



HAL
open science

Ordonnement de l'activité des noeuds dans les réseaux ad hoc et les réseaux de capteurs sans fil

Pascale Minet, Ridha Soua, Ichrak Amdouni, Erwan Livolant, Saoucene Mahfoudh

► **To cite this version:**

Pascale Minet, Ridha Soua, Ichrak Amdouni, Erwan Livolant, Saoucene Mahfoudh. Ordonnement de l'activité des noeuds dans les réseaux ad hoc et les réseaux de capteurs sans fil. École d'Été Temps Réel, Aug 2013, Toulouse, France. pp.15. hal-00913303

HAL Id: hal-00913303

<https://hal.science/hal-00913303>

Submitted on 3 Dec 2013

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

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

Ordonnancement de l'activité des noeuds dans les réseaux ad hoc et les réseaux de capteurs sans fil

Pascale Minet, Ridha Soua, Ichrak Amdouni, Erwan Livolant et Saoucene Mahdoudh

Résumé—L'efficacité énergétique est une exigence majeure pour les réseaux sans fil où certains noeuds opèrent sur batterie. L'ordonnancement de l'activité des noeuds permet de distinguer périodes actives où la communication radio est possible et périodes inactives où la radio est arrêtée. Cet ordonnancement contribue largement à améliorer l'efficacité énergétique : d'une part en évitant les collisions entre transmissions conflictuelles et donc les retransmissions associées et d'autre part en permettant aux noeuds non concernés par la transmission de dormir pour économiser leur énergie. Parmi les solutions possibles, nous étudierons plus particulièrement le coloriage des noeuds. Après avoir défini le problème et ses différentes déclinaisons, nous donnerons sa complexité et proposerons SERENA, un algorithme de coloriage distribué qui s'adapte à la collecte de données. Nous présenterons OSERENA, l'optimisation de SERENA pour les réseaux denses et son utilisation dans le réseau de capteurs sans fil OCARI. Lorsque les noeuds ont des charges de trafic fortement hétérogènes, il devient plus intéressant d'effectuer une assignation de slots. Disposer d'un accès au médium multicanal et d'un puits multi-interfaces permet de gagner en nombre de slots nécessaires à la collecte de données, de réduire les interférences et d'améliorer la résistance aux perturbations. Nous présenterons une formalisation en ILP (Integer Linear Programming) du problème d'assignation de slots visant à minimiser le nombre de slots en profitant d'un environnement mono ou multicanal et d'un puits mono ou multi-interfaces. Nous donnerons des bornes théoriques sur le nombre optimal de slots dans diverses configurations et divers environnements (mono ou multicanal, puits mono ou multi-interfaces). Nous présenterons MODESA un algorithme centralisé d'allocation conjointe de canaux et slots temporels. Nous terminerons par quelques questions ouvertes.

Index Terms—Réseaux ad hoc, réseaux de capteurs sans fil, efficacité énergétique, ordonnancement, activité, coloriage, assignation de slots, multicanal, multi-interfaces, collecte de données.

I. INTRODUCTION

Les réseaux ad hoc et réseaux de capteurs sans fil RCF sont le support de nombreuses applications de type collecte de données. Les exigences de ces applications concernent :

- la qualité de service, notée QoS, qui devra être maximisée. Ainsi les interférences qui ont un effet négatif sur la QoS devront être minimisées.
- la fiabilité des transmissions, qui devra être maximisée, malgré un environnement qui n'est pas toujours très coopératif.
- l'efficacité énergétique, qui devra être assurée en raison de l'énergie limitée dont disposent les noeuds.
- la bande passante, qui devra être utilisée efficacement, ainsi que la mémoire et la capacité de traitement des noeuds qui sont des ressources limitées.

Nous considérons des applications de type collecte de données : chaque noeud produit des données qui doivent être transmises au puits. Nous supposons qu'un arbre de routage ayant pour racine le puits est donné. Le rôle de tout noeud est alors de transmettre les données dont il dispose (données générées localement ou produites par ses descendants dans l'arbre de routage) à son parent dans l'arbre. Nous distinguons deux types de collecte de données :

- avec agrégation : chaque noeud agrège les données reçues de ses enfants et les transmet avant de les transmettre à son parent. Ceci suppose que la taille des slots temporels utilisés pour la transmission de ces données est suffisante pour permettre cette agrégation.
- sans agrégation : (en anglais : raw data convergecast) où aucune agrégation n'est effectuée sur les noeuds intermédiaires (i.e. noeuds entre le noeud producteur et le puits).

Dans les réseaux sans fil supportant ce type d'application, les accès au médium sont organisés selon une supertrame. Cette supertrame est délimitée par une balise comprenant une période avec contention avec accès en CSMA/CA et une période sans contention où les accès sont effectués dans des slots selon un ordonnancement préétabli. Lorsque cette supertrame se répète, la durée entre deux balises successives détermine la longueur du cycle. La Figure 1 illustre la structure de la supertrame.

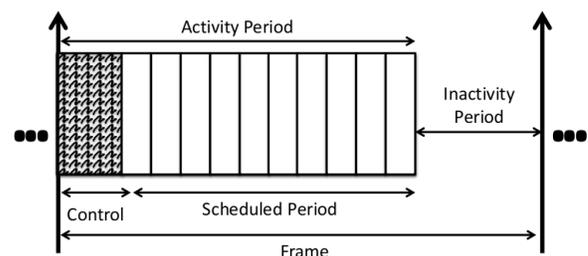


FIGURE 1. Structure d'une supertrame.

Notre objectif est de minimiser le nombre de slots nécessaires à la collecte de données. En effet, minimiser le nombre de slots permet en effet de diminuer les délais de collecte de données. Considérons une supertrame telle que chaque noeud dispose des slots nécessaires pour transmettre ses données locales et les données reçues de ses enfants. Supposons de plus que les noeuds produisent des données périodiquement et maintiennent une file FIFO des messages à transmettre. Le pire délai de collecte de données est obtenu par le noeud qui génère ses données juste après le dernier slot qui lui a été alloué dans le cycle et donc doit attendre

le prochain cycle pour transmettre. Notons cependant que sur les noeuds intermédiaires, l'ordonnancement des slots a été conçu de telle manière que tout message reçu peut être propagé dans le même cycle. Le dernier message de cette collecte est remis au puits dans le dernier slot de la période sans contention. Il s'en suit que le pire délai de remise est égal à la durée du cycle plus la durée de la période sans contention. Remarquons que par ailleurs, plus ce délai est petit, plus les données collectées dans cette supertrame seront cohérentes temporellement.

Par ailleurs, le multicanal permet d'utiliser simultanément plusieurs canaux dans un même réseau. Notons qu'au niveau de l'accès au médium, une condition nécessaire pour qu'un message soit bien reçu est que l'émetteur et le récepteur soient sur le même canal. Les progrès technologiques permettent à un noeud de switcher dynamiquement d'un canal à un autre. Ceci est utilisé par exemple pour protéger les transmissions des perturbations sur un canal, perturbations pouvant être causées par une source extérieure. Ainsi le multicanal permet d'améliorer les performances de la collecte de données en termes de débit supporté, délais de remise, taux de remise et en robustesse face aux perturbations. Si le puits est équipé de plusieurs interfaces, il peut alors recevoir simultanément sur chacune de ces interfaces. Ceci a pour effet d'améliorer encore le parallélisme et donc les performances de la collecte de données. C'est pourquoi dans ce papier, nous nous intéressons aux ordonnancements minimisant le nombre de slots nécessaires à la collecte de données, dans un réseau ad hoc ou un réseau de capteurs sans fil, multicanal avec un puits pouvant être équipé de plusieurs interfaces.

II. ETAT DE L'ART

A. Efficacité énergétique

Nous pouvons identifier cinq grandes classes de techniques visant l'efficacité énergétique [1], [2] :

- 1) *la réduction des données* produites, traitées et transmises (ex. : compression, agrégation de données) ;
- 2) *la réduction du surcoût des protocoles* (ex. : adapter la périodicité des messages de contrôle selon la stabilité du réseau) ;
- 3) *le routage économe en énergie* : (ex. : protocoles opportunistes, hiérarchiques, géographiques....)
- 4) *le contrôle de topologie* : il ajuste la puissance de transmission en fonction du récepteur.
- 5) *l'ordonnancement de l'activité des noeuds* : il permet aux noeuds d'alterner les périodes actives et les périodes de sommeil, durant lesquelles la radio est arrêtée afin d'économiser l'énergie. L'idée consiste à revenir à cet état sommeil dès que possible, tout en satisfaisant les exigences de l'application.

La Table III montre sur quelle perte énergétique agit chaque classe de technique. Le symbole 'M' indique un impact majeur et 'S' un impact secondaire.

TABLE I
IMPACT DE CES TECHNIQUES SUR LES PERTES ÉNERGÉTIQUES.[1]

	capture trt	comm.	collision	ecoute active	message controle	ecoute passive	interf.
Reduction donnees	M	M	S	S	-	-	S
Reduction surcout	-	M	S	S	M	-	S
Routage energie	-	M	S	M	S	-	M
Ordo activite	M	M	M	M	S	M	M
Controle topologie	-	M	M	M	-	-	M

B. Ordonnancement de l'activité des noeuds

L'ordonnancement de l'activité des noeuds peut être appliqué à un niveau de granularité haut ou bas.

1) Haut niveau de granularité:

A haut niveau de granularité, l'ordonnancement consiste à sélectionner parmi les noeuds déployés dans le réseau ceux qui doivent rester actifs pour satisfaire les exigences applicatives. Les autres noeuds, dits redondants, peuvent dormir pour économiser leur énergie. Dans [8], [9] la sélection de l'ensemble minimum des noeuds actifs nécessaires à la couverture de la zone surveillée est obtenu par programmation linéaire. Dans GAF [10], la zone surveillée est considérée comme une grille virtuelle découpée en cellules. Dans chaque cellule, un seul noeud est actif, les autres noeuds dorment. Dans ce papier, seule la connectivité est prise en compte. SPAN [11] assure également la connectivité par le biais des coordinateurs. Ceux-ci sont chargés du routage multi-saut tandis que les autres capteurs peuvent être arrêtés. Dans [12], les critères de sélection des noeuds actifs sont basés sur la couverture et la connectivité. Les auteurs de [13] montrent que la durée de vie du réseau est maximisée quand les capteurs déployés sont répartis dans des ensembles non-disjoints, de telle sorte qu'à tout instant un seul ensemble est actif.

2) Bas niveau de granularité:

Avec un ordonnancement de l'activité des noeuds à un niveau bas, même les noeuds qui ont été sélectionnés comme actifs pour satisfaire les exigences applicatives peuvent dormir lorsqu'ils n'ont aucun message à transmettre et à recevoir. C'est pourquoi cet ordonnancement est fortement couplé avec l'accès au médium. Nous distinguons trois types de protocoles MAC :

- *basés sur TDMA* : Cette technique assure aux capteurs un accès au médium sans collision. Le temps est divisé en slots assignés aux noeuds. Ce protocole est bien adapté au trafic périodique. TRAMA [14] est un des premiers protocoles proposés qui s'adapte au trafic. Pour chaque slot, un seul transmetteur dans le voisinage jusqu'à 2 sauts est sélectionné selon un algorithme distribué. FLAMA [15] est dérivé de TRAMA et conçu pour les applications de collecte de données. FLAMA évite les échanges périodiques d'information entre voisins jusqu'à 2 sauts en transmettant sur demande uniquement. FlexiTP [16], pro-

posé également pour les applications de collecte, construit un arbre de collecte et utilise un algorithme de parcours de l'arbre en profondeur pour assigner les slots. Les noeuds peuvent demander ou libérer des slots en fonction des informations dans leurs tables. Un protocole récent, appelé TDMA-ASAP, [17], proposé pour les applications de collecte de données, intègre un algorithme de coloriage dans l'accès au médium. En autorisant un noeud à voler à son frère un slot inutilisé, ce protocole peut s'adapter à des conditions de trafic variées.

- *basés sur la contention* : S-MAC [21] essaie de forcer les noeuds voisins à adopter le même ordonnancement des périodes d'activité et de sommeil. Pour cela, les noeuds voisins doivent échanger leurs ordonnancements avant de pouvoir échanger des données. Toutefois, les périodes d'écoute passive et de sommeil ne peuvent plus changer après le déploiement des noeuds. T-MAC [22] améliore S-MAC en autorisant les noeuds à se rendormir s'ils n'ont reçu aucun message dans une durée spécifiée. D-MAC [23] ordonnance les périodes d'activité et de sommeil selon la profondeur du noeud dans l'arbre de collecte, afin d'assurer que les noeuds intermédiaires seront actifs durant le transfert des messages vers le puits.
- *hybrides* : ces protocoles se comportent comme TDMA ou comme CSMA pour s'adapter à la variabilité du trafic. Le plus connu est Z-MAC [18]. Il agit comme CSMA en situation de faible trafic et comme TDMA en fort trafic.

C. Coloriage des noeuds ou des liens

Le coloriage des graphes peut être utilisé pour améliorer les performances de TDMA en autorisant tous les noeuds (respectivement les liens) de même couleur à transmettre simultanément. Nous distinguons le coloriage des noeuds et le coloriage des liens.

1) Coloriage des noeuds:

Colorier les noeuds d'un graphe consiste à assigner une couleur à chaque noeud de telle sorte que deux noeuds ayant un lien commun n'ont pas la même couleur, tout en minimisant le nombre de couleurs utilisées. Colorier les noeuds d'un graphe G avec k couleurs est équivalent à partitionner les noeuds de ce graphe en k sous-ensembles tels que tous les noeuds d'un même sous-ensemble ont la même couleur. Le plus petit nombre de couleurs utilisé pour colorier les noeuds d'un graphe G est appelé nombre chromatique. Les premiers algorithmes de coloriage étaient centralisés, comme par exemple [24], [25], [26] et D-satur [27] qui colorie en premier le noeud de plus grand degré. Ils peuvent être aléatoires comme [24] où les noeuds choisissent aléatoirement leur couleur. Si un noeud a choisi la même couleur que son voisin, il doit choisir une autre couleur. cet algorithme est simple et rapide : il s'exécute en $O(\log n)$ mais nécessite $\Delta + 1$ couleurs avec Δ le degré maximum d'un noeud. Dans les algorithmes distribués déterministes comme DLF (Distributed Largest First) [28], les noeuds de plus fort degré se colorient les premiers. La complexité de cet algorithme est en $O(\Delta^2 \log n)$. Une autre heuristique bien

connue est First Fit [32] qui assigne la première couleur disponible au noeud non coloré de plus grande priorité. Dans Z-MAC [18], l'algorithme de coloriage utilisé est DRAND [19]. Il s'agit d'un coloriage de noeuds randomisé à 2-sauts. Il a l'avantage de ne pas dépendre du nombre de noeuds dans le réseau au prix d'une convergence asymptotique.

D'autres algorithmes procèdent par améliorations successives d'un coloriage initial [33], [34]. Parmi ces algorithmes, figure TabuCOL [29]. Ces algorithmes fournissent des résultats remarquables pour des petits graphes uniquement [34].

Certains travaux [30], [31] s'intéressent aux performances pire cas des algorithmes de coloriage : ils prouvent que pour tout graphe d'une certaine famille, l'algorithme utilise un nombre de couleurs au plus égal à α fois le nombre optimal de couleurs. Un tel algorithme est dit être une α -approximation. Plus généralement, la complexité en temps [48] d'un algorithme distribué de coloriage est évaluée en nombre de rounds. Par définition, dans un round, un noeud reçoit les messages émis par ses voisins à un saut, les traite et diffuse son message à ses voisins à un saut. La complexité spatiale est donnée par le nombre et la taille des messages transmis par noeud.

2) Coloriage des liens:

Le coloriage des liens d'un graphe est une assignation de couleurs aux liens du graphe de telle sorte que les liens incidents en un même noeud ont des couleurs différentes. Le coloriage des liens peut être considéré comme un coloriage des noeuds du graphe de liens. Colorier les liens d'un graphe G avec k couleurs est équivalent à partitionner ces liens en k sous-ensembles.

Durand et al. [35], [36] appliquent le coloriage des liens dans des graphes bipartites pour ordonnancer les transferts de données entre serveurs et clients. L'algorithme de coloriage est distribué et randomisé, comme l'algorithme [24] : chaque noeud sélectionne aléatoirement une couleur pour chacun de ses liens ; il ne conserve la couleur choisie que si elle n'est utilisée par aucun des liens conflictuels. Sinon, le noeud de plus faible degré doit changer de couleur. Marathe et al. [37] proposent un algorithme distribué simple où chaque lien dispose d'une palette de $(1 + \epsilon)\Delta$ couleurs. Chaque lien choisit une couleur de la palette et vérifie qu'il n'entre en conflit avec aucun des liens voisins. Si tel est le cas, la couleur est adoptée et retirée des couleurs disponibles. Sinon, le lien fait une nouvelle tentative dans le round suivant. Comme le coloriage des liens est un problème NP-complet, des heuristiques sont utilisées, plus particulièrement pour les grands graphes. Les auteurs de [38] comparent les performances de différentes heuristiques de coloriage des liens pour des benchmarks standards sur des petits graphes (< 500 noeuds). Dans [39], les liens de réseaux planaires sont coloriés avec $\Delta + 4$ couleurs, avec $\Delta \geq 5$, les liens de plus haute priorité choisissent leurs couleurs en premier.

3) Classification des algorithmes de coloriage:

Nous proposons une classification des algorithmes de coloriage dans le Tableau II.

TABLE II
CLASSIFICATION DES ALGORITHMES DE COLORIAGE

	central. distrib.	déterm. aléa.	noeud lien	comm.	pour
TRAMA [14]	distrib.	déterm.	lien	unicast	général
FLAMA [15]	distrib.	déterm.	lien	unicast arbre	collecte
ZMAC DRAND [19]	distrib.	aléa.	noeud	unicast +diff.	général
TDMA- ASAP [17]	central.*	déterm.	noeud	unicast arbre	collecte
FlexiTP [16]	distrib.	déterm.	lien	unicast arbre+diff.	collecte
SERENA [49]	distrib.	déterm.	noeud	unicast* +diff.	général collecte
OSERENA [50]	distrib.	déterm..	noeud	unicast* +diff.	collecte*

Légende : unicast* : unicast avec acquittement immédiat.
collecte* : optimisé pour des réseaux denses.
central* : seule la version centralisée est décrite dans [17].

4) Applications du coloriage de graphes aux réseaux:

De nombreux algorithmes de coloriage sont utilisés dans les réseaux (filaires et sans fil) pour rendre l'accès au médium plus efficace par le biais d'une allocation des fréquences ou des slots temporels. L'allocation des fréquences basée sur un coloriage est utilisée dans :

- les réseaux cellulaires [40] pour limiter le nombre de fréquences allouées aux stations de base ;
- les réseaux mesh [41] et les réseaux de capteurs [42] pour minimiser les interférences ;
- les réseaux radio cognitive [43], [44] pour une utilisation opportuniste des fréquences disponibles dans le spectre.

L'allocation de slots permet d'allouer des slots aux noeuds [17], [49] ou aux liens [45], [46] pour ordonnancer les activités des noeuds ou des liens. Chaque couleur utilisée par l'algorithme de coloriage est associée à un ou plusieurs slots temporels. Dans un slot temporel sont ordonnancés tous les noeuds (respectivement les liens qui partagent cette couleur. Ainsi les collisions sont évitées. Notons que l'utilisation de la bande passante est également améliorée par la réutilisation spatiale. Dans le coloriage de liens, les slots de transmission sont alloués à des liens et donc seuls les émetteurs et récepteurs de ces liens sont actifs dans ce slot. Dans le coloriage des noeuds, les slots sont alloués aux noeuds émetteurs qui utilisent le slot selon leurs besoins (ex. : transmission en unicast ou en broadcast) ; le corollaire est que tous les voisins d'un noeud émetteur doivent être éveillés au cas où une transmission leur serait destinée.

D. Communications multicanal

Dans le but d'établir une classification des protocoles d'assignation de canaux, nous posons trois questions :

- (1) A quelle fréquence est réalisée l'assignation des canaux ? La réponse permet de distinguer les assignations statiques, semi-dynamiques et dynamiques (voir Section II-D1).
- (2) Quel canal est sélectionné ? La réponse est donnée par la politique de sélection des canaux (voir Section II-D2).
- (3) Comment les noeuds se coordonnent-ils ? La réponse est donnée par la technique de coordination utilisée (voir Section

II-D3).

1) Méthode d'assignation des canaux :

Nous distinguons trois catégories de méthodes d'assignation des canaux :

- *statique* : l'assignation des canaux est effectuée une fois, lors de l'initialisation du réseau. Ceci a l'avantage de n'induire qu'un faible surcoût mais peut donner de faibles performances si l'environnement dans lequel le RCSF est placé évolue (e.g. niveau excessif d'interférence). Dans un tel cas, une nouvelle assignation de canal est effectuée et sera utilisée pendant une longue période. Avec une assignation statique, le surcoût induit par l'assignation des canaux est minimisé. TMCP [54] et MCRT [55] adoptent une assignation statique de canaux.
- *semi-dynamique* : lorsque l'assignation de canaux est effectuée à une fréquence moyenne pour s'adapter aux changements de conditions des canaux ou du trafic. Cette assignation peut être faite périodiquement ou sur événement. De nombreux protocoles appartiennent à cette famille : MMSN [52], TACA [56], EM-MAC [57], RMCA [58] et ARM [59].
- *dynamique* : lorsque l'assignation de canaux est effectuée très fréquemment, typiquement avant chaque transmission. Une assignation dynamique (e.g. Y-MAC [60] and MuChMAC [61]) permet au protocole de prendre de meilleures décisions (i.e. sélection du meilleur canal basé sur une information plus à jour) mais avec une augmentation du surcoût.

La tendance dans les protocoles d'assignation de canaux est de favoriser les assignations semi-dynamiques qui représentent le meilleur compromis entre faible surcoût et adaptativité élevée.

2) Politique de sélection des canaux :

Généralement, le canal est sélectionné dans une liste des canaux préférés. Pour éviter d'utiliser un canal occupé du côté du récepteur, cette liste est calculée soit par le récepteur comme dans EM-MAC [57] or en échangeant les listes de l'émetteur et du récepteur comme dans MMAC [62]. Une sélection dynamique des canaux est cruciale pour éviter les canaux surchargés ou perturbés par des interférences. De plus, elle permet d'équilibrer la charge entre tous les canaux disponibles, assurant ainsi une meilleure adaptation aux changements de trafic durant la vie du réseau.

3) Coordination des noeuds sur les canaux:

La communication entre noeuds autorisés à switcher d'un canal à un autre n'est possible que si l'émetteur et le récepteur sont sur le même canal durant la transmission. Ceci nécessite une coordination sur le canal. Les protocoles de coordination diffèrent dans la manière dont les noeuds négocient le canal à utiliser et résolvent la contention au médium. Cette coordination peut être :

- *implicite* lorsque les noeuds sont implicitement d'accord sur quel canal switcher et quand le changement de canal intervient. Parmi les protocoles à coordination implicite, nous distinguons :

TABLE III
CLASSIFICATION ET ANALYSE DES PROTOCOLES D'ASSIGNATION DE CANAUX.

		Static		Semi-dynamic					Dynamic	
		TMCP [54]	MCRT [55]	MMSN [52]	TACA [56]	EM-MAC [57]	RMCA [58]	ARM [59]	Y-MAC [60]	MuChMAC [61]
Goals	Parallel transm.	✓		✓		✓				
	Capacity				✓					
	Robustness				✓	✓	✓			
	Delays	✓	✓							
Properties	Adaptive				✓					
	Broadcast	✓		✓				✓	✓	✓
	QoS		✓		✓		✓			
	Energy							✓	✓	✓
	Channel switch	✓	✓							
	Reduced interf.	✓	✓		✓	✓				
	Scalable							✓	✓	
	RR					✓			✓	✓
Channel selection	least chosen			✓						
	least loaded				✓					
	proba.						✓	✓		
	Control channel							✓	✓	
Channel coordin.	Splitting phase									
	Game theory						✓			
	Node coloring			✓	✓					
	Comm. island	✓	✓							
	Freq. hopping					✓			✓	✓
	MAC	CSMA/CA	CSMA/CA	CSMA/CA	CSMA/CA			CSMA/CA	TDMA	TDMA XMAC
Solution maturity	simul testbed	simul	simul	simul	testbed	simul	simul testbed	simul	testbed	

- *ceux basés sur les îlots de communication* : Chaque îlot de communication utilise son propre canal pour ses communications. L'avantage de ces protocoles est d'éviter les changements de canal, coûteux en temps, tout au long du chemin suivi par un message. La réutilisation spatiale de la bande passante au sein d'un îlot doit être fournie par le protocole MAC, comme dans OCARI [6] avec le coloriage des noeuds.
- *ceux utilisant le saut de fréquence* : les noeuds sautent de canal en canal selon soit une séquence commune de sauts (e.g. Bluetooth [63]), soit des séquences indépendantes de sauts (e.g. EM-MAC [57]). Le principal avantage du saut de fréquences réside dans la meilleure immunité au bruit. Son inconvénient est la nécessité d'une forte synchronisation entre noeuds.
- *explicite* lorsque les noeuds négocient la sélection du canal ou l'ordonnancement pour coordonner les changements de canal. Il existe différentes techniques :
 - *canal de contrôle dédié* : un canal est dédié exclusivement au trafic de contrôle, tandis que les autres canaux sont utilisés pour le trafic de données (e.g. ARM [59]). Cette méthode n'exige pas une synchronisation temporelle fine entre les noeuds, mais peut réduire fortement l'efficacité de l'utilisation de la bande passante si les données échangées après chaque rendez-vous ou le nombre de canaux disponibles est petit.
 - *alternance de phases* : les phases de contrôle et de données alternent. Dans la phase de contrôle, tous les noeuds écoutent le canal de contrôle pour parvenir à un accord. Dans la phase de données, les noeuds switchent sur leurs canaux respectifs qui ont été négociés durant la phase de contrôle (voir par exemple MMSN [52]

- pour un accès MAC avec contention et Y-MAC [60] pour un accès ordonnancé). Notons que les canaux dédiés aux données sont inutilisés en phase de contrôle.
- *théorie des jeux* : Tous les noeuds sont des joueurs et les canaux disponibles sont les actions des joueurs. Le but du jeu est de réduire les interférences dans le réseau. Chaque joueur sélectionne un canal différent de ceux choisis par les joueurs interférents. Les stratégies des joueurs constituent la coordination et l'assignation des canaux [65], [58]. Cette technique innovante demande l'échange d'une information limitée pour assigner les canaux. Cependant la convergence du jeu basé sur les protocoles n'est pas triviale.
- *coloriage des noeuds* : Un problème d'assignation de canaux peut être modélisé par un problème de coloriage de graphe. L'objectif est de maximiser le parallélisme des transmissions et minimiser les interférences. Cette technique consiste à assigner un canal à chaque noeud de sorte que deux noeuds adjacents ont des canaux différents et le nombre de canaux utilisés est minimum (e.g. MMSN [52] et TACA [56]). Cette technique peut imposer un grand nombre de switches lors de la transmission de bout-en-bout d'un message.

Indépendamment de ces critères, la coordination des canaux est dite orientée émetteur lorsque le canal est assigné à l'émetteur, le récepteur devant switcher sur le canal de l'émetteur. De façon similaire, la coordination peut être orientée récepteur. Elle peut également être orientée émetteur et récepteur lorsque tous deux doivent switcher sur le canal assigné pour la transmission (e.g. saut de fréquences).

4) Taxonomie des protocoles d'assignation de canaux:

La taxonomie des protocoles d'assignation de canaux que nous proposons est basée sur quatre questions : 1) quel est le but recherché ? 2) quand l'assignation de canaux est-elle effectuée ? 3) quel canal est sélectionné ? and 4) comment est effectuée la coordination sur les canaux ? Elle est illustrée par le Tableau III.

III. ORDONNANCEMENT DE L'ACTIVITÉ DES NOEUDS : COLORIAGE DES NOEUDS

Nous généralisons l'étude du problème de coloriage des noeuds en définissant le coloriage des noeuds à h -sauts pour tout entier $h \geq 1$.

A. Définitions et complexité du coloriage de noeuds à h -sauts

Nous donnons tout d'abord quelques définitions :

Définition 1: Le coloriage des noeuds à h -sauts en mode général consiste à assigner des couleurs aux noeuds de telle sorte que : (i) il n'existe pas deux noeuds à k -sauts, $1 \leq k \leq h$, qui utilisent la même couleur ;

(ii) le nombre de couleurs utilisées est minimisé.

De façon similaire, nous définissons le mode stratégique pour les applications de collecte de données.

Définition 2: Le coloriage des noeuds à h -sauts en mode stratégique consiste à assigner des couleurs aux noeuds de telle sorte que : (i) il n'existe pas deux noeuds à k -sauts, $1 \leq k \leq h$, qui utilisent la même couleur ;

(ii) le nombre de couleurs utilisées est minimisé.

(iii) et satisfaisant la Contrainte 1 : tout noeud doit sélectionner une couleur strictement supérieure à la couleur de son parent dans l'arbre de collecte de données.

Définition 3: Pour tout entier $h > 0$, un coloriage des noeuds à h -sauts en mode général (respectivement en mode stratégique) est dit valide si et seulement si il satisfait la définition 1 (respectivement la définition 2).

Définition 4: Un coloriage valide des noeuds à h -sauts en mode général (respectivement en mode stratégique) est dit optimal si et seulement si aucun coloriage valide des noeuds à h -sauts en mode général (respectivement en mode stratégique) utilise moins de couleurs que ce coloriage.

Il a été montré dans [47] que le coloriage des noeuds à un saut est NP-complet. Nous avons généralisé ce résultat pour tout coloriage à h sauts avec $h \geq 1$ et pour les modes général et stratégique.

Théorème 1: Le problème de décision du coloriage des noeuds à h -sauts ($h \geq 1$) en mode général et en mode stratégique est NP-complet.

Pour la preuve, voir [7].

B. SERENA : un algorithme distribué de coloriage

L'objectif de SERENA est d'ordonner l'activité des noeuds d'un réseau mobile ad hoc ou d'un réseau de capteurs sans fil RCSF en utilisant les couleurs. En effet, SERENA est utilisé avec un accès au médium basé sur un cycle composé de slots temporels. Chaque couleur est associée à un slot

temporel. Durant un slot, tous les noeuds ayant la couleur associée à ce slot peuvent transmettre. Chaque noeud est réveillé dans les slots de sa couleur pour transmettre ses messages, mais aussi dans les slots de ses voisins pour recevoir les messages qui lui sont destinés. Le reste du temps, le noeud dort (arrêt de sa radio).

SERENA est un algorithme distribué de coloriage qui utilise le cross-layering avec les couches MAC et applicatives pour fournir le coloriage le plus adapté. Cette adaptation s'effectue par le biais de la détermination des ensembles des noeuds conflictuels. L'ensemble des noeuds conflictuels d'un noeud u quelconque est noté $\mathcal{N}(u)$. Il est déterminé par :

- le type de communications supportées : point-à-point et/ou diffusion ;
- le type d'application considérée : général lorsqu'un noeud est susceptible d'échanger des données avec n'importe quel voisin ou au contraire de type arbre lorsqu'un noeud n'échange de l'information qu'avec son parent et ses enfants dans l'arbre de collecte des données.
- le type d'acquiescement des transmissions point-à-point : immédiat ou différé. L'acquiescement est dit immédiat si et seulement si il est effectué dans le même slot que le message qu'il acquitte. En d'autres termes, le récepteur acquitte le message dans le slot de l'émetteur.

1) Hypothèses:

Nous supposons un environnement idéal où :

Hypothèse H0 : Tous les liens sont symétriques et stables.

Hypothèse H1 : Chaque noeud a une adresse unique dans le réseau.

Hypothèse H2 : Aucun noeud n'empêche la réception correcte d'un autre noeud hors de sa portée radio.

Hypothèse H3 : Il n'y a pas de perte de message.

2) Choix de conception:

Un coloriage des noeuds est préféré au coloriage des liens pour différentes raisons. Lorsque les liens sont coloriés, chaque noeud doit transmettre un message qu'il souhaite diffuser autant de fois que de liens sortants. Avec un coloriage des noeuds, tous les voisins de l'émetteur sont actifs durant la diffusion du message et peuvent donc recevoir correctement le message diffusé. Rappelons que dans les réseaux ad hoc, la découverte des voisins est faite par diffusion périodique de messages *Hello*.

Considérons l'exemple illustré dans la Figure 2 : les noeuds A et D ont la même couleur, résultat d'un coloriage à 2 sauts. Ils peuvent donc transmettre simultanément. Ceci va résulter en une collision sur le noeud B, lorsque l'acquiescement immédiat est utilisé. En effet, A transmet un message à B tandis que C acquitte le message reçu de D. Cet exemple montre que l'acquiescement immédiat crée des collisions lorsqu'un coloriage à 2 sauts est utilisé.

C'est pourquoi, $\mathcal{N}(u)$, l'ensemble des noeuds conflictuels d'un noeud u quelconque, est inclus dans l'ensemble des voisins jusqu'à 3 sauts de u , lorsque l'acquiescement immédiat est utilisé.

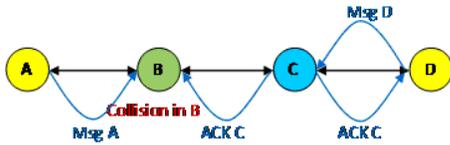


FIGURE 2. Collision avec un coloriage à 2 sauts et acquittement immédiat.

3) Principes de SERENA:

Dans SERENA, chaque noeud u procède comme suit pour se colorier :

- 1) Le noeud u détermine l'ensemble $\mathcal{N}(u)$ des noeuds qui ne peuvent pas avoir la même couleur que lui. Ainsi dans un coloriage à 3-sauts, l'ensemble $\mathcal{N}(u)$ contient tous les voisins jusqu'à 3-sauts de u .
- 2) Le noeud u calcule sa priorité, $priority(u)$. Dans une application de collecte de données, la priorité d'un noeud est égal à son nombre de descendants dans l'arbre de collecte.
- 3) Le noeud u applique les deux règles suivantes :
 - **Règle R1** : Le noeud u se colorie si et seulement si il a une priorité strictement supérieure à la priorité de tout noeud non colorié dans $\mathcal{N}(u)$.
 - **Règle R2** : Pour se colorier, le noeud u prend la plus petite couleur inutilisée dans $\mathcal{N}(u)$. Pour une application de collecte de données, la couleur prise par u doit être supérieure à celle de son parent.

La règle R2 permet de minimiser le délai de remontée des données au puits pour les applications de collecte de données. En effet, ordonnancer les couleurs par ordre décroissant permet d'assurer qu'un parent accède au médium après ses enfants et donc peut faire remonter dans le même cycle les données transmises par ses enfants. Ainsi, SERENA assure la remontée des données collectées en un seul cycle pour ces applications.

SERENA, algorithme distribué de coloriage, procède par itérations ou rounds, où les noeuds échangent leur message *Color*. Ce message *Color* inclut l'adresse, la priorité et la couleur du noeud émetteur u et de ses noeuds conflictuels (jusqu'à 2 sauts pour un coloriage à 3 sauts). Les données stockées localement par u contiennent ces informations (jusqu'à 3 sauts pour un coloriage à 3-sauts). Notons que pour des raisons de robustesse, un coloriage à 3 sauts peut être retenu pour une application de collecte de données où toutes les transmissions de données se font selon l'arbre. Ceci permet de remplacer à tout noeud de remplacer son parent défaillant par un autre noeud voisin à un saut sans invalider le coloriage.

4) Propriétés de SERENA:

En environnement idéal, nous avons prouvé les propriétés suivantes de SERENA, algorithme distribué de coloriage des noeuds :

- Le coloriage fourni par SERENA en mode général est identique au coloriage FirstFit centralisé utilisant la même priorité des noeuds.
- Le coloriage fourni par SERENA en mode stratégique

est identique au coloriage FirstFit centralisé utilisant la même priorité des noeuds et affectant à un noeud la plus petite couleur disponible supérieure à celle de son parent.

- Pour les applications générales, SERENA en mode général fournit un coloriage valide à h -sauts, avec $h \geq 1$.

- Pour les applications de collecte de données, SERENA en mode stratégique fournit un coloriage valide à 2 et 3 sauts, et les données collectées sont remontées au puits en un seul cycle.

Remarquons que l'équivalence à First Fit centralisé est très importante car elle permet de prédire facilement les couleurs des noeuds. Il est donc plus facile de prouver la validité du coloriage fourni. Par ailleurs, nous avons réalisé de nombreuses simulations et montré que :

- Les performances de SERENA en mode général à 2 sauts sont meilleures (moins de messages échangés et nombre de couleurs sensiblement plus faible) que celles de DLF étendu à 2 sauts (voir [3]).

- Les performances de SERENA en mode stratégique à 2 sauts sont meilleures que celles :

- * de DRAND, coloriage utilisé dans Z-MAC, car SERENA permet une remontée au puits des informations collectées en un seul cycle (voir [6]);

- * du coloriage utilisé dans TDMA-ASAP pour les applications de collecte de données, car SERENA utilise sensiblement moins de couleurs (voir [5]).

C. OSERENA : une optimisation pour les réseaux denses

Le surcoût de SERENA en termes du nombre d'octets échangés et de la taille mémoire nécessaire pour stocker la priorité et la couleur de tout noeud jusqu'à 3-sauts trois sauts peut être inacceptable pour des réseaux de capteurs sans fil denses avec des capacités de stockage et traitement limitées, une énergie résiduelle des noeuds limitée également et une petite taille de trame MAC. C'est pourquoi, nous proposons une optimisation de SERENA, appelée OSERENA, réduisant la taille des messages échangés ainsi que la taille des données maintenues localement, tout en conservant une faible complexité. Nous montrons également que la réduction du surcoût n'induit pas un accroissement du temps de convergence. Ainsi, OSERENA est adapté aux réseaux denses.

1) Principes d'optimisation:

Les principes d'optimisation d'OSERENA sont basés sur les remarques suivantes :

- Tout noeud u quelconque n'a pas à savoir qui parmi ses noeuds conflictuels jusqu'à 3 sauts a quelle priorité mais seulement s'il existe à un saut, deux sauts ou trois sauts, un noeud conflictuel avec une priorité supérieure à la sienne. De même, u ne doit pas savoir quel noeud conflictuel a quelle couleur, mais seulement quelles sont les couleurs utilisées par ses noeuds conflictuels à un saut, à deux sauts et à trois sauts.

- En conséquence, il suffit que le noeud u envoie :

* sa priorité, la plus haute priorité de ses noeuds conflictuels non coloriés à 1 saut et la plus haute priorité de ses noeuds conflictuels non coloriés à 2 sauts.

* sa couleur, les couleurs prises par ses noeuds conflictuels à 1 saut et les couleurs prises par ses noeuds conflictuels à 2 sauts.

Remarquons cependant qu'envoyer seulement une priorité pour les noeuds conflictuels à un saut et une priorité pour les noeuds conflictuels à 2 sauts ne suffit pas à garantir le même nombre de rounds que SERENA. En effet, considérons un noeud u qui s'est colorié au round r , un noeud v à 3 sauts de u ne sait pas au round $r + 2$ qu'il a une priorité plus grande que ses noeuds conflictuels non coloriés. C'est pourquoi, il faut transmettre plus d'une priorité pour les noeuds conflictuels à 2 sauts et plus d'une priorité pour les noeuds conflictuels à 1 saut, sinon le nombre de rounds nécessaires au coloriage sera augmenté par rapport à celui de SERENA.

2) Principes d'OSERENA:

Nous notons $color_bitmap1$, $color_bitmap2$ et $color_bitmap3$ les bitmaps des couleurs utilisés par les noeuds conflictuels respectivement à 1, 2 et 3 sauts. Nous définissons :

- $max_prio1(u)$ comme étant les 4 plus grandes priorités des noeuds conflictuels non coloriés à 1 saut de u , si 4 tels noeuds existent ; sinon les priorités manquantes sont \emptyset .

- $max_prio2(u)$ comme étant les 3 plus fortes priorités des noeuds conflictuels à 1 saut non coloriés des noeuds conflictuels à un saut de u .

- $max_prio3(u)$ comme la plus forte priorité des noeuds conflictuels à 1 saut non coloriés des noeuds conflictuels à un saut des noeuds conflictuels à un saut de u .

Dans OSERENA, tout noeud u émet un message *Color* contenant sa priorité et sa couleur ainsi que $max_prio1(u)$, $max_prio2(u)$, et les bitmaps des couleurs utilisées par ses noeuds conflictuels à 1 et 2 sauts.

Pour se colorier dans OSERENA, tout noeud u procède comme suit :

- 1) Le noeud u détermine l'ensemble $\mathcal{N}(u)$ des noeuds qui ne peuvent pas avoir la même couleur que lui. Ainsi dans un coloriage à 3-sauts, l'ensemble $\mathcal{N}(u)$ contient tous les voisins jusqu'à 3-sauts de u .
- 2) Le noeud u calcule sa priorité, $priority(u)$. Dans une application de collecte de données, la priorité d'un noeud est égale à son nombre de descendants dans l'arbre de collecte.
- 3) Le noeud u applique les cinq règles suivantes :
 - **Règle R'1** : Le noeud u se colorie si et seulement si $priority(u) = \max \{ max_prio1(u), max_prio2(u), max_prio3(u) \}$.
 - **Règle R'2** : Pour se colorier, le noeud u prend la plus petite couleur inutilisée dans $bitmap1(u) \cup bitmap2(u) \cup bitmap3(u)$. Pour une application de collecte de données, la couleur prise par u doit être supérieure à celle de son parent.
 - **Règle R'3** : Lorsqu'un noeud u reçoit le message *Color* d'un noeud voisin v , il insère dans la

liste *implicit_node_colored_list* tout noeud w dont la priorité dans $previous_max_prio1(v)$ précédemment reçu de v est supérieure à la plus forte priorité dans $max_prio1(v)$ reçu dernièrement. Ce noeud w vient de se colorier. Le noeud u procède de même avec $max_prio2(v)$.

- **Règle R'4** : Sur réception d'un message *Color*, tout noeud u calcule :
 - max_prio1 en éliminant tout noeud colorié (i.e. appartient à la liste *implicit_node_colored_list*).
 - max_prio2 en éliminant tout noeud colorié de priorité p reçue dans $max_prio1(v)$ d'un émetteur v si et seulement si :
 - a) soit p est la plus forte priorité dans $max_prio1(v)$.
 - b) soit p est la deuxième plus forte priorité dans $max_prio1(v)$ et la 4ème plus forte priorité dans $max_prio1(v)$ est égale à \emptyset .
 - c) soit p est la troisième plus forte priorité dans $max_prio1(v)$ et la 4ème plus forte priorité dans $max_prio1(v)$ est égale à \emptyset .
 - max_prio3 en éliminant tout noeud colorié de priorité p reçue dans $max_prio2(v)$ d'un émetteur v si et seulement si p est la plus forte ou la 2ème plus forte priorité dans $max_prio2(v)$.
- **Règle R'5** : Tout noeud u arrête d'envoyer son message *Color* dès qu'il est colorié, $max_prio1(u) = \emptyset$ et il a reçu de tous ses noeuds conflictuels à un saut v un message *Color* avec $max_prio1(v) = max_prio2(v) = \emptyset$.

3) Propriétés et performances d'OSERENA:

En environnement idéal, nous avons prouvé les propriétés suivantes [50] de OSERENA, algorithme distribué de coloriage des noeuds optimisé pour la collecte de données en réseaux denses :

- OSERENA fournit un coloriage valide à 3 sauts.
- Le coloriage fourni par OSERENA en mode stratégique est identique au coloriage FirstFit centralisé à 3 sauts utilisant la même priorité des noeuds et affectant à un noeud la plus petite couleur disponible supérieure à celle de son parent.
- OSERENA fournit les mêmes couleurs que SERENA. De plus, dans toutes les simulations effectuées, le nombre de rounds nécessaire au coloriage est identique.
- La taille du message *Color* est d'au plus $4 + 8 \cdot (size_address + size_prio) + size_color + size_bitmap1 + size_bitmap2$ octets avec $Size_max_prio1 = 4$ octets et $Size_max_prio2 = 3$ octets.

Nous illustrons maintenant les performances d'OSERENA, coloriage à 3 sauts. La Figure 3 montre que le nombre de couleurs dépend fortement de la densité des noeuds et beaucoup moins du nombre de noeuds. Intuitivement, la couleur d'un noeud dépend de la taille de son voisinage jusqu'à 3 sauts qui est proportionnelle à la densité.

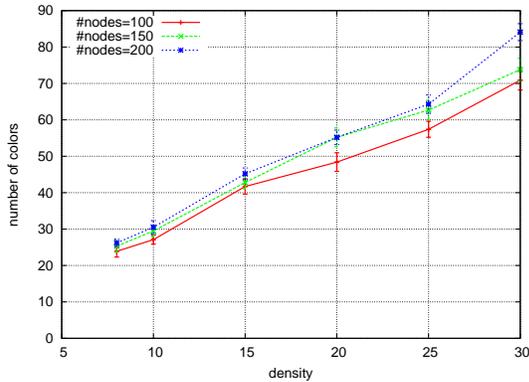


FIGURE 3. Nombre de couleurs.

Dans la Figure 4, nous observons que le nombre de rounds dépend plus du nombre de noeuds que de la densité. En effet, un noeud doit attendre que les noeuds conflictuels plus prioritaires soient coloriés ce qui risque d'induire une liste d'attente d'autant plus longue que le nombre de noeuds dans le réseau est grand.

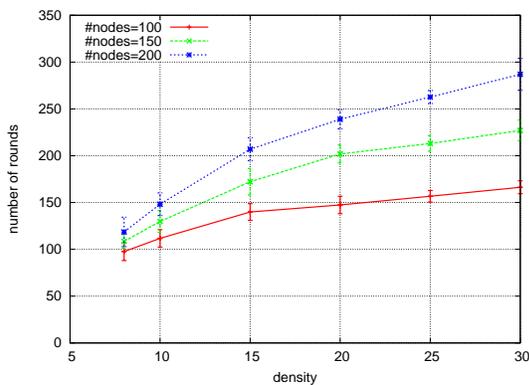


FIGURE 4. Nombre de rounds.

La Figure 5 illustre l'optimisation d'OSERENA en terme de nombre moyen de bits des messages *Color* échangés par noeud.

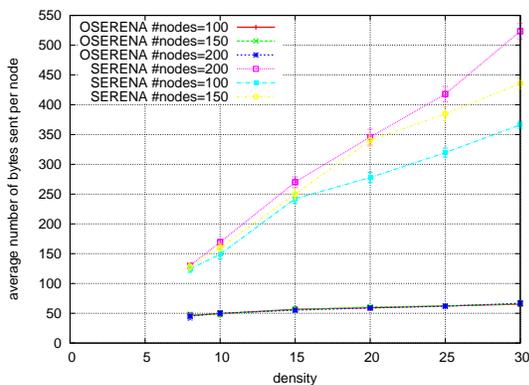


FIGURE 5. Nombre moyen d'octets envoyés par noeud avec SERENA et OSERENA.

Nous avons montré dans [51], [6] que le coloriage par OSERENA et son couplage à un accès slotté offre les avantages suivants :

- Gain en bande passante grâce à la réutilisation spatiale de la fréquence radio : les noeuds de même couleur peuvent transmettre simultanément. De plus, comme il n'y a pas de collision, il n'y a pas de bande passante perdue en collision et retransmission suite aux pertes engendrées par les collisions.
- Gain en énergie où un noeud peut dormir dans tout slot où ni lui ni ses voisins ne transmettent. La perte d'énergie suite aux collisions est évitée.
- Gain en délais de collecte des données grâce à un coloriage permettant la remontée des données collectées en un seul cycle.

D. Discussion

1) Mise en oeuvre dans un réseau de capteurs sans fil:

Le coloriage OSERENA est mis en oeuvre dans le réseau OCARI, réseau de capteurs sans fil en environnement industriel (voir <http://www.ocari.org> et <http://en.wikipedia.org/wiki/OCARI> pour plus de détails).

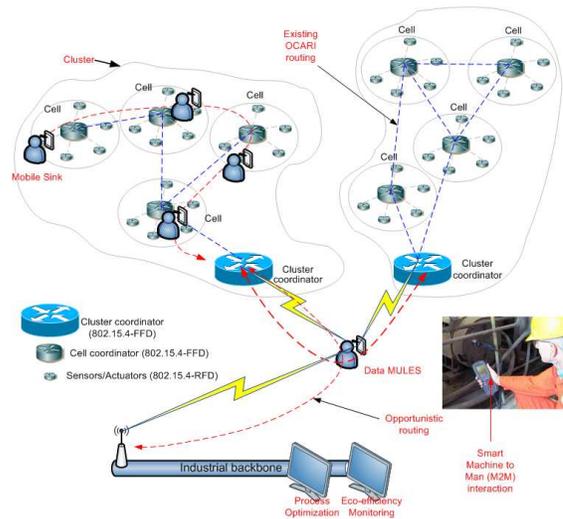


FIGURE 6. Topologie d'un réseau OCARI.

La topologie OCARI est illustrée par la Figure 6. Il s'agit d'un réseau maillé organisé en clusters. Chaque cluster est géré par un coordinateur de cluster, noté CPAN, responsable de l'assignation des adresses, du contrôle d'accès, etc. Il assure également l'interconnexion avec le réseau industriel. Chaque cellule a une topologie en étoile incluant un coordinateur de cellule assurant les fonctions de routage et des terminaux (RFDs) avec des ressources limitées. Le réseau OCARI peut inclure des puits mobiles qui peuvent être des agents équipés de PDA (Personal Digital Assistant) collectant les données lorsqu'ils passent à proximité des capteurs.

OSERENA est utilisé par la couche MAC pour ordonnancer les transmissions des noeuds selon les couleurs qui leur ont été attribuées. Pour réaliser le coloriage, OSERENA s'appuie sur la topologie fournie par le protocole de routage EOLSR. Plus particulièrement, dans OSERENA chaque noeud a besoin de connaître :

- ses voisins à un saut et les voisins à un saut de ses voisins,

informations fournies par les messages *Hello* échangés dans la découverte du voisinage, module du routage économe en énergie EOLSR.

○ son parent et ses enfants dans l'arbre de routage utilisé pour la collecte des données par le puits, informations fournies par la construction des routes effectuée dans EOLSR. Rappelons que dans l'arbre de routage, seuls les liens symétriques sont utilisés.

Avant de lancer le coloriage, il faut attendre la stabilité des liens. En effet, un changement tel que la création d'un nouveau lien peut invalider le coloriage. C'est pourquoi, EOLSR détecte la stabilité de l'arbre et des liens et en informe SERENA qui peut alors lancer le coloriage. De même, au cours de la vie du réseau, EOLSR, informe OSERENA de tout changement de topologie. Ainsi, OSERENA n'attend pas indéfiniment le message de *Color* d'un noeud qui n'est plus voisin. Remarquons que la rupture d'un lien n'invalide pas le coloriage fourni par SERENA et OSERENA.

La règle R'6 a été introduite pour tolérer les pertes de message. Ainsi la non-réception d'un message *Color* à un round r introduit un round supplémentaire dans le coloriage mais ne compromet pas la validité du coloriage obtenu.

Règle R'6 : Si à un round $r > 1$ de OSERENA, un noeud u quelconque ne reçoit pas un message de son voisin à un saut v , il utilise l'information reçue de v au round $r - 1$. Après $n \geq 2$ rounds successifs sans recevoir un message *Color* de v , u ne considère plus v comme un voisin à un saut.

2) Noeuds avec des charges fortement hétérogènes:

Dans une application de collecte de données, les noeuds près du puits ont un trafic plus important que les autres. Pour écouler ce trafic, ils ont besoin de plus de slots. L'idée est d'avoir une allocation de slots adaptée au trafic soumis par le noeud. Il est possible d'étendre OSERENA pour permettre une allocation adaptée au trafic. L'allocation de slots adaptée au trafic permet un bon dimensionnement du réseau et est plus efficace au niveau de l'utilisation des ressources. L'inconvénient réside dans une éventuelle perte en flexibilité : comment ordonnancer les retransmissions ?

IV. ORDONNANCEMENT DE L'ACTIVITÉ DES NOEUDS : ASSIGNATION DE SLOTS EN MONO/MULTICANAL

Dans cette section, nous nous intéressons au problème d'assignation de slot dans une application de collecte de données en environnement mono ou multicanal. Chaque noeud (ou de façon équivalente chaque lien montant dans l'arbre de collecte) doit recevoir un nombre de slots adapté au trafic généré (i.e. données produites localement + données reçues des enfants à propager). L'allocation de slots doit assurer la collecte des données en un seul cycle afin de minimiser les délais de collecte et garantir la cohérence temporelle des données collectées. Par ailleurs, la solution tirera avantage du contexte multicanal pour augmenter le parallélisme et la robustesse des transmissions.

A. Formalisation du problème d'assignation de slots

1) Hypothèses:

Nous supposons un *environnement idéal* où :

○ *Hypothèse H0* : Tous les liens sont symétriques et stables.

○ *Hypothèse H1* : Chaque noeud a une adresse unique dans le réseau.

○ *Hypothèse H2* : Aucun noeud n'empêche la réception correcte d'un autre noeud hors de sa portée radio. ○ *Hypothèse H3* : Il n'y a pas de perte de message.

○ *Hypothèse H4* : *Demandes hétérogènes des noeuds* : Le puits est chargé de collecter les données générées par les autres noeuds. Dans chaque cycle de collecte, chaque noeud ordinaire (i.e. qui n'est pas le puits) transmet ses propres données (i.e. données générées localement dans un cycle) et propage les données reçues de ses enfants à son parent dans l'arbre de collecte. Soit d_u le nombre de slots nécessaires au noeud ordinaire u pour transmettre ses données. Le noeud u recevra en plus des d_u slots demandés les slots qui lui sont nécessaires pour propager les données reçues de ses enfants.

○ *Hypothèse H5* : *Interface radio d'un noeud* : Le puits est le seul noeud à avoir $n_{interf} \geq 1$ interfaces radio. Tout noeud ordinaire a une seule interface radio.

○ *Hypothèse H6* : *Canaux disponibles* : Dans un souci de simplicité, nous supposons que chaque noeud dispose de $n_{channel} > 1$ canaux disponibles, numérotés de 1 à $n_{channel}$. Nous supposons aussi que l'arbre de routage utilisé pour la collecte est le même sur tous les canaux.

○ *Hypothèse H7* : *Taille du slot* : Tous les slots ont la même taille. Dans un souci de simplicité, nous supposons que cette taille permet la transmission d'un paquet.

○ *Hypothèse H8* : *Topologie* : Pour toute topologie, les entrées de l'algorithme d'ordonnancement sont les liens entre noeuds, les noeuds conflictuels et l'arbre de routage.

○ *Hypothèse H9* : *Liens topologiques* : Les seuls liens existant dans la topologie sont les liens de l'arbre de routage.

○ *Hypothèse H10* : *Noeuds conflictuels* : Nous supposons un modèle d'interférence basé sur un graphe : deux noeuds quelconques u et v qui sont voisins à 1 ou 2 sauts sont conflictuels. En conséquence, ils ne peuvent pas transmettre à leur parent respectif dans le même slot et sur le même canal. Nous supposons également que l'ensemble des noeuds conflictuels d'un noeud ordinaire u quelconque est le même sur chaque canal.

2) Formalisation en programmation linéaire:

Le réseau est modélisé par un graphe $G = (V, E)$ où V est l'ensemble des sommets représentant les noeuds du réseau et E est l'ensemble des arcs représentant les liens de communication. Soit $V = V_s \cup V_g$, où V_s est l'ensemble des noeuds ordinaires, encore appelés sources et V_g représente l'ensemble des puits, avec $V_s \cap V_g = \emptyset$.

Pour chaque noeud $v \in V$, nous définissons $I(v)$ l'ensemble des noeuds qui interfèrent avec v (i.e. noeuds conflictuels de v) lorsqu'ils transmettent sur le même canal dans le même slot. Soit i_v le nombre d'interfaces radio disponibles sur le noeud v .

Pour toute source s , soit p_s le nombre de paquets que s doit transmettre en un cycle (i.e. paquets générés localement + paquets propagés) Pour tout lien e , soit $f_{e,s}$ le nombre de paquets générés par s que s transmet sur le lien e en un cycle. Soit $E^+(v)$ l'ensemble des liens sur lesquels le noeud v peut transmettre. Soit $E^-(v)$ l'ensemble des liens sur lesquels le noeud v peut recevoir.

Soit C l'ensemble des $n_{channel}$ canaux disponibles. Nous définissons $a_{e,c,t}$ l'activité du lien e sur le canal c dans le slot t , ie $a_{e,c,t} = 1$ si et seulement si il y a une transmission sur le lien e sur le canal c dans le slot t et $a_{e,c,t} = 0$ sinon. De plus, soit u_t l'utilisation du slot t , en d'autres termes $u_t = 1$ signifie qu'au moins un lien est actif dans le slot t et $u_t = 0$ signifie que le slot est inutilisé.

Nous pouvons calculer T_{max} , un majorant de la longueur du cycle. Ce majorant est atteint lorsque tous les noeuds transmettent en séquence sur le même canal. Nous avons alors : $T_{max} = \sum_s \sum_e f_{e,s} * depth_s$ où $depth_s$ est la profondeur du noeud s dans l'arbre de collecte des données. L'objectif est de minimiser le nombre de slots $t \leq T_{max}$.

$$\min \sum_{t \leq T_{max}} u_t$$

avec les contraintes suivantes :

$$\begin{aligned} a_{e,c,t} &\leq u_t \\ \forall e \in E, \forall c \in C, \\ t &\leq T_{max} \end{aligned} \quad (1)$$

$$\begin{aligned} a_{e,c,t} + a_{e',c,t} &\leq 1 \\ \forall v \in V, \forall e \in E^+(v), \\ \forall w \in I(v), \forall e' \in E^+(w), \\ \forall c \in C, t &\leq T_{max} \end{aligned} \quad (2)$$

$$\begin{aligned} \sum_{c \in C} \sum_{e \in E^+(v)} a_{e,c,t} + \sum_{c \in C} \sum_{e' \in E^-(v)} a_{e',c,t} &\leq i_v \\ \forall v \in V, t &\leq T_{max} \end{aligned} \quad (3)$$

$$\begin{aligned} \sum_{s \in V_s} f_{e,s} &= \sum_{c \in C} \sum_{t \leq T_{max}} a_{e,c,t} \\ \forall e \in E \end{aligned} \quad (4)$$

$$\begin{aligned} \sum_{e \in E^+(s)} f_{e,s} &= p_s \\ \forall s \in V_s \end{aligned} \quad (5)$$

$$\begin{aligned} \sum_{g \in V_g} \sum_{e \in E^-(g)} f_{e,s} &= p_s \\ \forall s \in V_s \end{aligned} \quad (6)$$

$$\begin{aligned} \sum_{s \in V_s} \sum_{e \in E^+(i)} f_{e,s} &= p_i + \sum_{s \in V_s} \sum_{e \in E^-(i)} f_{e,s} \\ \forall i \in V_s \end{aligned} \quad (7)$$

$$\begin{aligned} \sum_{c \in C} \sum_{e \in E^+(i)} a_{e,c,t} &\leq \sum_{c \in C} \sum_{e \in E^-(i)} \sum_{t' \in \{1..t-1\}} a_{e,c,t'} + p_i \\ &\quad - \sum_{c \in C} \sum_{e \in E^+(i)} \sum_{t' \in \{1..t-1\}} a_{e,c,t'} \\ \forall i \in V_s, t &\leq T_{max} \end{aligned} \quad (8)$$

La contrainte 1 relie l'utilisation d'un slot à l'activité d'au moins un lien sur un canal dans ce slot. La contrainte 2 assure que deux noeuds conflictuels ne transmettent pas sur le même

canal dans le même slot. La contrainte 3 garantit que le nombre de communications simultanées d'un noeud quelconque est limité à son nombre d'interfaces radio. La contrainte 4 assure la correspondance entre les activités sur les canaux et les paquets émis sur les liens. Les contraintes 5, 6 et 7 expriment la conservation des messages entre les sources et les puits. La dernière contrainte garantit qu'un noeud génère ou reçoit un paquet avant de le transmettre.

B. Bornes théoriques du nombre optimal de slots

Nous donnons maintenant les bornes inférieures pour la collecte de données sans agrégation avec des demandes hétérogènes. Ces bornes sont établies sous les hypothèses données précédemment. Pour les démonstrations, se référer à [67].

Lemma 1: Dans tout RCSF avec demandes hétérogènes des noeuds, une borne inférieure du nombre de slots nécessaires à une collecte de données sans agrégation est $\lceil \frac{\sum_{u \neq sink} d_u}{g} \rceil$, où $g = \min(n_{interf}, n_{child}, n_{channel})$, avec $n_{interf} \geq 1$ le nombre d'interfaces radio du puits, n_{child} le nombre d'enfants du puits, $n_{channel} > 1$ le nombre de canaux disponibles sur chaque noeud et d_u le nombre de slots nécessaires à un noeud ordinaire quelconque u pour transmettre ses propres données à son parent.

1) Réseaux linéaires:

Lemma 2: Dans tout RCSF linéaire avec demandes hétérogènes des noeuds où chaque noeud dispose de $n_{channel} > 1$ canaux, le nombre minimum de slots nécessaires à une collecte de données sans agrégation est $d_{child(sink)} + 2 \sum_{u \neq sink, u \neq child(sink)} d_u$, quelque soit le nombre d'interfaces radio du puits, où d_u est le nombre de slots nécessaires à un noeud ordinaire quelconque u pour transmettre ses propres données à son parent.

Lemma 3: Dans tout RCSF linéaire avec demandes hétérogènes des noeuds où chaque noeud dispose de $n_{channel} > 1$ canaux, l'algorithme qui ordonnance :

- dans tout slot impair, tout noeud à $(2h+1)$ sauts du puits, avec $h \in [0, \lfloor \frac{N-1}{2} \rfloor]$, où N est le nombre de noeuds,
- dans tout slot pair, tout noeud à $2h$ sauts du puits, avec $h \in [1, \lfloor \frac{N-1}{2} \rfloor]$,

fournit le nombre minimum de slots nécessaires à une collecte de données sans agrégation.

2) Réseaux multi-lignes:

Théorème 2: Dans tout RCSF multi-ligne avec demandes hétérogènes des noeuds où chaque noeud dispose de $n_{channel} > 1$ canaux, le nombre minimum de slots nécessaires à une collecte de données sans agrégation est $\text{Max}(\lceil \frac{\sum_{u \neq sink} d_u}{g} \rceil, \text{max}_{i=child(sink)}(d_i + 2 \sum_{v \neq i, v \in \text{subline}(i)} d_v + \delta))$, où $g = \min(n_{interf}, n_{child}, n_{channel})$, d_u est le nombre de slots nécessaires à un noeud ordinaire quelconque u pour transmettre ses propres données à son parent, $n_{interf} \geq 1$ le nombre d'interfaces radio du puits, n_{child} le nombre d'enfants

du puits, et $\delta = 1$ si et seulement si le $(g + 1)^{eme}$ enfant du puits requiert le même nombre total de slots que le premier, en supposant que les enfants du puits sont ordonnés par ordre décroissant du nombre de slots requis, le nombre total de slots requis par l'enfant i du puits étant $d_i + 2 \sum_{v \neq i, v \in \text{subline}(i)} d_v$, et $\delta = 0$ sinon.

3) Réseaux en arbre:

Théorème 3: Dans tout RCSF en arbre avec demandes hétérogènes des noeuds où chaque noeud dispose de $n_{channel} > 1$ canaux, le nombre minimum de slots nécessaires à une collecte de données sans agrégation est $\text{Max}(\lceil \frac{\sum_{u \neq \text{sink}} d_u}{g} \rceil, \text{max}_{i=\text{child}(\text{sink})} (d_i + 2 \sum_{v \neq i, v \in \text{subtree}(i)} d_v + \delta))$, où $g = \min(n_{interf}, n_{child}, n_{channel})$, d_u est le nombre de slots nécessaires à un noeud ordinaire quelconque u pour transmettre ses propres données à son parent, $n_{interf} \geq 1$ le nombre d'interfaces radio du puits, n_{child} le nombre d'enfants du puits, et $\delta = 1$ si et seulement si le $(g + 1)^{eme}$ enfant du puits requiert le même nombre total de slots que le premier, en supposant que les enfants du puits sont ordonnés par ordre décroissant du nombre de slots requis, le nombre total de slots requis par l'enfant i du puits étant $d_i + 2 \sum_{v \neq i, v \in \text{subline}(i)} d_v$, et $\delta = 0$ sinon.

4) Configurations T_s et T_n :

Toute configuration est définie par une topologie (ensemble de noeuds et liens) et les demandes de slots de chaque noeud correspondant aux données générées localement.

Pour les réseaux multi-lignes et en arbre, nous définissons deux types de configurations, T_s et T_n :

- Une configuration est dite T_s si et seulement si le nombre optimal de slots est imposé par le sous-arbre le plus demandeur de slots, sous-arbre de racine un enfant i du puits. La demande du sous-arbre est égale à $d_i + 2 \sum_{v \neq i, v \in \text{subtree}(i)} d_v$.
- Une configuration est dite T_n si et seulement si le nombre optimal de slots dépend uniquement du nombre total de demandes des noeuds et de $g = \min(n_{interf}, n_{child}, n_{channel})$. Ce nombre optimal est égal à $\lceil \frac{\sum_{u \neq \text{sink}} d_u}{g} \rceil$. Remarquons qu'une configuration T_n correspond à un arbre couvrant minimum où les branches ont des demandes équilibrées [70], [71].

C. Exemple illustratif

Une assignation optimale de slots en environnement multicanal peut être obtenue par des outils de programmation linéaire tels que GLPK (GNU Linear Programming Kit) [72]. La Figure 7 montre le nombre optimal de slots nb_s pour différentes topologies avec un seul puits (réseaux linéaires 7(a), réseaux multi-lignes 7(b), réseaux en arbres équilibrés 7(c) et réseaux en arbre quelconque 7(d)) avec différents nombres d'interfaces radio n_{interf} du puits et différents nombres de canaux $n_{channel}$. Remarquons que ces nombres optimaux de slots obtenus avec GPLK sont également obtenus avec l'algorithme MODESA que nous présentons maintenant.

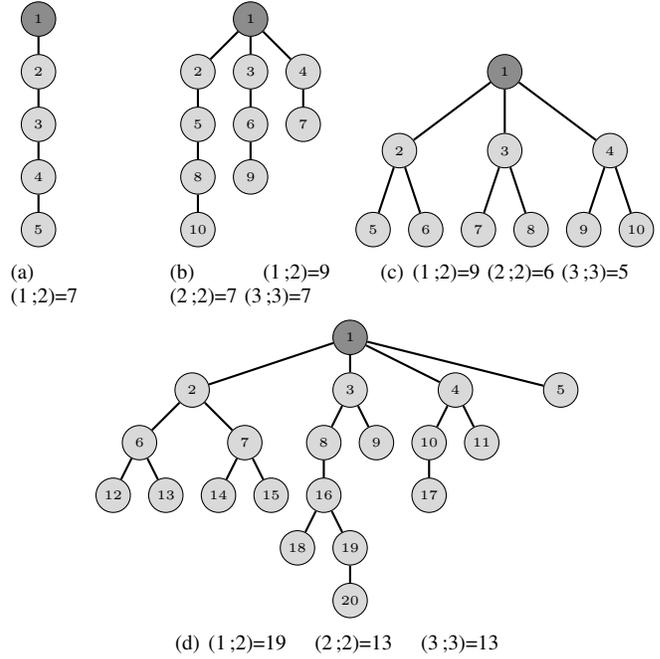


FIGURE 7. Nombre optimal de slots nb_s pour des topologies variées avec différents nombres d'interfaces radio du puits n_{interf} et nombres de canaux $n_{channel}$ avec la notation : $(n_{interf}; n_{channel})=nb_s$.

D. MODESA

1) Principes de MODESA:

MODESA, un ordonnancement centralisé pour la collecte de données sans agrégation, a été proposé dans [67]. Il prend en compte la disponibilité de plusieurs canaux pour réduire la période d'activité dans la supertrame de collecte de données, tout en assurant un accès au médium proportionnel au trafic du noeud. MODESA construit l'ordonnancement après avoir remonté au puits l'information nécessaire (e.g. liens de topologie, demandes des noeuds, arbre de routage, noeuds conflictuels). L'ordonnancement est ensuite diffusé à tous les noeuds en multi-sauts suivant l'arbre de routage. Notons qu'avec MODESA, les liens de la topologie et les noeuds conflictuels sont définis par canal, ils peuvent différer d'un canal à l'autre.

MODESA construit l'ordonnancement multicanal slot par slot. A chaque itération, MODESA alloue le slot courant à un ou plusieurs tuples (*numero de slot, émetteur, receveur, canal*) en appliquant les règles suivantes :

- 1) Chaque noeud a une *priorité dynamique*. Cette priorité est égale à $remPckt * parentRcv$ où $remPckt$ est le nombre de paquets que le noeud a dans son buffer à l'itération courante. $parentRcv$ est le nombre total de paquets que le parent du noeud doit recevoir dans un cycle. Cette heuristique vise à réduire le nombre de paquets bufférisés en favorisant les noeuds qui ont des paquets à transmettre au parent avant recevoir un grand nombre de paquets.
- 2) Seuls les noeuds ayant des données à transmettre entrent en compétition pour le slot courant.
- 3) De plus, pour être autorisé à transmettre dans un slot,

un noeud et son parent doivent avoir une interface disponible.

- 4) Pour tout slot, le premier noeud ordonnancé est le noeud de plus forte priorité parmi tous les noeuds ayant des données à transmettre. Si plusieurs noeuds ont la même priorité, MODESA choisit le noeud de plus petit identificateur. Le noeud choisi est ordonnancé sur le premier canal disponible c .
- 5) Tout noeud en compétition peut être ordonnancé dans le slot courant sur le canal c si et seulement s'il n'est pas conflictuel avec les noeuds déjà ordonnancés sur ce canal et dans ce slot.
- 6) Un noeud conflictuel avec les noeuds déjà ordonnancés sur ce canal est ordonnancé sur un autre canal, s'il en existe. Sinon dans le slot suivant.

2) Propriétés et performances de MODESA:

Un arbre est dit équilibré si et seulement si à chaque niveau n , le nombre d'enfants d'un parent est le même pour tous les noeuds du niveau n .

Théorème 4: En environnement multicanal, $n_{channel} > 1$, MODESA est optimal pour des demandes homogènes dans des topologies linéaires, multi-lignes et les arbres équilibrés. La Figure 8 montre que dans un réseau de 100 noeuds avec demandes hétérogènes, trois canaux et un puits disposant d'une seule interface radio, MODESA est optimal dans 50% des configurations T_s et 87% des configurations T_n qui sont mieux équilibrées au niveau des demandes des sous-arbres. De plus, pour les configurations où MODESA n'est pas optimal nous avons évalué son écart par rapport à l'ordonnancement optimal. Cet écart est calculé comme suit : $(nb\ slots\ MODESA - nb\ optimal) / nb\ optimal$. Comme illustré par la Figure 9, cet écart moyen est de 8.5% pour les configurations T_s et 2.9% pour les configurations T_n . Ceci montre l'importance d'avoir un arbre de routage qui équilibre la charge de trafic entre ses différentes branches.

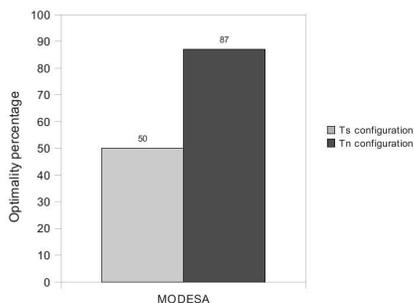


FIGURE 8. Optimalité de MODESA.

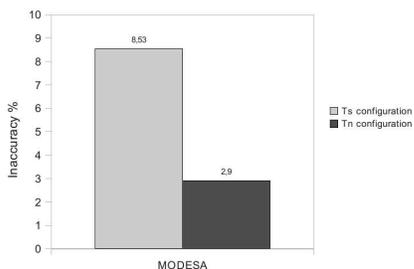


FIGURE 9. Distance de MODESA à l'optimal.

E. Discussion

Remarquons tout d'abord que les algorithmes de coloriage peuvent être utilisés pour réaliser une assignation de slots proportionnelle au trafic soumis. Il suffit pour cela de créer un motif avec toutes les couleurs obtenues par le coloriage, d'instancier ce motif, puis de répéter le motif dans lequel chaque couleur n'est instanciée que si le nombre total de slots dont elle a besoin n'est pas atteint.

Il existe de nombreux algorithmes de coloriage distribués, mais peu sont adaptés aux réseaux denses.

Les algorithmes centralisés d'allocation de canaux et assignation de slots ont l'avantage d'approcher finement les bornes théoriques mais ne passent pas à l'échelle.

L'adaptivité de ces algorithmes aux changements de topologie reste un défi.

Les nouvelles approches traitent conjointement du routage, de l'allocation des canaux et de l'ordonnancement des liens dans les RCSF [74], [75]. Il existe également des algorithmes prometteurs tels que [76] appliqués à la radio logicielle, mais dont la complexité n'est pas encore compatible avec les capteurs aux ressources très limitées.

V. CONCLUSION

Ce papier traite de l'ordonnancement de l'activité des noeuds dans les réseaux ad hoc et les réseaux de capteurs sans fil, RCSF. Deux problèmes ont été traités : le coloriage et l'assignation de slots. Les algorithmes de coloriage ont été introduits dans les RCSF pour permettre aux capteurs d'économiser leur énergie, d'utiliser plus efficacement la bande passante et d'améliorer la fiabilité des communications. En effet, les collisions sont totalement évitées et les noeuds peuvent dormir lorsqu'ils n'ont ni message à transmettre, ni message à recevoir. Toutefois, l'utilisation des algorithmes de coloriage dans les réseaux denses n'est possible que s'ils ont été conçus pour limiter les ressources (mémoire, capacité de traitement et bande passante) dont ils ont besoin pour s'exécuter. Les limitations en ressources et le nombre potentiellement élevé de voisins constituent un véritable défi pour la conception de protocoles de communication efficaces. C'est pourquoi nous avons proposé OSERENA. Cette version optimisée fournit le même coloriage que la version non optimisée, mais avec une taille de messages très réduite. Par conséquent, OSERENA permet des gains considérables en bande passante et consommation énergétique.

Nous nous sommes également intéressés à l'assignation de slots pour les applications de collecte de données. Les noeuds proches du puits demandent plus de bande passante que les noeuds feuilles dans l'arbre de collecte. L'algorithme d'assignation de slots doit minimiser le nombre de slots tout en assurant la remontée des données générées par les capteurs en un seul cycle. Ceci a pour avantage de réduire les délais de collecte et assurer une meilleure cohérence temporelle des données collectées. Nous avons déterminé les bornes inférieures du nombre de slots nécessaires à la collecte de données en environnement multicanal lorsque le puits est équipé de plusieurs interfaces radio et les noeuds ont des demandes hétérogènes : les nombres de paquets générés localement par

les noeuds peuvent différer. Ces bornes sont à affiner pour une topologie donnée, lorsqu'il existe des liens supplémentaires à ceux utilisés dans l'arbre de collecte. Pour les topologies en arbre, nous avons distingué deux types de configuration. Les configurations T_n , équilibrées en termes de trafic entre les différentes branches du sous-arbre, sont moins gourmandes en slots que les configurations T_s . Il s'en suit que le routage doit s'efforcer d'équilibrer le trafic transmis vers le puits par les branches de l'arbre de routage. L'optimisation conjointe du routage, de l'allocation des canaux, de l'assignation de slots et de l'ordonnement des liens reste un problème ouvert dans les RCSF.

RÉFÉRENCES

- [1] R. Soua, P. Minet, *A survey on energy efficient techniques in wireless sensor networks*, WMNC 2011, Toulouse, France, October 2011.
- [2] I. Amdouni, *Wireless self-adaptive ad hoc and sensor networks : energy efficiency and spatial reuse*, Thèse de l'université Pierre et Marie Curie, Paris, Février 2013.
- [3] S. Mahfoudh, *Energy efficiency in wireless ad hoc and sensor networks : routing, node activity scheduling and cross-layering*, Thèse de l'université Pierre et Marie Curie, Paris, Février 2010.
- [4] I. Amdouni, P. Minet, C. Adjih, *Adaptivity of a Coloring Algorithm to Unreliable Communications for Data Gathering in Wireless Sensor Networks*, International Journal of Digital Information and Wireless Communications, (IJDIWC), <http://www.sdiwc.net/ijdiwc/>, March 2013.
- [5] S. Mahfoudh, P. Minet, I. Amdouni, "Energy efficient routing and node activity scheduling in the OCARI wireless sensor network", *Future Internet*, 2(3), 308-340, August 2010.
- [6] S. Mahfoudh, G. Chalhoub, P. Minet, M. Misson, I. Amdouni, "Node Coloring and Color Conflict Detection in Wireless Sensor Networks", *Future Internet*, vol. 2(4), pp. 469-504, 2010.
- [7] I. Amdouni, P. Minet, C. Adjih, *Node Coloring in Wireless Networks : Complexity Results and Grid Coloring*, WMNC, Toulouse, France, October 2011.
- [8] S. Meguerdichian, M. Potkonjak, "Low power 0/1 coverage and scheduling techniques in sensor networks", *UCLA Technical Reports 030001*, January 2003.
- [9] K. Chakrabarty et al., "Grid coverage for surveillance and target location in distributed sensor networks", *IEEE Transactions on Computers*, vol. 51 (12), 2002, pp. 1448-1453.
- [10] K. Akkaya, M. Younis, "A survey on routing protocols for wireless sensor networks", *Ad Hoc Networks* Volume : 3, Issue : 3, Pages : 325-349, 2005.
- [11] B. Chen, K. Jamieson, H. Balakrishnan, R. Morris, "Span : an energy efficient coordination algorithm for topology maintenance in ad hoc wireless networks", *ACM Wireless Networks*, vol. 8 (5), 2002.
- [12] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks", in *Proc. ACM SenSys03*, Los Angeles, California, USA, November 2003.
- [13] M. Cardei, M. Thai, Y. Li, W. Wu, "Energy-efficient target coverage in wireless sensor networks", in *Proc. IEEE INFOCOM 2005*, Miami, Florida, March 2005.
- [14] Rajendran V.; Obraczka K.; Garcia-Luna-Aceves J.J., *Energy-efficient, collision-free medium access control for wireless sensor networks*, SenSys'03, Los Angeles, California November 2003.
- [15] Rajendran V.; Garcia-Luna-Aceves J.J.; Obraczka K., *Energy-efficient, application-aware medium access for sensor networks*, IEEE MASS 2005, Washington, November 2005.
- [16] Winnie L.L.; Amitava D.; Rachel C.O., *FlexiTP : A Flexible Schedule-Based TDMA Protocol for Fault-Tolerant and Energy-Efficient Wireless Sensor Networks*, IEEE Transactions on Parallel and Distributed Systems, vol. 19, pp. 851-864, June 2008.
- [17] Gabriel S.; Mosse D.; Cleric R., *TDMA-ASAP : sensor network TDMA scheduling with adaptive slot stealing and parallelism*, ICDCS 2009, Montreal, Canada, June 2009.
- [18] Rhee I.; Warrier I.; Aia M.; Min J., *Z-MAC : a hybrid MAC for wireless sensor networks*, SenSys'05, San Diego, California, November 2005.
- [19] Rhee I.; Warrier A.; Xu L., *Randomized dining philosophers to TDMA scheduling in wireless sensor networks*, Technical Report TR-2005-21, Dept of Computer Science, North Carolina State University, April 2005.
- [20] V. Rajendran, "Energy-efficient, application aware medium access for sensor networks", in *Proc. 2nd IEEE Conf. on Mobile Ad-hoc and Sensor Systems (MASS'05)*, 2005.
- [21] W. Ye, J. Heidemann, D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks", *IEEE/ACM Transactions on Networking*, vol. 12(3), pp. 493-506, June 2004.
- [22] T. van Dam and K. Langendoen. "An adaptive energy efficient MAC protocol for wireless sensor networks", in *Proc. 1st International Conference on Embedded Networked Sensor Systems*, pages 171-180. ACM, November 2003.
- [23] G. Lu, B. Krishnamachari, C.S. Raghavendra, "An adaptive energy efficient and low-latency Mac for data gathering in wireless sensor networks", in *Proc. 18th International Parallel and Distributed Processing Symposium*, New Mexico, USA, April 2004, pp. 224, 26-30.
- [24] Johansson O., *Simple distributed ($\Delta + 1$) -coloring of graphs*, Information processing Letters, 70, 1999.
- [25] Finoccho I.; Panconesi A.; Silvestri R., *Experimental analysis of simple distributed vertex coloring algorithms*, SODA 2002, San Francisco, California, January 2002.
- [26] Busch C.; Magdon-Ismael M.; Sivrikaya F.; Yener B., *Contention-free MAC protocols for wireless sensor networks*, DISC 2004, Amsterdam, Netherlands, October 2004.
- [27] Brelaz D., *New methods to color the vertices of a graph*, Communications of the ACM, 22(4), 1979.
- [28] Hansen J.; Kubale M.; Kuszner L.; Nadolski A., *Distributed largest-first algorithm for graph coloring*, EURO-PAR 2004, Dresden, Germany, August 2004.
- [29] Glover F., *Tabu search-part I*, ORSA Journal on Computing, vol. 1, pp. 190-205, 1989.
- [30] Bermond J.C.; Havet F.; Huc F.; Linhares-Sales C., *Improper colouring of weighted grid and hexagonal graphs*, Discrete Mathematics, Algorithms and Applications, vol. 2, pp. 395-411, 2010.
- [31] Cargiannis I.; Fishkin A.; Kaklamani C.; Papaioannou E., *A tight bound for online coloring of disk graphs*, Theoretical Computer Science, 2007.
- [32] Kierstead H. A.; Qin J., *Coloring interval graphs with First-Fit*, Discrete Math, vol. 144, pp. 47-57, September 1995.
- [33] Galinier P.; Hertz A., *A survey of local search methods for graph coloring*, Computers & Operations Research, vol. 3, pp. 2547-2562, September 2006.
- [34] Wu Q.; Hao J.K., *Coloring large graphs based on independent set extraction*, to appear in Computers and Operations Research, Elsevier, 2011.
- [35] Durand D.; Jain R.; Tseytlin D., *Distributed Scheduling Algorithms to Improve the Performance of Parallel Data Transfers*, Proc. IPPPS'94 Workshop on Input/Output in Parallel Computer Systems, April 1994.
- [36] Durand D.; Jain R.; Tseytlin D., *Applying Randomized Edge Coloring Algorithms to Distributed Communication : An Experimental Study*, ACM Symposium of Parallel Algorithms and Architectures, 1995.
- [37] Marathe M. V.; Panconesi A.; Risinger L.D., *An Experimental Study of a Simple, Distributed Edge-Coloring Algorithm*, ACM Journal of Experimental Algorithmics, vol. 9, 2004.
- [38] Hilgemeier M.; Drechsler N.; Drechsler R., *Fast heuristics for the edge coloring of large graphs*, Euromicro Symposium on Digital system design, DSD'03, IEEE computer Society, 2003.
- [39] Tzeng C.H.; Jehn-Ruey J.; Shing-Tsaan H., *A self-stabilizing ($\Delta + 4$)-edge-coloring algorithm for planar graphs in anonymous uniform systems*, Journal Information Processing Letters, vol. 101, pp. 168-173, February 2007.
- [40] Even G.; Lotker Z.; Ron D.; Smorodinsky S., *Conflict-Free Colorings of Simple Geometric Regions with Applications to Frequency Assignment in Cellular Networks*, SIAM J. Comput, vol. 33, pp 94-136, 2004.
- [41] Subramanian A.P.; Gupta H.; Das S.R., *Minimum Interference Channel Assignment in Multi-Radio Wireless Mesh Network*, The 4th Annual IEEE Communications Society Conference on Mesh and Ad Hoc Communications and Networks SECON'07, San Diego, California, USA, June 2007.
- [42] Chowdhury K.R.; Chanda P.; Agrawal D.P.; Zeng Q.A., *DCA - A Distributed Channel Allocation Scheme for Wireless Sensor Networks*, The 16th International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC'05, Berlin, Germany, September 2005.

- [43] Wang W.; Liu X., *List-coloring based channel allocation for open-spectrum wireless networks*, The 62nd Vehicular Technology Conference, VTC'05, Dallas, Texas, USA, September 2005.
- [44] Zheng H.; Peng C., *Collaboration and fairness in opportunistic spectrum access*, IEEE International Conference on Communications, ICC'05, Seoul Korea, March 2005.
- [45] Gandham S.; Dawande M.; Prakash R., *Link scheduling in sensor networks : distributed edge coloring revisited*, The 24th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM'05, Miami, March 2005.
- [46] Cheng M.; Yin L., *Transmission Scheduling in Sensor Networks via Directed Edge Coloring*, IEEE International Conference on Communications, ICC'07, Glasgow, Scotland, June 2007.
- [47] M. Garey, D. Johnson, *Computers and intractability : a guide to theory of NP-completeness*, W.H. Freeman, San Francisco, California, 1979.
- [48] Schneider J.; Wattenhofer R., *Coloring unstructured wireless multi-hop networks*, The 28th ACM symposium on Principles of distributed computing, PODC'09, USA, 2009.
- [49] S. Mahfoudh, P. Minet, I. Amdouni, *Energy efficient routing and node activity scheduling in the OCARI wireless sensor network*, Future Internet 2010, 2(3), 308-340, August 2010.
- [50] I. Amdouni, P. Minet, C. Adjih, *OSERENA : a coloring algorithm optimized for dense wireless networks*, the International Journal of Networked and Distributed Computing (IJNDC), Volume 1, Issue 1, November 2012.
- [51] S. Mahfoudh, P. Minet, *Maximization of energy efficiency in wireless ad hoc and sensor networks with SERENA*, Journal on Mobile Information Systems, IOS Press, Vol.5 No1, 2009.
- [52] G. Zhou, C.D. Huang, T. Yan, T. He, J.A. Stankovic, T.F. Adbelzahr, "MMSN : Multi-frequency media access control for wireless sensor networks," *In Proc. IEEE INFOCOM'06*, Barcelona, Catalunya, Spain, 2006.
- [53] <http://www.ieee802.org/15/pub/TG4.html>
- [54] Y. Wu, J. Stankovic, T. He, S. Lin, "Realistic and efficient multi-channel communications in wireless sensor networks," *In Proc. INFOCOM'08*, pp. 1193-1201, Phoenix, AZ, USA, 2008.
- [55] X. Wang, X. Wang, X. Fu, G. Xing, N. Jha, "Flow-Based Real-Time Communication in Multi-Channel Wireless Sensor Networks," *In Proc. EWSN'09*, Cork, Ireland, 2009.
- [56] Y. Wu, M. Keally, G. Zhou, W. Mao, "Traffic-Aware Channel Assignment in Wireless Sensor Networks," *Wireless Algorithms, Systems, and Applications, Lecture Notes in Computer Science*, vol. 5682, 2009.
- [57] L. Tang, Y. Sun, O. Gurewitz and David B. Johnson, "EM-MAC : A Dynamic Multichannel Energy-Efficient MAC Protocol for Wireless Sensor Networks," *In Proc. ACM MobiHoc'11*, Paris, France, 2011.
- [58] Q. Yu, J. Chen, Y. Sun, Y. Fan, W. Shen, "Regret Matching Based Channel Assignment for Wireless Sensor Networks," *In Proc. IEEE ICC'10*, Cape Town, South Africa, 2010.
- [59] J. Li, D. Zhang, L. Guo, S. Ji, Y. Li, "ARM : An asynchronous receiver-initiated multichannel MAC protocol with duty cycling for WSNs," *In Proc. IEEE IPCCC'10*, Albuquerque, NM, 2010.
- [60] Y. Kim, H. Shin, H. Cha, "Y-MAC : An Energy-Efficient Multi-channel MAC Protocol for Dense Wireless Sensor Networks," *In Proc. IPSN'08*, St. Louis, Missouri, USA, 2008.
- [61] J. Borms, K. Steenhaut, B. Lemmens, "Low-Overhead Dynamic Multi-channel MAC for Wireless Sensor Networks," *In Proc. EWSN'10*, Coimbra, Portugal, 2010.
- [62] J. So, N. Vaidya, "Multi-channel MAC for ad-hoc networks : handling multi-channel hidden terminal using a single transceiver," *In Proc. ACM MobiHoc'04*, Tokyo, Japan, 2004.
- [63] <http://www.ieee802.org/15/>
- [64] A. Tzamaloukas, J. Garcia-Luna-Aceves, "Channel-hopping multiple access with packet trains for ad hoc networks," *In Proc. IEEE MoMuC'00*, Tokyo, 2000.
- [65] Q. Yu, J. Chen, Y. Fan, X. Shen, Y. Sun, "Multi channel assignment in wireless sensor networks : a game theoretic approach," *In Proc. INFOCOM'10*, San Diego, CA, 2010.
- [66] R. Soua, P. Minet, *A survey on multichannel assignment protocols in wireless sensor networks*, IFIP Wireless Days, Niagara Falls, Ontario, Canada, October 2011.
- [67] R. Soua, P. Minet, E. Livolant, *MODESA : an Optimized Multichannel Slot Assignment for Raw Data Convergecast in Wireless Sensor Networks*, 31st IEEE International Performance Computing and Communications Conference, IPCCC 2012, Austin, Texas, December 2012.
- [68] R. Soua, E. Livolant, P. Minet, *Adaptive Strategy for an Optimized Collision-Free Slot Assignment in Multichannel Wireless Sensor Networks*, submitted to Journal of Sensor and Actuator Networks, Special Issue on Advances in Sensor Network Operating Systems, February 2013.
- [69] R. Soua, E. Livolant, P. Minet, *MUSIKA : a Multichannel Multi-sink Data Gathering Algorithm in Multichannel Wireless Sensor Networks*, IEEE IWCMC 2013, Cagliari, Sardinia, Italy, July 2013.
- [70] C. H. Papadimitriou, *The Complexity of the Capacitated Tree Problem*, Networks, vol. 8, n 3, 1978.
- [71] A. Ghosh, O. D. Incel, A. Kumar, B. Krishnamachari, *Multi-Channel Scheduling Algorithms for Fast Aggregated Convergecast in Sensor Networks*, The 6th IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS), 2009.
- [72] <http://www.gnu.org/software/glpk/>
- [73] <http://www.gnu.org/software/octave/>
- [74] D. Fotue, F. Melakessou, H. Labiod, T. Engel, *A Distributed Hybrid Channel selection and Routing technique for Wireless sensor Networks*, IEEE VTC Fall 2011.
- [75] J. Li, X. Guo, L. Guo, *Joint Routing, Scheduling and Channel-Assignment in multi-Power multi-Radio Wireless Sensor Networks*, IPCCC 2011, Orlando, Florida, November 2011.
- [76] B. Han, V.S Anil Kumar, M. V. Marathe, S. Parthasarathy, A. Srinivasan, *Distributed Strategies for Channel Allocation and Scheduling in Software-Defined Radio Networks*, Infocom 2009, Rio de Janeiro, Brazil, 2009.