HAL
open science

Algorithme distribué pour l'assignation de rôles dans Molecular MAC, une couche MAC multicanal

Benoit Darties, Fabrice Theoleyre, Andrzej Duda

To cite this version:

Benoit Darties, Fabrice Theoleyre, Andrzej Duda. Algorithme distribué pour l'assignation de rôles dans Molecular MAC, une couche MAC multicanal. CFIP'2009, Oct 2009, Strasbourg, France. inria-00419467

HAL Id: inria-00419467

https://inria.hal.science/inria-00419467v1

Submitted on 23 Sep 2009

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.

Algorithme distribué pour l'assignation de rôles dans Molecular MAC, une couche MAC multicanal

Benoît Darties, Fabrice Theoleyre, Andrzej Duda

Laboratoire d'Informatique de Grenoble (LIG)
CNRS / Grenoble INP
681 rue de la passerelle, BP72
38402 Saint Martin d'Hères Cedex, France
firstame.lastname@imag.fr

RÉSUMÉ. *Une couche MAC multicanal semble une approche intéressante afin d'optimiser le débit d'un réseau maillé radio en multiplexant les transmissions sur des canaux orthogonaux. En particulier, Molecular MAC proposé récemment étends* IEEE *802.11 DCF afin d'offrir une telle fonctionnalité. Cependant, ce protocol requiert une assignation de rôles et canaux : certains noeuds utilisent un canal statique tandis que les autres sautent continuellement de canal. Nous étendons ici la notion d'ensemble dominant faiblement connecté (WCDS) en introduisant de nouvelles contraintes pour constituer ce que nous appelons un* reversible-WCDS *(r-WCDS), directement utilisable par Molecular MAC. Nous proposons de partitionner le réseau en zones, un chef par zone optimisant localement l'assignation des rôles grâce à un* PLNE. *En définissant de façon appropriée les rôles pour les bordures des zones, nous maintenons une connexité globale dans le r-WCDS. Finalement, les simulations montrent que l'approche proposée a des performances comparables à un algorithme centralisé optimal.*

ABSTRACT. *A multi-channel MAC seems to be an interesting approach for improving network throughput by multiplexing transmissions over orthogonal channels. In particular, Molecular MAC has recently proposed to extend the standard* IEEE *802.11 DCF access method to provide such a feature. However, this MAC layer requires role and channel assignment to nodes: some of them use a static channel, while others dynamically switch to neighbor channels on-demand. We extend here the notion of the Weakly Connected Dominating Set by introducing new constraints to define what we call a* reversible WCDS*, directly used by Molecular MAC. We parition the network into regions (clusters), one leader per cluster optimizes the role assignment by solving a* MILP. *By appropriately defining the roles at the border of clusters, we maintain global connectivity in the r-WCDS. Finally, our simulations show that the performance of the propose scheme is close to a centralized optimal algorithm.*

MOTS-CLÉS : *Molecular MAC, multi-canal, réseaux maillés, r-WCDS, algorithmes distribués*

KEY WORDS: *Molecular MAC, multichannel, mesh networks, r-WCDS, distributed algorithms*

1. Introduction

Nous considérons dans cet article les *réseaux maillés sans-fil* utilisant la norme IEEE 802.11 pour interconnecter les routeurs maillés. Lorsque ces routeurs utilisent seulement une interface et un seul canal radio, les performances du réseau se dégradent rapidement avec le nombre de sauts sur lesquels les paquets sont relayés, à cause de la contention sur la canal et des problèmes spatiaux tels que les terminaux bloqués, cachés ou exposés[CHA 05]. Afin d'améliorer la capacité, plusieurs auteurs ont proposé d'utiliser des routeurs avec des interfaces pouvant commuter dynamiquement de canal radio pour chaque trame.

Certains protocoles MAC proposent de changer de canal à la demande. Parmi ces approches, MMAC (Multi-Channel MAC) [SO 04] utilise un mécanisme de Rendez-Vous pour effectuer des réservations de canaux, utilisés ultérieurement pour l'échange des données. Une autre approche, proposée par Molecular MAC [NAS 09] étends IEEE 802.11 en tirant parti d'une architecture moléculaire, basée sur une distribution hétérogène de rôles et de canaux. Ces rôles différents permettent à la couche MAC de multiplexer efficacement les transmissions, résolvant ainsi le problème classique de surdité aux transmissions sur les autres canaux. Des simulations ont montré que Molecular MAC offrait des performances bien supérieures à celles de type IEEE 802.11 mono-canal et MMAC.

Nous nous intéressons ici au problème de construction de la structure associée à Molecular MAC. En effet, [NAS 09] se focalise sur les aspects protocolaires de la couche MAC multicanal, le problème d'attribution distribuée des rôles ayant été laissée à un travail ultérieur. Nous souhaitons garantir une connexité globale (un noeud peut communiquer avec tout autre noeud via une communication multisaut) et maximiser la capacité du réseau.

La contribution de ce papier est double. Dans un premier temps, nous définissons formellement le problème de la construction de la structure associée à Molecular MAC sous la forme d'un r-WCDS. Par ailleurs, nous proposons une approche distribuée diviser-pour-régner, permettant d'allouer de façon efficace les rôles dans un réseau maillé. Les simulations montrent que cette approche présente des performances proches d'une approche centralisée optimale.

Dans une première partie, nous introduisons la formulation de notre problème, et expliquons son rapport avec Molecular MAC. La section 3 détaillera la résolution centralisée optimale sous forme de programme linéaire en nombre entiers. La section suivante introduira notre algorithme distribué, dont les performances seront évaluées dans la section 5. La section 6 introduira l'état de l'art du domaine, puis nous conclurons cet article en donnant quelques perspectives.

2. Formulation du problème

Nous modélisons le réseau comme un graphe non dirigé $G = (V, E)$ dont les sommets $V(G)$ sont les noeuds du réseau et les arêtes $E(G)$ sont toutes les paires de noeuds capables de communiquer directement. Nous adoptons la notation classique suivante :

– $n = |V|$ définit le nombre de noeuds dans le réseau,
– $N(u)$ représente les voisins de u avec la cardinalité $\Delta(u) = |N(u)|$,
– $\{u, v\}$ représente le lien entre u et v, i.e. $\{u, v\} \in E$,

– BW représente la capacité du canal,

– CH est l'ensemble des canaux disponibles $nb_{CH} = |CH|$ (IEEE 802.11a offre par exemple 13 canaux orthogonaux en Europe).

2.1. *Le problème de Reversible Weakly Connected Dominating Set*

L'ensemble dominant faiblement connexe (WCDS) [DUN 97] est une structure largement utilisée dans les réseaux radio multisauts. Formellement, un WCDS est défini par l'emsemble $D \subseteq V$ tel que :

$$\forall u \in \{V - D\} \quad , \quad \exists v \in D | v \in N(u) \tag{1}$$

$$G = (V, E') \; connecté \quad | \quad E' = \big\{\{u, v\}, u \in D, v \in V\big\} \tag{2}$$

Les noeuds de D sont souvent appelés *dominants* tandis que les noeuds de $V - D$ sont appelés *dominés*.

Nous définissons un *WCDS réversible* (noté r-WCDS dans le reste du papier) en adaptant la définition des WCDS : nous gardons seulement les liens $(dominant, dominé)$, i.e. les liens entre dominants sont enlevés. Nous verrons que Molecular MAC se base sur une telle structure. Formellement, nous transformons la deuxième propriété de l'équation 2 en :

$$G = (V, E') \; connecté \quad | \quad E' = \big\{\{u, v\}, u \in D, v \in \{V - D\}\big\} \tag{3}$$

2.2. *Lien avec Molecular MAC*

Molecular MAC requiert une structure de r-WCDS pour fonctionner. Ce protocole nécessite de diviser un réseau maillé radio en atomes répartis spatialement de telle sorte que chaque atome utilise un canal différent des atomes interférents. Un atome est composé d'un *noyau* central et d'*électrons*. Le noyau choisit le canal qui sera utilisé pour toutes les transmissions au sein de son atome, et reste en écoute sur ce canal. Au contraire, les électrons se trouvent à la bordure des atomes et sautent dynamiquement de canal : ils relaient le trafic d'un atome à l'autre.

Nous pouvons comparer Molecular MAC à un WLAN étendu : un atome corresponds à la cellule WLAN et un noyau joue le rôle de *point d'accès virtuel*, interconnectant les clients (électrons). Dans un réseau maillé, un client communique avec différentes cellules. Cependant, un noeud ne doit pas manquer des paquets venant de son point d'accès virtuel, par exemple parce qu'il est en transmission dans une autre cellule, sur un autre canal. Aussi, Molecular MAC implémente des mécanismes spécifiques : l'AP place toutes les trames à destination des électrons dans une file d'attente et annonce périodiquement la liste des destinations en attente. Un électron n'a plus qu'à *tirer* explicitement les trames de l'AP lorsqu'il sait que certains des paquets sont en attente pour lui.

Dans un réseau maillé moléculaire, nous avons besoin d'assigner un rôle à chaque noeud (soit noyau, soit électron) de telle sorte que le réseau résultant possède les propriétés suivantes :

1) un noeud peut communiquer avec tout autre noeud (potentiellement en multisaut) ;

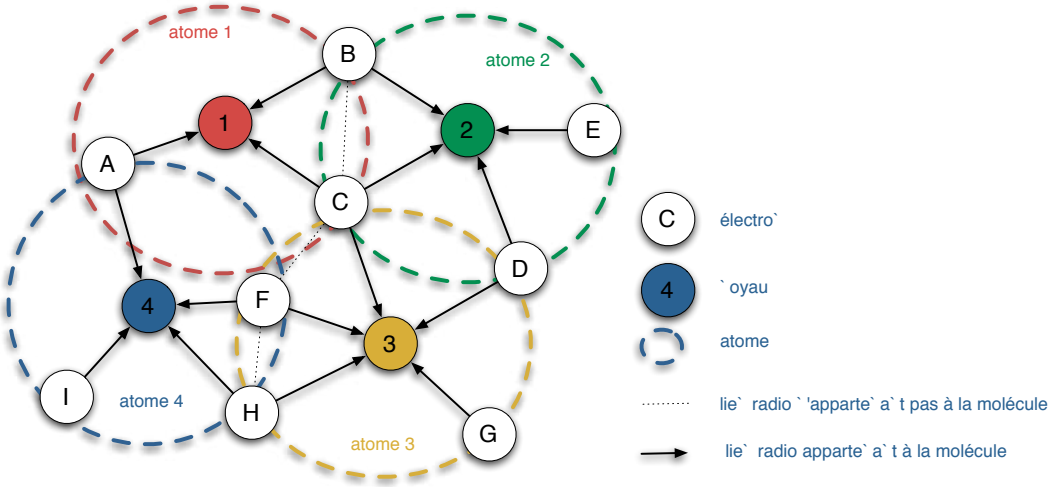

Figure 1. *Exemple de topologie associée à molecular MAC*

2) seuls les noyaux et électrons peuvent communiquer directement, i.e. il n'existe pas de communication directe entre deux électrons ou entre deux noyaux. En effet, deux noyaux ne peuvent pas s'entendre car ils utilisent un canal différent. De façon similaire, deux électrons sautent continuellement de canal et pourraient souffrir de surdité ;

3) la capacité du réseau doit être maximale. En particulier, deux atomes voisins ne devraient pas interférer et devraient utiliser des canaux orthogonaux pour leurs transmissions.

La figure 1 illustre une telle organisation. Les identifiants des noyaux sont représentés par des nombres tandis que ceux des électrons le sont par des lettres. Le lecteur pourra noter que les liens entre électrons ne sont pas utilisés puisque Molecular MAC les interdit pour éviter toute surdité. Cependant, il faut minimiser le nombre de ces liens radio interdits par Molecular MAC : nous maintiendrons ainsi plus de chemins redondants, et augmenterons ainsi la tolérance aux fautes. Il apparaît clairement que les contraintes pour l'assignation de tels rôles mène à un r-WCDS : chaque dominant corresponds à un noyau, et chaque dominé à un électron.

Par ailleurs, chaque noyau doit sélectionner un *bon* canal pour son atome. Cependant, une assignation efficace conjointe des rôles et canaux est complexe, les deux problèmes étant très interdépendants. Nous avons donc choisi de procéder de façon séquentielle, afin d'offrir un meilleur compromis entre performances et rapidité de calcul. Un noyau nouvellement élu choisit son canal selon une approche gloutonne : il sélectionne le canal le moins chargé après en avoir mesuré l'activité.

3. Résolution centralisée sous forme d'un Programme Linéaire en Nombres Entiers

Nous proposons dans cette section une formulation en PLNE (programmation linéaire en nombres entiers) du problème consistant à trouver un r-WCDS maximisant la capacité du réseau, dont la résolution centralisée donne une affectation de rôles (noyau ou électron) aux nœuds du réseau. Pour se faire, nous définissons les variables suivantes :

– Le rôle de chaque nœud $u \in V$ est désigné par la variable $role(u) \in \{0, 1\}$ qui prend 1 si u est dominant (i.e. noyau), et 0 s'il est dominé (i.e. électron).

– nous supposons qu'il existe une commnuication entre chaque paire de nœuds, soit $n(n-1)$ flots multi-sauts. Notre objectif est de maximiser T_{min}, le débit minimum affecté à chaque flot.

– $T(u, v, d)$ correspond au trafic transmis par u sur le lien $\{u, v\}$ à destination de d. Notons que $\forall \{u, v\} \in E,\ T(u, v, u) = 0$ (i.e. un nœud ne génère pas de trafic pour lui-même) ;

Tout d'abord, un lien est utilisable si et seulement si ses deux extrémités ont des rôles différents. Sa capacité (la somme des $T(u, v, d)$ pout toute les destinations d) est nulle si ses deux extrémités sont des dominants (Eq. 4) ou des dominés (Eq. 5) :

$$\forall \{u, v\} \in E, \sum_{d \in V} \big(T(u, v, d) + T(v, u, d)\big) \leq BW\,(2 - role(u) - role(v)) \tag{4}$$

$$\forall \{u, v\} \in E, \sum_{d \in V} \big(T(u, v, d) + T(v, u, d)\big) \leq BW\,(role(u) + role(v)) \tag{5}$$

Les équations 6 et 7 expriment la loi de conservation des flots[1], tandis que l'équation 7 représente le fait qu'une destination doit recevoir $(n-1).T_{min}$ unités de trafic (T_{min} pour chaque source) :

$$\forall u, d \in V, d \neq u, \quad \sum_{v \in N(u)} T(u, v, d) = \sum_{v \in N(u)} T(v, u, d) + T_{min} \tag{6}$$

$$\forall u \in V, \quad \sum_{v \in N(u)} (n-1).T_{min} = T(v, u, u) \tag{7}$$

Le trafic sortant et entrant d'un noeud partage la bande passante BW :

$$\forall u \in V, \quad \sum_{v \in N(u)} \sum_{d \in V} \big(T(u, v, d) + T(v, u, d)\big) \leq BW \tag{8}$$

Si u est dominant, cette équation signifie que tout le trafic à l'intérieur d'un atome (le noyau est soit source, soit destination) partage le même canal. Symétriquement, lorsque u est dominé, il doit partager son temps entre les atomes auxquels il appartient.

La résolution du PLNE est coûteuse dans des réseaux étendus : plusieurs heures sont nécessaires pour trouver l'assignation d'un réseau maillé de 40 noeuds.

1. la somme du trafic vers d envoyé par u est égale à la somme du trafic vers d entrant dans u et du trafic généré par u vers d ($= T_{min}$)

4. Une approche diviser-pour-régner : `potatoes`

Nous proposons ici une approche de type *diviser pour régner* : nous partitionnons le réseau en clusters dans chacun desquels un *leader* résout le problème PLNE associé au cluster. Une fois le résultat calculé, il assigne les rôles aux autres membres du cluster. La taille limitée d'un cluster permets de résoudre rapidement le PLNE. Cependant, nous devons ajouter des contraintes additionnelles de telle sorte que l'union des différents r-WCDS locaux forme bien globalement un r-WCDS.

Nous proposons tout d'abord de créer un *arbre de clusters enraciné* (fig. 2), un arbre dans lequel chaque sommet corresponds à un cluster, et une arête existe entre deux clusters ssi ils partagent un noeud commun. Nous supposons que le réseau est connecté : si plusieurs composantes connexes existent, chacune exécutera l'algorithme indépendamment.

Il existe un chef par cluster, récupérant la topologie de son cluster, i.e. les noeuds appartenant à ce cluster, et les liens radio unissant deux membres. Chaque leader résout le PLNE basé sur cette topologie, attribuant les rôles de façon optimale si on ne considère que les liens intra-clusters. Une fois le PLNE résolu, le chef envoie les rôles calculés à tous les membres du cluster.

Afin de maintenir le r-WCDS globalement connexe, nous ajoutons des contraintes à l' *arbre de clusters enraciné* : un noeud appartient à au plus deux clusters et l'intersection de deux clusters contient exactement un noeud. Ainsi, les rôles peuvent être attribués hiérarchiquement : un noeud appartenant à deux clusters est chef pour un des clusters (le plus bas dans l'arbre de clusters) et simple membre pour l'autre. Un chef de cluster résout donc le PLNE en tenant compte du fait que son propre rôle a déjà été attribué par son chef, plus haut dans la hiérarchie. La forme hiérarchique de l'arbre de clusters assure qu'il n'existe pas de conflit dans l'attribution.

4.1. *Notations*

Afin de décrire plus formellement l'algorithme, nous adoptons les notations suivantes :

– un cluster est un sous-graphe connexe $\mathcal{C}luster$ du graphe initial G induit par le sous-ensemble des noeuds $S \subseteq V(G)$ avec $V(\mathcal{C}luster) = S$ et $E(\mathcal{C}luster) = \big\{\{u,v\} | u,v \in V(G) \; et \; \{u,v\} \in E(G)\big\}$;

– $\mathcal{T}$ représente un arbre recouvrant utilisé pour l'arbre de clusters ;

– le chef global $\mathcal{R}oot\mathcal{L}eader$ est la racine de l'arbre $\mathcal{T}$. Le cluster contenant $\mathcal{R}oot\mathcal{L}eader$ représente la racine de l'arbre de clusters ;

– $V(\mathcal{C}luster)$ représente l'ensemble des noeuds appartenant à $\mathcal{C}luster$;

– $\mathcal{L}eader(\mathcal{C}luster)$ représente le chef de $\mathcal{C}luster$. $leader(\mathcal{C}luster) = V(\mathcal{C}luster) \cap V(parent(\mathcal{C}luster))$ avec $parent(\mathcal{C}luster)$ étant le cluster plus faut dans l'arbre (excepté pour le cluster contenant $\mathcal{R}oot\mathcal{L}eader$ puisque ce noeud ne possède pas de père) ;

– les liens *intra-cluster* sont les liens unissant deux noeuds d'un même cluster tandis que les autres sont appelés des liens *inter-cluster* ;

– $role(u)$ représente le rôle de u (soit dominant, soit dominé).

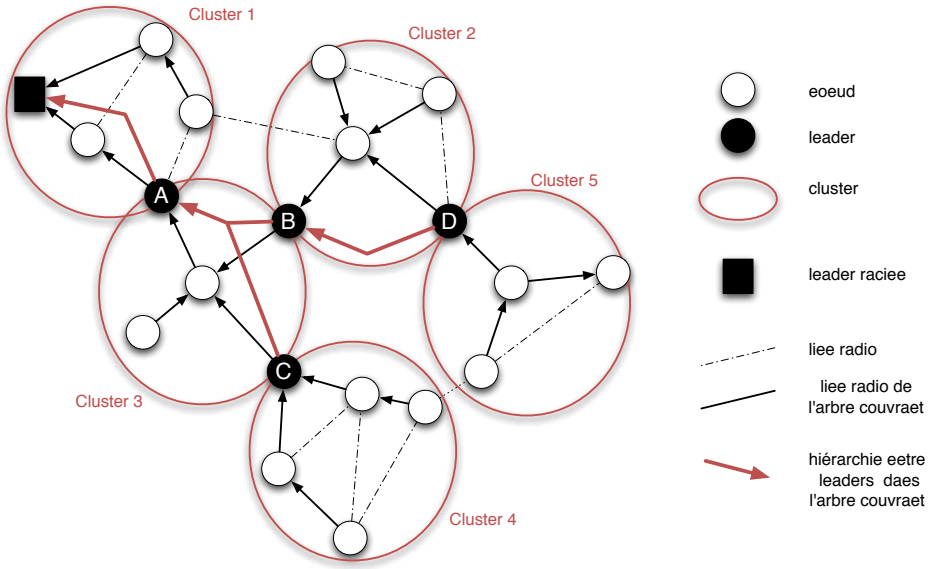

Figure 2. *Un exemple de structure en arbre de clusters*

4.2. *Construction de l'arbre de clusters*

Nous supposons que chaque noeud envoie périodiquement des paquets `hellos` pour découvrir son voisinage. Ainsi, nous ajoutons juste des champs supplémentaires aux `hellos`.

La construction commence en élisant tout d'abord un chef global ($\mathcal{R}oot\mathcal{L}eader$) : chacun propage l'identifiant minimum reçu auparavant. Afin d'éviter tout problème de convergence, chaque noeud inclus dans ses `hellos` les champs $< min_id, distance, seq_nb >$. Un noeud met à jour ces informations ssi le `hello` contient un numéro de séquence supérieur au numéro de séquence courant. Puisque seul le $\mathcal{R}oot\mathcal{L}eader$ incrémente le numéro de séquence dans ses paquets `hellos`, chaque noeud est capable de détecter les déconnexions. Ainsi, si le numéro de séquence n'est pas incrémenté durant T_{dead}, tout noeud se réélit comme $\mathcal{R}oot\mathcal{L}eader$, et l'élection repart du début. Par ailleurs, les noeuds mettant à jour leur distance à la racine, chacun peut choisir un père parmi les noeuds s'annonçant d'un saut plus proche du $\mathcal{R}oot\mathcal{L}eader$. Chaque noeud annonce dans ses `hellos` le père choisi de telle sorte que tout noeud de l'arbre puisse connaître ses fils.

Nous définissons ensuite l'arbre de clusters et les chefs en se basant sur la profondeur de chaque noeud dans l'arbre : un chef est élu lorsque sa distance au chef plus haut dans la hiérarchie est d'exactement $D + 1$ sauts (pour former des clusters de rayon D sauts). Pour ce faire, tout noeud s'élit chef de cluster s'il est à exactement $\equiv 0\ [D]$ sauts du $\mathcal{R}oot\mathcal{L}eader$ est ($[D]$ signifie ici *modulo D*). Exception est faite pour un noeud ne possédant aucun fils dans l'arbre, car il serait chef d'un cluster qui ne contiendrait que lui même. Un tel noeud ne sera donc que simple membre dans le cluster le plus haut dans la hiérarchie.

Considérons la topologie de la figure 2 avec un rayon de cluster de 2. La figure représente l'arbre recouvrant utilisé pour la construction de l'arbre de clusters. Chaque noeud est à au plus 2 sauts de son chef, tous les noeuds de profondeur paire s'élisant chef. Par ailleurs, `potatoes` crée bien une hiérarchie : B est leader du cluster 2 et constitue un des membres du cluster 3, coordonné par le chef A. Dans cette topologie, A assignera les rôles dans son cluster (incluant le rôle de B) de telle sorte que le cluster 3 reste connexe (il existera un chemin reliant C et B alternant les rôles). Récursivement, B assignera les rôles, en sachant que le sien a déjà été assigné par A. Le lecteur pourra vérifier que nous trouverons toujours un chemin alternant les rôles dans un même cluster. Comme tous les clusters sont organisé en arbre, nous avons bien globalement un r-WCDS.

4.3. *Topologie du cluster*

Comme chaque chef de cluster doit résoudre le PLNE, il doit connaître la topologie de son cluster. L'arbre nous aide à optimiser la diffusion d'informations. Initialement, chaque noeud utilise des `hellos` pour découvrir ses voisins. Nous proposons que par ailleurs chaque noeud maintienne une *table de topologie* contenant tous les liens dont une des extrémités est un fils dans l'arbre $\mathcal{T}$. Cette *table de topologie* est restreinte aux fils du même cluster et est alimentée récursivement : chaque noeud inclut sa *table de topologie* dans les `hellos`, mise à jour en fonction des informations reçues de ses fils.

Pour être tolérant aux fautes, chaque noeud met à jour sa *table de topologie*. Il inclue notamment pour chaque entrée le fils dont l'entrée vient. Un nouveau paquet `hello` de ce même fils écrasera naturellement toutes ses précédentes entrées, devenues obsolètes.

4.4. *Assignation des rôles*

Lorsque sa *table de topologie* reste stable pendant suffisamment longtemps, un chef de cluster résout le PLNE associé et envoie les rôles calculés aux membres de son cluster.

Afin d'accélérer la convergence de `potatoes`, nous forçons chaque chef de cluster à posséder un rôle prédéfini en ajoutant une contrainte au PLNE :

– $\mathcal{R}oot\mathcal{L}eader$ devient le premier dominant ;

– Si le rayon de cluster (D) est pair, tout chef de cluster peut devenir dominant. En effet, un chemin alternant dominants et dominés peut toujours relier tout nœud (et notamment un chef pour un cluster plus bas dans l'arbre de clusters) à son chef de cluster[2] ;

– si le rayon de cluster est impair, un chef de cluster doit avoir le rôle inverse[3] de son propre chef de cluster (plus haut dans la hiérarchie). . En alternant dominants et dominés, nous aurons un chemin valide.

2. le PLNE pourra toujours trouver une assignation faisable en colorant tous les noeuds de profondeur paire en dominant, les autres en dominés
3. le rôle inverse de dominant (respt. dominé) est dominé (respvt. dominant).

Chaque chef de cluster $\mathcal{L}$ saura donc immédiatement après son élection le rôle qu'il doit prendre en fonction de sa profondeur dans l'arbre et de la parité du rayon de cluster. Le rôle est fixé dans le PLNE en ajoutant une contrainte linéaire $role(\mathcal{L}) = dominant$ ou $role(\mathcal{L}) = dominé$ selon le rôle choisi. Ainsi, un chef n'a pas à attendre la décision d'autres chefs de cluster, et peut exécuter de façon distribuée le PLNE dés qu'il connaît la topologie de son cluster.

Cette optimisation simple accélère la convergence et mène à de bons résultats. De plus, nous gardons une connexité grâce à l'alternance des rôles dans chaque cluster. Considérons la topologie de la figure 2. Le PLNE pourra trouver une solution faisable si nous obligeons tous les chefs de cluster à être dominants. Par exemple, A dans le cluster 3 exécutera son PLNE avec les rôles de A, B et C fixés à dominants.

`potatoes` est distribué, et permet d'augmenter le passage à l'échelle en trouvant une solution faisable en un temps raisonnable. Il est également tolérant aux fautes, prends en compte les changements de topologie (en élisant une nouvelle racine, en mettant à jour l'arbre recouvrant, en réexécutant le PLNE sur la topologie modifiée d'un cluster, etc.) Nous pouvons ajouter un mécanisme permettant de filtrer la qualité des liens : seuls les liens de qualité suffisante sont reportés au leader du cluster. Par ailleurs, un lien ne comportant que des liens radio faibles sera ajouté comme feuille à la topologie intra-cluster.

5. Évaluation de performances

Nous avons simulé le protocole proposé dans WsNet [CHE 06] en utilisant la librairie de programmation linéaire COIN-CBC [Coi]. Les noeuds sont placés aléatoirement sur la surface de simulation. Ils utilisent IEEE 802.11a pour communiquer, avec un rayon de portée radio de 10 unités et un rayon d'interférence de 30 unités. Par défault, le réseau compte 50 noeuds avec en moyenne 10 voisins. Nous ajustons la surface de simulation pour maintenir le degré constant.

Nous présentons sur les graphes les moyennes de 10 simulations différentes sur 240 secondes. Nous avons également représenté les intervalles de confiance de 95%. Nous avons comparé les performances de la formulation PLNE centralisée (OPT) et `potatoes` avec l'ensemble indépendant maximum (MIS) et un arbre recouvrant auto-stabilisant (ST) présentés dans [THE 09]. Nous avons mesuré :

– le débit minimum T_{min} garanti pour chaque flot, extrait du PLNE. Nous considérons une capacité du canal normalisée (i.e. $BW = 1$) ;

– le stretch factor moyen : le ratio moyen de la route la plus courte via le r-WCDS et celle via le graphe d'origine. Un stretch factor de 1 signifie que le routage utilise les plus courts chemins.

5.1. *Stretch factor de la route*

Nous avons tout d'abord mesuré le stretch factor de la route (fig. 3). Pour MIS, nous avons retiré les noeuds isolés sans quoi le stretch factor aurait été infini. Nous avons donc tendance à le sous-estimer pour le MIS. OPT obtient un stretch factor d'environ 1,3 quel que soit le nombre de noeuds. Cependant, OPT ne passe pas à l'échelle et les résultats n'ont pas pu être obtenus pour des réseaux de

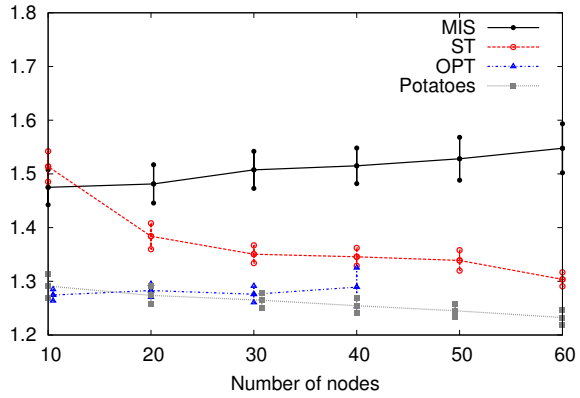

Figure 3. *Stretch factor moyen de la route pour un nombre variable de noeuds*

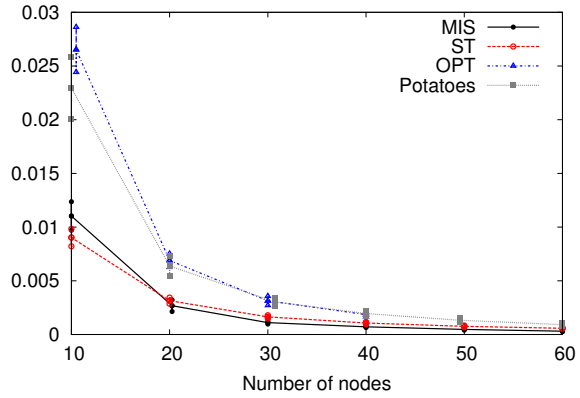

Figure 4. *Minimum throughput T_{min} for a varying number of nodes in a random network.*

plus de 40 noeuds (50 noeuds requièrent plus de 3 heures). Ceci explique que les points représentés s'arrêtent à 40 noeuds pour OPT.

Les performances de `potatoes` et OPT sont très similaires. `potatoes` arrive à construire un r-WCDS et maximise le nombre de liens radio maintenus dans le r-WCDS : les routes sont souvent les plus courtes. Par ailleurs, `potatoes` passe bien mieux à l'échelle. Une approche diviser-pour-régner est donc bien pertinente pour notre problème.

ST utilise des routes plus longues mais le stretch factor a tendance à diminuer lorsqu'il existe plus de noeuds. Le stretch factor de MIS est trop large : une route longue consommera plus de bande passante, car elle nécessitera plus de transmissions.

5.2. *Capacité*

Finalement, nous avons mesuré le débit minimum T_{min} (fig 4). Cette métrique corresponds au débit que nous pourrions obtenir avec Molecular MAC.

Naturellement, OPT donne la borne supérieure car cet algorithme optimise justement l'assignation de rôle pour maximiser cet objectif. Cependant, les problèmes de passage à l'échelle rendent une telle stratégie inapplicable en pratique. `potatoes` obtient par ailleurs des résultats très proches de OPT, tout en distribuant les décisions.

Nous pouvons noter que MIS et ST obtiennent un débit bien moindre : ces algorithmes ne maximisent pas le nombre de liens restant dans le r-WCDS, ce qui diminue le débit possible. Les performances obtenues sont inférieures d'au moins 50% à celles obtenues avec `potatoes`.

6. État de l'art

Le clustering permets de grouper des noeuds, concept particulièrement intéressant pour introduire une hiérarchie (e.g. pour un protocole de routage). Il est souvent basé sur le concept de domination : des noeuds élisent des chefs de cluster et tous leurs voisins deviennent membre du cluster associé [LIN 97]. Cependant, des chefs de clusters peuvent dans certains cas être séparés par exactement 2 sauts, nécessitant l'élection de passerelles pour les interconnecter.

L'ensemble dominant faiblement connecté (WCDS) est une structure souvent utilisée, par exemple pour le clustering ou pour distribuer des clés partagées dans un MANET [PAT 06]. Cependant, trouver un WCDS en minimisant sa cardinalité a été prouvé comme NP difficile [DUN 97]. Domke et al. [DOM 05] ont étendu ce résultat en caractérisant les graphes possédant la même cardinalité minimale pour un WCDS et un DS. Cependant, les auteurs se focalisent sur des graphes particuliers (e.g. des arbres possédant des propriétés spécifiques) et ne résolvent pas le problème dans un graphe général.

Chen et al. [CHE 02] ont étendu l'algorithme centralisé de Guha et al. [GUH 98] en sélectionnant à chaque itération les meilleurs noeuds à ajouter au WCDS, i.e. la composante formant le WCDS grossit à chaque étape. Dubashi et al. [DUB 05] proposent d'étayer les liens appartenant à un cycle, i.e. ils créent un réseau moins dense. Bien que l'ensemble forme un ensemble dominant connecté, il ne peut pas être utilisé pour construire un r-WCDS, les propriétés recherchées étant différentes.

Alzoubi et al. [ALZ 03] proposent de construire un ensemble indépendant maximum, les chefs de clusters étant élus selon leur profondeur dans un arbre recouvrant. Ainsi, cet algorithme est proche de *ST*, auquel `potatoes` a été comparé.

Dans notre approche, nous nous sommes inspirés des idées de Chen et al. [CHE 05] : les auteurs partitionnent le réseau en zones dans chacune desquelles s'exécute l'algorithme. Cependant, leur algorithme est indépendamment appliqué à chaque zone. Aussi, il ne peut être directement appliqué pour construire un r-WCDS. Par ailleurs, nous avons également pris en compte d'autres critères de performances comme la capacité réseau. Han et al. [HAN 07] adoptent également une approche de partitionnement, mais leur approche présente les mêmes limites.

7. Conclusion et perspectives

Nous avons proposé ici une définition formelle du problème visant à construire la structure associée à Molecular Mac, assignant les rôles au niveau de la couche MAC. Après avoir présenté une

résolution centralisée sous forme de PLNE, conduisant à un résultat optimal pour de petites instances, nous avons introduit un nouvel algorithme –`potatoes`– permettant de distribuer les décisions. Cet algorithme permets d'optimiser l'assignation des rôles et mène à de bonnes performances, proches de l'optimal. Nous souhaitons maintenant nous intéresser à l'optimisation conjointe des rôles et canaux, afin d'obtenir des solutions avec garantie de performance incluant l'affectation de canaux.

8. Bibliographie

[ALZ 03] ALZOUBI K. M., WAN P.-J., FRIEDER O., « Weakly-Connected Dominating Sets and Sparse Spanners in Wireless Ad Hoc Networks », *ICDCS*, Providence, USA, May 2003, IEEE.

[CHA 05] CHAUDET C., DHOUTAUT D., GUERIN LASSOUS I., « Performance Issues with IEEE 802.11 in Ad Hoc Networking », *IEEE Communications Magazine*, vol. 43, n° 7, 2005.

[CHE 02] CHEN Y. P., LIESTMAN A. L., « Approximating minimum size weakly-connected dominating sets for clustering mobile ad hoc networks », *MOBIHOC*, Lausanne, Switzerland, June 2002, ACM, p. 165–172.

[CHE 05] CHEN Y. P., LIESTMAN A. L., « Maintaining weakly-connected dominating sets for clustering ad hoc networks », *Ad Hoc Networks*, vol. 3, 2005, p. 629–642.

[CHE 06] CHELIUS G., FRABOULET A., FLEURY E., « WSNet : a modular event-driven wireless network simulator », `http://www.worldsens.net/`, 2006.

[Coi] COIN-OR BRANCH AND CUT, « `https://projects.coin-or.org/Cbc` ».

[DOM 05] DOMKE G. S., HATTINGH J. H., MARKUS L. R., « On weakly connected domination in graphs II », *Discrete Mathematics*, vol. 305, n° 1–3, 2005, p. 112–122.

[DUB 05] DUBHASHI D., MEI A., PANCONESI A., RADHAKRISHNAN J., SRINIVASAN A., « Fast Distributed Algorithms for (Weakly) Connected Dominating Sets and Linear-Size Skeletons, », *Journal of Computer and System Sciences*, vol. 71, n° 4, 2005, p. 467–479.

[DUN 97] DUNBAR J. E., GROSSMAN J. W., HATTINGH J. H., HEDETNIEMI S. T., MCRAE A. A., « On weakly connected domination in graphs », *Discrete Mathematics*, vol. 167-168, 1997.

[GUH 98] GUHA S., KHULLER S., « Approximation algorithmsforconnecteddominatingsets. », *Algorithmica*, vol. 20, n° 4, 1998, p. 374–387.

[HAN 07] HAN B., JIA W., « Clustering wireless ad hoc networks with weakly connected dominating set », *Journal of Parallel and Distributed Computing*, vol. 67, n° 6, 2007, p. 727–737.

[LIN 97] LIN C. R., GERLA M., « Adaptive Clustering for Mobile Wireless Networks », *IEEE Journal of Selected Areas in Communications*, vol. 15, n° 7, 1997, p. 1265–1275.

[NAS 09] NASSIRI M., THEOLEYRE F., HEUSSE M., DUDA A., « Molecular Architecture for Autonomic Wireless Mesh Networks », submitted, `http://membres-liglab.imag.fr/theoleyre/uploads/molecular_research_report.pdf`, 2009.

[PAT 06] PATHAN A.-S. K., HONG C. S., « A Key-Predistribution-Based Weakly Connected Dominating Set for Secure Clustering in DSN », LNCS, Ed., *HPCC*, vol. 4208, Munich, Germany, September 2006, p. 270–279.

[SO 04] SO J., VAIDYA N. H., « Multi-channel mac for ad hoc networks : handling multi-channel hidden terminals using a single transceiver. », *MOBIHOC*, Japan, 2004, ACM.

[THE 09] THEOLEYRE F., DARTIES B., DUDA A., « Assignment of Roles and Channels for a Multichannel MAC in Wireless Mesh Networks », *ICCCN*, San Francisco, USA, August 2009, IEEE.