : graph # le graphe modélisant le réseau  
           routTree # l'arbre de routage  
**Output** : slotAlloc # la matrice d'allocation (le TDMA)  
**Local** : currentNode # nœud courant  
 1: allocate(1, getRoot(routTree)) # allouer le premier slot pour la racine de l'arbre

```

2: for each node, currentNode, having a scheduled parent do
    # suivant l'ordre des pères
3:   descendingSlotAllocMethod(currentNode, graph, routTree, slotAlloc)
4: end for
5: reverse(slotAlloc)

```

---

## 6.4/ APPROCHES BASÉES SUR DES INFORMATIONS DE L'ARBRE DE ROUTAGE ET DU GRAPHE DU RÉSEAU

Les approches TDMA précédentes utilisent des informations de l'arbre de routage pour l'ordonnancement du TDMA mais ignorent complètement le graphe modélisant le réseau. Les informations venant du graphe peuvent aussi influencer les latences de communications car certains nœuds peuvent être plus critiques que d'autres en termes d'interférence, ce qui n'apparaît pas dans l'arbre de routage. L'ordonnancement d'un nœud présentant beaucoup d'interférences vers la fin du processus d'ordonnancement peut amener facilement à ne pas lui trouver de slot compatible, proche de son prédécesseur ou successeur dans l'arbre de routage.

Dans cette deuxième classe d'heuristiques, les parcours appliqués sur les arbres de routage dépendent aussi d'une information du graphe du réseau que nous appelons *le degré d'interférence* (interference degree). Le degré d'interférence d'un nœud est égal à la somme du nombre des ses voisins à un saut et du nombre de ses voisins à deux sauts. Si un nœud est un voisin à un saut et à deux sauts en même temps, il n'est considéré qu'une seule fois. Cette métrique permet de privilégier les nœuds présentant le plus d'interférences dans le réseau.

### 6.4.1/ IDEG-LO (INTERFERENCE DEGREE LEAVES ORDERING)

L'idée de IDeg-LO est de trier les feuilles selon leur degré d'interférence afin de privilégier celles qui présentent le plus d'incompatibilité durant les communications.

Fonctionnement de IDeg-LO :

1. Sélectionner les feuilles de l'arbre.
2. Trier les feuilles, en ordre décroissant, selon leur degré d'interférence dans le graphe.
3. Pour chaque feuille, trouver un slot compatible en appliquant la méthode ascendante d'allocation de slots.
4. Sélectionner les pères des nœuds déjà alloués dans les slots en gardant le même ordre de tri des fils.
5. Pour chaque père, trouver un slot en appliquant la méthode ascendante d'allocation de slots.
6. Répéter à partir de l'étape 4 jusqu'à ce que tous les nœuds du réseau aient des slots dans le TDMA.

---

**Algorithm** IDeg-LO(graph, routTree, slotAlloc)

---

**Input** : graph # le graphe modélisant le réseau  
routTree # l'arbre de routage  
**Output** : slotAlloc # la matrice d'allocation (le TDMA)  
**Local** : currentNode # le noeud courant  
listNodes # liste des noeuds sélectionnés

```

1: add(getLeaves(routTree), listNodes)
2: descendingSortIDeg(listNodes, routTree)
3: while listNodes ≠ ∅ do
4:   currentNode = removeFirst(listNodes)
5:   ascendingSlotAllocMethod(currentNode, graph, routTree, slotAlloc)
6: end while
7: for each node, currentNode, having a scheduled child do
   # suivant l'ordre des fils
8:   ascendingSlotAllocMethod(currentNode, graph, routTree, slotAlloc)
9: end for

```

---

#### 6.4.2/ IDEG-RELO (INTERFERENCE DEGREE REMAINING LEAVES ORDERING)

Les nœuds internes d'un chemin de routage peuvent présenter plus d'interférence que les feuilles d'autres chemins. IDeg-ReLO donne la priorité à ces nœuds internes durant le processus d'ordonnement.

Fonctionnement de IDeg-ReLO :

1. Sélectionner la feuille ayant le plus grand degré d'interférence dans le graphe.
  2. Trouver un slot compatible pour cette feuille en appliquant la méthode ascendante d'allocation de slots.
  3. Supprimer la feuille de l'arbre.
  4. Répéter à partir de l'étape 1 jusqu'à ce que tous les nœuds du réseau aient des slots dans le TDMA.
- 

**Algorithm** IDeg-ReLO(graph, routTree, slotAlloc)

---

**Input** : graph # le graphe modélisant le réseau  
routTree # l'arbre de routage  
**Output** : slotAlloc # la matrice d'allocation (le TDMA)  
**Local** : currentNode # le noeud courant

```

1: while routTree ≠ ∅ do
2:   currentNode = findMaxIDegLeaf(routTree, graph)
3:   ascendingSlotAllocMethod(currentNode, graph, routTree, slotAlloc)
4:   remove(currentNode, routTree)
5: end while

```

---

### 6.4.3/ IDEG-BFS (INTERFERENCE DEGREE BFS)

Comme BFS, IDeg-BFS commence l'allocation des slots par les nœuds appartenant à la partie haute de l'arbre de routage tout en privilégiant les nœuds ayant un degré d'interférence important.

Fonctionnement de IDeg-BFS :

1. Sélectionner la racine de l'arbre et appliquer la méthode descendante d'allocation de slots.
2. Sélectionner les nœuds ayant déjà un père alloué dans le TDMA en gardant le même ordre des pères.
3. Trier ces nœuds, en ordre décroissant, selon leur degré d'interférence dans le graphe.
4. Suivant l'ordre du tri, pour chaque nœud, appliquer la méthode descendante d'allocation de slots.
5. Répéter à partir de l'étape 2 jusqu'à ce que tous les nœuds du réseau aient des slots dans le TDMA.
6. Inverser le TDMA obtenu.

---

**Algorithm** IDeg-BFS(graph, routTree, slotAlloc)

---

**Input** : graph # le graphe modélisant le réseau  
 routTree # l'arbre de routage  
**Output** : slotAlloc # la matrice d'allocation (le TDMA)  
**Local** : childrenList # liste des fils d'un noeud  
 currentNode # le noeud courant

- 1: allocate(1, getRoot(routTree)) # allouer le premier slot pour la racine de l'arbre
- 2: childrenList = findChildren(getRoot(routTree), routTree)
- 3: descendingSortIDeg(childrenList, graph)
- 4: **for** each unscheduled node, currentNode, having a scheduled parent **do**
- 5: # suivant l'ordre des pères
- 6: descendingSlotAllocMethod(currentNode, graph, routTree, slotAlloc)
- 7: **end for**
- 8: reverse(slotAlloc)

---

## 6.5/ ORDONNANCEMENT OPTIMAL

Les précédentes approches d'ordonnement se constituent comme heuristiques essayant de minimiser la latence des communications. L'évaluation comparative de leurs performances peut être insuffisante si l'ordonnement optimal (par rapport à la latence) n'est pas connu. Nous nous sommes donc également intéressés à l'obtention de l'ordonnement optimal par rapport à la latence engendrée lors des communications suivant l'arbre de routage. L'algorithme proposé permet, d'une part, d'argumenter le besoin des heuristiques précédemment décrites, car nous démontrons la difficulté de

l'obtention de l'ordonnancement optimal (même pour des instances de réseaux de faibles tailles) grâce à un calcul de complexité. D'autre part, l'ordonnancement optimal construit permet d'obtenir une borne inférieure de la latence. Celle-ci sert de référentiel pour évaluer les heuristiques.

### 6.5.1/ PRINCIPE ET ALGORITHME

Afin d'obtenir le TDMA optimisant le mieux la latence des communications, nous énumérons toutes les allocations possibles et valides et nous retenons celle qui génère la valeur minimale pour la métrique visée i.e. la latence.

L'énumération de tous les TDMA candidats dépend de la longueur des TDMA (en nombre de slots) bornée par le degré maximal des nœuds dans le graphe, incrémenté, et le nombre total des nœuds. Pour chaque longueur possible et valide, nous recherchons la meilleure allocation.

---

**Algorithm** findOptSlotAllocation( $n$ , graph, routTree, fSlotAlloc)

---

```

Input   :  $n$            # le nombre de nœuds
           : graph       # le graphe modélisant le réseau
           : routTree    # l'arbre de routage
Output  : fSlotAlloc  # la matrice d'allocation finale (le TDMA)
Local   : minFrameLg  # la longueur minimale du TDMA est le degré maximal
           :             # des nœuds dans le graphe incrémenté
           : cSlotAlloc  # la matrice d'allocation actuelle
           : metricRefV  # la valeur de la métrique pour l'allocation obtenue
           : cMetric     # la valeur de la métrique pour l'allocation actuelle
           : iFrameLg    # compteur

1: minFrameLg = getMaxNodeDegree(graph) + 1
2: metricRefV = maxValueMetric()
3: # essayer toutes les longueurs de TDMA pour trouver l'allocation minimisant la métrique
4: for iFrameLg from minFrameLg to  $n$  do
5:   findOptSlotAllocationRec(iFrameLg, 1,  $n$ , graph, routTree, metricRefV,
                             cSlotAlloc, cMetric)
6:   if cMetric < metricRefV then # une meilleure solution
7:     fSlotAlloc = cSlotAlloc
8:     metricRefV = cMetric
9:   end if
10: end for

```

---

L'énumération de tous les TDMA possibles pour une longueur de TDMA fixée se fait récursivement en utilisant une méthode de backtracking. L'idée du backtracking est d'essayer d'allouer chaque slot possible pour le nœud actuel ; ensuite, il continue l'allocation de façon récursive pour les autres nœuds. Les nœuds sont ordonnés selon leurs identifiants. Le premier nœud avec lequel l'allocation débute est le sink (le nœud 1).

L'algorithme produit en sortie la meilleure allocation de slots et sa valeur de métrique en utilisant la longueur du TDMA, le nœud actuel, la taille du réseau, le graphe modélisant le réseau, l'arbre de routage et la valeur de référence pour la métrique comme paramètres

d'entrée.

---

**Algorithm** findOptSlotAllocationRec(frameLength,  $i$ ,  $n$ , graph, routTree, metricRef, slotAlloc, metricV)

---

**Input** : frameLength # la longueur actuelle du TDMA  
 $i$  # le nœud actuel pour lequel l'allocation de slot est recherchée  
 $n$  # le nombre de nœuds  
graph # le graphe modélisant le réseau  
routTree # l'arbre de routage  
metricRef # la valeur de référence pour la métrique

**Output** : slotAlloc # la matrice d'allocation (le TDMA)  
metricV # la valeur de la métrique pour l'allocation obtenue

**Local** : k # compteur  
flag # booléen, à faux = pas de slot compatible pour le nœud courant  $i$

```

1: # allouer la mémoire pour la matrice d'allocation slotAlloc( $n \times$  frameLength)
2: if  $i = n+1$  then
3:   metricV = computeMetric(slotAlloc, routTree)
4:   if metricV < metricRef then
5:     save(slotAlloc)
6:   end if
7: else
8:   flag=faux
9:   for  $k$  from 1 to frameLength do
10:    if isCollisionFree( $i$ ,  $k$ , slotAlloc) then
11:      flag=vrai
12:      slotAlloc[ $i,k$ ] =  $k$  # le nœud  $i$  transmet durant le slot  $k$ 
13:      impactAllocNeighbors( $i$ , graph, slotAlloc, frameLength)
14:      # trouver l'allocation de slot pour le nœud  $i+1$ 
15:      findOptSlotAllocationRec(frameLength,  $i+1$ ,  $n$ , graph, routTree,
                               metricRef, slotAlloc, metricV)
16:      slotAlloc[ $i,k$ ] = 0 # libérer l'allocation de slot pour le nœud  $i$ 
17:    end if
18:  end for
19:  if non flag then #la longueur actuelle ne permet pas d'allouer des slots
20:    #à tous les nœuds du réseau
21:    metricV = maxValueMetric()
22:  end if
23:  impactFreeNeighbors( $i$ , graph, slotAlloc, frameLength)
24: end if

```

---

La procédure *computeMetric* permet de trouver la valeur de la métrique choisie (la latence dans notre cas) pour toute allocation obtenue. Si cette valeur est la minimale, l'allocation ayant cette valeur est retournée en paramètre de sortie.

La procédure *isCollisionFree* permet de trouver un slot compatible dans l'allocation courante pour chaque nœud. Nous rappelons qu'un slot est considéré compatible pour le nœud  $i$  s'il n'est pas encore alloué, ou si les nœuds utilisant le slot en transmission ne sont pas des voisins à un saut ou à deux sauts du nœud  $i$ .

La procédure *impactAllocNeighbors* considère le slot alloué pour le nœud  $i$  et modifie les TDMA de tous les nœuds impactés (le voisinage à un et à deux sauts). Les voisins à un saut marqueront ce slot en réception. Formellement, si le slot  $t$  est alloué en transmission

au nœud  $i$ , nous obtenons  $slotAlloc[i,t] = i$ . Par conséquent,  $slotAlloc[j,t] = i$ , pour chaque nœud  $j$  dans le voisinage à un saut ou à deux sauts du nœud  $i$ .

La procédure *impactFreeNeighbors* effectue le processus inverse de la procédure *impactAllocNeighbors*. Pendant le processus d'allocation de slot pour le nœud  $i$ , la procédure libère le slot et modifie les TDMA de tous les nœuds impactés dans le voisinage à un saut et à deux sauts du nœud  $i$ . Formellement, si le slot  $t$  est alloué en transmission au nœud  $i$ , nous obtenons  $slotAlloc[i,t] = i$ . La suppression de l'allocation pour le nœud  $i$  est faite par  $slotAlloc[i,t] = 0$ . Par conséquent,  $slotAlloc[j,t] = 0$ , pour chaque nœud  $j$  dans le voisinage à un saut ou à deux sauts du nœud  $i$ .

### 6.5.2/ EXEMPLE

Nous illustrons sur un exemple l'espace de recherche parcouru par la méthode de *backtracking*.

Étant donnée la croissance exponentielle de la taille de l'espace de recherche en fonction du nombre de nœuds, nous considérons comme exemple un réseau très simple, composé de 4 nœuds (figure 6.3). Son degré maximal est égal à 2. Par conséquent, la longueur des ordonnancements TDMA testés  $l$  est égale à 3 ou à 4.

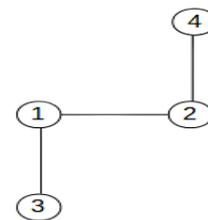


FIGURE 6.3 – Exemple d'un réseau linéaire composé de quatre nœuds

Nous illustrons, dans la figure 6.4, l'arbre des solutions d'allocation de nœuds sur les slots. Pour une raison de taille de la figure, nous ne présentons qu'un extrait de cet arbre. De même, nous n'illustrons pas les ordonnancements de tous les nœuds comme nous le faisons précédemment sous forme de matrice ; nous ne représentons que le slot de transmission de chaque nœud.

Par exemple, l'allocation à droite indique que le nœud 1 utilise le premier slot pour la transmission, le nœud 2 utilise le troisième slot pour la transmission et les nœuds 3 et 4 utilisent le quatrième slot pour la transmission. Le deuxième slot noté par - car il n'est utilisé par aucun nœud.

1	-	2	3
			4

Les solutions d'ordonnancement générées par l'algorithme d'ordonnancement optimal ainsi que les étapes intermédiaires sont entourées dans la figure 6.4. En effet, certaines branches de l'arbre représentant l'espace de recherche parcouru par l'algorithme ne sont pas explorées à cause des contraintes imposées par les liens de communication entre capteurs (donc pas d'arêtes dans le graphe : les voisins directs ou à deux sauts ne peuvent pas se voir alloués le même slot).

Pour les graphes donnés comme exemple, si le slot 1 a été alloué au nœud 1, le même slot ne peut être alloué au nœud 2. Toute possibilité d'allocation des slots pour les nœuds 3 et 4 n'est pas explorée.

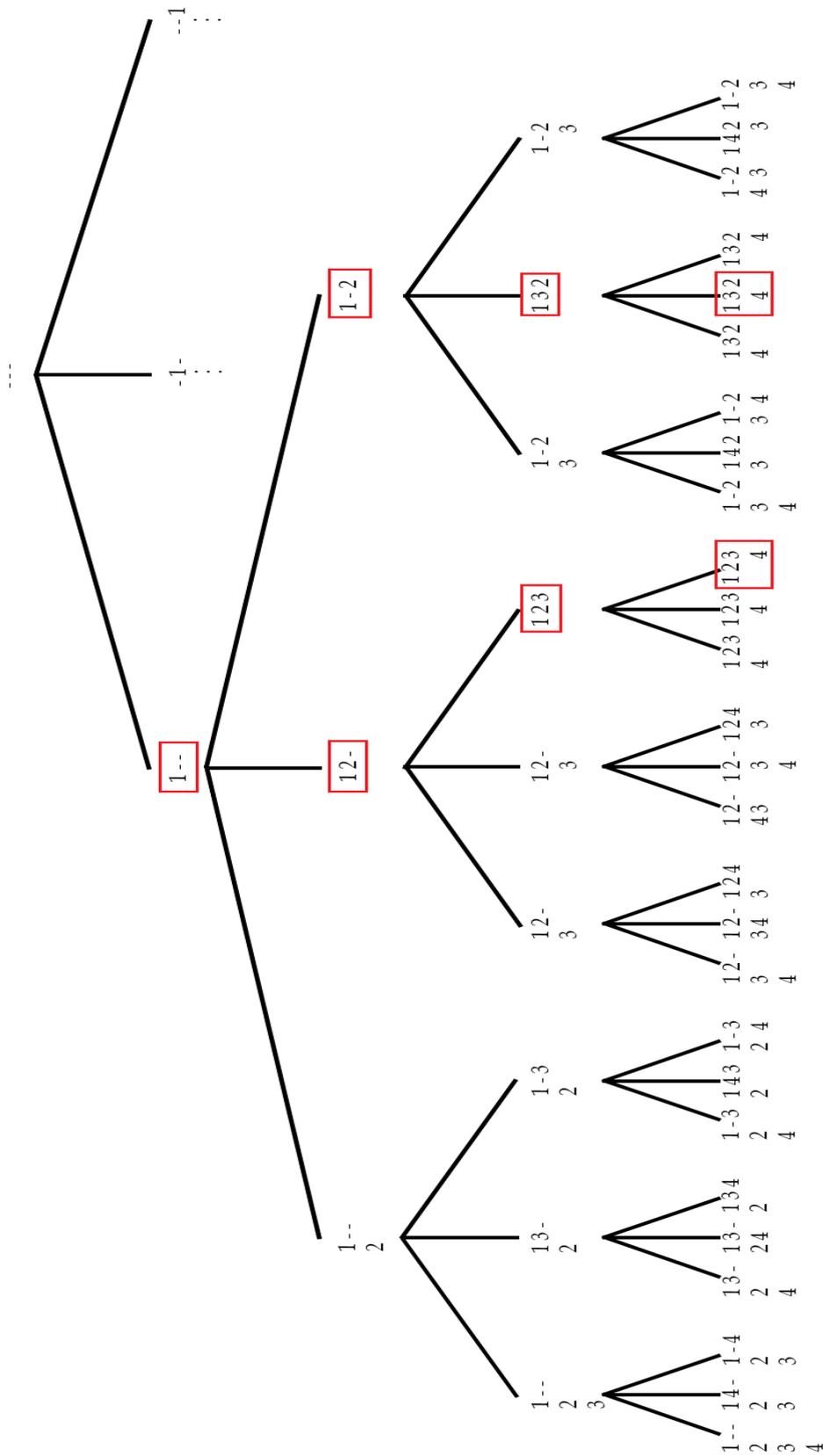


FIGURE 6.4 – Extrait de l'arbre des allocations possibles pour les nœuds de la figure 6.3

**Remarque.** L'algorithme de *backtracking* peut générer des ordonnancements dans lesquels un ou plusieurs slots ne sont utilisés par aucun nœud. C'est le cas dans l'exemple précédent quand  $l=4$ . Nous avons, par exemple, les ordonnancements suivants qui sont valides, évalués par rapport à la latence.

$$\begin{array}{cccc} 1 & 2 & 3 & - \\ & & 4 & \end{array} \qquad \begin{array}{cccc} 1 & - & 2 & 3 \\ & & & 4 \end{array}$$

Il est évident que la suppression du slot non utilisé permettrait de diminuer la latence engendrée mais l'algorithme ne peut pas considérer cette situation car l'obtention d'un ordonnancement se fait dans le cas d'arrêt de l'algorithme récursif.

### 6.5.3/ REMARQUES SUR LA COMPLEXITÉ

Nous souhaitons analyser la complexité temps de l'algorithme proposé pour l'obtention de l'ordonnancement optimal.

Soit  $n$  le nombre de nœuds du réseau et  $l$  la longueur du TDMA. Nous notons par  $C(nb, m)$  le nombre d'opérations nécessaires pour l'algorithme d'ordonnancement optimal afin d'allouer des slots pour  $nb$  nœuds parmi les  $m$  slots possibles.

Nous avons la relation de récurrence suivante :  $C(nb, m) \leq m * C(nb - 1, m) \forall nb \geq 1$ .

En effet, à chaque appel récursif, au maximum  $m$  possibilités différentes se présentent pour l'allocation d'un slot pour le nœud courant.

Au cas d'arrêt, quand tous les nœuds ont reçu un slot, l'évaluation de la latence est réalisée, en  $O(n)$ , car elle porte sur tous les nœuds du réseau, sauf le sink.

$C(0, m) = c \times (n - 1)$  où  $c$  est une constante.

Nous obtenons  $C(n, l) = l^n$ .

La longueur de TDMA est bornée par  $n$  (lorsque le graphe modélisant le réseau est complet). Par conséquent, la complexité de l'algorithme d'ordonnancement optimal est majorée par  $O(n^n)$ .

Cette majoration s'impose car la totalité des possibilités d'ordonnancement TDMA est traitée par l'algorithme seulement si toutes les feuilles de l'arbre représentant l'espace de recherche sont analysées.

Il faudrait, dans ce cas, qu'il n'y ait jamais d'incompatibilité de slots. Ce type de réseau ne présente aucun lien de communication. Ce n'est pas le cas des réseaux que nous traitons, qui sont tous connexes.

La complexité en moyenne de l'algorithme est difficile à calculer car elle dépend du nombre de branches à éliminer au cours du processus de *backtracking*.

L'analyse de la complexité qui suit considère que le générateur aléatoire utilisé pour obtenir la modélisation d'un réseau de capteurs construit, avec une certaine probabilité, tout graphe possible et connexe. C'est donc une analyse indépendante des caractéristiques des générateurs que nous avons implémentés.

Nous constatons qu'une génération aléatoire d'un graphe peut construire, avec une faible probabilité, un graphe complet. Dans ce cas, l'algorithme de calcul de l'ordonnancement optimal a une complexité factorielle en  $n$  :  $O(n!)$ . En effet, seule la longueur d'un ordonnancement égale à  $n$  permet de trouver des slots pour tous les nœuds. L'ensemble

d'ordonnements valides correspond à toutes les permutations de  $n$ .

De même, grâce au caractère aléatoire de génération d'un réseau, un graphe linéaire peut être construit, avec une certaine probabilité. Sans restreindre la généralité, un graphe de ce type est illustré dans la figure 6.5.



FIGURE 6.5 – Exemple d'un réseau linéaire de  $n$  nœuds

Dans ce cas, la complexité de l'algorithme de génération de l'ordonnement optimal a une complexité exponentielle en  $n$  :  $O(n^n)$ . Démontrons cette affirmation.

Considérons le cas d'une longueur d'ordonnement égale à  $n$ . Pour l'allocation de slot au nœud 1,  $n$  possibilités sont testées. Le nœud 2 ne peut pas avoir le même slot que le nœud 1 (car voisin direct). Il y aura  $n-1$  possibilités d'allocation pour ce nœud. Le nœud 3 ne peut pas se voir alloué le même slot que les nœuds 1 et 2 (qui sont à 2, respectivement 1 saut). Par conséquent,  $(n-2)!$  possibilités d'allocation existent pour ce nœud. Le même nombre de possibilités d'allocation se présente pour tous les autres nœuds, de 4 à  $n$ .

Au total, nous avons  $n(n-1)(n-2)^{n-2}$  solutions possibles d'ordonnement à évaluer par rapport à la latence. La complexité dans ce cas est  $O(n^n)$ .

Malgré le fait que la complexité dans le cas général ne puisse pas être calculée, nous obtenons  $\Omega(n!) \leq C(\text{findOptSlotAlloc}) \leq O(n^n)$ , où  $n$  est le nombre de nœuds du réseau.

Nous avons également exécuté l'algorithme pour quatre types de réseaux générés aléatoirement, comportant de 8 à 11 nœuds avec des densités  $\delta \in [4, n]$ . Les temps moyens d'exécution de l'algorithme d'ordonnement optimal pour ces réseaux croissent exponentiellement comme montré dans la table 6.1

Nombre de nœuds	Temps d'exécution moyen
8	16 s (sur 100 réseaux)
9	3 mn (sur 100 réseaux)
10	2 h (sur 5 réseaux)
11	8 h (sur 2 réseaux)

TABLE 6.1 – Temps nécessaires pour obtenir l'Ordonnement Optimal

## 6.6/ ÉVALUATION DES PERFORMANCES DES APPROCHES TDMA UTILISANT DES INFORMATIONS DU ROUTAGE

Pour évaluer les différentes approches, nous comparons leurs performances avec celles de CoLaNet et de Random TDMA (décrit en détail à la page 44). Random TDMA est un TDMA qui n'utilise aucune information de l'approche de routage et dans lequel les nœuds sont sélectionnés aléatoirement afin de leur attribuer un slot dans le TDMA tout en assurant l'absence des collisions.

Les paramètres de simulation sont décrits dans la section 4.4 (page 53) et nous avons considéré que tous les nœuds ont des données à transmettre au début du TDMA. Les

métriques évaluées sont la latence moyenne, la latence normalisée moyenne, la longueur du TDMA et le duty cycle. Les différentes approches sont évaluées en utilisant le même arbre de routage, par conséquent la consommation énergétique de chaque nœud reste invariée quelque soit l'approche d'ordonnancement. Ainsi, cette métrique n'est pas évaluée dans cette partie et est remplacée par le duty cycle qui reflète l'état des nœuds dans le TDMA.

Les résultats présentés dans cette partie concernent l'arbre de routage MinDegree et l'arbre de routage géographique. Les performances obtenues avec l'arbre de routage Hop Count sont similaires à celles obtenues avec l'arbre géographique ; pour cette raison, les résultats obtenus avec l'arbre Hop Count sont présentés dans l'annexe B (page 139).

### 6.6.1/ APPROCHES BASÉES SUR DES INFORMATIONS DE L'ARBRE DE ROUTAGE

La figure 6.6 montre que, indépendamment du type de l'arbre de routage, Random TDMA génère la plus grande latence car il n'utilise aucune information du routage, ainsi CoLaNet est meilleur que Random TDMA.

I-CoLaNet est meilleur que CoLaNet car ce dernier ne reflète pas la direction des communications dans le processus de construction de l'ordonnancement alors que I-CoLaNet le fait en coloriant les fils d'un nœud coloré au lieu de colorer tous ses voisins.

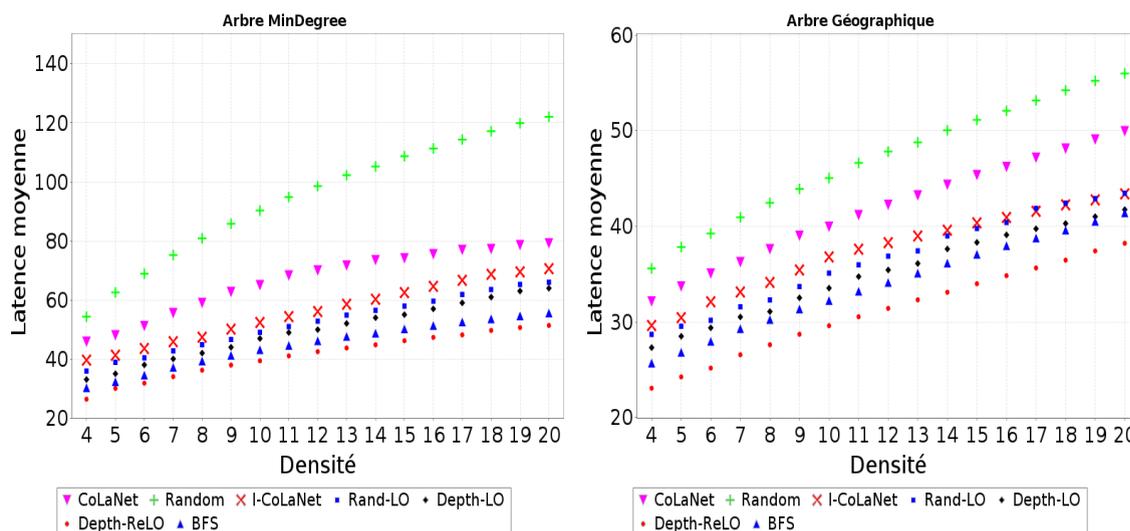


FIGURE 6.6 – Latence moyenne des approches basées sur des informations de l'arbre de routage MinDegree (à gauche) et de l'arbre géographique (à droite)

Les approches utilisant un ordonnancement des nœuds obtenu par un certain parcours de l'arbre de routage génèrent de meilleures latences que Random, CoLaNet et I-CoLaNet. Parmi ces approches, Rand-LO est en dernière position à cause de la sélection aléatoire des feuilles de l'arbre de routage. Ce défaut est corrigé avec Depth-LO et Depth-ReLO car ces derniers privilégient les nœuds qui sont éloignés du sink.

BFS utilise un parcours de haut en bas pour les nœuds de l'arbre de routage pendant l'allocation des slots et inverse le TDMA final ce qui lui permet d'être meilleur que Rand-LO et Depth-LO.

Depth-ReLO produit la meilleure latence car il permet aux nœuds internes de l'arbre de routage et qui se trouvent dans des chemins longs d'avoir les premiers slots du TDMA.

Les résultats présentés dans la figure 6.7 complètent les résultats précédents. Les deux méthodes d'allocation de slots, ascendante et descendante, permettent de rapprocher le slot d'un nœud des slots de ses fils dans l'arbre de routage ce qui minimise la latence normalisée. Par conséquent, Rand-LO, Depth-LO, Depth-ReLO et BFS sont meilleurs que Random, CoLaNet et I-CoLaNet.

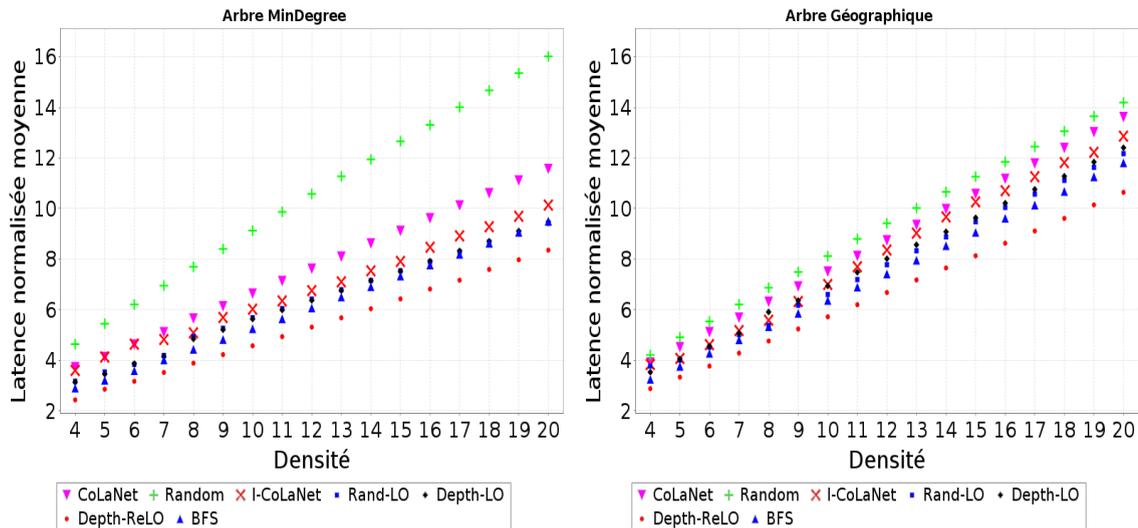


FIGURE 6.7 – Latence normalisée moyenne des approches basées sur des informations de l'arbre de routage MinDegree (à gauche) et de l'arbre géographique (à droite)

Depth-ReLO reste le meilleur car il privilégie les nœuds internes se trouvant dans des chemins longs.

Comme présenté dans la figure 6.8, CoLaNet et I-CoLaNet ont la meilleure longueur de TDMA grâce à l'algorithme de coloriage utilisé. Les autres approches ont des TDMA plus longs de seulement un ou deux slots mais qui sont utilisés pour minimiser la latence des communications.

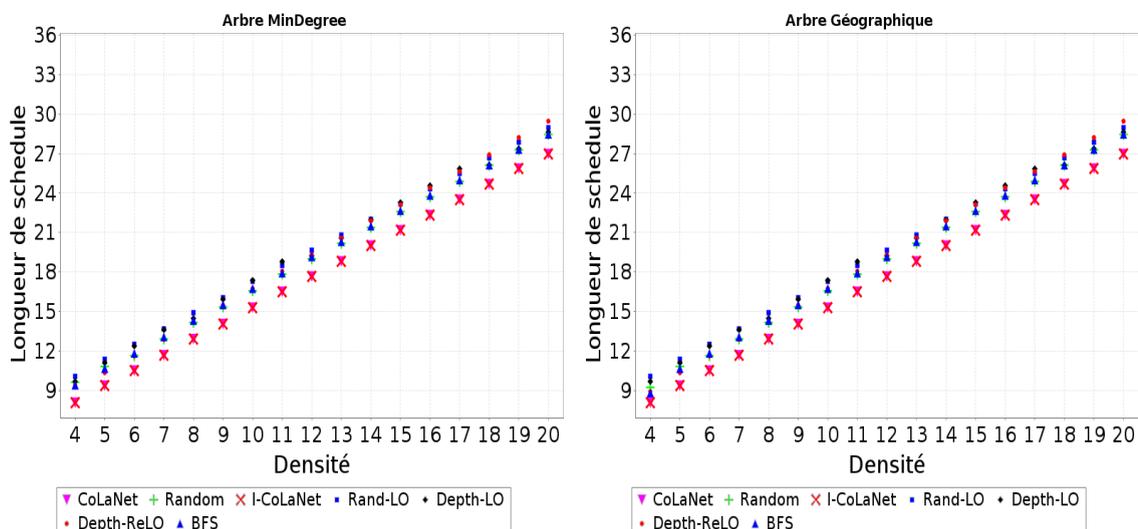


FIGURE 6.8 – Longueur de TDMA des approches basées sur des informations de l'arbre de routage MinDegree (à gauche) et de l'arbre géographique (à droite)

### 6.6.2/ HEURISTIQUES BASÉES SUR DES INFORMATIONS DE L'ARBRE DE ROUTAGE ET DU GRAPHE DU RÉSEAU

La figure 6.9 montre que les approches MAC utilisant des informations de l'arbre de routage et du graphe ont une meilleure latence que CoLaNet et ce indépendamment du type de l'arbre de routage.

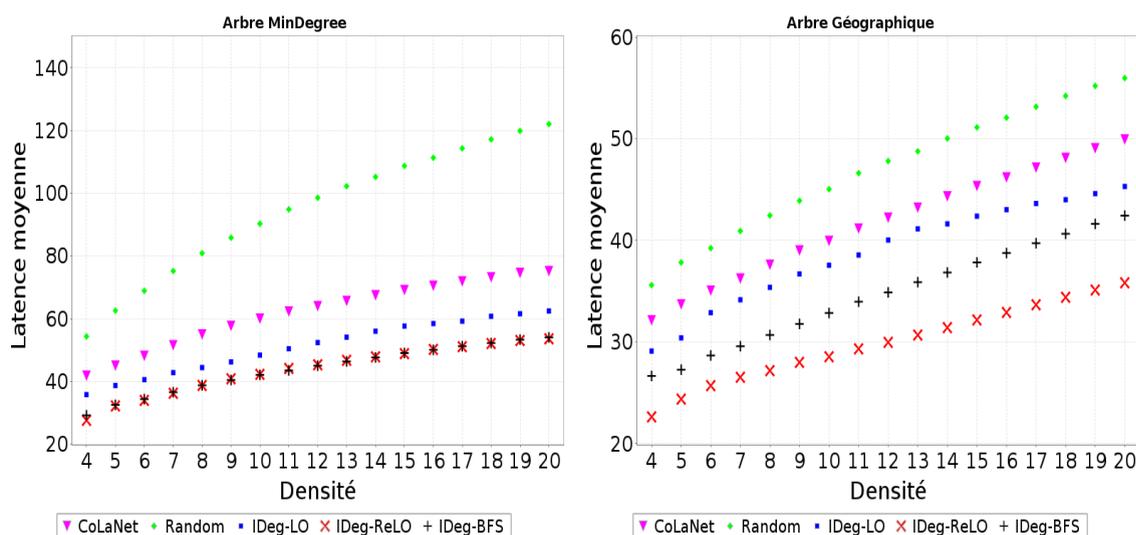


FIGURE 6.9 – Latence moyenne des approches basées sur des informations du graphe et de l'arbre de routage MinDegree (à gauche) et de l'arbre géographique (à droite)

IDeg-LO privilégie les feuilles de l'arbre de routage présentant beaucoup d'interférences alors que IDeg-ReLO privilégie les nœuds internes présentant beaucoup d'interférences.

En utilisant l'arbre MinDegree, IDeg-ReLO et IDeg-BFS ont les mêmes performances, alors qu'avec les arbres géographique et Hop Count, IDeg-ReLO est le meilleur. Ceci est dû au fait que les arbres géographique et Hop Count sont plus équilibrés que l'arbre MinDegree. Ce dernier contient des chemins (reliant le sink à une feuille dans l'arbre de routage) de longueurs différentes alors que les deux autres arbres contiennent des chemins quasiment de la même longueur. Le parcours de bas en haut des arbres géographique et Hop Count privilégie les nœuds qui sont éloignés du sink et génère une meilleure latence.

Les méthodes d'allocation de slots descendante et ascendante permettent de rapprocher un nœud de ses fils dans le TDMA ce qui génère une meilleure latence normalisée, pour nos approches, que celle générée par Random et CoLaNet comme montré dans la figure 6.10.

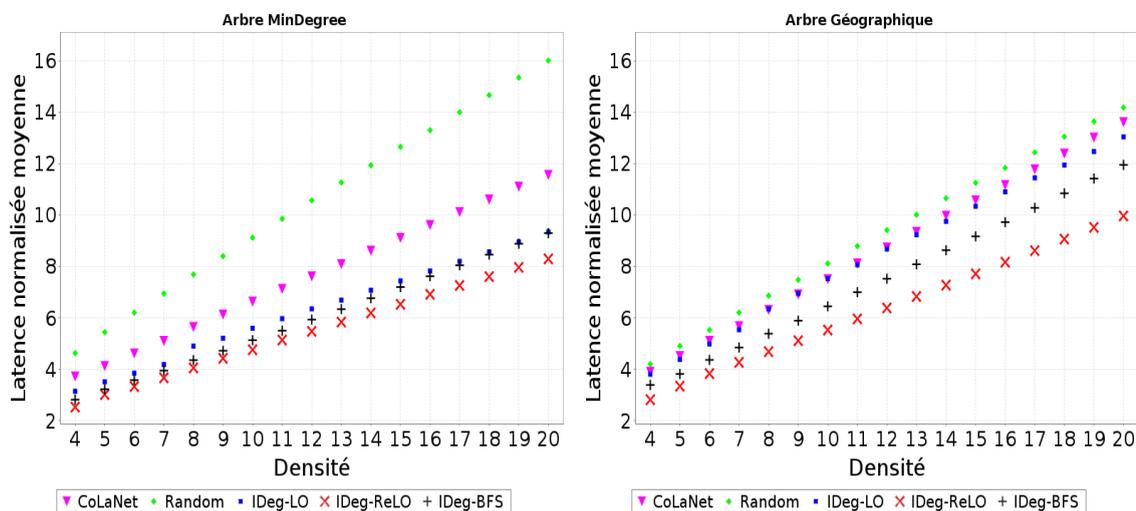


FIGURE 6.10 – Latence normalisée moyenne des approches basées sur des informations du graphe et de l'arbre MinDegree (à gauche) et de l'arbre géographique (à droite)

La figure 6.11 montre que CoLaNet a la meilleure longueur de TDMA grâce à son algorithme de coloriage, les autres approches ont des TDMA plus longs de un ou deux slots utilisés dans l'optimisation de la latence.

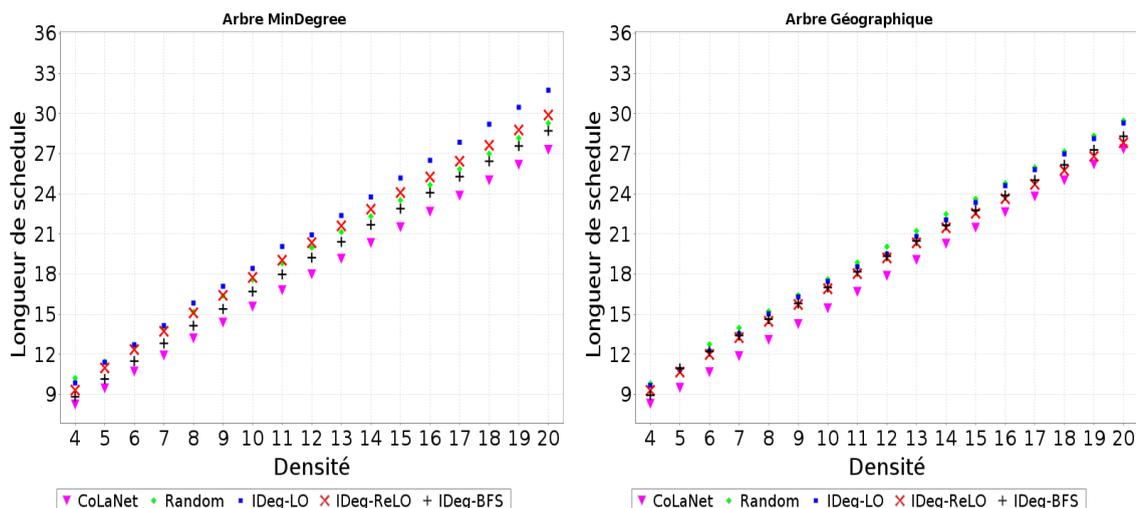


FIGURE 6.11 – Longueur de TDMA des approches basées sur des informations du graphe et de l'arbre MinDegree (à gauche) et de l'arbre géographique (à droite)

### 6.6.3/ MÉTHODE ASCENDANTE D'ALLOCATION DE SLOTS VS MÉTHODE DESCENDANTE D'ALLOCATION DE SLOTS

Pour comprendre la différence entre les performances des approches utilisant les méthodes d'allocation de slots ascendante et descendante, nous analysons la distribution des nœuds sur les slots des TDMA obtenus. Cette évaluation permettra une meilleure comparaison d'une part entre Depth-ReLO et BFS pour les approches utilisant des informations de l'arbre de routage, et d'autre part entre IDeg-ReLO et IDeg-BFS pour les approches utilisant des informations du graphe et de l'arbre de routage.

Soit  $k_i$  le nombre de nœuds alloués dans le slot  $i$ ,  $\forall i, 1 \leq i \leq l$  ( $l$  est la longueur du TDMA).

Nous avons  $\sum_{i=1}^l k_i = n$ , où  $n$  est le nombre total des nœuds dans le réseau.

Pour Depth-ReLO et IDeg-ReLO, nous avons constaté que les nœuds sont distribués de manière quasi décroissante sur les slots du TDMA. Par la suite, nous supposons que

$$0 \leq k_i - k_{i+1} = \Delta k_i, \forall i, 1 \leq i < l$$

Nous considérons que la différence entre les  $k_i$  consécutifs est égale, en moyenne, à  $\Delta k$ . Par conséquent, l'équation générale de la distribution des nœuds est :

$$k_i = k_1 - (i - 1) * \Delta k, \quad 1 \leq i \leq l \quad (6.1)$$

D'autre part, nous avons constaté que BFS et IDeg-BFS ont une distribution différente, les nœuds sont distribués de façon quasi croissante sur les slots des TDMA.

$$-\Delta k_i = k_i - k_{i+1} \leq 0, \forall i, 1 \leq i < l$$

L'équation de cette distribution est la suivante :

$$k_i = k_1 + (i - 1) * \Delta k, \quad 1 \leq i \leq l \quad (6.2)$$

où  $\Delta k$  est la moyenne de  $\Delta k_i$ .

Les figures 6.12 et 6.13 présentent, d'une part, la courbe théorique pour chaque distribution, et d'autre part, des distributions de nœuds sur les slots (calculées à partir de trois ordonnancements TDMA issus respectivement, des approches Depth-ReLO, BFS, IDeg-ReLO et IDeg-BFS). Les courbes théoriques des parties gauches correspondent à l'équation 6.1 et les courbes théoriques des parties droites correspondent à l'équation 6.2.

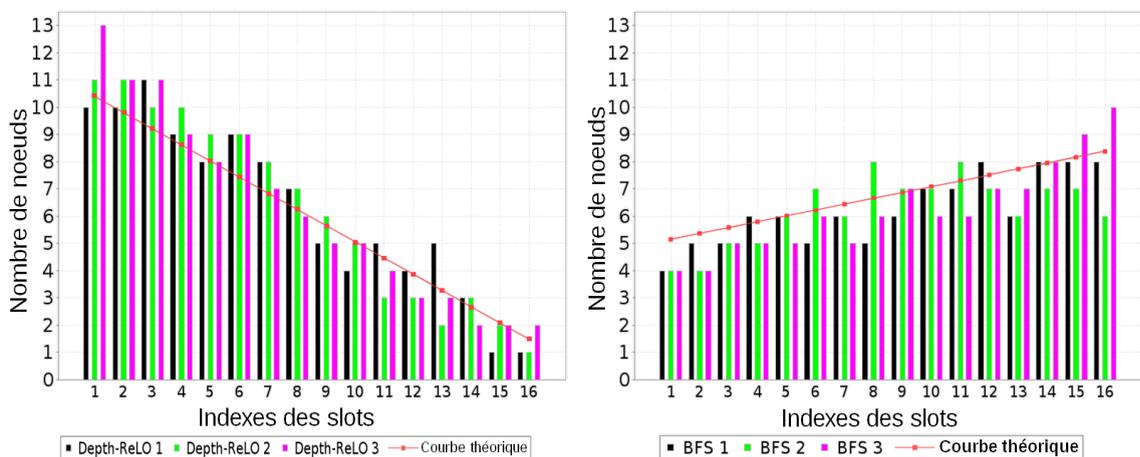


FIGURE 6.12 – Distributions des nœuds sur les slots pour Depth-ReLO (à gauche) et BFS (à droite)

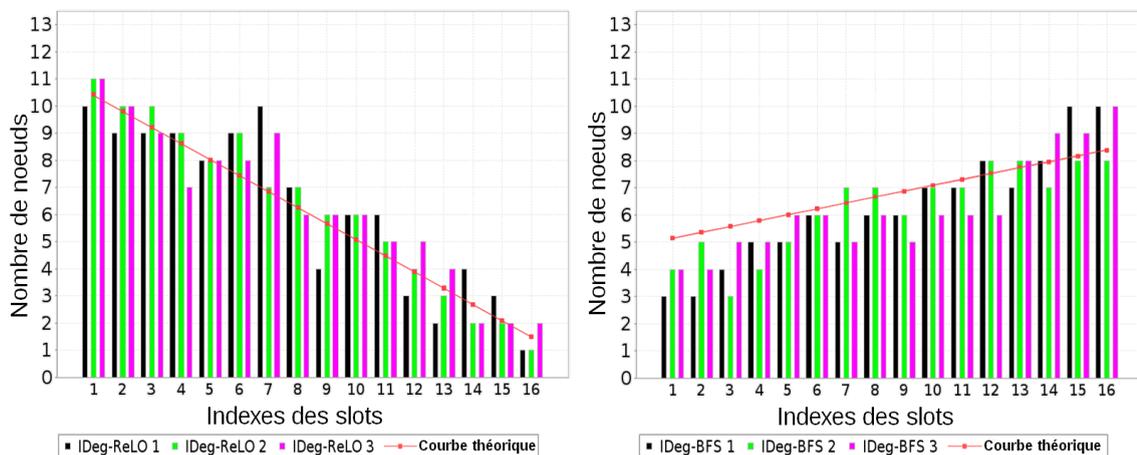


FIGURE 6.13 – Distributions des nœuds sur les slots pour IDeg-ReLO (à gauche) et IDeg-BFS (à droite)

Dans les parties gauches des figures, IDeg-ReLO et Depth-ReLO ont une distribution quasi décroissante parce qu'ils commencent l'allocation des slots par les feuilles de l'arbre de routage. Les feuilles sont, généralement, dispersées dans le réseau, loin du sink et donc elles présentent moins d'interférences ; par conséquent, la plupart d'entre elles peuvent utiliser les premiers slots du TDMA et la latence générée est inférieure à celle générée par BFS et IDeg-BFS. Ces deux derniers commencent l'allocation par les nœuds de la partie haute de l'arbre qui sont généralement proches du sink et présentent plus d'interférences et par conséquent, ils ne peuvent pas utiliser les mêmes slots.

Cette analyse de la distribution des nœuds sur les slots montre la raison pour laquelle la méthode ascendante permet de générer une meilleure latence que la méthode descendante. C'est l'argument qui explique les résultats des sections 6.6.1 et 6.6.2 où Depth-ReLO présentait une meilleure latence que BFS pour la première classe d'approches et IDeg-ReLO qui présentait une meilleure latence que IDeg-BFS pour la deuxième classe d'approches.

#### 6.6.4/ SYNTHÈSE DES SIMULATIONS

Les résultats précédents ont montré que Depth-ReLO est la meilleure approche TDMA parmi celles utilisant des informations de l'arbre de routage et que IDeg-ReLO est la meilleure approche TDMA parmi celles utilisant des informations de l'arbre de routage et du graphe modélisant le réseau.

La figure 6.14 montre qu'en utilisant l'arbre MinDegree, qui n'est pas équilibré car il contient des chemins (reliant le sink aux feuilles de l'arbre de routage) de longueurs différentes, Depth-ReLO a une meilleure performance parce qu'il alloue les premiers slots du TDMA aux nœuds qui se trouvent dans des chemins longs. Par contre, avec les arbres géographique et Hop Count, IDeg-ReLO est le meilleur car il privilégie les nœuds critiques présentant beaucoup d'interférences dans le réseau.

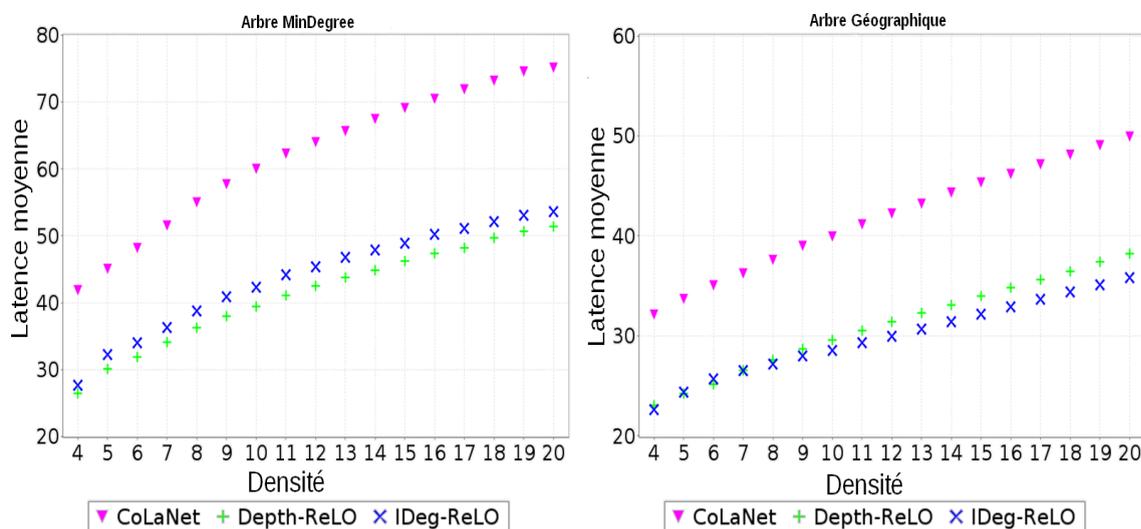


FIGURE 6.14 – Synthèse des résultats de la latence moyenne avec l'arbre MinDegree (à gauche) et avec l'arbre géographique (à droite)

La table 6.2 résume les gains en latence obtenus par IDeg-ReLO et Depth-ReLO par rapport à CoLaNet.

Arbre de routage \ Heuristique	IDeg-ReLO	Depth-ReLO
MinDegree	29.87%	33.74%
Géographique	29.49%	27.73%
Hop count	29.11%	27.75%

TABLE 6.2 – Synthèse des gains en latence moyenne par rapport à CoLaNet

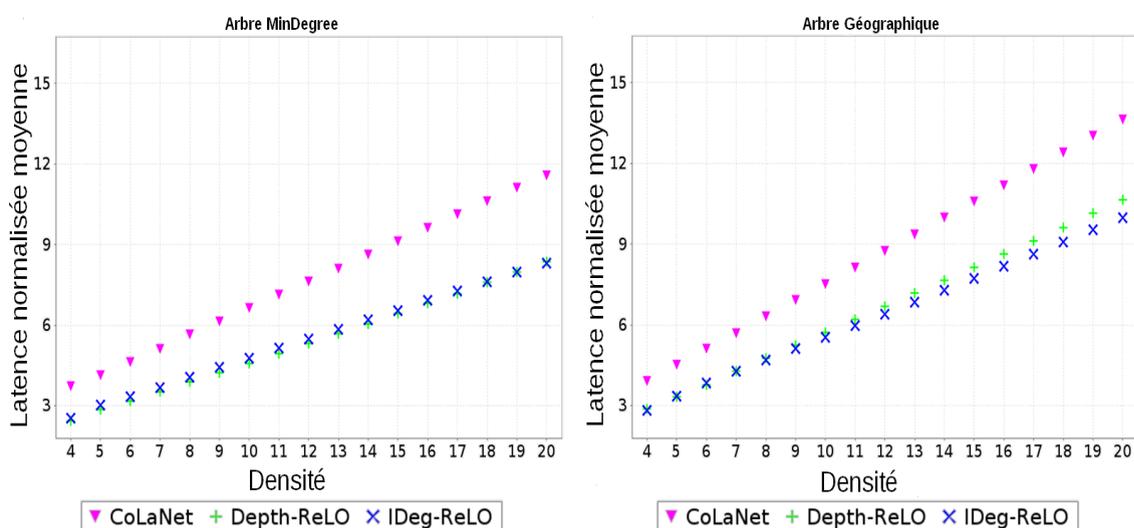


FIGURE 6.15 – Synthèse des résultats de la latence normalisée moyenne avec l'arbre MinDegree (à gauche) et avec l'arbre géographique (à droite)

Depth-ReLO et IDeg-ReLO génèrent presque la même latence normalisée grâce à la méthode ascendante d'allocation de slots, comme montré dans la figure 6.15.

CoLaNet a des longueurs de TDMA inférieures à celles de Depth-ReLO et IDeg-ReLO (figure 6.16) grâce à son algorithme de coloriage. Depth-ReLO et IDeg-ReLO ont des TDMA plus longs de un ou deux slots, mais l'ajout de ces slots supplémentaires est une conséquence de l'objectif de minimisation de la latence.

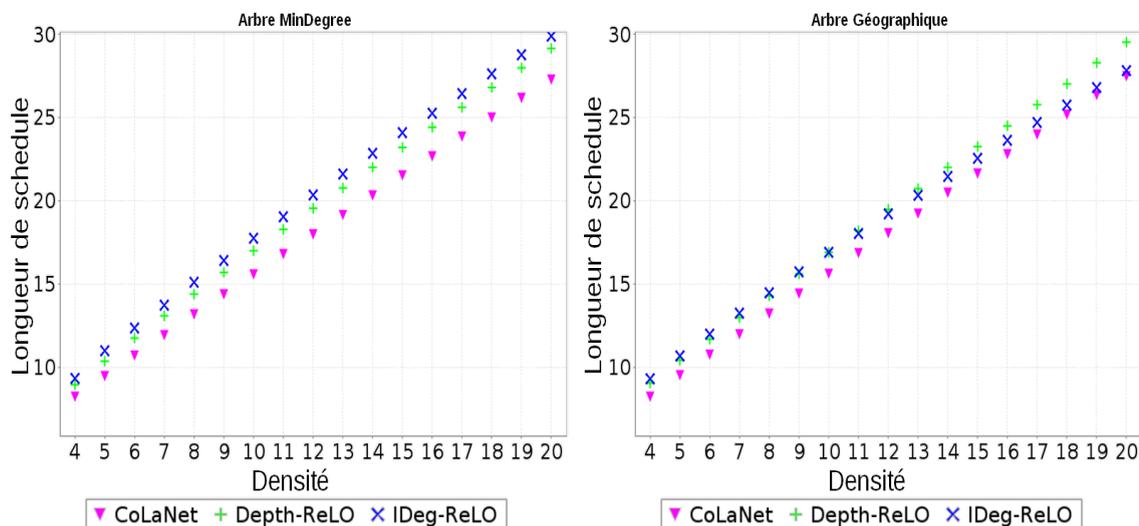


FIGURE 6.16 – Synthèse des résultats de la longueur de TDMA avec l'arbre MinDegree (à gauche) et avec l'arbre géographique (à droite)

La table 6.3 résume les gains de IDeg-ReLO et Depth-ReLO, en terme de duty cycle, par rapport à CoLaNet. Les deux approches ont un meilleur duty cycle ; par conséquent, ils consomment moins d'énergie que CoLaNet.

Arbre de routage \ Heuristique	IDeg-ReLO	Depth-ReLO
MinDegree	11.5%	9.6%
Géographique	10.92%	11.54%
Hop count	7.69%	11.5%

TABLE 6.3 – Synthèse des gains en duty cycle par rapport à CoLaNet

La table 6.4 présente les coefficients de variation CV (voir description à la page 55) des trois métriques (latence moyenne, latence normalisée moyenne et longueur de l'ordonnancement TDMA), toutes densités confondues, en considérant l'arbre de routage MinDegree, pour les approches Random TDMA, CoLaNet, Depth-ReLO et IDeg-ReLO.

[CVmin, CVmax]	latence moyenne	latence normalisée moyenne	longueur du TDMA
Random	[8.9 ; 11.2]%	[5.8 ; 7.2]%	[3.1 ; 5.2]%
CoLaNet	[8.8 ; 9.5 ]%	[5.5 ; 7.6]%	[3.6 ; 4.8]%
Depth-ReLO	[10.3 ; 11.0]%	[7.6 ; 9.8]%	[3.3 ; 5.9]%
IDeg-ReLO	[9.4 ; 11.9 ]%	[7.7 ; 9.4]%	[3.6 ; 5.4]%

TABLE 6.4 – Les intervalles des coefficients de variation des approches d’ordonnement TDMA utilisant des informations de l’arbre de routage MinDegree, toutes densités confondues

Les coefficients obtenus sont inférieurs à 11.9% ; par conséquent, les moyennes reflètent bien la tendance des valeurs.

### 6.6.5/ ÉVALUATION DES PERFORMANCES DES APPROCHES PAR RAPPORT À L’ORDONNANCEMENT OPTIMAL

L’Ordonnement Optimal énumère tous les TDMA possibles et valides et retourne celui qui génère la meilleure latence. Nous évaluons les performances de nos approches par rapport à cet algorithme en utilisant les mêmes métriques décrites précédemment.

À cause des temps d’exécution importants générés par l’algorithme d’ordonnement optimal (voir table 6.1), les résultats suivants ont été obtenus en utilisant des réseaux comportant seulement 8 nœuds.

La figure 6.17 montre que les performances des deux approches Depth-ReLO et IDeg-ReLO, en terme de latence moyenne, sont proches de l’Optimal et que IDeg-ReLO reste meilleur que Depth-ReLO.

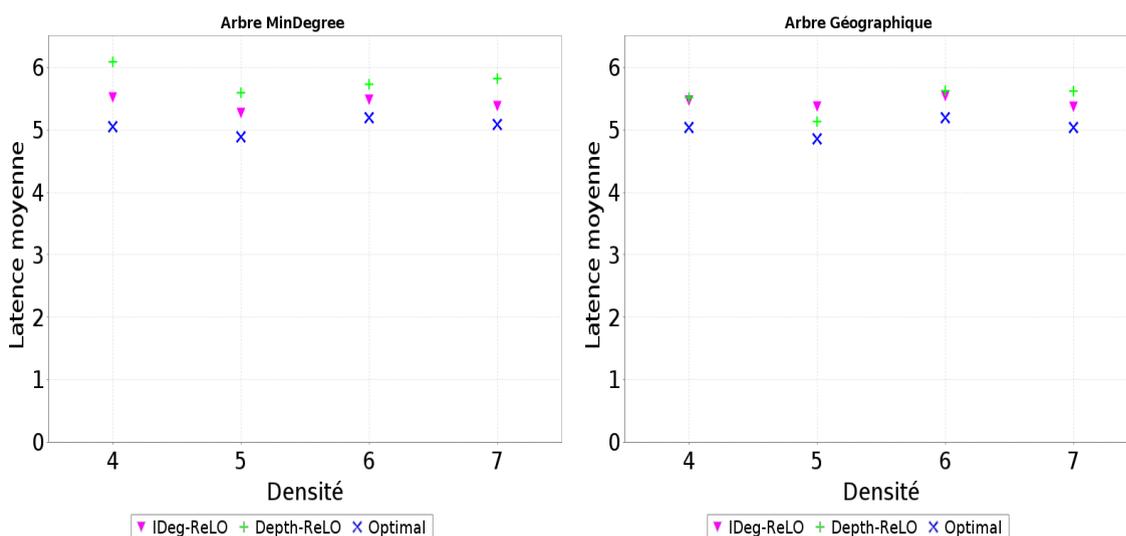


FIGURE 6.17 – Latence moyenne de IDeg-ReLO et Depth-ReLO vs l’algorithme d’ordonnement optimal avec l’arbre MinDegree (à gauche) et avec l’arbre géographique (à droite)

La table 6.5 résume les pertes des approches IDeg-ReLO et Depth-ReLO par rapport aux

performances de l'algorithme d'ordonnement optimal concernant la latence moyenne.

Arbre de routage \ Heuristique	IDeg-ReLO	Depth-ReLO
MinDegree	8.13%	17.76%
Géographique	6.02%	6.35%
Hop count	6.31%	8.78%

TABLE 6.5 – Synthèse des pertes en latence moyenne par rapport à l'Optimal

De même, pour la latence normalisée moyenne, la figure 6.18 montre que IDeg-ReLO est le plus proche de l'Optimal suivi par Depth-ReLO.

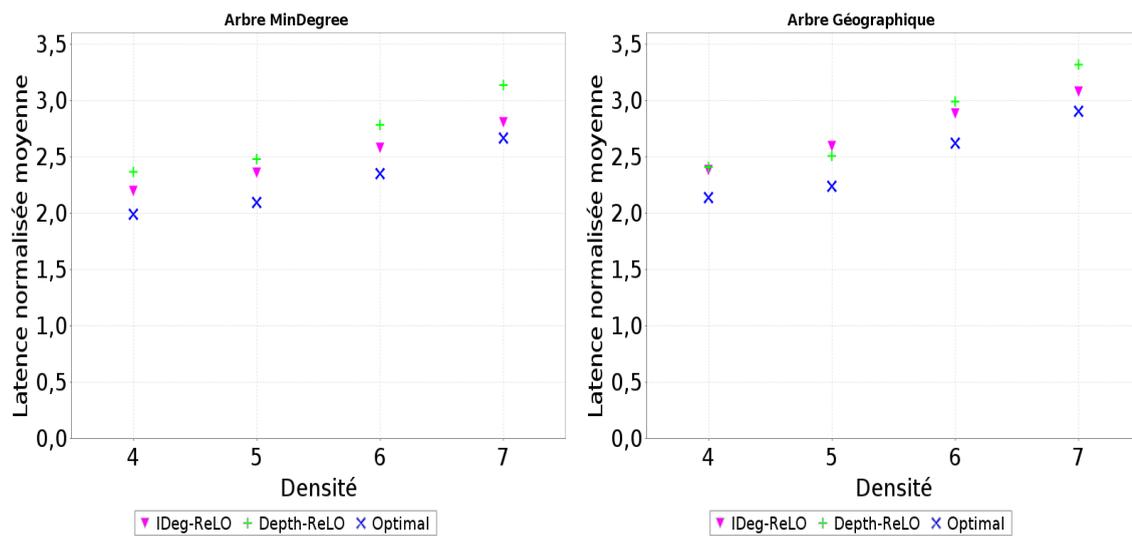


FIGURE 6.18 – Latence normalisée moyenne de IDeg-ReLO et Depth-ReLO vs l'algorithme d'ordonnement optimal avec l'arbre MinDegree (à gauche) et avec l'arbre géographique (à droite)

## 6.7/ CONCLUSION

Dans ce chapitre, nous avons proposé différentes approches cross-layer définissant un algorithme d'ordonnement MAC sans contention basé sur des informations du routage. Nous avons distingué deux classes principales d'approches : la classe des TDMA utilisant des informations de l'arbre de routage, et la classe des TDMA utilisant des informations du graphe modélisant le réseau et de l'arbre de routage.

Un algorithme d'Ordonnement Optimal est aussi donné, qui parcourt tous les TDMA possibles afin de retourner celui ayant la meilleure latence. Cet algorithme est utilisé pour l'évaluation de nos heuristiques afin de connaître l'ampleur entre les latences de nos approches et celles de l'Optimal.

Étant donnée la complexité exponentielle de l'algorithme d'ordonnement optimal, seules des petites instances de graphes ont pu servir dans ce comparatif. En revanche, les approches que nous avons proposées ont une complexité de  $O(n^2)$ , où  $n$  est le nombre de nœuds.

Les résultats des simulations ont montré que nos approches ont une meilleure latence que CoLaNet qui est aussi une approche MAC basée sur des informations du routage. En moyenne pour les résultats obtenus avec les trois arbres de routage, nos approches ont un gain de 29% en terme de latence moyenne de communication. Les performances de nos approches sont aussi proches, en moyenne de 8.89%, des performances de l'algorithme d'Ordonnancement Optimal pour des réseaux de petites tailles.

La figure 6.19 résume les propositions des approches MAC utilisant des informations de l'approche de routage présentées dans ce chapitre.

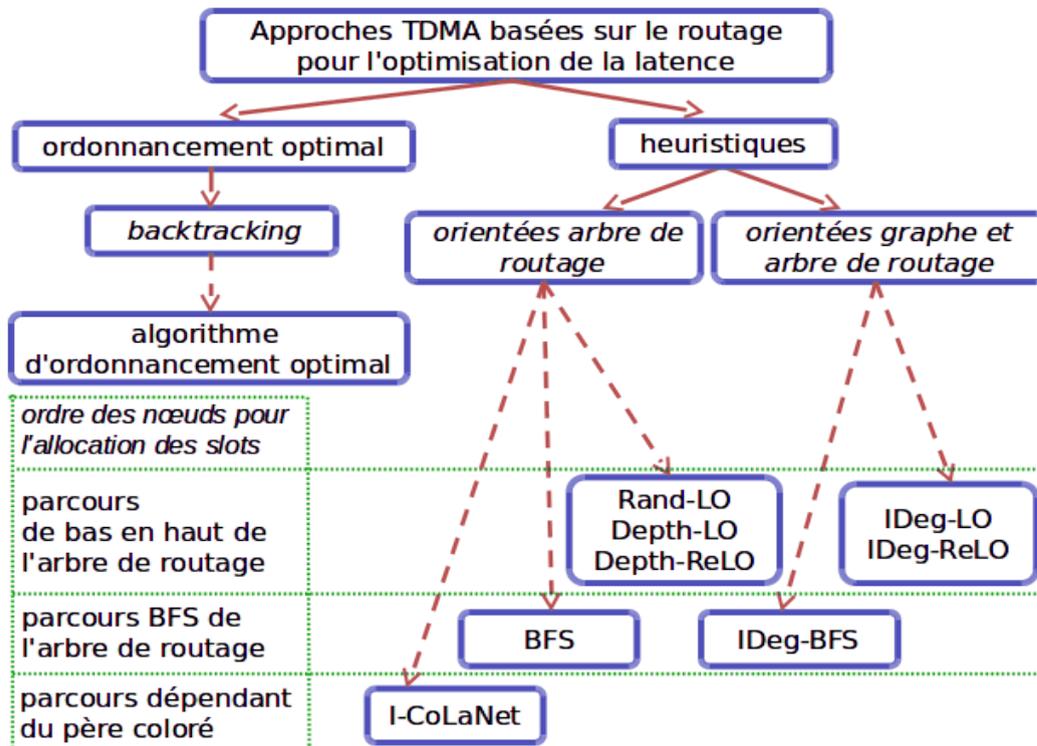


FIGURE 6.19 – Synthèse des approches TDMA proposées, utilisant des informations du routage

Les approches que nous avons proposées dans ce chapitre s'appuient sur des algorithmes de routage et sur des graphes modélisant des réseaux de topologie statique. Cependant, les réseaux présentent, au cours de leur fonctionnement, des topologies dynamiques. L'apparition d'un nouveau nœud du réseau (pour des raisons de pannes ou d'énergie faible) peuvent faire évoluer la topologie du réseau. L'impact de ces modifications sur nos algorithmes doit être analysé.

La disparition d'un nœud a le moins de conséquences si une optimisation n'est pas souhaitée. En effet, seulement si le nœud en extinction est le seul à utiliser son slot, ce dernier devient obsolète dans l'ordonnancement et peut être supprimé. Cette suppression provoque des messages en vague de mise à jour. Dans les autres cas, aucune action n'est exigée.

L'apparition d'un nœud nécessite de lui trouver un slot compatible. Cette action peut être réalisée en distribué, grâce à un échange de messages de contrôle avec son voisinage et à la connaissance de son père dans l'arbre de routage. Si l'allocation n'exige pas le rajout d'un nouveau slot, seuls les ordonnancements des voisins sont impactés. Dans le cas contraire, tous les nœuds doivent être informés de la nouvelle longueur du TDMA.



# APPROCHE COMBINÉE ROUTAGE ET MAC

## Sommaire

7.1	Motivation	103
7.2	IDeg-Routing&MAC	104
7.3	Évaluation des performances de IDeg-Routing&MAC	106
7.4	Longueur théorique de l'ordonnancement TDMA	108
7.5	Performances de IDeg-Routing&MAC vs MAR-WSN, IDeg-ReLO et Depth-ReLO	113
7.6	Conclusion	115

La fusion de deux approches de couches différentes dans le modèle en couches consiste à regrouper les informations et les fonctionnalités des deux approches en une seule nouvelle approche qui remplacera et assurera les rôles des deux anciennes approches [42].

Dans ce chapitre, nous présentons une approche cross-layer, appelée IDeg-Routing&MAC [81], qui combine une approche de routage et une approche MAC en une seule approche assurant les deux rôles. Le routage utilisé est un routage géographique dans lequel, lors de l'envoi des données, un nœud choisit parmi ses voisins celui qui le rapproche le plus du sink, en terme de distance géographique. Pour la partie MAC, un TDMA est établi en même temps que le routage en essayant de rapprocher les slots de transmission des nœuds des slots de transmission de leurs père. L'objectif est le même que dans les approches précédentes, à savoir minimiser la latence des communications.

Les approches des deux classes cross-layer, Routing-aware MAC et Joint MAC & Routing, construisent des ordonnancements TDMA. Nous avons pu constater, expérimentalement, une dépendance linéaire entre la longueur d'un ordonnancement TDMA et la densité du réseau. Dans ce chapitre, nous essayons de démontrer cette dépendance et de la chiffrer.

## 7.1/ MOTIVATION

Les deux chapitres précédents ont présenté des exemples de décisions non corrélées entre les couches MAC et Réseau et des solutions où chaque couche profite des informations de l'autre couche afin d'éliminer ces décisions non corrélées. Au lieu de concevoir, d'un côté, une approche de routage utilisant des informations du TDMA et, de l'autre côté,

un TDMA basé sur des informations du routage, nous proposons de les faire en même temps en combinant les deux couches MAC et Réseau en une seule nouvelle couche.

## 7.2/ IDEG-ROUTING&MAC

Le but de IDeg-Routing&MAC est de trouver, pour chaque nœud du réseau, un slot compatible dans le TDMA et un nœud voisin relais pour transmettre les données au sink. Pour cela, nous considérons les nœuds du réseau dans un ordre particulier donné par un parcours BFS (Breadth First Search) du graphe et la métrique de degré d'interférence (voir sa définition à la page 82). Le parcours BFS permet de trier les nœuds selon leur rapprochement du sink et le degré d'interférence permet de privilégier les nœuds présentant plus d'interférences dans le réseau.

1. Initialement, l'arbre de routage et le TDMA sont vides (la longueur initiale du TDMA est égale au degré maximal dans le graphe incrémenté).
2. Construire un arbre BFS en appliquant un parcours en largeur sur le graphe et en commençant par le sink.
3. Sélectionner la feuille, de l'arbre BFS, ayant le plus grand degré d'interférence dans le graphe.
4. Trouver pour cette feuille le voisin le plus proche du sink, en terme de distance, et le choisir comme père.
5. Ajouter le couple (feuille, prochain saut) à l'arbre de routage.
6. Vérifier si la feuille est déjà choisie comme prochain saut dans l'arbre de routage.
  - Si oui, chercher les slots de ses fils dans l'arbre et trouver le premier slot compatible dans le TDMA en commençant la recherche par le dernier slot de ses fils dans l'arbre de routage et en faisant une recherche circulaire.
  - Sinon, trouver un slot pour la feuille à partir du début du TDMA.
  - Dans les deux cas, si aucun slot compatible n'est trouvé, ajouter un slot à la fin du TDMA et l'allouer au nœud.
7. Supprimer la feuille de l'arbre BFS.
8. Répéter à partir de l'étape 3 jusqu'à ce que l'arbre BFS devienne vide.

À la fin de la procédure, un arbre de routage géographique et un TDMA sont obtenus.

La figure 7.1 présente un exemple d'un réseau de 8 capteurs (A à H) utilisant un TDMA.

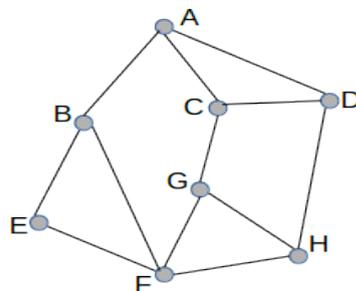


FIGURE 7.1 – Exemple d'un réseau de 8 capteurs

L'arbre BFS obtenu à partir de ce graphe est présenté dans la figure 7.2.

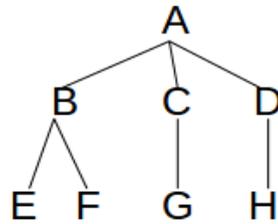


FIGURE 7.2 – Arbre BFS du graphe de la figure 7.1

Les degrés d'interférence des nœuds du graphe de la figure 7.1 sont les suivants : A-7, B-7, C-6, D-6, E-5, F-7, G-7, H-7.

Considérant l'arbre BFS et les degrés d'interférence des nœuds du graphe, l'ordre dans lequel les nœuds seront considérés pour trouver leurs slots et leurs prochains sauts est : H-G-F-D-C-E-B-A.

La table 7.1 et la figure 7.3 présentent, respectivement, le TDMA et l'arbre de routage obtenus en appliquant l'approche IDeg-Routing&MAC sur le réseau de la figure 7.1.

	slot1	slot2	slot3	slot4	slot5	slot6	slot7
node A	-	-	-	<i>D</i>	<i>C</i>	<i>B</i>	<b>A</b>
node B	-	-	<i>F</i>	<i>E</i>	-	<b>B</b>	<i>A</i>
node C	-	<i>G</i>	-	<i>D</i>	<b>C</b>	-	<i>A</i>
node D	<i>H</i>	-	-	<b>D</b>	<i>C</i>	-	<i>A</i>
node E	-	-	<i>F</i>	<b>E</b>	-	<i>B</i>	-
node F	<i>H</i>	<i>G</i>	<b>F</b>	<i>E</i>	-	<i>B</i>	-
node G	<i>H</i>	<b>G</b>	<i>F</i>	-	<i>C</i>	-	-
node H	<b>H</b>	<i>G</i>	<i>F</i>	<i>D</i>	-	-	-

TABLE 7.1 – Le TDMA obtenu en appliquant IDeg-Routing&MAC sur le réseau de la figure 7.1

Nous rappelons que pour chaque nœud, le slot en gras indique le slot de transmission et les slots en italique indiquent les slots de réception. Un tiret note le fait que le nœud peut passer en mode veille car il n'effectue ni transmission, ni réception.

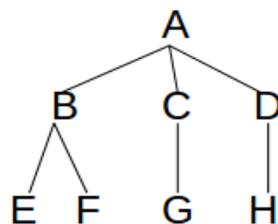


FIGURE 7.3 – Arbre de routage obtenu en appliquant IDeg-Routing&MAC sur le réseau de la figure 7.1

À cause de la simplicité de l'exemple, l'arbre BFS et l'arbre de routage sont identiques dans ce cas.

### 7.3/ ÉVALUATION DES PERFORMANCES DE IDEG-ROUTING&MAC

Les approches cross-layer combinant MAC et routage existant dans la littérature considèrent des approches MAC avec contention. Aucune approche ne propose la fusion du routage avec une approche MAC sans contention. La comparaison de notre approche avec une approche utilisant un type différent d'approche MAC ne semble pas faisable. Par conséquent, et puisque notre approche est la première approche de fusion d'une approche de routage avec un TDMA, nous procédons à son évaluation en la comparant avec une approche en couches contenant un routage géographique et un TDMA aléatoire qui n'utilise aucune information cross-layer.

Les paramètres de simulation sont décrits dans la section 4.4 (page 53) et nous avons considéré que tous les nœuds ont des données à transmettre au début du TDMA. Les métriques évaluées sont la latence moyenne, la latence normalisée moyenne, la longueur du TDMA et le duty cycle.

La figure 7.4 montre que IDeg-Routing&MAC génère une meilleure latence de communication que le routage et MAC indépendants, toutes densités confondues. Premièrement, parce que le parcours de l'arbre BFS permet d'ordonner les nœuds de façon pratique. Les nœuds proches du sink sont placés dans la partie haute de l'arbre BFS et les nœuds éloignés du sink sont placés dans la partie basse de l'arbre BFS. Ensuite, privilégier les nœuds ayant un degré d'interférence important permet de donner la priorité aux nœuds qui présentent le plus de contraintes durant les communications.

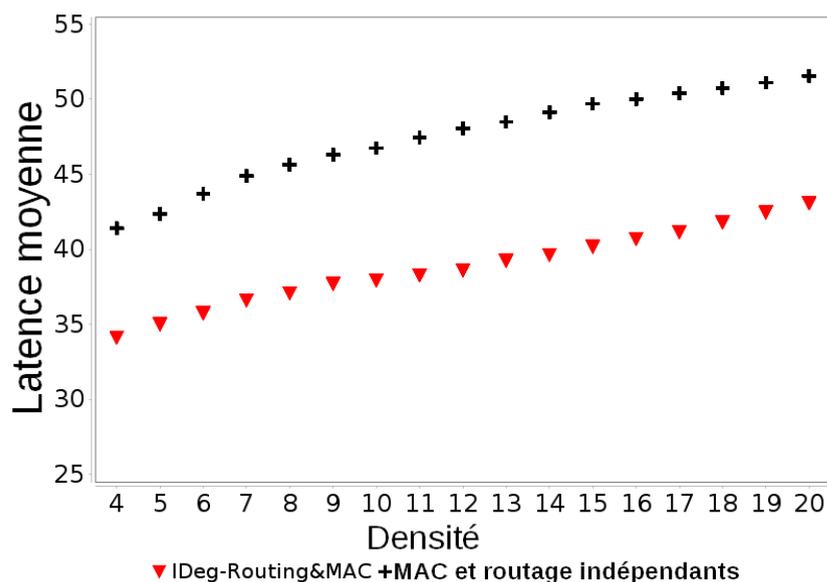


FIGURE 7.4 – Latence moyenne de IDeg-Routing&MAC vs MAC et routage indépendants

La figure 7.5 complète les résultats précédents ; IDeg-Routing&MAC génère une meilleure latence normalisée car il essaye de rapprocher le slot d'un nœud des slots

de ses fils.

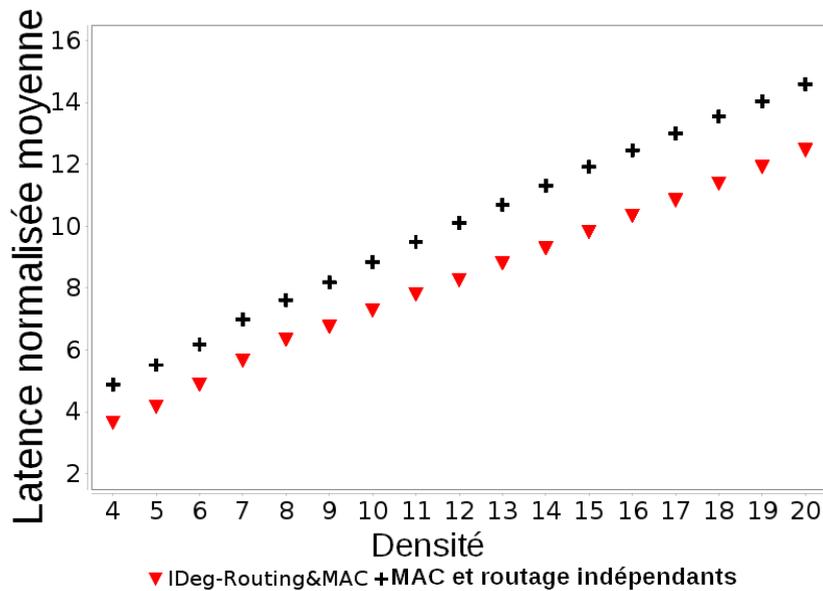


FIGURE 7.5 – Latence normalisée moyenne de IDEG-Routing&MAC vs MAC et routage indépendants

Les augmentations progressives des latences sont causées par la croissance de la densité, car le nombre moyen de voisins, pour chaque noeud, augmente et donc l'ordonnement TDMA nécessite plus de slots et devient ainsi plus long.

La recherche circulaire d'un slot compatible dans l'ordonnement permet l'obtention de TDMA plus courts que les TDMA obtenus aléatoirement comme montré dans la figure 7.6.

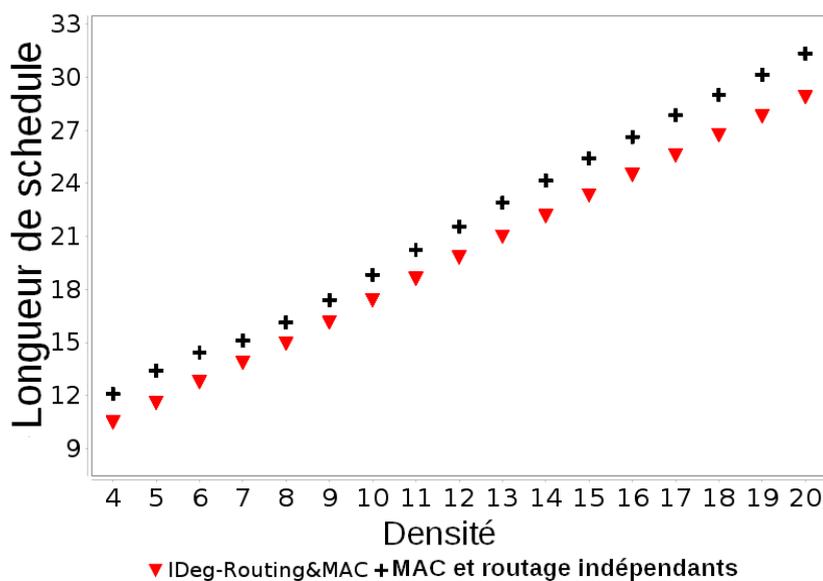


FIGURE 7.6 – Longueur de TDMA de IDEG-Routing&MAC vs MAC et routage indépendants

Un TDMA plus court entraîne un duty cycle plus important. Notre perte en duty cycle (figure 7.7) est à seulement 7% mais elle est utilisée dans et compensée par dans le gain en latence des communications.

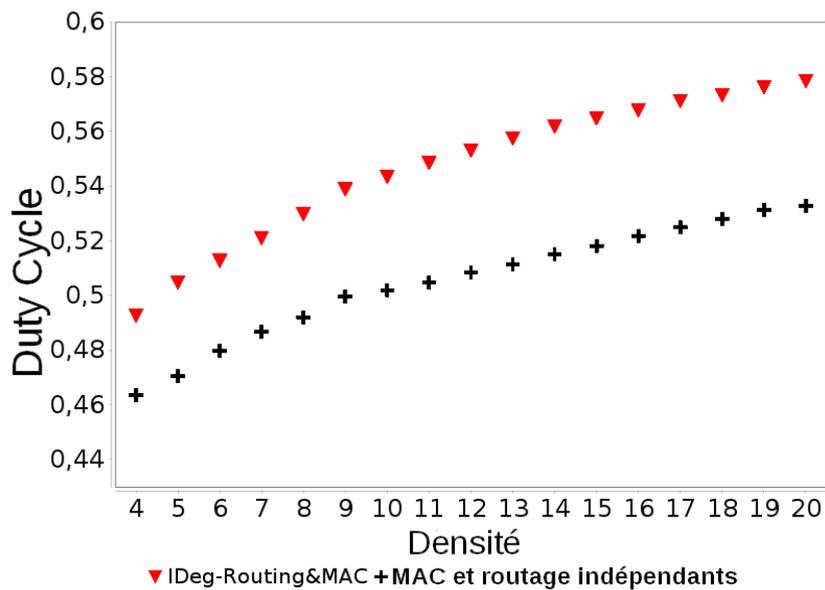


FIGURE 7.7 – Duty cycle de IDeg-Routing&MAC vs MAC et routage indépendants

La table 7.2 présente les coefficients de variation CV (voir description à la page 55) des trois métriques (latence moyenne, latence normalisée moyenne et longueur de l'ordonnancement TDMA), pour toutes densités confondues, pour IDeg-Routing&MAC et pour l'approche MAC et routage indépendants.

[CVmin, CVmax]	latence moyenne	latence normalisée moyenne	longueur du TDMA
IDeg-Routing&MAC	[9.2 ; 11.0]%	[7.6 ; 9.2]%	[4.2 ; 5.8]%
MAC et routage indépendants	[8.5 ; 10.3]%	[7.3 ; 9.8]%	[3.6 ; 5.3]%

TABLE 7.2 – Les intervalles des coefficients de variation de IDeg-Routing&MAC et de l'approche MAC et routage indépendants pour toutes densités confondues

Les coefficients obtenus sont inférieurs à 11% ; par conséquent, les moyennes reflètent bien la tendance des valeurs.

## 7.4/ LONGUEUR THÉORIQUE DE L'ORDONNANCEMENT TDMA

Les différents TDMA obtenus avec les approches de fusion entre MAC et routage et les approches MAC utilisant des informations du routage, nous ont permis d'observer une dépendance linéaire entre les longueurs des TDMA et les densités des réseaux. Afin d'exprimer cette dépendance, nous analysons les collisions possibles entre les nœuds

ce qui permettra d'identifier le besoin en nombre de slots différents dans le processus d'allocation de slots.

La figure 7.8 présente une partie du réseau divisée en carrés de taille  $2r$ . Dans chaque carré, la portée des communications est représentée avec un cercle de rayon  $r$ . Ensuite, chaque carré est divisé en quatre parties. Nous considérons dans cette analyse, que chaque carré  $2r \times 2r$  contient  $\delta$  nœuds (représentés par des cercle gris dans la figure 7.8). C'est une hypothèse plus forte que celle donnée par le calcul de la densité  $\delta = \pi r^2 n / a^2$ , car cette dernière affirme qu'en moyenne un nœud a  $\delta-1$  voisins. Équivalent à l'affirmation que dans un cercle de rayon  $r$  inscrit dans un carré  $2r \times 2r$ , il y a en moyenne  $\delta$  nœuds. Les nœuds dans un carré  $2r \times 2r$  peuvent produire des collisions lorsque les communications sont générées simultanément. Certaines paires de nœuds, à distance strictement supérieure à  $2r$  (comme ceux placés dans des coins opposés d'un carré  $2r \times 2r$ ) ne génèrent pas de collisions mais nous ignorons dans l'analyse menée ces situations.

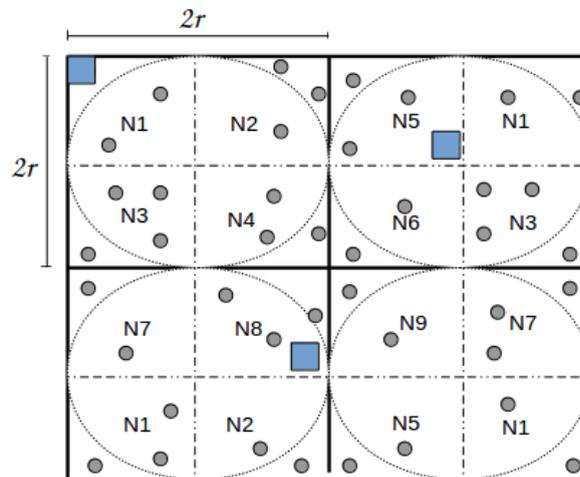


FIGURE 7.8 – Distribution des nœuds avec le modèle UDG pour l'allocation des slots

Selon le modèle UDG, les nœuds dans chaque carré  $r \times r$  ne peuvent pas utiliser le même slot de communication que ses voisins à un saut ou ses voisins à deux sauts. Par conséquent, ils ne peuvent utiliser les mêmes slots de communication que les nœuds dans les carrés adjacents et les nœuds dans les carrés adjacents aux derniers. Nous obtenons alors, neuf classes de slots notées de N1 à N9 dans la figure 7.8. Une classe est un ensemble de slots pouvant être utilisé par plusieurs nœuds en assurant l'absence des collisions. La contrainte précédente dans l'allocation des slots est trop forte. Par exemple, les nœuds carrés de la figure 7.8 peuvent utiliser les mêmes slots de la classe N1 (car les distances directes qui les séparent sont strictement supérieures à  $2r$ ).

Chaque classe de slots nécessite, au plus,  $\delta/4$  slots différents qui seront alloués aux nœuds se trouvant dans un carré  $r \times r$ ; par conséquent, la longueur du TDMA est :  $lg = 9 * \delta/4$ , ce qui génère un facteur de multiplication égal à 2.25 appliqué à la densité du réseau pour obtenir la longueur du TDMA.

Ce raisonnement peut être étendu sur tout le réseau et les neuf classes de slots peuvent être réutilisées pour l'allocation des slots aux nœuds restants dans le réseau (voir la figure 7.8).

Le facteur de multiplication précédent est une borne supérieure qui peut être diminuée

encore plus. Ce raisonnement est basé sur une distribution uniforme des nœuds sur la zone de déploiement. Quand la densité  $\delta$  croît, la probabilité d'avoir plus de nœuds dans chaque carré augmente aussi. Dans ce cas, les slots peuvent être réutilisés plus tôt, comme montré dans la figure 7.9. Dans la partie gauche de la figure, nous considérons des réseaux de densités  $8 \leq \delta < 16$ . Dans ce cas, chaque carré  $r \times r$  contient, en moyenne, un ou deux nœuds et peut être divisé en deux parties rectangulaires  $r \times r/2$ . Un maximum de 15 classes de slots différentes est alors nécessaire pour l'ordonnancement de tous les nœuds du réseau, et le facteur de multiplication devient, dans ce cas,  $lg/\delta = 15/8$ . Notons que dans cet exemple nous avons considéré une division verticale, et que l'approche horizontale est similaire.

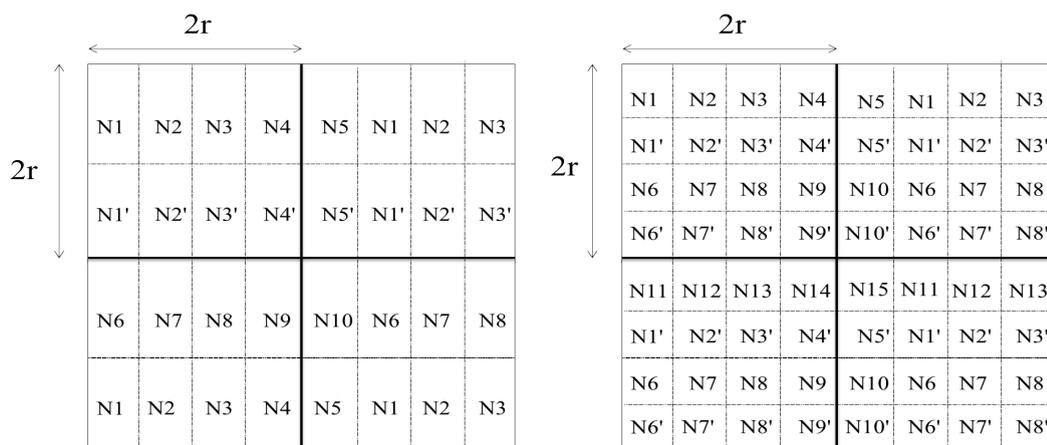


FIGURE 7.9 – Allocation des slots pour des réseaux de densités de 8 à 16 (à gauche) et supérieure à 16 (à droite)

Quand la densité dépasse 16, la division peut être faite sur les deux dimensions (verticalement et horizontalement) pour que chaque carré contienne, en moyenne, au moins un nœud. Dans ce cas, 25 classes de slots sont nécessaires pour l'ordonnancement de tous les nœuds du réseau (partie droite de la figure 7.9) et le facteur de multiplication est égal à  $lg/\delta = 25/16$ .

La borne supérieure estimée pour la longueur du TDMA est alors :

$$lg = \begin{cases} 9/4 * \delta & \text{si } 4 \leq \delta < 8 \\ 15/8 * \delta & \text{si } 8 \leq \delta < 16 \\ 25/16 * \delta & \text{si } 16 \leq \delta \leq 20 \end{cases} \quad (7.1)$$

La figure 7.10 présente la corrélation des bornes supérieures théoriques avec les résultats obtenus par simulation pour l'approche de fusion MAC et routage IDeg-Routing&MAC et pour l'approche MAC utilisant des informations de routage présentée dans le chapitre précédent, IDeg-ReLO. IDeg-ReLO est choisie parmi les différentes approches MAC décrites précédemment car elle présente les meilleures performances en termes de latence de communication.

Nous remarquons que pour les faibles densités,  $4 \leq \delta < 8$ , le facteur de multiplication 2.25 est le plus adéquat car les résultats théoriques sont cohérents avec les résultats expérimentaux. Pour des densités  $8 \leq \delta < 16$  et  $\delta \geq 16$ , respectivement, la longueur

théorique est cohérente avec les résultats de simulation au début de chaque groupe de densités ; ensuite, le calcul théorique surestime largement la longueur du TDMA à la fin de chaque groupe de densités. Ceci montre qu'il y a encore des nœuds qui peuvent réutiliser des slots déjà alloués et donc les facteurs de multiplication 15/8 et 25/16 peuvent être encore diminués.

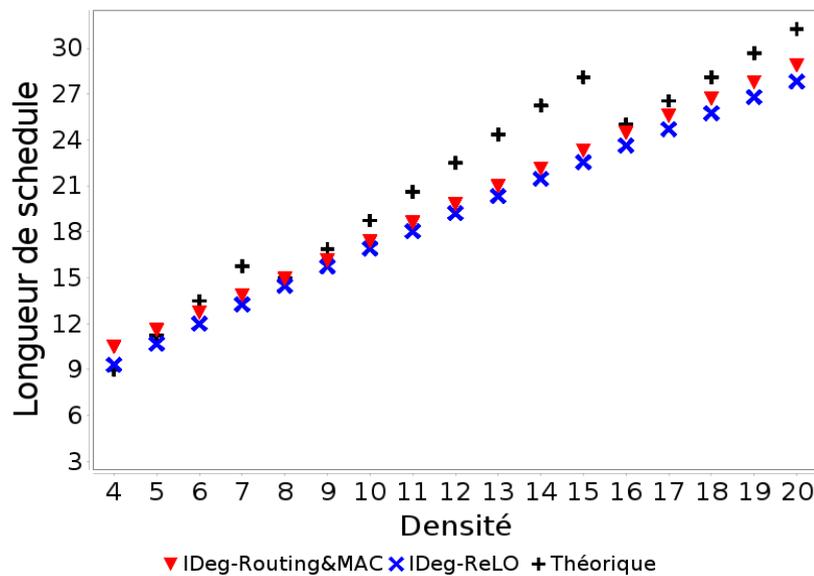


FIGURE 7.10 – Longueur de TDMA de IDeg-Routing&MAC et IDeg-ReLO vs longueur de TDMA théorique

Nous avons remarqué que le facteur de multiplication obtenu précédemment diminue lorsque la densité du réseau augmente. De manière plus générale et pour un réseau de densité quelconque, nous souhaitons trouver une borne inférieure pour ce facteur.

Il s'agit d'identifier le nombre de classes de slots nécessaires à l'ordonnancement TDMA, chaque classe désignant des slots ne pouvant être utilisés simultanément et chaque paire de deux classes n'ayant pas de slots en commun.

Pour une distribution de nœuds considérée uniforme, i.e. les carrés contiennent, approximativement, le même nombre de nœuds, nous essayons d'identifier les nœuds pouvant réutiliser des slots déjà alloués. Afin de réaliser cette analyse, nous appliquons une division virtuelle, itérative, de la superficie de déploiement en carrés de taille  $l_k \times l_k$ . Ce quadrillage sera affiné à chaque itération  $k$ .

Traitons une dimension (l'horizontale). Soit  $h$  (un nombre entier strictement positif) le nombre de classes à l'itération  $k$  nécessaires pour allouer des slots aux nœuds appartenant aux carrés, de 1 à  $h$ , de dimension  $l_k \times l_k$  (voir figure 7.11).

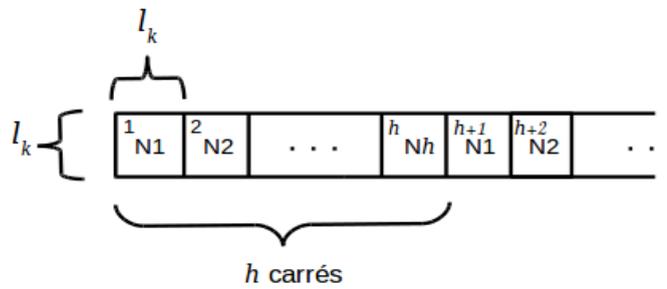


FIGURE 7.11 – Exemple de découpage en carrés et de classes de slots utilisées pour les nœuds des carrés à l'itération  $k$

Les nœuds du carré  $h + 1$  peuvent utiliser les mêmes slots que les nœuds du carré 1 (la classe de slots N1) à condition qu'ils soient distancés de plus de  $2 \times r$ . Nous obtenons donc la condition :

$$h \times l_k > 2r \text{ ou } h > \frac{2r}{l_k}$$

Étant donné que  $h$  est entier,  $h = \lceil \frac{2r}{l_k} \rceil + 1$ .

La même condition s'applique sur la dimension verticale. Par conséquent,  $h \times h$  classes de slots sont nécessaires pour l'ordonnement.

Pour commencer, à la première itération, la taille du carré est  $l_0 = 2r$ . En effet, les nœuds distribués dans un carré  $2r \times 2r$  ont besoin de slots différents, sauf pour ceux dans les coins opposés du carré, en dehors du cercle inscrit dans le carré (voir figure 7.12). Nous n'intégrons pas cette optimisation ici, pour traiter le cas général.

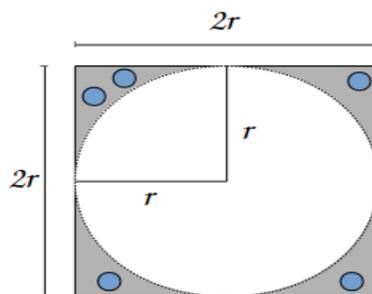


FIGURE 7.12 – Exemple de nœuds se trouvant en dehors du cercle mais inscrits dans le carré

À chaque itération, nous considérons que nous pouvons encore diviser chaque carré sur deux dimensions, en gardant au moins un nœud dans chaque nouveau carré obtenu. Par conséquent,  $l_{k+1} = l_k/2$ .

Les équations  $l_0 = 2r$  et  $l_{k+1} = l_k/2$  impliquent  $l_k = \frac{2r}{2^k}, \forall k \geq 0$ .

Nous avons donc

$$h = \lceil \frac{2r}{l_k} \rceil + 1 \implies h = 2^k + 1$$

En plus, le nombre de classes de slots est également conditionné par la présence d'au moins un nœud dans chaque carré. Nous notons par  $\delta_k$  la densité moyenne pour un carré de côté  $l_k$ .

Nous avons  $\delta_0 = \pi/4$ , car chaque carré  $2r \times 2r$  contient au moins un nœud, et l'ensemble total de nœuds est  $n$ . Nous avons  $(\frac{a}{l_0})^2$  carrés différents.

En effet,

$$\left(\frac{a}{l_0}\right)^2 \times 1 = n \implies \delta_0 = \frac{n\pi r^2}{a^2} = \frac{(\frac{a}{l_0})^2 \pi r^2}{(2r)^2} = \pi/4$$

Nous avons  $\delta_{k+1} = \delta_k \times 4$ , car à chaque itération, le carré  $l_k \times l_k$  est divisé en quatre sous-carrés  $l_{k+1} \times l_{k+1}$ , qui doivent à leur tour, chacun, contenir au moins 1 nœud. Nous obtenons  $\delta_k = \delta_0 \times 4^k, \forall k \geq 0$ .

Le rapport de la longueur de l'ordonnancement  $lg_k$  par rapport à la densité  $\delta_k$  est

$$\frac{lg_k}{\delta_k} = \frac{h^2 \times |N|}{\delta_k}$$

ou  $|N|$  est le cardinal moyen d'une classe de slots.

Ayant considéré le processus itératif jusqu'à ce qu'au moins un nœud appartient aux carrés,  $|N| = 1$ .

$$\frac{lg_k}{\delta_k} = \frac{(2^k + 1)^2}{\delta_0 \times 4^k} = \frac{2^{2k} + 2^{k+1} + 1}{\delta_0 \times 4^k} = \frac{2^{2k} + 2^{k+1} + 1}{\pi 4^{k-1}} = \frac{1}{\pi} \left(4 + \frac{1}{2^{k-3}} + \frac{1}{2^{2k-2}}\right)$$

$$\lim_{k \rightarrow +\infty} \frac{lg_k}{\delta_k} = \frac{4}{\pi}$$

La valeur  $4/\pi$  représente donc la borne inférieure du facteur de multiplication.

Ce raisonnement a été mené sous l'hypothèse d'une division successive d'un carré  $l_k$  en 4 carrés de côté  $l_{k+1} = l_k/2$ , en assurant l'existence d'au moins un nœud dans les carrés obtenus. Cette hypothèse est forte, le découpage pouvant se faire, dans certains cas, sur une seule dimension.

## 7.5/ PERFORMANCES DE IDEG-ROUTING&MAC VS MAR-WSN, IDEG-RELO ET DEPTH-RELO

Durant le travail de cette thèse, nous avons abordé les trois types de cross-layering et nous avons proposé de nouvelles approches permettant de minimiser la latence des communications dans les réseaux de capteurs sans fil.

Comme IDeg-Routing&MAC produit simultanément un ordonnancement TDMA et un arbre de routage géographique, nous proposons dans cette section de comparer ses performances, d'une part avec MAR-WSN car celui-ci produit un arbre de routage et, d'autre part, avec IDeg-ReLO et Depth-ReLO car elles produisent des ordonnancements TDMA. IDeg-ReLO et Depth-ReLO sont choisies parmi toutes les approches d'ordonnancement TDMA utilisant des informations du routage car elles présentaient les meilleures performances dans cette catégorie (voir section 6.6.4).

### 7.5.1/ IDEG-ROUTING&MAC vs MAR-WSN

Les deux approches IDeg-Routing&MAC et MAR-WSN permettent d'obtenir des arbres de routage. IDeg-Routing&MAC produit un arbre de routage géographique et un ordonnancement TDMA simultanément.

MAR-WSN construit un arbre de routage permettant le rapprochement vers le sink en utilisant des informations de l'ordonnancement TDMA et en profitant de la connaissance du voisinage à deux sauts. Comparer les performances du routage des deux approches (IDeg-Routing&MAC et MAR-WSN) revient à comparer l'arbre de routage géographique et l'arbre de routage de MAR-WSN. Cette comparaison est déjà faite dans la section 5.3.2.

### 7.5.2/ IDEG-ROUTING&MAC vs IDEG-RELO ET DEPTH-RELO

IDeg-ReLO et Depth-ReLO ont présenté les meilleures performances dans leur catégorie (respectivement la catégorie des approches MAC utilisant des informations de l'arbre de routage et la catégorie des approches MAC utilisant des informations de l'arbre de routage et du graphe modélisant le réseau). Puisque IDeg-Routing&MAC, IDeg-ReLO et Depth-ReLO produisent des ordonnancements TDMA, nous évaluons leurs performances en termes de la latence moyenne, la latence normalisée moyenne et la longueur du TDMA.

Les figures 7.13, 7.14 et 7.15 présentent les résultats obtenus.

Comme le montrent les figures 7.13 et 7.14, en termes de latence moyenne et de latence normalisée moyenne de communication, IDeg-ReLO et Depth-ReLO présentent de meilleures performances que IDeg-Routing&MAC. L'avantage de IDeg-ReLO et Depth-ReLO est la vue globale qu'elles ont sur l'arbre de routage ce qui leur permet d'ordonnancer les nœuds de la meilleure façon et ensuite de faire l'ordonnancement. IDeg-Routing&MAC n'a pas cette vue globale sur l'arbre de routage car ce dernier n'est pas complètement construit.

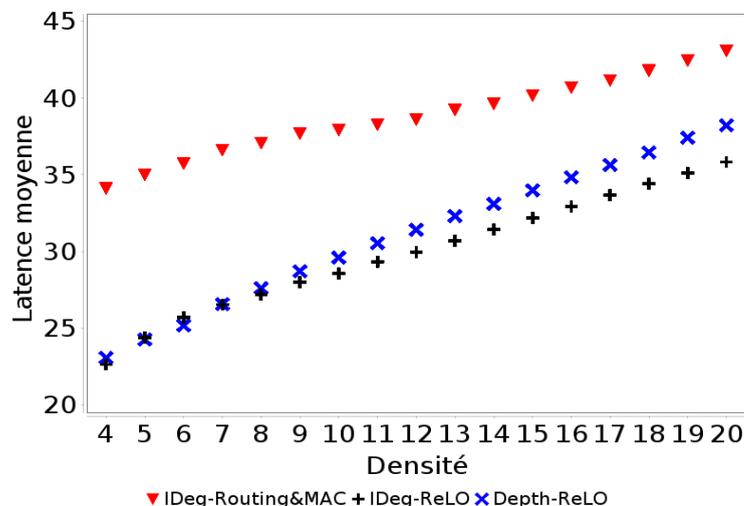


FIGURE 7.13 – Latence moyenne de IDeg-Routing&MAC vs IDeg-ReLO et Depth-ReLO

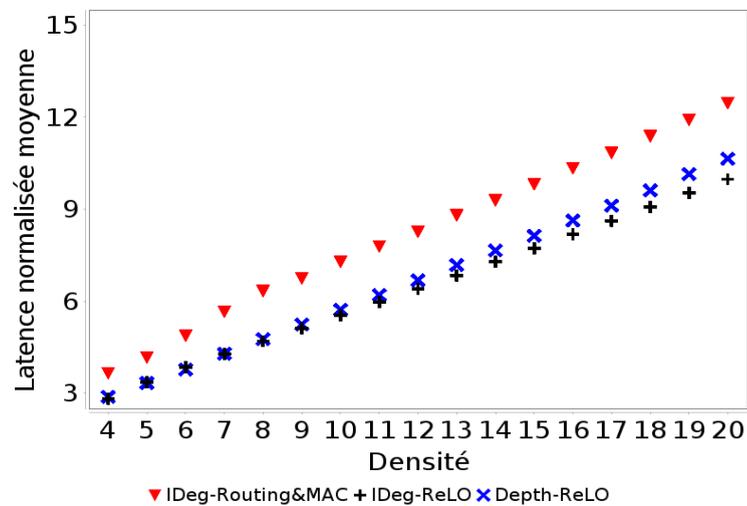


FIGURE 7.14 – Latence normalisée moyenne de IDeg-Routing&MAC vs IDeg-ReLO et Depth-ReLO

La figure 7.15 présente les longueurs de TDMA obtenues avec IDeg-Routing&MAC, IDeg-ReLO et Depth-ReLO. Les trois approches produisent des ordonnancements TDMA de longueurs très proches grâce à la méthode d'allocation de slots circulaire qui permet de réduire la longueur du TDMA.

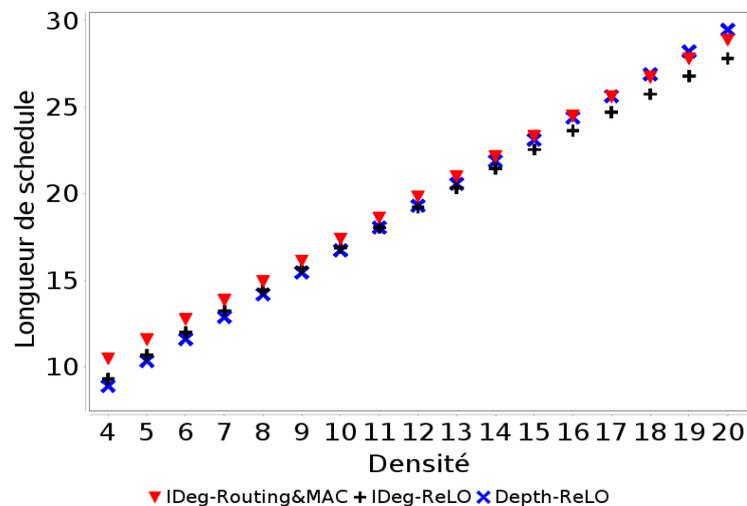


FIGURE 7.15 – Longueur de TDMA de IDeg-Routing&MAC vs IDeg-ReLO et Depth-ReLO

## 7.6/ CONCLUSION

Dans ce chapitre, nous avons proposé une approche cross-layer combinant un TDMA et le routage géographique, appelée IDeg-Routing&MAC. Les travaux existants combinant MAC et routage se focalisent sur les approches MAC avec contention et donc aucune approche ne propose de combiner une approche MAC sans contention avec du routage.

La fusion des deux couches Liaison de données et Réseaux dans IDeg-Routing&MAC permet, d'après les résultats obtenus, d'atteindre un gain en performance en termes de latence de communication. L'utilisation des informations des deux couches en même temps permet aussi de surmonter les décisions non corrélées qui proviennent des deux couches.

L'observation des longueurs de TDMA obtenus avec les approches MAC combinées avec le routage et les approches MAC utilisant des informations du routage a permis de trouver une dépendance entre cette longueur et la densité des réseaux utilisés. Cette dépendance est linéaire et peut s'exprimer sous forme de facteur de multiplication qui a comme borne inférieure  $4/\pi$ .

# CONCLUSION GÉNÉRALE ET PERSPECTIVES

## CONCLUSION GÉNÉRALE

Dans cette thèse, nous avons traité le problème de décisions non corrélées entre les protocoles de routage et les protocoles MAC sans contention de type TDMA, qui affectent les métriques liées à plusieurs couches telles que la latence des communications. Le concept de cross-layering permet de résoudre ce problème en exploitant les dépendances entre la couche Réseau et la couche Liaison de données.

Nous avons présenté d'abord des généralités sur les réseaux de capteurs sans fil et le modèle en couches assurant les communications dans de tels réseaux. Puis, nous avons proposé une classification des approches cross-layer entre routage et MAC et nous avons présenté les travaux existants dans cet axe de recherche. Cette classification comporte trois classes principales : la classe des approches de routage utilisant des informations du TDMA, la classe des approches MAC utilisant des informations du routage et la classe combinant les deux approches MAC et routage en une seule nouvelle approche. Nous nous sommes ensuite concentrés sur le routage permettant d'acheminer les données de chaque nœud du réseau vers le sink en passant par d'autres nœuds, appelé routage multi-sauts, et sur l'approche MAC sans contention de type TDMA. À partir de ces deux protocoles nous avons proposé nos différentes contributions, résumées dans la figure 7.16.

Notre première contribution, MAR-WSN, consiste en un algorithme de routage utilisant des informations du TDMA et exploitant le voisinage à deux sauts afin de trouver un chemin relayant un nœud émetteur au sink tout en minimisant la latence des communications.

La deuxième contribution comporte plusieurs approches d'ordonnancement TDMA utilisant des informations de l'arbre de routage afin d'établir des TDMA minimisant la latence de communication dans l'arbre de routage donné. En effet, nous avons proposé deux classes d'approches TDMA ; la première classe comporte les approches d'ordonnancement TDMA utilisant uniquement des informations de l'arbre de routage (I-CoLaNet, Rand-LO, Depth-LO, Depth-ReLO et BFS) ; la deuxième classe comporte les approches d'ordonnancement TDMA utilisant des informations de l'arbre de routage et du graphe modélisant le réseaux (IDeg-LO, IDeg-ReLO et IDeg-BFS).

Nous présentons aussi un algorithme d'ordonnancement optimal qui permet de trouver l'ordonnancement TDMA ayant la latence minimale.

La troisième contribution, IDeg-Routing&MAC, consiste en une approche fusionnelle MAC et routage dans laquelle un seul algorithme produit un ordonnancement TDMA et un arbre de routage simultanément, qui minimise la latence des communications.

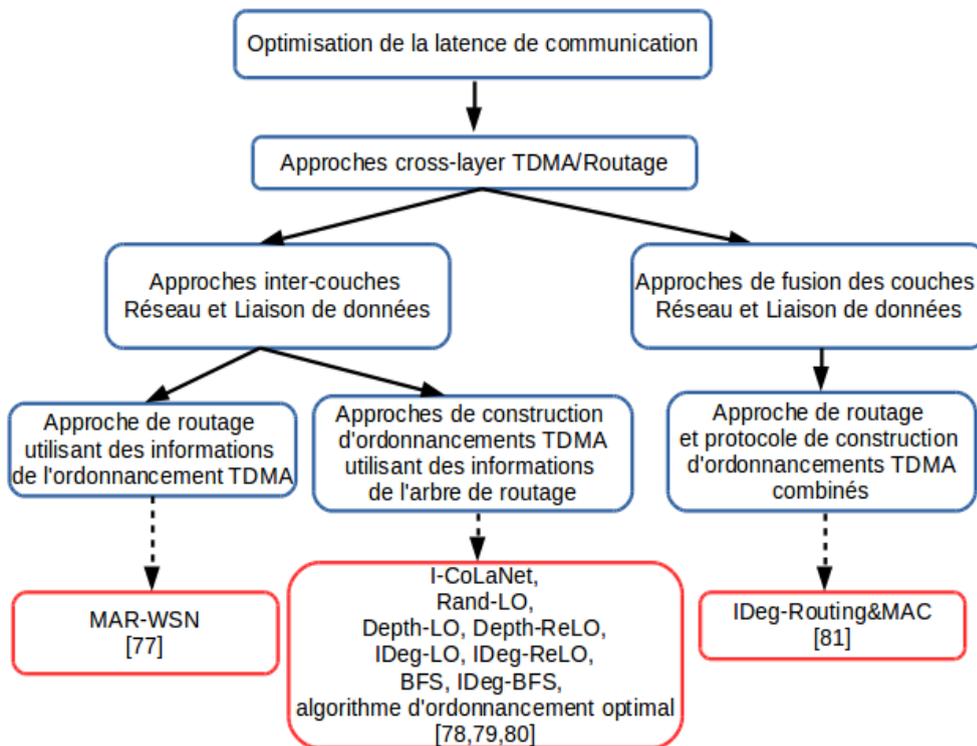


FIGURE 7.16 – Synthèse des contributions

À travers des simulations extensives, les différentes contributions sont évaluées et comparées avec des travaux similaires existants. Les résultats obtenus sont satisfaisants en termes de latence de communication. D'autres métriques ont été aussi considérées telles que la consommation énergétique des nœuds, le nombre de sauts des chemins de routage ou le duty cycle des nœuds.

## PERSPECTIVES

Le but de la thèse était de proposer des approches cross-layer entre les deux couches Réseaux et Liaison de données afin de minimiser la latence des communications dans les réseaux de capteurs sans fil. Ce but étant atteint dans un contexte précis (selon les hypothèses précisées) nous énonçons quelques perspectives intéressantes de ce travail.

**Topologie des réseaux** Nous sommes persuadés de la nécessité d'étendre nos simulations à d'autres modèles de communication. Le modèle UDG est connu pour être idéaliste et ne reflète pas le modèle de communication réel des capteurs. L'utilisation du modèle de disque quasi-unitaire (QUDG), par exemple, permettrait d'analyser les performances de nos différentes approches dans un contexte moins idéaliste.

De même, les simulations utilisant d'autres topologies du réseau, guidée par des densités différentes par zone ou par un placement différent pour le sink dans le réseau par exemple, semblent être intéressantes afin d'observer les performances de nos différentes approches.

**Sous-couche MAC** Dans ce travail, nous n'avons considéré que des protocoles MAC sans contention (TDMA) parce que le TDMA ne présente pas de collisions et par conséquent, la latence ne provient pas du mécanisme de synchronisation des accès au canal de communication malgré la nécessité d'une synchronisation en amont pour aligner les slots de communication. Avec un ordonnancement TDMA, la latence provient directement de la communication multi-sauts induite par l'arbre de routage. Cette latence pouvant être calculée explicitement, le choix du TDMA nous a paru pertinent pour maîtriser au mieux l'optimisation des communications. En plus, le TDMA n'est pas coûteux, en terme de consommation énergétique, lorsqu'il est utilisé de manière centralisée. Cependant, les protocoles MAC sans contention de type CSMA sont beaucoup plus utilisés dans le domaine des réseaux de capteurs sans fil à cause de leur simplicité lors du passage à l'échelle. Des travaux récents tels que [82] et [83] proposent des approches cross-layer entre les couches Réseaux et Liaison de données en se basant sur des approches MAC avec contention.

**Mobilité des nœuds capteurs** Il est également nécessaire de considérer l'aspect de mobilité des nœuds dans les réseaux de capteurs sans fil. En effet, les capteurs mobiles permettent d'élargir la couverture du réseau et d'améliorer le suivi du phénomène physique visé par l'application de ce réseau. Cependant, plusieurs aspects sont à considérer avec l'ajout de la mobilité. Entre autres, des techniques d'auto-organisation du réseau doivent être utilisées pour que le réseau soit capable de reconnaître ses capteurs. Dans [84], les auteurs proposent une approche permettant de minimiser la consommation énergétique en adoptant le TDMA avec la mobilité des nœuds. À partir de cette idée, nous pouvons considérer les approches cross-layer que nous avons déjà proposées en ajoutant la mobilité des nœuds.

**Communications multi-sink** Toutes nos simulations ont été réalisées en considérant un seul sink qui collecte toutes les données du réseau. L'inconvénient de ce choix est que les nœuds proches du sink relayent tout le trafic du réseau vers le sink ; par conséquent ces nœuds épuisent leur énergie rapidement, formant des trous d'énergie (energy-holes) autour du sink et réduisant la durée de vie du réseau. Les approches multi-sink permettent de corriger ces problèmes et la consommation énergétique est mieux équilibrée au sein d'un réseau. Des travaux comme [85] et [86] proposent des algorithmes d'organisation des communications dans les réseaux de capteurs avec plusieurs sinks. D'autres travaux comme [87] considèrent le cross-layering avec des réseaux de capteurs multi-sink.



# LISTE DES PUBLICATIONS PERSONNELLES

## REVUE INTERNATIONALE

- Latency optimization through routing-aware time scheduling protocols for wireless sensor networks  
Lemia Louail, Violeta Felea  
in press by Computers & Electrical Engineering, Elsevier. Available online 12 July 2016.

## CONFÉRENCES INTERNATIONALES

- MAC-aware routing in wireless sensor networks  
Lemia Louail, Violeta Felea, Julien Bernard, Hervé Guyennet  
3rd International IEEE Black Sea Conference on Communications and Networking (BlackSeaCom) 2015, Constanta, Romania, pages 225-229.
- Routing-aware TDMA scheduling for Wireless Sensor Networks  
Lemia Louail, Violeta Felea  
12th IEEE/IFIP Wireless On-demand Network systems and Services Conference (WONS) 2016, Cortina d'Ampezzo, Italy, pages 9-16.
- Routing-Aware Time Slot Allocation Heuristics in Contention-Free Sensor Networks  
Lemia Louail, Violeta Felea  
14th International Conference on Wired & Wireless Internet Communications (WWIC) 2016, Thessaloniki, Greece, pages 271-283.
- Routing and TDMA Joint Cross-Layer Design for Wireless Sensor Networks  
Lemia Louail, Violeta Felea  
International Conference on Ad Hoc Networks and Wireless (ADHOC-NOW) 2016, Lille, France, pages 111-123.



# BIBLIOGRAPHIE

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci. Wireless sensor networks : a survey. *Computer Networks*, 38 :393–422, 2002.
- [2] S. Schmid and R. Wattenhofer. Algorithmic models for sensor networks. In *14th WPDRTS*, pages 450–459, 2006.
- [3] B. N. Clark, C. J. Colbourn and D. S. Johnson. Unit disk graphs. *Discrete mathematics*, 86(1-3) :165–177, 1990.
- [4] F. Kuhn, R. Wattenhofer and A. Zollinger. Ad-hoc networks beyond unit disk graphs. In *Proceedings of the 2003 joint workshop on Foundations of mobile computing*, pages 69–78. ACM, 2003.
- [5] M. Mehrandish. *On routing, backbone formation and barrier coverage in wireless Ad Hoc and sensor networks*. PhD thesis, University Montréal, Québec, Canada, 2011.
- [6] H. Zimmermann. OSI reference model–The ISO model of architecture for open systems interconnection. *IEEE Transactions on Communications*, 28(4) :425–432, 1980.
- [7] I. Demirkol, C. Ersoy, F. Alagoz. MAC protocols for wireless sensor networks : a survey. *IEEE Communications Magazine*, 44(4) :115–121, 2006.
- [8] Huang, Pei and Xiao, Li and Soltani, Soroor and Mutka, Matt W and Xi, Ning. The evolution of MAC protocols in wireless sensor networks : A survey. *IEEE Communications Surveys & Tutorials*, 15(1) :101–120, 2013.
- [9] W. Ye, J. S. Heidemann and D. Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Trans. Netw.*, 12(3) :493–506, 2004.
- [10] T. van Dam and K. Langendoen. An adaptive energy-efficient MAC protocol for wireless sensor networks. In *SenSys*, pages 171–180, 2003.
- [11] P. Lin, C. Qiao and X. Wang. Medium access control with a dynamic duty cycle for sensor networks. In *WCNC*, pages 1534–1539, 2004.
- [12] G. Lu, B. Krishnamachari and C. S. Raghavendra. An Adaptive Energy-Efficient and Low-Latency MAC for Data Gathering in Wireless Sensor Networks. In *IPDPS*, pages 224–235, 2004.
- [13] R. Injong, W. Ajit, A. Mahesh, M. Jeongki and M. L. Sichitiu. Z-mac : a hybrid mac for wireless sensor networks. *IEEE/ACM Transactions on Networking (TON)*, 16(3) :511–524, 2008.
- [14] W. Ye, F. Silva and J. Heidemann. Ultra-low duty cycle MAC with scheduled channel polling. In *Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 321–334. ACM, 2006.
- [15] K. Jamieson, H. Balakrishnan and Y. C. Tay. Sift : A MAC Protocol for Event-Driven Wireless Sensor Networks. In *EWSN*, pages 260–275, 2006.

- [16] P. Jung. Time Division Multiple Access (TDMA). *Encyclopedia of Telecommunications*, 2003.
- [17] L. Bao and J. J. Garcia-Luna-Aceves. A new approach to channel access scheduling for Ad Hoc networks. In *MOBICOM*, pages 210–221, 2001.
- [18] V. Rajendran, K. Obraczka and J. J. Garcia-Luna-Aceves. Energy-efficient collision-free medium access control for wireless sensor networks. In *SenSys*, pages 181–192, 2003.
- [19] S. Gobriel, D. Mosse and R. Cleric. TDMA-ASAP : Sensor network TDMA scheduling with adaptive slot-stealing and parallelism. In *29th IEEE International Conference on Distributed Computing Systems*, pages 458–465. IEEE, 2009.
- [20] T. Herman and S. Tixeuil. A distributed TDMA slot assignment algorithm for wireless sensor networks. In *International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics*, pages 45–58. Springer, 2004.
- [21] V. Rajendran, K. Obraczka and J. J. Garcia-Luna-Aceves. Energy-efficient, collision-free medium access control for wireless sensor networks. *Wireless Networks*, 12(1) :63–78, 2006.
- [22] R. Prasad. *CDMA for wireless personal communications*. Artech House, Inc., 1996.
- [23] B. H. Liu, Chun. T. Chou, J. Lipman and S. Jha. Using frequency division to reduce MAI in DS-CDMA wireless sensor networks. In *WCNC*, pages 657–663, 2005.
- [24] J. N. Al-Karaki and A. E. Kamal. Routing techniques in wireless sensor networks : a survey. *Wireless communications, IEEE*, 11(6) :6–28, 2004.
- [25] C. Intanagonwiwat, R. Govindan and D. Estrin. Directed diffusion : a scalable and robust communication paradigm for sensor networks. In *MOBICOM*, pages 56–67, 2000.
- [26] W. Heinzelman, J. Kulik and H. Balakrishnan. Adaptive Protocols for Information Dissemination in Wireless Sensor Networks. In *MOBICOM*, pages 174–185, 1999.
- [27] J. Kulik, W. R. Heinzelman and H. Balakrishnan. Negotiation-Based Protocols for Disseminating Information in Wireless Sensor Networks. *Wireless Networks*, 8(2-3) :169–185, 2002.
- [28] D. Braginsky and D. Estrin. Rumor routing algorithm for sensor networks. In *WSNA*, pages 22–31, 2002.
- [29] F. Ye, A. Chen, S. Lu, and L. Zhang. A scalable solution to minimum cost forwarding in large sensor networks. In *In proceedings of the 10th International Conference on Computer Communications and Networks*, pages 304–309. IEEE, 2001.
- [30] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In *HICSS, USA*, pages 3005–3014, 2000.
- [31] S. Lindsey and C. S. Raghavendra. PEGASIS : Power-efficient gathering in sensor information systems. In *Aerospace conference proceedings*, volume 3, pages 1125–1130. IEEE, 2002.
- [32] A. Manjeshwar and D. P. Agrawal. TEEN : A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks. In *IPDPS*, page 189, 2001.

- [33] A. Manjeshwar and D. P. Agrawal. APTEEN : A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks. In *IPDPS*, pages 49–55, 2002.
- [34] V. Rodoplu and T. H. Meng. Minimum energy mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, 17(8) :1333–1344, 1999.
- [35] B. Hofmann-Wellenhof, H. Lichtenegger and J. Collins. *Global positioning system : theory and practice*. Springer Science & Business Media, 2012.
- [36] N. Bulusu, J. Heidemann and D. Estrin. GPS-less low-cost outdoor localization for very small devices. *IEEE personal communications*, 7(5) :28–34, 2000.
- [37] B. Karp. *Geographic Routing for Wireless Networks*. PhD thesis, Harvard University, 2000.
- [38] B. Karp and H. Kung. GPSR : Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243–254. ACM, 2000.
- [39] X. Ya, J. S. Heidemann and D. Estrin. Geography-informed energy conservation for Ad Hoc routing. In *MOBICOM*, pages 70–84, 2001.
- [40] Y. Yan, G. Ramesh and D. Estrin. Geographical and energy aware routing : A recursive data dissemination protocol for wireless sensor networks. Technical report, 2001.
- [41] K. Sohrabi, J. Gao, V. Ailawadhi and G. J. Pottie. Protocols for self-organization of a wireless sensor network. *IEEE Personal Commun.*, 7(5) :16–27, 2000.
- [42] S. Fedor, M. Collier and C. J. Sreenan. Cross-layer routing and time synchronisation in wireless sensor networks. *Int. J. Signal and Imaging Systems Engineering*, 10(3) :143–159, 2011.
- [43] I. F. Akyildiz and X. Wang. Cross-layer design in wireless mesh networks. *IEEE Transactions on Vehicular Technology*, 57(2) :1061–1076, 2008.
- [44] N. El Rachkidy. *Cross-Layering et routage dans un réseau ad hoc : politique de relais de trame sur un réseau de capteurs sans fil organisé selon une topologie en arbre*. PhD thesis, Université Blaise Pascal-Clermont-Ferrand II, 2011.
- [45] ITU-R International Telecommunication Union - Radiocommunication Sector . Cross-layer QoS provisioning in IP-based hybrid satellite-terrestrial networks.
- [46] O. Landsiedel, E. Ghadimi, S. Duquennoy and M. Johansson. Low Power, Low Delay : Opportunistic Routing meets Duty Cycling. In *IPSN*, pages 185–196, 2012.
- [47] C. Suh, Y. Ko and D. Son. An Energy Efficient Cross-Layer MAC Protocol for Wireless Sensor Networks. In *APWeb Workshops*, pages 410–419, 2006.
- [48] S. Du, A. Kumar Saha and D. B. Johnson. RMAC : A Routing-Enhanced Duty-Cycle MAC Protocol for Wireless Sensor Networks. In *INFOCOM*, pages 1478–1486, 2007.
- [49] M. Hefaida, T. Canli and A. A. Khokhar. CL-MAC : A Cross-Layer MAC protocol for heterogeneous Wireless Sensor Networks. *Ad Hoc Networks*, 11(1) :213–225, 2013.
- [50] K. Heurtefeux, F. Maraninchi and F. Valois. AreaCast : A cross-layer approach for a communication by area in Wireless Sensor Networks. In *ICON*, pages 112–117, 2011.

- [51] C. Chou and K. Chuang. CoLaNet : A Cross-Layer Design of Energy-Efficient Wireless Sensor Networks. In *ICW/ICHSN/ICMCS/SENET*, pages 364–369, 2005.
- [52] O. D. Incel, A. Ghosh, B. Krishnamachari and K. Chintalapudi. Fast data collection in tree-based wireless sensor networks. *IEEE Transactions on Mobile computing*, 11(1) :86–99, 2012.
- [53] S. Pemmaraju and S. Skiena. *Computational Discrete Mathematics : Combinatorics and Graph Theory with Mathematica*. Cambridge University Press, NY, USA, 2003.
- [54] W. Masri. *Dérivation d'exigences de qualité de service dans les réseaux de capteurs sans fils basés sur TDMA*. PhD thesis, Université Toulouse III-Paul Sabatier, 2009.
- [55] W. Masri and Z. Mammeri. QoS Mapping in TDMA tree based clustered WSN between accuracy, density and bandwidth. In *International Conference on Communication Theory, Reliability, and Quality of Service*, pages 71–76. IEEE, 2008.
- [56] S. S. Kulkarni, A. Iyer and C. Rosenberg. An address-light, integrated MAC and routing protocol for wireless sensor networks. *IEEE/ACM Trans. Netw.*, 14(4) :793–806, 2006.
- [57] A. Awang, X. Lagrange and D. Ros Sanchez. A Cross-Layer Medium Access Control and Routing Protocol for Wireless Sensor Networks. In *10èmes Journées Doctorales en Informatique et Réseaux, 2-4 février, Belfort, France, 2009*.
- [58] A. Awang, X. Lagrange and D. Ros. RSSI-based forwarding for multihop wireless sensor networks. In *Meeting of the European Network of Universities and Companies in Information and Communication Engineering*, pages 138–147. Springer, 2009.
- [59] D. Chen, J. Deng and P. K. Varshney. A state-free data delivery protocol for multihop wireless sensor networks. In *WCNC*, pages 1818–1823, 2005.
- [60] I. Rhee, A. Warriar, J. Min and L. Xu. DRAND : distributed randomized TDMA scheduling for wireless ad-hoc networks. In *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, pages 190–201. ACM, 2006.
- [61] K. Moriyama and Y. Zhang. An Efficient Distributed TDMA MAC Protocol for Large-Scale and High-Data-Rate Wireless Sensor Networks. In *29th International Conference on Advanced Information Networking and Applications*, pages 84–91. IEEE, 2015.
- [62] F. Clad, A. Gallais and P. Mérindol. Energy-efficient data collection in wsn : A sink-oriented dynamic backbone. In *International Conference on Communications (ICC)*, pages 276–280. IEEE, 2012.
- [63] B. Romdhani. *Exploitation de l'hétérogénéité des réseaux de capteurs et d'actionneurs dans la conception des protocoles d'auto-organisation et de routage*. PhD thesis, INSA de Lyon, 2012.
- [64] P. E. Greenwood and M. S. Nikulin. *A guide to chi-squared testing*, volume 280. John Wiley & Sons, 1996.
- [65] D. Knuth and others. *The Art of Computer Programming, Volume 2 : Semi numerical Algorithms*, 1998.
- [66] Z. Wang S. Zhuo, Y. Song and Z. Wang. Queue-MAC : A queue-length aware hybrid CSMA/TDMA MAC protocol for providing dynamic adaptation to traffic and duty-cycle variation in wireless sensor networks. In *9th IEEE WFCS*, pages 105–114, May 2012.

- [67] J. O'Madadhain, D. Fisher, P. Smyth, and S. White and Y. Boey. Analysis and visualization of network data using JUNG. *Journal of Statistical Software*, 10(2) :1–35, 2005.
- [68] D. L. Guidoni, R. Mini and A. Loureiro. On the design of heterogeneous sensor networks based on small world concepts. In *Proceedings of the 11th international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 309–314. ACM, 2008.
- [69] F. A. Onat and I. Stojmenovic. Generating Random Graphs for Wireless Acuator Networks. In *IEEE WoWMoM*, pages 1–12, 2007.
- [70] F. Kuhn, R. Wattenhofer, Y. Zhang and A. Zollinger. Geometric adhoc routing : of theory and practice. In *22nd Annual Symposium on Principles of distributed computing*, pages 63–72, 2003.
- [71] P. Bose, P. Morin, I. Stojmenovic and J. Urrutia. Routing with Guaranteed Delivery in Ad Hoc Wireless Networks. *Wireless Networks*, 7 :609–616, 2001.
- [72] P. Ardilly. *Les techniques de sondage*. Technip, 2006.
- [73] R. Seljak. Standard errors presentation and dissemination at the statistical office of the republic of Slovenia. 2008.
- [74] T. Badrick and P. Stewart. Measurement uncertainty. *Common Sense Pathology*, 2014. 8 pages.
- [75] A. Pryseley and K. Mintiens and K. Knapien and Y. Van der Stede and G. Molenberghs. Estimating Precision, Repeatability, and Reproducibility From Gaussian and Non-Gaussian Data : A Mixed Models Approach. *Journal of Applied Statistics*, 37(10) :1729–1747, 2010.
- [76] Statistics Canada. Guide to the Labour Force Survey, 2010.
- [77] L. Louail, V. Felea, J. Bernard and H. Guyennet. MAC-aware routing in wireless sensor networks. In *International Black Sea Conference on Communications and Networking (BlackSeaCom)*, pages 225–229. IEEE, 2015.
- [78] L. Louail and V. Felea. Routing-Aware Time Slot Allocation Heuristics in Contention-Free Sensor Networks. In *International Conference on Wired/Wireless Internet Communication*, pages 271–283. Springer, 2016.
- [79] L. Louail and V. Felea. Latency optimization through routing-aware time scheduling protocols for wireless sensor networks. *Computers & Electrical Engineering*, Available online 12 July 2016.
- [80] L. Louail and V. Felea. Routing-aware TDMA scheduling for wireless sensor networks. In *12th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, pages 9–16. IEEE, 2016.
- [81] L. Louail and V. Felea. Routing and TDMA Joint Cross-Layer Design for Wireless Sensor Networks. In *International Conference on Ad-Hoc Networks and Wireless*, pages 111–123. Springer, 2016.
- [82] S. Ozen and S. Oktug. Adaptive sink selection for WSNs using forwarder set based dynamic duty cycling. In *Eleventh Annual IEEE International Conference on Sensing, Communication, and Networking Workshops (SECON Workshops)*, pages 7–12. IEEE, 2014.

- [83] A. Hiromori, A. Uchiyama, H. Yamaguchi and T. Higashino. Deadline-Aware Data Collection in CSMA/CA-based Multi-Sink Wireless Sensor Networks. In *Proceedings of the Sixth International Conference on Mobile Computing and Ubiquitous Networking, ICMU*, volume 12, pages 284–294, 2012.
- [84] P. Poornima and R. Rajesh. Mobility Control Based Dual Data Transmission for Load Balancing in WSN. *Middle-East Journal of Scientific Research*, 2016.
- [85] P. Chatterjee and N. Das. Multiple sink deployment in multi-hop wireless sensor networks to enhance lifetime. In *Applications and Innovations in Mobile Computing (AIMoC)*, pages 48–54. IEEE, 2015.
- [86] C. Sha, J. Qiu, S. Li, M. Qiang and R. Wang. A Type of Low-Latency Data Gathering Method with Multi-Sink for Sensor Networks. *Sensors*, 16(6) :923, 2016.
- [87] R. Soua, E. Livolant and P. Minet. MUSIKA : A multichannel multi-sink data gathering algorithm in wireless sensor networks. In *9th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 1370–1375. IEEE, 2013.

# TABLE DES FIGURES

1.1	Composants d'un capteur intelligent . . . . .	8
1.2	Exemple d'un réseau de capteurs sans fil . . . . .	9
1.3	Modèle de communication UDG . . . . .	10
1.4	Modèle de communication GG . . . . .	11
1.5	Modèle de communication QUDG . . . . .	11
2.1	Modèle en couches utilisé dans les réseaux de capteurs sans fil . . . . .	14
2.2	Classification des protocoles MAC . . . . .	15
2.3	Classification des protocoles de routage . . . . .	17
3.1	Les deux types de Cross-Layering . . . . .	24
3.2	Classification des approches cross-layer concernant les couches Réseau et Liaison de données . . . . .	25
3.3	Transmission des paquets RTS/CTS avec MAC-CROSS . . . . .	28
3.4	Période DATA de RMAC . . . . .	29
3.5	Phase de configuration de AIMRP . . . . .	33
3.6	Configuration de SIF . . . . .	35
4.1	Un graphe modélisant un réseau comportant 8 capteurs (dont un sink) . . . . .	44
4.2	Exemple d'un arbre de routage pour le réseau de la figure 4.1 . . . . .	46
4.3	Les cadences pour les données et pour les transmissions . . . . .	47
4.4	Exemples de graphes générés avec les densités 4, 10 et 20 . . . . .	55
5.1	Exemple d'un réseau de capteurs utilisant TDMA . . . . .	57
5.2	Exemple d'un TDMA d'un nœud avec un voisin à un saut et un voisin à deux sauts . . . . .	59
5.3	MAR-WSN : latence des transmissions entre l'émetteur et le sink ( $\delta=10$ ) . . . . .	60
5.4	MAR-WSN : consommation énergétique des nœuds du chemin entre l'émetteur et le sink ( $\delta=10$ ) . . . . .	61
5.5	MAR-WSN : nombre de sauts entre l'émetteur et le sink ( $\delta=10$ ) . . . . .	61
5.6	MAR-WSN : latence des transmissions entre l'émetteur et le sink ( $\delta \in [6, 20]$ ) . . . . .	62

5.7	MAR-WSN : consommation énergétique des nœuds du chemin entre l'émetteur et le sink ( $\delta \in [6, 20]$ ) . . . . .	63
5.8	MAR-WSN : nombre de sauts entre l'émetteur et le sink ( $\delta \in [6, 20]$ ) . . . . .	63
5.9	Latence de MAR-WSN vs routage géographique . . . . .	67
5.10	Nombre de sauts de MAR-WSN vs routage géographique . . . . .	68
5.11	Consommation énergétique de MAR-WSN vs routage géographique . . . . .	68
5.12	Gains et pertes de MAR-WSN par rapport au routage géographique . . . . .	69
5.13	Latence de MAR-WSN vs ORW . . . . .	70
5.14	Consommation énergétique de MAR-WSN vs ORW . . . . .	70
5.15	Nombre de sauts de MAR-WSN vs ORW . . . . .	71
5.16	Gains et pertes de MAR-WSN par rapport à ORW . . . . .	72
6.1	Exemples d'un graphe représentant un réseau de capteurs (à gauche) et son arbre de routage (à droite) . . . . .	74
6.2	Exemples de deux TDMA pour un réseau de capteurs . . . . .	74
6.3	Exemple d'un réseau linéaire composé de quatre nœuds . . . . .	87
6.4	Caption . . . . .	88
6.5	Exemple d'un réseau linéaire de $n$ nœuds . . . . .	90
6.6	Latence moyenne des approches basées sur des informations de l'arbre de routage MinDegree (à gauche) et de l'arbre géographique (à droite) . . . . .	91
6.7	Latence normalisée moyenne des approches basées sur des informations de l'arbre de routage MinDegree (à gauche) et de l'arbre géographique (à droite) . . . . .	92
6.8	Longueur de TDMA des approches basées sur des informations de l'arbre de routage MinDegree (à gauche) et de l'arbre géographique (à droite) . . . . .	92
6.9	Latence moyenne des approches basées sur des informations du graphe et de l'arbre de routage MinDegree (à gauche) et de l'arbre géographique (à droite) . . . . .	93
6.10	Latence normalisée moyenne des approches basées sur des informations du graphe et de l'arbre MinDegree (à gauche) et de l'arbre géographique (à droite) . . . . .	94
6.11	Longueur de TDMA des approches basées sur des informations du graphe et de l'arbre MinDegree (à gauche) et de l'arbre géographique (à droite) . . . . .	94
6.12	Distributions des nœuds sur les slots pour Depth-ReLO (à gauche) et BFS (à droite) . . . . .	95
6.13	Distributions des nœuds sur les slots pour IDeg-ReLO (à gauche) et IDeg-BFS (à droite) . . . . .	96
6.14	Synthèse des résultats de la latence moyenne avec l'arbre MinDegree (à gauche) et avec l'arbre géographique (à droite) . . . . .	97

6.15 Synthèse des résultats de la latence normalisée moyenne avec l'arbre Min-Degree (à gauche) et avec l'arbre géographique (à droite) . . . . .	97
6.16 Synthèse des résultats de la longueur de TDMA avec l'arbre MinDegree (à gauche) et avec l'arbre géographique (à droite) . . . . .	98
6.17 Latence moyenne de IDeg-ReLO et Depth-ReLO vs l'algorithme d'ordonnement optimal avec l'arbre MinDegree (à gauche) et avec l'arbre géographique (à droite) . . . . .	99
6.18 Latence normalisée moyenne de IDeg-ReLO et Depth-ReLO vs l'algorithme d'ordonnement optimal avec l'arbre MinDegree (à gauche) et avec l'arbre géographique (à droite) . . . . .	100
6.19 Synthèse des approches TDMA proposées, utilisant des informations du routage . . . . .	101
7.1 Exemple d'un réseau de 8 capteurs . . . . .	104
7.2 Arbre BFS du graphe de la figure 7.1 . . . . .	105
7.3 Arbre de routage obtenu en appliquant IDeg-Routing&MAC sur le réseau de la figure 7.1 . . . . .	105
7.4 Latence moyenne de IDeg-Routing&MAC vs MAC et routage indépendants	106
7.5 Latence normalisée moyenne de IDeg-Routing&MAC vs MAC et routage indépendants . . . . .	107
7.6 Longueur de TDMA de IDeg-Routing&MAC vs MAC et routage indépendants	107
7.7 Duty cycle de IDeg-Routing&MAC vs MAC et routage indépendants . . . . .	108
7.8 Distribution des nœuds avec le modèle UDG pour l'allocation des slots . . . . .	109
7.9 Allocation des slots pour des réseaux de densités de 8 à 16 (à gauche) et supérieure à 16 (à droite) . . . . .	110
7.10 Longueur de TDMA de IDeg-Routing&MAC et IDeg-ReLO vs longueur de TDMA théorique . . . . .	111
7.11 Exemple de découpage en carrés et de classes de slots utilisées pour les nœuds des carrés à l'itération $k$ . . . . .	112
7.12 Exemple de nœuds se trouvant en dehors du cercle mais inscrits dans le carré . . . . .	112
7.13 Latence moyenne de IDeg-Routing&MAC vs IDeg-ReLO et Depth-ReLO . . . . .	114
7.14 Latence normalisée moyenne de IDeg-Routing&MAC vs IDeg-ReLO et Depth-ReLO . . . . .	115
7.15 Longueur de TDMA de IDeg-Routing&MAC vs IDeg-ReLO et Depth-ReLO . . . . .	115
7.16 Synthèse des contributions . . . . .	118
B.1 Latence moyenne des approches basées sur des informations de l'arbre de routage Hop Count . . . . .	139

B.2	Latence normalisée moyenne des approches basées sur des informations de l'arbre de routage Hop Count . . . . .	140
B.3	Longueur de schedule des approches basées sur des informations de l'arbre de routage Hop Count . . . . .	140
B.4	Latence moyenne des approches basées sur des informations du graphe et de l'arbre Hop Count . . . . .	141
B.5	Latence normalisée moyenne des approches basées sur des informations du graphe et de l'arbre Hop Count . . . . .	141
B.6	Longueur de schedule des approches basées sur des informations du graphe et de l'arbre Hop Count . . . . .	142
B.7	Synthèse des résultats de la latence moyenne avec l'arbre Hop Count . . .	142
B.8	Synthèse des résultats de la latence normalisée moyenne avec l'arbre Hop Count . . . . .	143
B.9	Synthèse des résultats de la longueur de schedule avec l'arbre Hop Count	143
B.10	Latence moyenne de IDeg-ReLO et Depth-ReLO vs l'algorithme Optimal avec l'arbre Hop Count . . . . .	144
B.11	Latence normalisée moyenne de IDeg-ReLO et Depth-ReLO vs l'algorithme Optimal avec l'arbre Hop Count . . . . .	144

# LISTE DES TABLES

3.1	Bilan des approches cross-layer . . . . .	36
3.2	Table résumant les caractéristiques des approches cross-layer existantes ainsi que leurs inconvénients . . . . .	38
4.1	Exemple d'un ordonnancement TDMA pour le réseau de la figure 4.1 . . . . .	45
4.2	Nombre de TDMA rejetés par le test $\chi^2$ d'ajustement à la loi de distribution uniforme . . . . .	50
5.1	La valeur de $\alpha$ pour chaque $\delta$ , pour laquelle la valeur minimale de chaque métrique est obtenue . . . . .	64
5.2	Nombre d'occurrences des valeurs de $\alpha$ pour de chaque métrique . . . . .	65
5.3	Les rangs des métriques minimales et les valeurs de $rg_{min}^\alpha$ . . . . .	65
5.4	La moyenne de chaque métrique pour toutes les densités . . . . .	66
5.5	Les rangs des $\alpha$ selon la moyenne de chaque métrique . . . . .	66
5.6	Les valeurs de $rg^\alpha$ . . . . .	67
5.7	Les intervalles des coefficients de variation de MAR-WSN et ORW pour toutes densités confondues . . . . .	71
6.1	Temps nécessaires pour obtenir l'Ordonnancement Optimal . . . . .	90
6.2	Synthèse des gains en latence moyenne par rapport à CoLaNet . . . . .	97
6.3	Synthèse des gains en duty cycle par rapport à CoLaNet . . . . .	98
6.4	Les intervalles des coefficients de variation des approches d'ordonnancement TDMA utilisant des informations de l'arbre de routage MinDegree, toutes densités confondues . . . . .	99
6.5	Synthèse des pertes en latence moyenne par rapport à l'Optimal . . . . .	100
7.1	Le TDMA obtenu en appliquant IDeg-Routing&MAC sur le réseau de la figure 7.1 . . . . .	105
7.2	Les intervalles des coefficients de variation de IDeg-Routing&MAC et de l'approche MAC et routage indépendants pour toutes densités confondues . . . . .	108
A.1	Extrait de la table de distribution $\chi^2$ . . . . .	138



# IV

## ANNEXES



## LE TEST DU $\chi^2$ D'AJUSTEMENT

Le test du khi-deux ( $\chi^2$ ) d'ajustement [64, 65] permet de vérifier qu'une distribution observée suit une loi de distribution théorique.

Soit une variable qualitative à  $p$  modalités sur un échantillon de taille  $N$ . Pour le test du  $\chi^2$  d'ajustement, le tableau suivant est constitué :

variable	effectifs observés	effectifs théoriques
modalité 1	$o_1$	$e_1$
.	.	.
.	.	.
.	.	.
modalité $p$	$o_p$	$e_p$
total	$N$	$N$

où  $o_i$  désigne les effectifs observés et  $e_i$  désigne les effectifs théoriques.

L'unique condition d'application du test exige que les effectifs théoriques  $e_i$  soient supérieurs ou égaux à 5. Si ce n'est pas le cas, certaines modalités doivent être regroupées.

Le principe de l'application du test  $\chi^2$  d'ajustement est le suivant :

- formuler les hypothèses statistiques  
 $H_0$  : la distribution observée suit la loi théorique (hypothèse nulle).  
 $H_1$  : la distribution observée ne suit pas la loi théorique.
- calculer la valeur du  $\chi^2$  sur la base des effectifs observés :

$$\chi^2_{calc} = \sum_{i=1}^p \frac{(o_i - e_i)^2}{e_i}$$

- définir la zone de rejet de l'hypothèse nulle : pour un seuil de confiance  $\alpha$ , la région critique conduisant au rejet de l'hypothèse nulle est définie par l'intervalle  $[\chi^2_{\alpha, v}, +\infty]$  où  $\chi^2_{\alpha, v}$  correspond au quantile d'ordre  $\alpha$  de la loi du  $\chi^2$  à  $v$  degré de liberté. Un extrait de la table de distribution  $\chi^2$  permettant d'identifier le seuil  $\chi^2_{\alpha, v}$  est présenté dans la table A.1.

$\alpha \backslash v$	0.100	0.050	0.010
1	2.706	3.841	6.635
2	4.605	5.991	9.21
3	6.251	7.815	11.345
4	7.779	9.488	13.277
5	9.236	11.070	15.086
6	10.645	12.592	16.812
7	12.017	14.067	18.475
8	13.362	15.507	20.09
9	14.684	16.919	21.666
10	15.987	18.307	23.209
11	17.275	19.675	24.725
12	18.549	21.026	26.217

TABLE A.1 – Extrait de la table de distribution  $\chi^2$ 

4. décider du rejet : la décision du rejet de l'hypothèse nulle se base sur l'évaluation de  $\chi_{calc}^2$  par rapport au seuil  $\chi_{\alpha,v}^2$ . Intuitivement si la somme des différences excède la limite  $\chi_{\alpha,v}^2$ , cela signifie que les effectifs observés et ceux théoriques sont différents et par conséquent l'hypothèse d'égalité des distributions peut être remise en cause et l'hypothèse  $H_1$  est acceptée. Au cas contraire, l'hypothèse nulle  $H_0$  est conservée : il ne peut pas être affirmé dans ce cas que les effectifs observés et théoriques diffèrent essentiellement.

Dans notre cas, nous nous intéressons à vérifier qu'une distribution donnée suit une loi de distribution uniforme (voir sous-section 4.3.1, page 48).

# B

## PERFORMANCES DES APPROCHES TDMA UTILISANT DES INFORMATIONS DU ROUTAGE AVEC L'ARBRE HOP COUNT

En complément aux résultats présentés dans le chapitre 5, nous présentons dans cette annexe les performances de nos différentes approches d'ordonnement TDMA utilisant des informations de l'arbre de routage Hop Count.

### B.1/ PERFORMANCES DES APPROCHES TDMA UTILISANT DES INFORMATIONS DE L'ARBRE DE ROUTAGE

Comme le montrent les figures B.1, B.2 et B.3, les approches d'ordonnement TDMA utilisant des informations de l'arbre de routage présentent de meilleures performances que CoLaNet. Depth-ReLo présente toujours les meilleures performances.

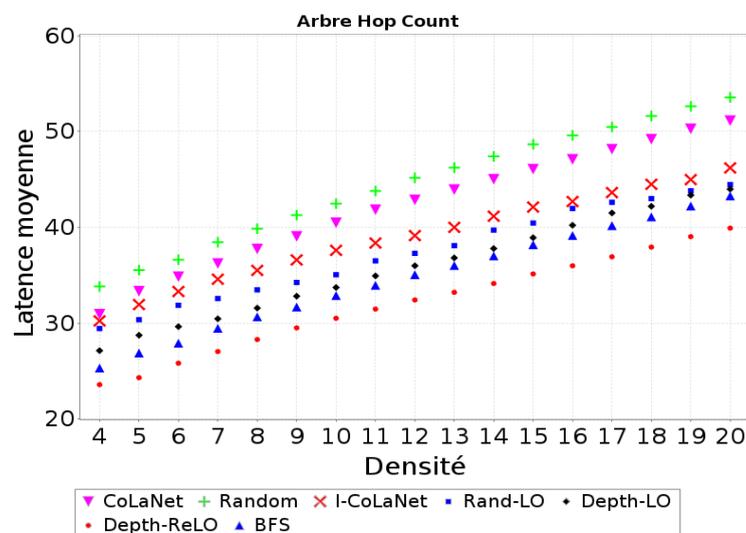


FIGURE B.1 – Latence moyenne des approches basées sur des informations de l'arbre de routage Hop Count

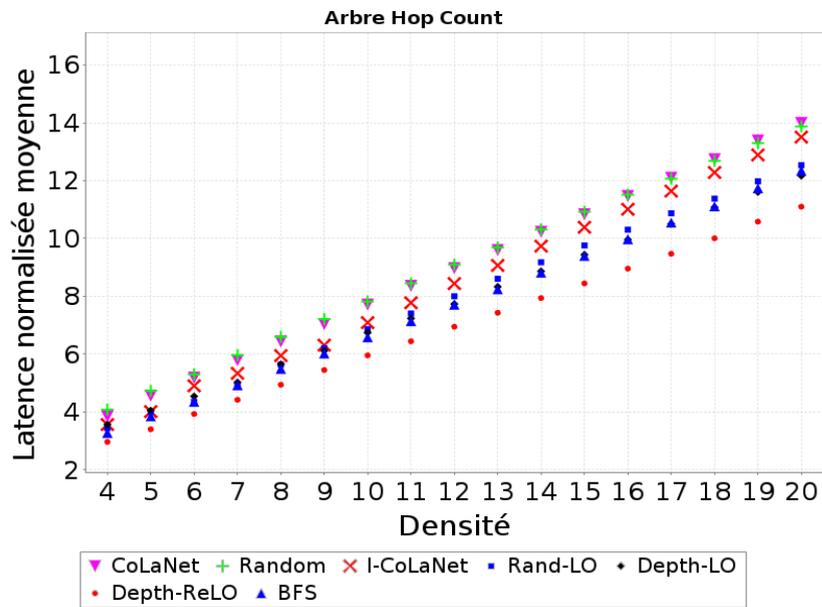


FIGURE B.2 – Latence normalisée moyenne des approches basées sur des informations de l'arbre de routage Hop Count

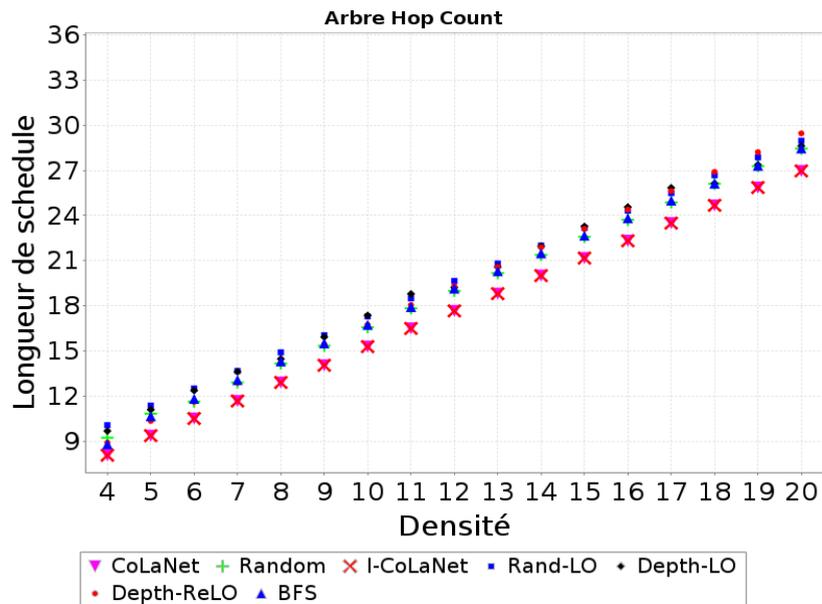


FIGURE B.3 – Longueur de schedule des approches basées sur des informations de l'arbre de routage Hop Count

## B.2/ PERFORMANCES DES APPROCHES TDMA UTILISANT DES INFORMATIONS DU GRAPHE ET DE L'ARBRE DE ROUTAGE

Les approches d'ordonnancement TDMA utilisant des informations de l'arbre de routage et du graphe présentent aussi de meilleures performances que CoLaNet. Depth-ReLo

présente aussi les meilleures performances dans cette catégorie (figures B.4, B.5 et B.7).

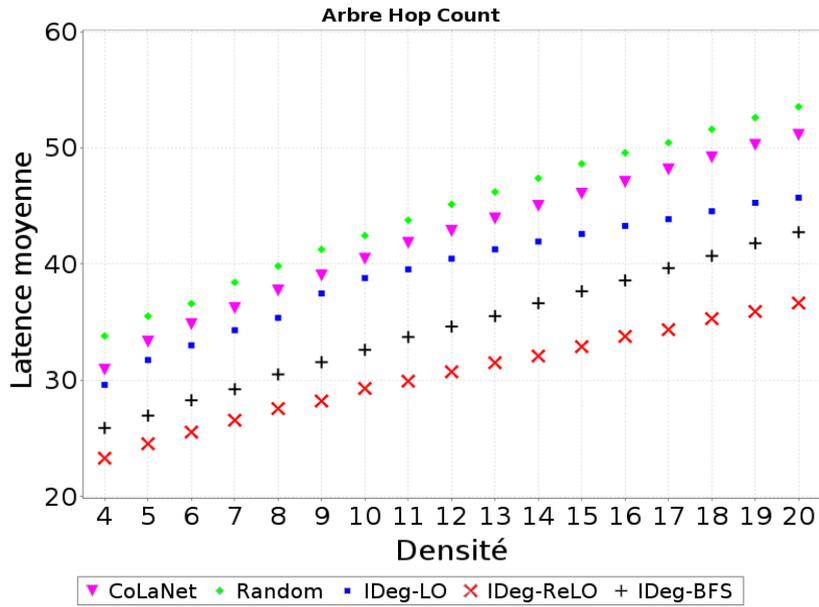


FIGURE B.4 – Latence moyenne des approches basées sur des informations du graphe et de l'arbre Hop Count

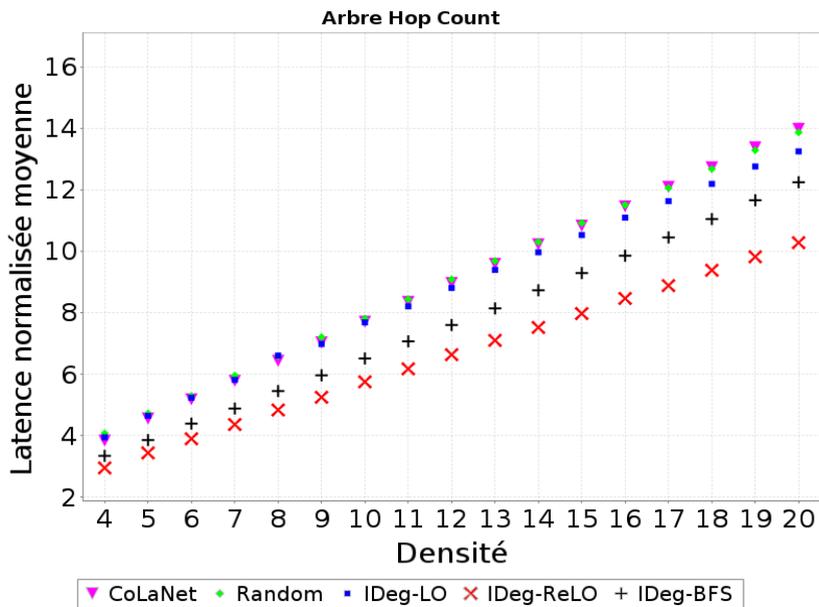


FIGURE B.5 – Latence normalisée moyenne des approches basées sur des informations du graphe et de l'arbre Hop Count

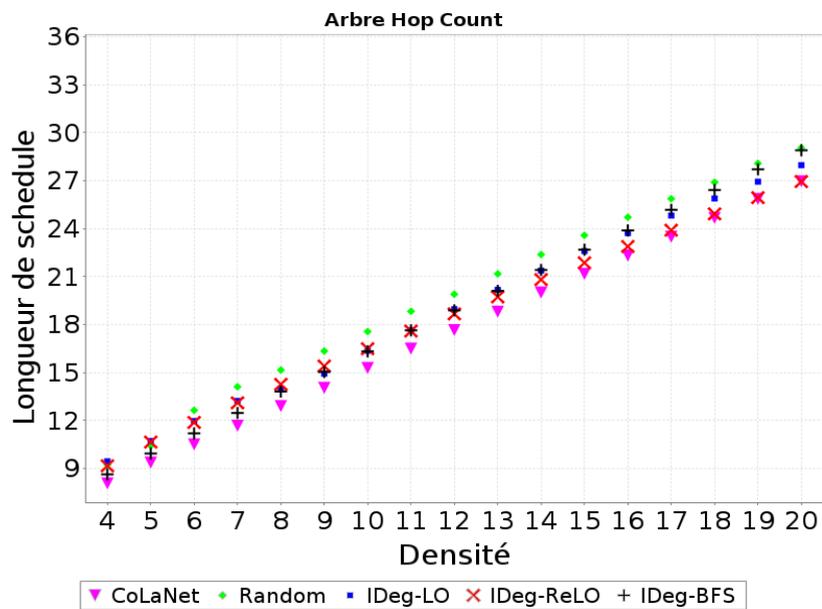


FIGURE B.6 – Longueur de schedule des approches basées sur des informations du graphe et de l'arbre Hop Count

### B.3/ SYNTHÈSE DES PERFORMANCES DES APPROCHES TDMA UTILISANT DES INFORMATIONS DU ROUTAGE

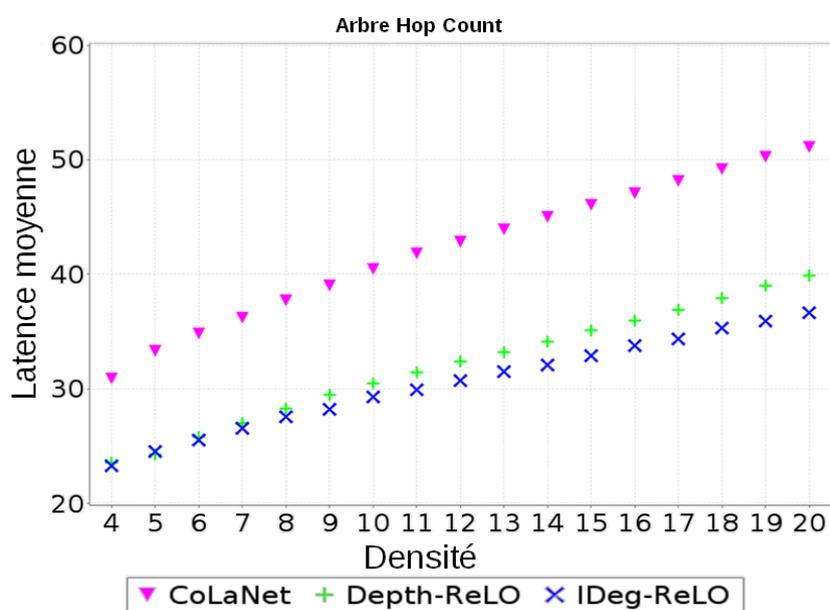


FIGURE B.7 – Synthèse des résultats de la latence moyenne avec l'arbre Hop Count

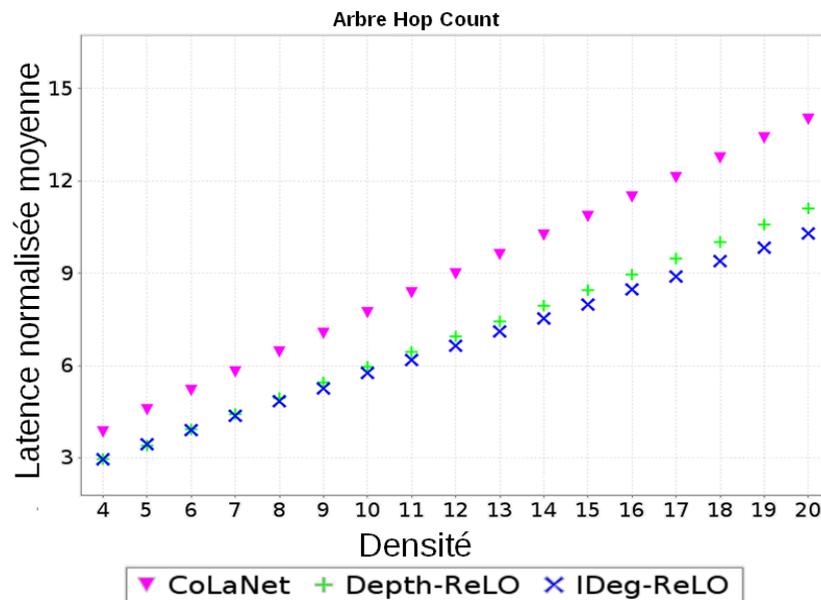


FIGURE B.8 – Synthèse des résultats de la latence normalisée moyenne avec l'arbre Hop Count

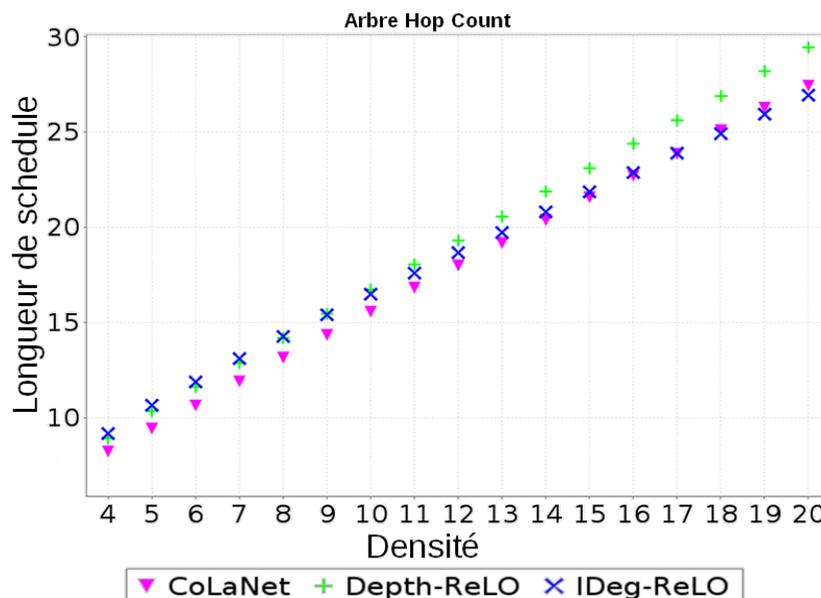


FIGURE B.9 – Synthèse des résultats de la longueur de schedule avec l'arbre Hop Count

#### B.4/ PERFORMANCES DES APPROCHES TDMA UTILISANT DES INFORMATIONS DE L'ARBRE DE ROUTAGE PAR RAPPORT À L'ORDONNANCEMENT OPTIMAL

Les figures B.10 et B.11 montrent que les performances de IDeg-ReLO et Depth-ReLo sont proches des performances de l'algorithme optimal.

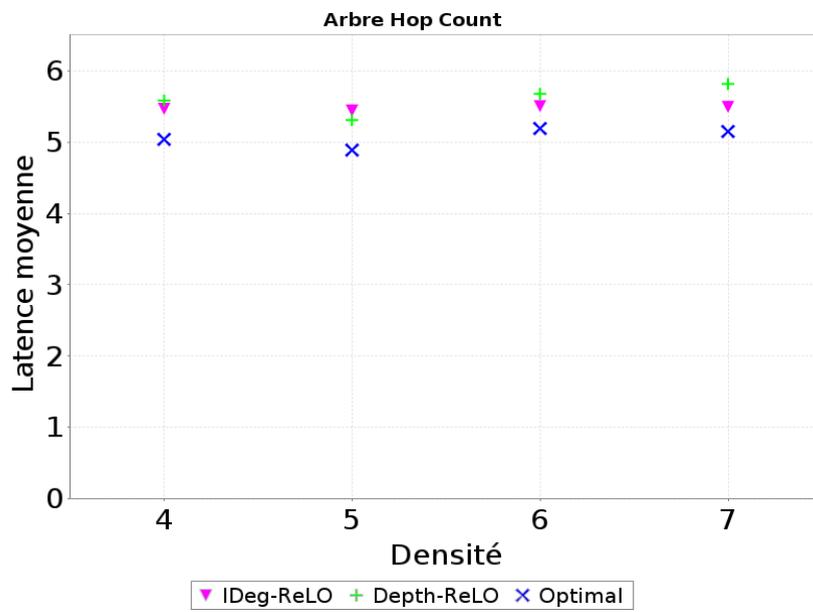


FIGURE B.10 – Latence moyenne de IDeg-ReLO et Depth-ReLO vs l'algorithme Optimal avec l'arbre Hop Count

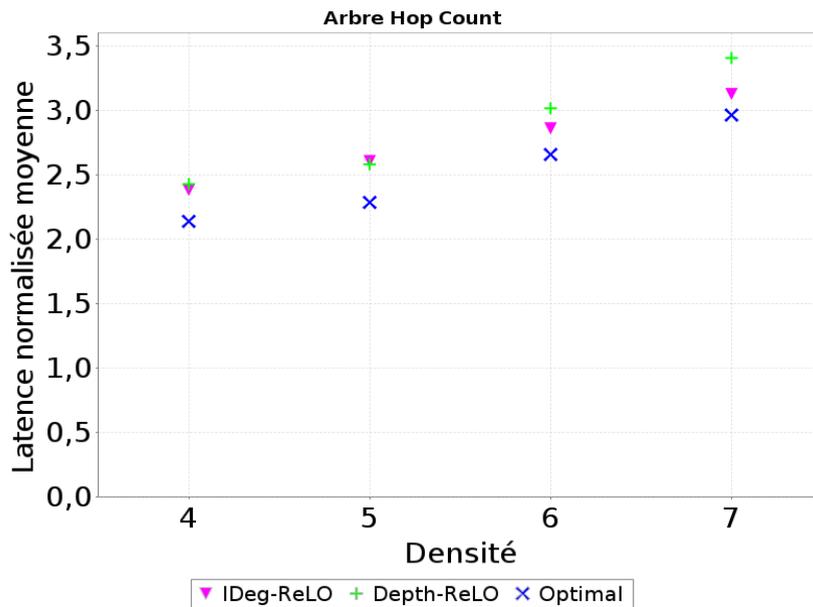


FIGURE B.11 – Latence normalisée moyenne de IDeg-ReLO et Depth-ReLO vs l'algorithme Optimal avec l'arbre Hop Count



## Résumé :

Traditionnellement, dans les réseaux de capteurs sans fil, les protocoles sont implémentés suivant le modèle en couches indépendamment les uns des autres; ainsi, les métriques impliquées dans plusieurs couches peuvent être affectées par les décisions incohérentes de ces couches. Par exemple, la latence des communications est affectée par certaines décisions du protocole de routage dans la couche Réseau et par les décisions du protocole MAC dans la couche Liaison de données. Dans ce cas, les approches cross-layer sont plus efficaces car elles exploitent la dépendance entre ces couches.

Dans ce contexte, ayant comme objectif la minimisation de la latence des communications, nous proposons différentes approches cross-layer concernant des arbres de routage et des ordonnancements TDMA. Tout d'abord, nous proposons une approche de routage utilisant des informations de l'ordonnancement TDMA. Ensuite, nous proposons plusieurs approches de construction d'ordonnancements TDMA utilisant des informations de l'arbre de routage. Enfin, nous proposons une approche qui produit un arbre de routage et un ordonnancement TDMA simultanément en combinant MAC et routage.

Des simulations extensives ont montré que nos contributions présentent de meilleurs résultats que l'état de l'art en terme de latence. D'autres métriques telles que la consommation énergétique des nœuds, le duty cycle des nœuds et le nombre de sauts des chemins de routage, ont également été considérées pour l'évaluation de nos contributions.

**Mots-clés :** réseaux de capteurs sans fil, approches cross-layer, ordonnancement TDMA, arbre de routage, latence des communications

## Abstract:

Traditionally, in Wireless Sensor Networks, protocols are designed independently in the layered protocol stack, and metrics involved in several layers can be affected. Communication latency is one metric example, impacted by both the routing protocol in the network layer and the MAC protocol in the data link layer. Cross-layer protocols are known to be more efficient when exploiting the dependencies between these layers.

In this context, and aiming to minimize the communication latency, we propose different cross-layer approaches concerning routing trees and TDMA schedules. First, we propose a routing approach using information of the TDMA schedule. Then, we propose different TDMA scheduling approaches which use information of the routing tree. Finally, we propose an approach that combines MAC and routing to produce a routing tree and a TDMA schedule simultaneously.

Extensive simulations have shown better results of our contributions compared to the state of the art in terms of latency. Other metrics such as the energy consumption, the duty cycle of the nodes and the number of hops in the routing paths were also considered during the evaluation of our contributions.

**Keywords:** wireless sensor networks, cross-layer approaches, TDMA schedules, routing tree, communication latency

The logo for SPIM (École doctorale SPIM) features a stylized 'S' followed by the letters 'P', 'I', and 'M' in a large, white, sans-serif font. A yellow horizontal bar is positioned to the left of the 'S'.