

## Mémoire

*présenté par*

**Fabrice VALOIS**

*en vue de l'obtention du diplôme*

**d'habilitation à diriger des recherches  
de l'INSA de Lyon et  
de l'Université Claude Bernard - Lyon 1  
{Numéro d'ordre HDR 2007 014}**

### **AUTO-ORGANISATION DE RÉSEAUX RADIO MULTI-SAUTS**

Soutenance le 19 novembre 2007

Après avis de :

Prof. Andrzej Duda  
Prof. Catherine Rosenberg  
Prof. David Simplot-Ryl

Devant la commission d'examen formée de :

Rapporteur	Pr. Andrzej Duda	INP Grenoble-Ensimag
Rapporteur	Pr. Catherine Rosenberg	Université de Waterloo, Canada
Rapporteur	Pr. David Simplot-Ryl	Université de Lille I
Examineur	Pr. Serge Fdida	Université Paris 6
Examineur	Pr. Jean-Michel Fourneau	Université de Versailles - Saint Quentin
Examineur	Pr. Isabelle Guérin-Lassous	Université de Lyon I
Examineur	Pr. Stéphane Ubéda	INSA Lyon



# Remerciements



# Table des matières

<b>1</b>	<b>Introduction, motivations, contexte</b>	<b>3</b>
<b>2</b>	<b>Auto-organisation pour réseaux radio multi-sauts</b>	<b>9</b>
2.1	Vue d'ensemble . . . . .	9
2.2	Définition et objectifs de l'auto-organisation . . . . .	10
2.2.1	Définition . . . . .	10
2.2.2	Principes et objectifs . . . . .	10
2.2.3	Décisions locales versus distribuées ? . . . . .	12
2.3	Quelques clefs pour l'auto-organisation . . . . .	12
2.3.1	Émergence d'un comportement global à partir de règles locales . . . . .	12
2.3.2	Interactions locales et coordination des nœuds . . . . .	13
2.3.3	Minimiser les informations sur l'état du réseau . . . . .	13
2.3.4	Adaptation dynamique à l'environnement . . . . .	14
2.4	Synthèse des propriétés clefs . . . . .	14
2.5	Entropie, auto-organisation et réseaux radio multi-sauts . . . . .	15
2.5.1	De la thermodynamique aux réseaux radios multi-sauts . . . . .	15
2.5.2	Entropie : application à l'auto-organisation . . . . .	16
2.5.3	Mesure de l'entropie . . . . .	18
2.5.4	Mesure de la variation de l'entropie . . . . .	20
2.5.5	Remarques et synthèse . . . . .	23
<b>3</b>	<b>Structure virtuelle pour réseaux ad hoc et hybrides</b>	<b>27</b>
3.1	Introduction . . . . .	27
3.2	Quelle structure d'auto-organisation ? . . . . .	27
3.3	Structure virtuelle : construction et maintenance . . . . .	28
3.3.1	Construction du backbone virtuel . . . . .	29
3.3.2	Construction des clusters . . . . .	31
3.3.3	Algorithmes de maintenance . . . . .	32
3.4	Auto-organisation : propriétés et impacts . . . . .	34
3.4.1	Évolution de la cardinalité . . . . .	34
3.4.2	Impact du degré . . . . .	35
3.4.3	Impact de la mobilité . . . . .	36
3.4.4	Passage à l'échelle . . . . .	37
3.4.5	Coût du trafic de contrôle . . . . .	38
3.5	Extensions . . . . .	38

3.5.1	D'un arbre à un treillis . . . . .	39
3.5.2	Mécanisme d'économie d'énergie . . . . .	40
3.5.3	Interconnexion à un réseau hybride et routage par défaut . . . . .	41
3.6	Routage . . . . .	43
3.6.1	Routage local aux clusters . . . . .	44
3.6.2	Routage inter-clusters . . . . .	44
3.6.3	Routage : vue d'ensemble . . . . .	45
3.6.4	Performances du routage basé sur une auto-organisation . . . . .	46
3.7	Synthèse . . . . .	49
<b>4</b>	<b>Auto-organisation de réseaux de capteurs</b>	<b>53</b>
4.1	Motivations, objectifs . . . . .	53
4.2	Auto-organisation pour réseaux de capteurs . . . . .	54
4.2.1	Préambule . . . . .	54
4.2.2	LEGOS : protocole d'auto-organisation pour réseaux de capteurs . . . . .	55
4.2.3	Analyse et complexité . . . . .	59
4.2.4	Évaluation de performances . . . . .	61
4.2.5	LEGOS : d'un arbre à un treillis . . . . .	65
4.3	Support pour l'auto-configuration d'adresses . . . . .	67
4.3.1	Allocation d'adresses à deux niveaux . . . . .	68
4.3.2	Évaluation de performances . . . . .	69
4.4	Gestion de puits mobiles et acheminement de données . . . . .	70
4.4.1	Prise en compte des requêtes . . . . .	71
4.4.2	Acheminement des données . . . . .	72
4.4.3	Stockage des données et points de rendez-vous . . . . .	72
4.4.4	Analyse du coût protocolaire . . . . .	73
4.4.5	Évaluation de performances . . . . .	75
4.5	Agrégation de données . . . . .	76
4.5.1	Modèle ARMA et prédiction de séries temporelles . . . . .	76
4.5.2	Agrégation temporelle locale . . . . .	77
4.5.3	Complexité du calcul local . . . . .	77
4.5.4	Précision du modèle . . . . .	79
4.5.5	Analyse du coût de communication . . . . .	79
4.5.6	Impact de mesures erronées . . . . .	80
4.6	Synthèse . . . . .	81
<b>5</b>	<b>Modélisation et évaluation de performances de couches MAC</b>	<b>85</b>
5.1	Introduction et motivations . . . . .	85
5.2	PEPA, un algèbre de processus stochastique . . . . .	86
5.2.1	Pourquoi utiliser PEPA ? . . . . .	86
5.2.2	Introduction au formalisme . . . . .	87
5.2.3	Graphe de transitions et propriétés . . . . .	88
5.2.4	Le modèle stochastique . . . . .	89
5.2.5	Résolution et calcul de performances . . . . .	90
5.2.6	L'agrégation exacte . . . . .	91
5.3	Modélisation et évaluation de performances de couches MAC . . . . .	92

5.3.1	Méthodologie de modélisation . . . . .	92
5.3.2	Le modèle de base . . . . .	95
5.3.3	Description du modèle de base . . . . .	95
5.3.4	Description du modèle PEPA associé . . . . .	96
5.4	Extension vers d'autres études . . . . .	99
5.4.1	Vers un autre mécanisme d'accès au médium . . . . .	99
5.4.2	Modélisation des variantes de BEB . . . . .	100
5.4.3	Investigation de topologies multi-sauts . . . . .	101
5.4.4	Modélisation de sources de trafic non saturé . . . . .	105
5.4.5	Tenir compte des propriétés du canal radio . . . . .	106
5.5	Évaluation de performances . . . . .	108
5.5.1	Mesures de performances . . . . .	108
5.5.2	Validation du modèle . . . . .	109
5.5.3	Étude des nœuds cachés . . . . .	109
5.5.4	Étude du scénario des trois paires . . . . .	111
5.5.5	Étude des nœuds transitifs . . . . .	112
5.5.6	Influence du médium sur les performances et l'équité . . . . .	114
5.6	Synthèse et conclusions . . . . .	116
<b>6</b>	<b>Conclusion, perspectives</b>	<b>121</b>
6.1	Synthèse sur l'auto-organisation . . . . .	121
6.2	Synthèse sur l'évaluation de performances . . . . .	123
6.3	Perspectives . . . . .	123



# Chapitre 1

## Introduction, motivations, contexte

Classiquement les thèses ainsi que les habilitations à diriger des recherches commencent par un état de l'art, un rappel des travaux similaires, concurrents ou alternatifs. J'épargnerai ici le lecteur en l'invitant à consulter les très nombreuses thèses soutenues ces dernières années [72, 39] ou à se reporter à quelques survols bien connus [36, 27, 6, 62]. Je préfère, au fil de l'eau, revenir sur quelques travaux qui m'ont marqués, influencés ou inspirés. Cela me permettra de me concentrer sur mes contributions principales.

Mais avant de présenter mes travaux, j'aimerais prendre un peu de temps pour expliquer comment j'en suis arrivé à considérer que l'auto-organisation dans les réseaux radio multi-sauts<sup>1</sup> est une clef pour bien des problèmes. Tant qu'à faire un peu d'histoire, je me vois obligé de revenir sur un petit changement de thématique scientifique.

Après des études universitaires classiques à osciller entre les protocoles réseaux et l'évaluation de performances de réseaux, le jour venu de faire une thèse, j'ai eu le plus grand mal à choisir où aller : une thèse en réseau ? une thèse en évaluation de performances ? Finalement, je suis resté entre les deux en ayant une contribution plus importante dans le domaine de l'évaluation de performances<sup>2</sup> que dans le domaine du réseau<sup>3</sup> [79]. En septembre 2000, j'ai rejoint l'INSA Lyon pour participer à un challenge particulièrement motivant : la création du laboratoire CITI. Le projet scientifique du CITI fût un peu plus tard repris par le projet INRIA ARES, localisé à l'INSA Lyon, au sein du laboratoire. Rapidement, les thématiques scientifiques ont eu pour application les réseaux ad hoc en allant de la couche radio au déploiement logiciel en passant par les protocoles de communication : un vrai projet transversal, *cross-layer*. C'est ainsi que j'ai mis entre parenthèses mes activités dans le domaine de l'évaluation de performances pour me concentrer sur le niveau 3 du monde OSI et que j'ai commencé à tisser quelques liens autour de l'auto-organisation de réseaux. J'y reviendrai. Mais, si mon habilitation gravite essentiellement autour de cette idée de réseaux et de protocoles autonomes, j'ai continué à m'intéresser à la modélisation et l'évaluation de performances de réseaux car, finalement, concevoir un protocole

---

<sup>1</sup>Sous le nom de réseaux radio multi-sauts, je sous-entend volontiers les réseaux ad hoc et les réseaux de capteurs. Ces réseaux en commun de minimiser (se passer ?) de l'intervention humaine. Toutefois, les réseaux radio multi-sauts ne sauraient former un cadre restreint d'application de la notion d'auto-organisation : les réseaux maillés sont également un vaste champ d'investigation.

<sup>2</sup>Ma thèse s'intitulait *Modélisation et évaluation de performances de réseaux cellulaires hiérarchiques* et faisait une large part aux outils d'évaluation de performances : chaînes de Markov, bornes stochastiques, algèbres de processus, etc.

<sup>3</sup>Plus précisément sur les algorithmes d'allocations de ressources pour réseaux cellulaires.

est un vrai challenge, mais mesurer ses performances, étudier un comportement asymptotique, comprendre des topologies et des scénarios emblématiques est au moins aussi important. C'est à travers une approche formelle que l'on peut énoncer des règles de dimensionnement ou de choix de protocoles en fonction des propriétés des topologies. Bref, finalement, rien n'a vraiment changé et je n'ai toujours pas tranché : j'oscille toujours entre le monde des protocoles de communications et celui de l'évaluation de performances de réseaux même si, cette fois-ci, les proportions sont inversées par rapport à ma thèse.

Donc, en 2000, quasi-vierge de connaissance sur les réseaux ad hoc, je me suis retrouvé à réfléchir sur les thématiques scientifiques du CITI. Jeune docteur, je me suis remis dans la peau d'un jeune... doctorant pour apprendre, découvrir, comprendre ce qu'il se faisait afin de lancer une activité scientifique au sein du CITI en cohérence avec le projet de recherche du laboratoire. Attaquer les réseaux ad hoc autour des années 2001 est un vrai challenge tant le nombre de publications était déjà conséquent. Oh, bien sûr, je n'ai pas étudié la centaine de protocoles de routage *unicast* ni la bonne dizaine de couches MAC sans parler des dizaines de modèles de mobilité. En revanche, un point m'a étonné dès le début de ma conversion ad hoc. Les réseaux ad hoc s'ils sont un formidable challenge pour les scientifiques n'en demeurent pas moins un objet étonnant : les noeuds ne sont pas coordonnés, ils évoluent de façon désynchronisées, les algorithmes sont distribués, les noeuds sont mobiles et de capacité (a priori) différente, la communication se fait via un lien radio ô combien versatile. Non pas que je fasse preuve d'un scepticisme pragmatique<sup>4</sup> à l'égard de ces réseaux, non, mais je me suis toujours étonné de voir des centaines de protocoles distribués qui se proposaient d'adresser un monde un peu chaotique sans y mettre un peu... d'ordre. J'ai toujours pensé qu'avant de configurer un noeud, qu'avant de router un paquet ou de localiser un service, il fallait *structurer, organiser* le réseau puis basé sur cette organisation, il s'agissait alors de proposer les protocoles de communications nécessaires. Bien entendu comme il n'est pas imaginable que cette organisation soit l'objet d'un décideur unique ni l'objet de quelques noeuds ayant une connaissance totale de la topologie, la notion d'*auto-organisation* s'impose alors. L'essentiel de mon travail de ces dernières années tient dans cette idée. Les sujets de masters et de thèses que j'ai élaboré, les collaborations que j'ai développé, les contrats que j'ai lancé ont toujours tendu vers ce but : étudier les protocoles d'*auto-organisation* puis montrer qu'un réseau basé sur une *auto-organisation* est plus efficace qu'un réseau à plat. On verra que ce n'est pas toujours le cas.

Je me dois de souligner un point de divergence avec une idée assez répandue dans la communauté scientifique, une idée majoritaire même. Qu'est-ce que l'*auto-organisation* ? Il est courant de parler d'*auto-organisation* dès que l'on adresse les réseaux ad hoc ou les réseaux de capteurs. Mais que cache cette notion ? En réalité, lorsqu'on parle d'*auto-organisation*, on fait souvent référence à des mécanismes autonomes s'organisant seuls pour rendre un service. Ainsi, il n'est pas rare de penser *auto-organisation* lorsque l'on étudie les protocoles de routage. La raison en est fort simple : les protocoles de routage, basés sur la découverte du voisinage, s'adaptent aux changements de topologie, tiennent compte de l'apparition / disparition de voisins, etc. Bref, ils s'*auto-organisent*. Dans ce contexte [30] propose une classification des protocoles d'*auto-organisation* : si l'initiative est intéressante, il est à regretter que tout y passe : couches MAC, routage, qualité de service, etc. La vision que je défend de l'*auto-organisation* n'est pas celle-ci. Pour moi, l'*auto-organisation* est à ranger au côté de la notion d'ordre, de la notion de structure. Alors, non, je ne suis pas le premier à vouloir structurer un réseau ad hoc ou de capteurs mais il me plaît à penser

---

<sup>4</sup>Et non pas scientifique : les réseaux ad hoc restent un formidable challenge pour tout chercheur.

que les travaux que j'ai mené ces dernières années ont poussé les idées assez loin dans l'auto-organisation et de ses impacts comme cela a pu être fait dans le cadre des réseaux pair-à-pair [33]. Un certain nombre de travaux me semblant important ont proposé de structurer le réseau, de l'organiser. C'est le cas de ZRP [37] par exemple mais également de travaux intéressants sur l'utilisation de *clusters* pour le routage [44, 11]. Toutefois ce que je reproche à ce type de travaux est *i)* d'être orienté source, c'est-à-dire que chaque nœud possède une vision de l'organisation indépendante et distincte de ses voisins *ii)* de n'avoir pour objectif que le routage de paquets sans s'intéresser à l'ensemble des protocoles de communications. Il me semble important, si l'on organise un réseau, que cette organisation soit un facteur commun de tous les nœuds (ou au moins la plupart) du réseau pour fédérer et structurer de façon efficace : avoir une organisation orientée source, c'est concevoir une structure d'organisation du réseau pour chaque nœud. Où est alors le gain ? Certes on pourra me répondre que considérer une structure commune dans le réseau c'est créer un goulot d'étranglement, c'est sur-solliciter certains nœuds plutôt que d'autres au détriment de l'énergie, par exemple. C'est sans compter sur l'utilisation de mécanismes efficaces comme l'*activity scheduling* [71] qui permet de faire tourner le rôle des nœuds mais aussi sur la notion de *self-healing* qui permet de maintenir l'organisation, qui permet de réagir aux changements de topologie et de masquer la dynamique du réseau pour les protocoles de communication. Je montrerais un peu plus loin comment ce mécanisme peut s'avérer efficace. Enfin, il me semble que l'auto-organisation doit être un socle sur lequel on développe les protocoles de communication et pas uniquement le routage, qui me semble trop restreint. Ainsi, à travers une auto-organisation exhibant de *bonnes* propriétés, des protocoles d'auto-configuration, de routage, d'agrégation de données, etc. peuvent être proposés : ils doivent alors tirer partie de l'auto-organisation fournie. Bref, l'auto-organisation se doit d'être un moyen et non une fin. Ainsi, alors que les réseaux radio multi-sauts affichent une dynamique et une versatilité importante, l'auto-organisation doit apporter une stabilité, une persistance dans le réseau, propriété importante pour envisager des protocoles de communications efficaces.

L'auto-organisation consiste à construire une vue logique du réseau tel que, basés sur cette vue logique les protocoles de communications seront plus efficaces. Ou du moins, l'espère-t'on. Ainsi, pour moi, un protocole d'auto-organisation est un protocole localisé<sup>5</sup> visant à construire une organisation logique, *virtuelle*. Basiquement, nous pouvons distinguer 4 stratégies d'auto-organisation (se reporter à la figure 1.1) : utilisation de tables de hachages distribuées [32], techniques de clusterisation [50, 54, 8], backbone virtuel [8, 18, 21], création d'*overlays* [19]. Finalement, l'objectif est le même : construire un espace logique de telle façon que les protocoles de niveau 3 et au-dessus soient plus efficace que sur le réseau à plat. Cet espace logique peut-être obtenu avec *n'importe quelle* fonction de l'espace physique vers un espace logique. Que ce soit dans le contexte des réseaux ad hoc ou des réseaux de capteurs, la littérature fourmille de publications proposant de construire des DHT, des clusters, des backbones virtuels, etc. Toutefois, le nombre de solutions proposant d'exploiter ces topologies logiques en vue de développer des protocoles de communications sont assez faible. Je proposerais, avec chaque stratégie d'auto-organisation, d'expliquer comment les protocoles de communications (routage dans le cas réseaux ad hoc) et les comportements applicatifs (agrégation de données dans le cas des réseaux de capteurs) peuvent tirer parti de telles structures afin de souligner la pertinence d'une telle approche.

Dans le premier chapitre de cette habilitation (**chapitre 2**), je reviendrai sur quelques notions clefs de l'auto-organisation telles que je les ai décrites dans [76]. L'objectif sera de préciser

<sup>5</sup>Un peu loin je reviendrai sur cette notion de protocole distribué ou localisé.

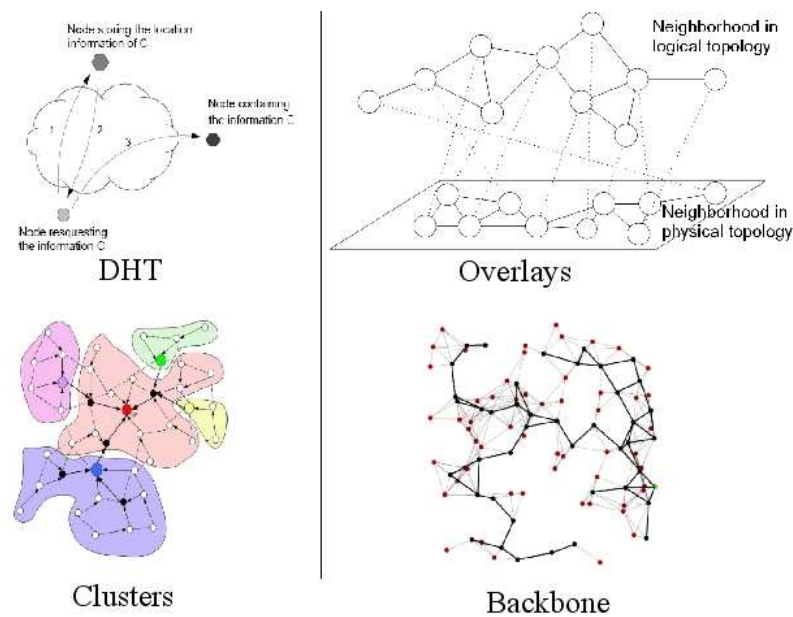


FIG. 1.1 – Auto-organisation : 4 stratégies principales

un peu plus ce que l'on attend d'une auto-organisation, de lister les grandes propriétés et les points clefs à aborder. Ce chapitre sera conclu avec une discussion sur l'évaluation des protocoles d'auto-organisation : bien qu'il est courant de rencontrer des travaux qui évaluent ces protocoles sous l'angle de la complexité, de la cardinalité, du surcoût protocolaire, etc. de toute évidence, il manque une métrique témoignant de l'efficacité d'une organisation. Quelle est cette métrique ? A bien y réfléchir, une organisation doit structurer le réseau, doit introduire un ordre au sein de la topologie. Il est alors logique que l'objectif d'une auto-organisation soit de diminuer le désordre. Ainsi, nous proposerons une mesure de l'entropie pour caractériser les protocoles d'auto-organisation. Dans les **chapitres 3 et 4**, je m'intéresserai à l'auto-organisation et ses impacts dans le cas des réseaux ad hoc puis des réseaux de capteurs. Dans le cas ad hoc (**chapitre 3**), nous regarderons comment une auto-organisation peut tirer partie de l'hétérogénéité des nœuds et de leur mobilité. À l'aide d'une structure logique couplant arbre et clusters, nous regarderons quels impacts a cette organisation sur les protocoles de routage et de localisation. Nous verrons que c'est également une solution peu coûteuse pour introduire facilement un mécanisme d'économie d'énergie dans les réseaux ad hoc. Dans le cas des réseaux de capteurs, nous n'avons que faire de la mobilité et un peu moins de l'hétérogénéité : le point central est la gestion et l'économie d'énergie. Ce point clef autorise à avoir une véritable démarche *cross-layer* à l'opposé des démarches classiques héritées des travaux de l'OSI. Dans le **chapitre 4**, nous nous intéresserons à l'auto-organisation pour les réseaux de capteurs à travers une topologie logique sous la forme d'un arbre : nous verrons que ce mécanisme est couplé à un mécanisme d'allocation d'adresses afin qu'au cours du déploiement les nœuds du réseau se configurent et s'organisent simultanément. Basées sur cette auto-organisation, des solutions applicatives pour la dissémination de données mais aussi l'agrégation de données spatiales et temporelles seront discutées.

Avant de conclure ce manuscrit d'habilitation, je terminerais par un dernier chapitre (**chap. 5**) s'intéressant à la modélisation et à l'évaluation de performances de réseaux sans fil multi-sauts

et d'infrastructure. L'objectif est d'étudier le comportement de différentes stratégies MAC afin de déterminer le *meilleur* compromis équité / performances. Nous nous intéresserons à plusieurs topologies emblématiques de celles que nous pouvons rencontrer dans d'autres études (routage, etc.). Plus précisément, l'objectif est double : certes il s'agit de mesurer des grandeurs quantitatives (débit, etc.) et qualitatives (équité) mais il s'agit aussi de proposer une modélisation générique permettant d'adresser relativement *facilement* plusieurs topologies, plusieurs hypothèses de canal radios, plusieurs algorithmes de gestion de collisions, etc. Je présenterais ici un travail tirant pleinement partie d'un algèbre de processus stochastique (PEPA [38]) en termes de modélisation compositionnelle et de réduction d'espace d'états. Nous verrons que les résultats obtenus ici permettent d'aider à la conception de couches MAC.

Ce document se terminera par une synthèse des travaux faits ces dernières années dans le **chapitre 6**, synthèse qui sera suivie par les pistes de recherches qu'il me semble nécessaire d'adresser dans le cadre des réseaux radio multi-sauts.

*Note : ce document se veut être une synthèse de mes travaux depuis 2000. Certaines activités n'y figurent pas notamment tout ce qui est lié à l'ingénierie de trafic dans les réseaux cellulaires et mes travaux sur les modèles de mobilité pour réseaux ad hoc. Je ne donnerais pas tous les résultats en essayant de me concentrer sur l'essentiel. J'invite alors le lecteur à consulter ma bibliographie pour y puiser des informations complémentaires et supplémentaires.*



## Chapitre 2

# Auto-organisation pour réseaux radio multi-sauts

### 2.1 Vue d'ensemble

Avant d'étudier l'auto-organisation dans les réseaux radio multi-sauts et de mesurer son impact en termes quantitatifs (performances) et qualitatifs (facilité de déploiement de protocoles réseaux), il nous semble important de la définir plus précisément. Pour nous, la notion d'auto-organisation est corrélée à celle de structuration du réseau à l'opposé de la vision plus classique qui consiste à considérer un réseau à *plat*. L'idée maîtresse est qu'une fois structuré *-organisé-* le comportement des protocoles tels que le routage, l'inondation, la localisation, etc. est plus efficace. Nous abordons donc la notion d'auto-organisation sous le point de vue d'un comportement émergent, structurant le réseau à partir d'interactions locales uniquement. Selon notre point de vue l'auto-organisation vise à tirer parti des propriétés des nœuds pour dégager une structure permettant l'organisation du réseau. Cette structure se doit d'être autonome et dynamique tel qu'un changement local ne doit que faiblement impacter la structure globale. Le recours à cette structure auto-organisante doit faciliter le déploiement de protocoles de communication. La notion d'interactions et de décisions locales seront les principes fondamentaux d'une telle auto-organisation. Par conséquent, nous attendons des propriétés d'adaptation dynamique à l'environnement, de robustesse et bien entendu que le concept de localisé entraîne un meilleur passage à l'échelle que le recours à des solutions distribuées et a fortiori centralisées.

L'objectif de ce chapitre est donc de définir de façon plus précise les concepts inhérents à l'auto-organisation. La dernière section (section 2.5) de ce chapitre est consacrée à l'évaluation des stratégies d'auto-organisation. La question que nous abordons est : finalement quelles sont les différentes possibilités pour évaluer les auto-organisations ? Après une discussion qui passe en revue les différentes opportunités, nous introduisons la notion d'entropie pour caractériser le comportement intrinsèque de ces stratégies d'auto-organisations.

## 2.2 Définition et objectifs de l'auto-organisation

### 2.2.1 Définition

Si l'on s'intéresse aux définitions classiques de l'organisation<sup>1</sup> un système est dit organisé s'il possède une structure et un ensemble de fonctions associés. La structure a pour objet de structurer toutes les entités et faciliter leurs interactions. Les fonctionnalités ont pour but de maintenir la structure et de favoriser son usage pour répondre à des besoins déterminés. La notion d'auto-organisation fait référence à l'organisation d'un système sans interaction avec une entité extérieure et sans contrôle centralisé. Ainsi, l'auto-organisation doit nécessairement être basée sur des interactions locales, de façon complètement distribuée.

C'est la définition que nous adopterons. Notons qu'elle diverge du point de vue assez commun rencontré : un réseau auto-organisé est souvent perçu comme un réseau où les nœuds s'adaptent au changement de topologie de façon localisée ou distribuée comme un mécanisme de découverte de routes. Il n'y a pas là d'idée de structure, d'organisation du réseau.

### 2.2.2 Principes et objectifs

Néanmoins, on attend d'un système auto-organisé plus que de simples interactions locales sans contrôle extérieur. En particulier nous souhaitons, ici, faire émerger un comportement global à partir des différentes interactions locales. Typiquement et dans le cas des réseaux radio multi-sauts, à partir d'échanges d'informations localisées nous souhaitons faire émerger une structure virtuelle permettant d'organiser le réseau. Plusieurs types de structures virtuelles peuvent nous intéresser : les *clusters*, les *backbones* (réseaux maillés, arbres) ou encore les DHT (voir figure 2.1). Plus généralement, toute fonction  $f$  qui permet de passer d'un espace physique à un espace logique peut s'avérer pertinente. Ainsi, nous arrivons à construire deux vues du réseau : une vue *microscopique*, locale, représentant la dynamique des nœuds et une autre, *globale*, possédant des propriétés que nous détaillerons par la suite. Bien évidemment, nous devons nous assurer que la vue *globale* est construite en temps borné et de façon cohérente.

Bien entendu, cette structure émergente doit s'adapter à l'environnement et réagir aux changements locaux. Plus précisément, un changement local ne doit entraîner qu'une modification locale de la structure et ne pas impacter sur son intégralité. Cette réaction dynamique aux changements locaux en un temps *suffisamment* court et borné entraîne la propriété d'adaptabilité. Le système apparaît alors comme robuste : la structure émergente (vue macroscopique) est stable car le mécanisme d'auto-organisation va permettre de réagir aux cas de défaut d'un lien ou d'un nœud du réseau (dus à la mobilité, au manque d'énergie, ...) : la structure va s'adapter et se reconstruire localement. Comme il n'y a pas d'entité centrale, dans un système auto-organisé il n'y a pas de nœuds critiques et le système peut être réparé sans aide extérieure. Nous regroupons les propriétés d'adaptabilité et de robustesse sous le terme d'auto-stabilisation [65] où les fautes sont associées aux liens et aux nœuds.

Si nous considérons l'auto-organisation sous l'angle de la construction d'une topologie stable fédérant le réseau radio multi-saut, cette topologie émergeant des interactions locales doit être partagée par l'ensemble des nœuds. Autrement dit, nous devons bannir toute construction orientée source comme, par exemple, l'utilisation des MPR (*Multi-point relay*) d'OLSR [28] et proposer une structure d'organisation commune à l'ensemble des nœuds du réseau. Comme cette structure

---

<sup>1</sup>e.g. <http://fr.wikipedia.org/>

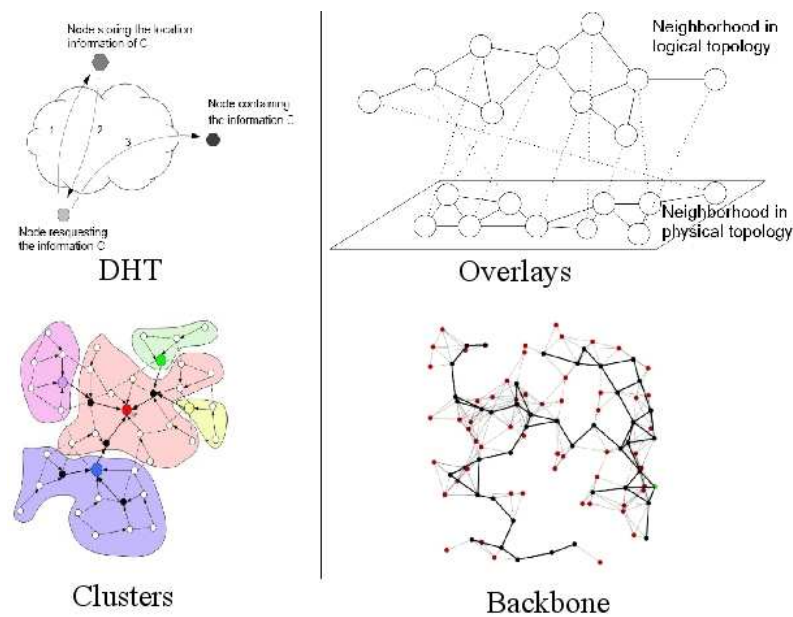


FIG. 2.1 – Différentes stratégies de structures d'auto-organisation

logique sera associée à des nœuds physiques, ces nœuds plus importants seront davantage sollicités pour mettre en place les services réseaux : dans une telle auto-organisation, il est nécessaire de veiller à ce que les nœuds participent à tour de rôle.

Finalement, la dernière propriété importante que doit posséder une structure auto-organisée c'est celle de supporter le passage à l'échelle : même avec un nombre de nœuds très important, le système doit continuer à fonctionner efficacement. Cette propriété découle de l'absence de contrôle centralisé et des interactions locales. Aucune congestion due à un nombre important de nœuds dans le réseau ne doit se produire. Les propriétés que doivent posséder un schéma d'auto-organisation sont donc :

- interactions locales uniquement,
- émergence d'une structure globale à partir d'informations locales,
- réactivité aux changements locaux et robustesse,
- structure d'auto-organisation non orientée source,
- passage à l'échelle.

La notion d'auto-organisation prend tout son sens dans le cadre des réseaux ad hoc et de capteurs. Elle offre un cadre pour faciliter la configuration des nœuds et la mise en place de protocoles de communications. Dans un réseau auto-organisé, l'intervention humaine est réduite au minimum ce qui facilite son déploiement.

L'auto-organisation doit donc faire émerger un comportement global. On peut considérer qu'un réseau radio multi-sauts est, par essence, un système auto-organisé car tous les protocoles utilisés sont basés sur des interactions locales entre les nœuds et sont répartis sur l'ensemble des membres du réseau. De notre point de vue, il ne s'agit pas de se limiter uniquement à la configuration d'adresses ou au routage mais bien de fournir une structure *virtuelle* dynamique et unique permettant de faciliter le déploiement de tout protocole réseau.

### 2.2.3 Décisions locales versus distribuées ?

La notion *locale* que nous exprimons fait référence au voisinage radio d'un nœud du réseau et à la vision partielle, locale, qu'il a du réseau. Dans une structure auto-organisée, il est préférable que les décisions soient prises via un processus purement local basé sur des informations et des interactions locales et non pas de façon distribuée. Rappelons la différence :

- un processus est localisé si chaque nœud prend une décision en fonction des informations (locales) qu'il possède et donc de la vue (partielle) du réseau qu'il a.
- d'un point de vue distribué, plusieurs nœuds peuvent interagir et converger vers une décision. La décision prise n'est donc plus issue des seules informations locales mais bien d'informations distribuées.

Choisir une stratégie locale c'est faire le choix de minimiser les informations connues sur le réseau, d'éviter un coût protocolaire pour synchroniser des décisions, de réagir plus rapidement aux changements locaux.

Après cette définition de l'auto-organisation et des propriétés associées, nous allons maintenant regarder quelles sont les règles permettant de concevoir un système auto-organisé.

## 2.3 Quelques clefs pour l'auto-organisation

[64] étudie les principes clefs de l'auto-organisation. Nous en reprenons ici une partie dans le but de concevoir une structure d'auto-organisation. Il est nécessaire de définir quelles sont les règles et les protocoles nécessaires pour tirer parti de l'interaction entre les nœuds du réseau. Quatre paradigmes vont être discutés :

- émergence d'un comportement global à partir de règles locales,
- interactions locales et coordination des nœuds,
- minimiser les informations sur l'état du réseau,
- s'adapter dynamiquement à l'environnement.

### 2.3.1 Émergence d'un comportement global à partir de règles locales

On s'intéresse à la conception d'un protocole réseau fournissant des propriétés globales comme la connectivité, la robustesse, etc. L'approche centralisée va proposer l'utilisation d'un nœud dédié à ce protocole ou ce service. Dans le cas de l'auto-organisation, cette fonction doit être partagée par l'ensemble des nœuds du réseau : aucune entité ne peut établir la fonction désirée mais chaque nœud doit contribuer à l'établissement d'un comportement global. Il est donc nécessaire de concevoir des règles locales qui tendent vers ce comportement global. Les protocoles de routage dans les réseaux radio multi-sauts illustrent ce paradigme : le comportement global c'est l'acheminement depuis une source vers une destination d'un paquet de données alors que la vision locale se contente de transférer le paquet à un voisin.

Quels sont les comportements locaux pouvant conduire à un tel comportement global ? Localement chaque nœud va collecter des informations qui seront agrégées et diffusées au voisinage. Ainsi, la vue globale n'est pas nécessaire et la connaissance locale permettra de faire émerger un comportement global. C'est effectivement ce qui se passe dans le cas des protocoles de routage à vecteur de distance comme DSDV [61].

Ainsi, la construction d'une vue locale et la mise en place d'interactions locales sont les deux premiers points clefs de l'auto-organisation. Notons que lorsque la vue globale émerge, elle

n'est pas nécessairement exacte mais tend à s'en approcher. De notre point de vue, ne pas avoir de solution optimale dans un contexte très dynamique n'est pas un inconvénient majeur : la topologie du réseau changeant fréquemment, la solution optimale évolue également et présente donc un coût important.

Baser les services et protocoles globaux uniquement sur des informations et des interactions locales conduit ces protocoles à être beaucoup plus robustes et plus stables car il n'y a pas de nœud(s) critique(s) et le réseau ne change pas d'état brutalement, les changements d'états locaux n'ayant qu'une portée locale. Néanmoins, le fait que chaque nœud ne possède qu'une vue locale du réseau peut entraîner des incohérences sur la perception du voisinage, objet du second paradigme.

### 2.3.2 Interactions locales et coordination des nœuds

Le résultat de décisions locales peut conduire à des situations d'incohérence. Dans le cas de l'auto-configuration d'adresses, l'incohérence entraîne des situations de conflits : deux nœuds peuvent posséder la même adresse après des décisions locales. Pour éviter ce phénomène, la première solution est d'utiliser une coordination explicite. Néanmoins, une telle solution est coûteuse d'un point de vue protocolaire car nécessite la mise en place d'un mécanisme de détection d'adresses par exemple. Ce type de coordination répond bien aux systèmes centralisés.

Dans le cas d'un réseau radio multi-sauts, l'utilisation d'une telle coordination explicite ne peut être appliquée à cause des ressources radio limitées et de la dynamique du réseau : les informations devant être mises à jour continuellement et les ressources radio étant limitées. Il faudra donc ici tolérer les conflits. Ils seront acceptables s'ils sont locaux, temporaires, facilement détectables et solubles. La notion de coordination implicite doit aussi être utilisée. Il s'agit d'exploiter toutes les informations circulant dans le voisinage local afin de détecter et de mettre à jour des incohérences. La détection de conflits permet également d'affiner, localement, la vue du réseau et donc d'améliorer les situations d'incohérences.

De toute évidence, ce second paradigme est lié au premier. La mise en place de règles locales va permettre de faire émerger un comportement global. Basé sur ces interactions locales, un service global sera mis en place au risque de présenter des incohérences. A travers ces interactions, des informations de voisinage seront échangées permettant de mettre à jour les informations de topologie ou d'adresses, résolvant les vues partielles mais erronées du réseau et du service associé.

### 2.3.3 Minimiser les informations sur l'état du réseau

Dans le cas général des réseaux, chaque nœud maintient une liste d'informations sur l'état du réseau comme : passerelles d'accès, serveurs (DNS, sécurité, etc.), etc. Ces informations connues par tous les nœuds doivent rester cohérentes et on doit donc utiliser un mécanisme de synchronisation. Autant dire que plus les informations d'états du réseau sont importantes et moins il sera facile de recourir au concept de l'auto-organisation (cf. point précédent).

Dans le cas des réseaux radio multi-sauts, il est nécessaire d'avoir recours à des mécanismes adaptatifs de maintiens et de découvertes d'informations sur l'état du réseau. [64] propose, comme dans le cas des protocoles de routage, deux approches : une réactive et une proactive. Dans le premier cas, un nœud va faire une demande explicite relayé en multicast ou en diffusion tandis que dans le second cas des annonces périodiques sont faites pour diffuser ces informations. Une fois de plus, le mécanisme d'auto-organisation rend le réseau plus réactif et plus robuste aux change-

ments de configuration et aux situations critiques. La dépendance à une information centrale est également évitée.

### 2.3.4 Adaptation dynamique à l'environnement

La dernière règle qui doit être établie dans les réseaux auto-organisés c'est la capacité à réagir aux changements de topologie du réseau et aux changements d'informations, résultant de la mobilité et/ou de la défaillance d'un nœud. Comme il n'y a pas d'entité centralisée qui peut diffuser un changement d'état, chaque nœud doit surveiller son voisinage local et réagir en conséquence. Surveiller son voisinage local est lié au paradigme 1 tandis que réagir à de nouvelles informations est lié au paradigme 2.

[64] identifie 3 niveaux d'adaptations :

- **1er niveau** : Adaptation aux changements de topologie résultant de la mobilité ou de l'apparition/disparition d'un nœud.
- **2ème niveau** : Adaptation des paramètres protocolaires (temporisateurs, connaissance du k voisinage, etc.) en fonction des changements d'états dans le but d'améliorer le comportement du réseau.
- **3ème niveau** : Un système auto-organisé doit être capable d'évaluer que l'environnement dans lequel il évolue ne peut pas conduire à un système stable et donc, remplacer un mécanisme par un autre.

Ces adaptations peuvent être combinées avec un contrôle adaptatif. Il faut néanmoins pondérer les mécanismes d'adaptation : un changement d'état local ne doit entraîner qu'un changement local et non pas impacter toute la topologie du réseau.

## 2.4 Synthèse des propriétés clefs

Dans la première partie de ce chapitre, nous avons défini l'auto-organisation telle que nous la considérons puis nous avons mis en avant les propriétés clefs de l'auto-organisation. Finalement l'auto-organisation doit structurer le réseau, pour cela nous pouvons nous intéresser à toute fonction  $f$  qui fait correspondre une topologie logique à la topologie physique. S'intéresser à l'auto-organisation c'est donc étudier les topologies dynamiques que l'on peut construire pour structurer le réseau. L'objectif étant bien entendu que cette structuration permette le développement plus efficace des protocoles réseaux. Les solutions d'auto-organisation ne doivent pas avoir recours à un système centralisé de coordination. Les points clefs qu'il est important de conserver à l'esprit lors de la proposition d'architectures auto-organisées sont :

- interactions locales uniquement,
- émergence d'une structure globale à partir d'informations locales,
- réactivité aux changements locaux et robustesse,
- structure d'auto-organisation non orientée source,
- passage à l'échelle.

Avant de nous intéresser à *comment* organiser un réseau radio multi-sauts, qu'il soit de type ad hoc ou de type capteurs, pour mieux comprendre *pourquoi* cela est nécessaire, nous allons nous attarder sur l'évaluation des protocoles d'auto-organisation. L'objectif est d'introduire une métrique qui reflète le comportement d'une auto-organisation, à savoir organiser.

## 2.5 Entropie, auto-organisation et réseaux radio multi-sauts

Après avoir défini la notion d'auto-organisation, une famille de questions demeure : comment évaluer une stratégie d'auto-organisation ? Quels sont les moyens et outils que nous avons à notre disposition ? Reflètent-ils tous le comportement *intrinsèque* d'une auto-organisation, à savoir **organiser, structurer** ? Comment montrer qu'une approche basée sur un réseau auto-organisé est nécessairement meilleure qu'une approche considérant un réseau à *plat* ? Tel est le but de cette section.

En réalité, nous avons au moins trois voies pour évaluer une auto-organisation :

- d'un point de vue algorithmique distribué : finalement les protocoles d'auto-organisation sont avant tout le résultat de l'exécution d'algorithmes localisés ou distribués. Il est donc possible de les caractériser sous l'angle de la complexité (en messages, en temps, en mémoire, etc.)
- d'un point de vue de l'auto-stabilisation [65]. L'auto-stabilisation s'intéresse à la convergence (en temps fini) d'algorithmes distribués et permet de montrer qu'en présence de fautes, les algorithmes convergent vers un objectif identifié. L'auto-stabilisation est à rapprocher de la robustesse. C'est un élément clef : montrer qu'un protocole d'auto-organisation finit bien par construire une topologie logique en temps borné et qu'il résiste aux imperfections des transmissions radio est un atout majeur.
- d'un point de vue *réseau* et protocoles de communications. Le passage entre les algorithmes distribués nécessaires à l'auto-organisation et leur mise en application nécessite l'élaboration de protocoles d'échanges de données pour procéder à des élections locales, pour maintenir un voisinage, diffuser un changement d'état, etc. Ainsi, il nous est possible de caractériser une stratégie d'auto-organisation par son coût protocolaire, par le surcoût de contrôle généré, etc.

A ces trois voies de caractérisation, il est important de rajouter la capacité en termes de flots induite par une auto-organisation : la restriction à un sous-ensemble de nœuds et de liens radios peut introduire un goulot d'étranglement au niveau du routage, par exemple. L'évaluation de la capacité induite par une auto-organisation est donc une mesure clef.

Si l'ensemble de ces métriques s'avèrent pertinentes, aucune d'elles ne reflètent le comportement intrinsèque d'une auto-organisation, *i.e.* structurer, organiser, ordonner le réseau. Est-ce qu'une *bonne* auto-organisation n'est-elle pas une auto-organisation qui minimise le désordre dans le réseau ? Dans le même temps, il nous faut une métrique pour montrer (ou pas...) qu'il est préférable d'organiser le réseau plutôt que de le considérer comme un réseau à plat.

### 2.5.1 De la thermodynamique aux réseaux radios multi-sauts

Comme nous l'avons précisé dans la première partie de ce chapitre, une auto-organisation est un comportement émergent basé sur des interactions locales : c'est l'expression macroscopique d'interactions microscopiques. Les interactions microscopiques sont décrites par les liens logiques entre les nœuds : un nœud physiquement voisin d'un autre peut communiquer avec lui suivant l'auto-organisation considérée (existence d'un lien logique) ou pas (non existence d'un lien logique). Au niveau macroscopique, ces interactions forment la topologie logique du réseau, *i.e.* l'auto-organisation considérée.

Nous pouvons faire ici un parallèle avec l'entropie statistique utilisée en thermodynamique : l'équilibre entre deux gaz parfaits (comportement macroscopique) est le résultat d'interactions

moléculaires (comportement microscopique). Plus l'entropie est faible et plus l'équilibre est stable et moins il y a de *désordre* observable au niveau moléculaire. C'est Boltzmann qui a défini l'entropie statistique [15] à partir de la combinaison de micro-états comme étant égale à :  $S = -k_B \sum_i p_i \ln p_i$ , où  $p_i$  est la probabilité que le micro-état  $i$  se produise pendant les variations du système et où  $k_B$  est la constante de Boltzmann. Cette notion peut s'appliquer pour caractériser l'ordre d'un système et comment le système s'auto-organise en tenant compte de ses différentes entités et de leurs interactions.

Nous proposons d'adapter cette définition de l'entropie au cas particulier de l'auto-organisation de réseaux radio multi-sauts. En construisant une analogie sur la dynamique des interactions locales et l'émergence d'un comportement global, une mesure de l'entropie adaptée à notre contexte de travail peut être proposée. Cette mesure permettra de caractériser les *bonnes* auto-organisations (faible entropie) des autres. Notons que plusieurs propositions d'entropie ont déjà été proposées [60, 52] mais elles ne s'intéressent pas à la caractérisation du principe d'auto-organisation.

### 2.5.2 Entropie : application à l'auto-organisation

L'entropie apparaît donc comme une évaluation d'un comportement macroscopique qui se base sur des interactions locales. Dans un réseau radio multi-sauts, l'objectif des protocoles d'auto-organisation est de simplifier la topologie physique en ne retenant, dans la topologie logique, qu'un sous-ensemble des liens et/ou un sous-ensemble des nœuds. Bref, une stratégie d'auto-organisation va restreindre ces interactions locales. Dans le but de quantifier l'impact d'une stratégie d'auto-organisation sur l'organisation interne du réseau, nous allons appliquer la notion d'entropie sur les liens entre les nœuds : là où l'entropie en thermodynamique considère  $p_i$  la probabilité que le micro-état  $i$  se produise, nous allons associer la notion de micro-état à celle de lien radio.

Nous supposons qu'un lien radio entre deux nœuds  $u$  et  $v$  existe avec la probabilité  $p(u, v)$  dépendant de l'état du lien radio, de l'état des deux nœuds et de la sélection par une auto-organisation de ce lien. L'entropie associée à ce lien est donné par :  $E = -p(u, v) \log p(u, v)$ . Une forte valeur de  $E$  témoigne d'une entropie élevée et donc d'un *désordre* important alors qu'une faible valeur de  $E$  est associée à un ordre plus important et donc, une meilleure organisation. Pour des valeurs de  $p(u, v)$  proches de 0, l'entropie est également proche de 0 : cela indique que le lien est bien organisé même si la probabilité que lien soit finalement présent est faible. Pour des valeurs de  $p(u, v)$  proche de 1, l'entropie est également proche de 0 ce qui indique une bonne organisation : le lien est très stable et donc peut être interprété comme bien organisé. Les valeurs intermédiaires témoignent d'un désordre plus important.

Nous supposons que la topologie d'un réseau radio multi-sauts est modélisable par un graphe géométrique aléatoire (noté  $G(X, r)$  où  $X$  est l'ensemble des nœuds dans le graphe et  $r$  la portée radio). En sommant sur chacun des liens du réseau, on obtient l'entropie globale du réseau :

$$E = \sum_{u,v \in X} -p(u, v) \log p(u, v). \quad (2.1)$$

Appliquons cette définition à un réseau  $G(X, r)$  et voyons comment calculer  $p(u, v)$ . Dans ce but, nous observons que la sélection d'un lien radio entre deux nœuds dans l'auto-organisation est le résultat de :

- deux comportements aléatoires indépendants : l'existence des deux nœuds à chaque bout du lien radio (à cause de mécanisme d'endormissement, etc.) et la disponibilité du lien radio ;
- un comportement déterministe : l'exécution du protocole d'auto-organisation sur la topologie qui conduit à une réalisation précise.

Nous supposons qu'un nœud est actif avec une probabilité  $p$  (et donc inactif (mort, etc.) avec une probabilité  $q = 1 - p$ ). De plus, nous supposons que le médium radio est disponible pour une bonne transmission avec une probabilité  $c$ .

Nous allons appliquer ce calcul de l'entropie à un réseau physique composé de 4 nœuds fortement connexes. Nous appliquerons le calcul de l'entropie dans le cas d'un réseau non organisé (figure 2.2) et dans le cas d'un réseau organisé suivant l'algorithme de LMST (*Local Minimum Spanning Tree* [53]) (figure 2.3).

**Application à un réseau à plat** D'après la figure 2.2, la probabilité que le lien (1) existe est  $p_1 = c \cdot (p^4 + 2qp^3 + q^2p^2) = c \cdot p^2$ . Quelque soit le lien  $i$ , la probabilité que le lien existe est égale à  $p_1$  (de façon plus générale, dans un *unit disk graph*, la probabilité d'existence d'un lien est la même quelque soit le lien). L'entropie d'un lien est donc :  $-p_1 \log p_1$ . Ce qui nous conduit à l'entropie globale du réseau égale à :  $E = -\sum_{i=1}^6 p_i \log p_i = -6 \cdot p_1 \log p_1$ .

Supposons maintenant que dans un réseau à plat, il existe  $m$  liens dans une topologie de  $N$  nœuds. La valeur de  $m$  dépend de la portée radio et de la distribution des nœuds. Pour une densité donnée, notée  $\rho$ ,  $m$  est défini par  $m = \rho \cdot N/2$ . L'entropie globale étant la somme des entropies locales à chacun des liens, l'entropie d'un réseau à plat est donné par :  $E = -m \cdot cp^2 \cdot \log(cp^2)$ .

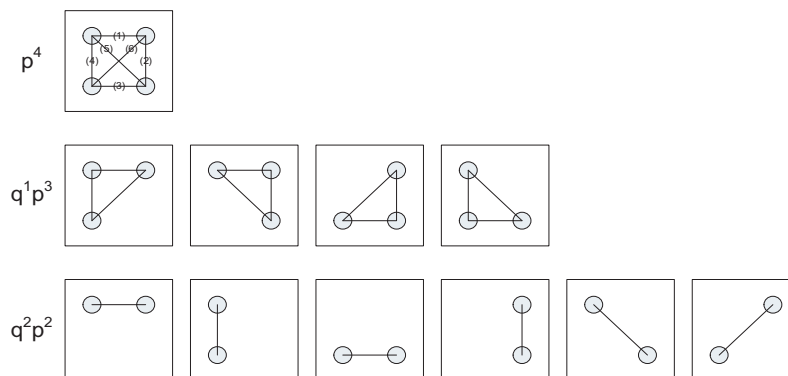


FIG. 2.2 – Réseau à plat

**Application à un réseau auto-organisé suivant LMST** L'algorithme LMST [53] propose de calculer un MST de façon purement localisé. Le LMST peut être perçu comme une stratégie d'auto-organisation basée sur la simplification de la topologie radio : l'ensemble des nœuds de la topologie physique est conservé mais seul un sous-ensemble des liens est maintenu. Plus exactement, chaque nœud  $u$  calcule un MST sur son graphe de voisinage à deux sauts. Si le nœud  $u$  est dans le MST de  $v$  et si le nœud  $v$  est dans le MST de  $u$  alors le lien  $(u, v)$  sera dans le LMST global.

La figure 2.3 illustre les différents LMST possible en partant de la topologie à 4 nœuds fortement connexe. D'après cette figure, la probabilité que chacun des liens (1), (2), (3), (4) soit dans la topologie d'auto-organisation est  $p_1 = c \cdot (p^4 + 2qp^3 + q^2p^2)$ . La probabilité que chacun les

liens (5), (6) soit présent est  $p_5 = c \cdot q^2 p^2$ . L'entropie globale du réseau auto-organisé est alors :  $E = - [4 \cdot p_1 \log p_1 + 2 \cdot p_5 \log p_5]$ .

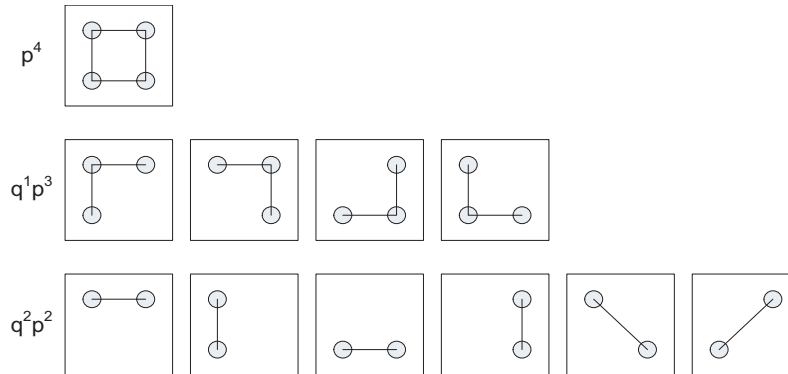


FIG. 2.3 – Réseau auto-organisé suivant LMST

De façon plus générale, et dans le cas de topologies aléatoires, le calcul des probabilités  $p_i$  n'est pas envisageable : de toute évidence le temps de calcul est exponentiel. Comment calculer ces probabilités ? Nous avons eu recours à d'intensives campagnes de simulation. Pour obtenir le résultat numérique de l'entropie, nous avons généré un réseau  $G(X, r)$  composé de  $N$  nœuds où la probabilité qu'un nœud soit actif est  $p$ . Sur cette topologie et sur chacun des liens, nous avons appliqué la probabilité  $c$  que le lien soit actif. Sur le graphe résultant, nous avons appliqué le protocole d'auto-organisation  $\Omega$  : nous avons obtenu une topologie finale notée  $G^\Omega(X, r, p, c)$  où les liens radio sélectionnés dans l'organisation sont connus. Pour calculer la probabilité qu'un lien soit effectivement sélectionné dans l'organisation, nous avons recommencé les mêmes étapes sur la même topologie jusqu'à obtenir une mesure de probabilité stationnaire pour chacun des liens. Puis nous avons recommencé pour obtenir des moyennes significatives sur plusieurs topologies.

### 2.5.3 Mesure de l'entropie

Nous avons comparé l'entropie d'un réseau à *plat* avec trois stratégies d'auto-organisation : deux stratégies orientées simplification de la topologie radio (LMST [53], RNG [77, 42]) et une stratégie orientée élection de dominants (CDS localisé [80]). Dans ce dernier cas, les règles suivantes sont appliquées pour la sélection des liens : *i*) les liens entre un dominant et ses dominés sont toujours conservés actifs *ii*) un seul dominé-dominant est conservé.

**Topologie exemple** Les courbes 2.4(a) et 2.4(b) indiquent le résultat analytique de l'entropie dans le cas du réseau composé de 4 nœuds fortement connexes pris en exemple précédemment. Dans un premier temps, nous avons fixé la probabilité que le lien radio existe ( $c = 1$ ) et nous faisons varier la probabilité qu'un nœud soit actif ( $p$ ). Dans un deuxième temps, nous avons fixé  $p$  et nous faisons varier  $c$ . L'allure des courbes obtenues est typique d'une mesure de l'entropie avec une entropie tendant vers 0 lorsque la probabilité ( $p$  ou  $c$ ) tend soit vers 0 soit vers 1 : l'entropie y est alors la plus faible et donc, l'organisation la plus forte. Notons également, sur cette topologie particulière, que l'entropie d'un réseau organisé suivant LMST est plus faible que si nous considérons le réseau à plat. Pour des probabilités proches de 0 et de 1, l'entropie de l'auto-organisation est proche de celle du réseau à plat.

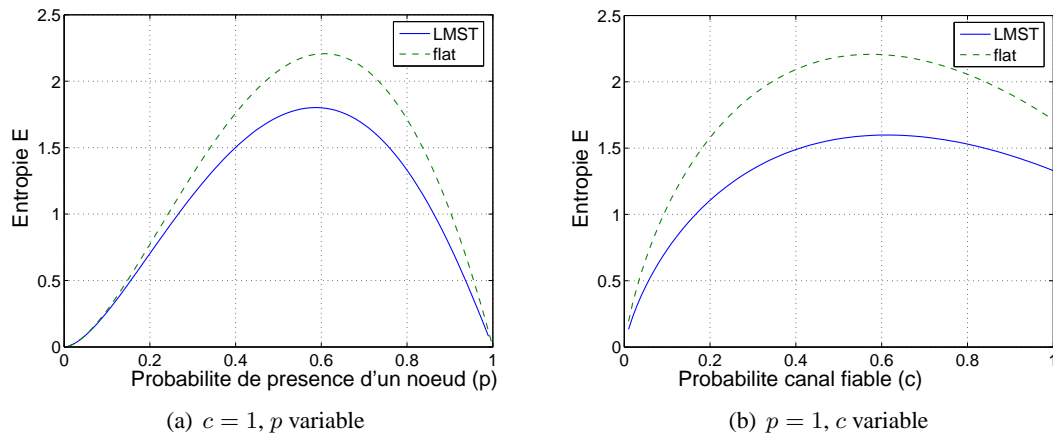
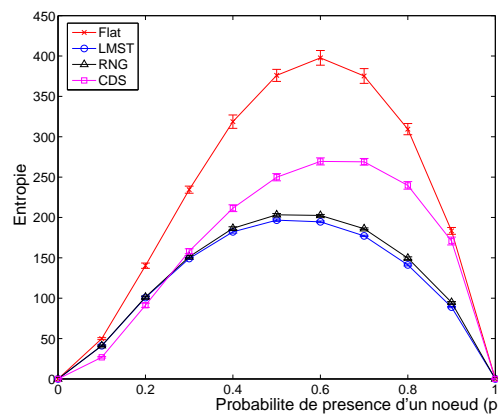


FIG. 2.4 – Entropie sur graphe fortement connexe à 4 nœuds : LMST vs réseau à plat

FIG. 2.5 – Entropie en fonction de  $p$  ( $c = 1, r = 0.14$ ) - Topologie aléatoire de 200 nœuds

**Topologie aléatoire** La figure 2.5 nous fournit l'entropie d'un réseau à plat et des auto-organisations LMST, RNG, CDS localisé pour des topologies aléatoires composées de 200 nœuds déployés sur carré  $1 \times 1$  et un rayon de transmission  $r = 0.14$ . Clairement, l'utilisation d'une auto-organisation fait décroître significativement l'entropie dans le réseau par rapport au réseau considéré comme à plat. C'est bien entendu, l'objectif de l'auto-organisation : introduire de l'ordre, structurer le réseau. Il est intéressant de noter que les stratégies orientées simplification de la topologie radio (LMST, RNG) ont un comportement très proche et introduisent un ordre plus important que l'utilisation d'un CDS localisé : cela est probablement due à la simplification de la topologie qui garde moins d'interactions avec le voisinage.

La figure 2.6 donne l'évolution de l'entropie en fonction d'une augmentation du degré : les probabilités  $p$  et  $c$  sont fixées, la cardinalité du réseau aussi, seule la portée radio ( $r$ ) varie. L'entropie d'un réseau à plat croît de façon importante tandis que l'entropie des auto-organisations LMST et RNG semblent insensible au degré :

- dans un réseau à plat, l'augmentation du degré se traduit par une augmentation du nombre de voisins. Cela signifie que les interactions locales augmentent et donc, que l'entropie

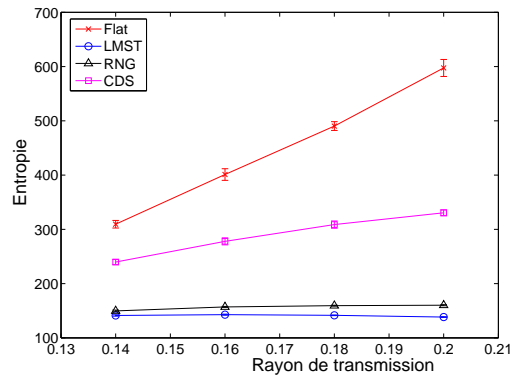


FIG. 2.6 – Entropie en fonction de  $r$  ( $c = 1, p = 0.8$ ) - Topologie aléatoire de 200 nœuds

globale augmente aussi.

- dans une auto-organisation, les interactions locales sont le résultat d'un algorithme local, exécuté sur chaque nœud et qui va simplifier le voisinage en sélectionnant certains liens et en supprimant d'autres. L'objectif est donc de maintenir moins d'interactions. Nous retrouvons également deux résultats connus de LMST et RNG : le degré de ces structures est faible (donc les interactions locales également) et relativement insensible à la densité du voisinage.

#### 2.5.4 Mesure de la variation de l'entropie

L'entropie calculée précédemment considérait des topologies fixées et stables. En réalité les topologies des réseaux radio multi-sauts sont dynamiques : de nouveaux nœuds peuvent apparaître, des nœuds peuvent disparaître faute d'énergie, des nœuds peuvent se réveiller et s'endormir de façon sporadique, des nœuds peuvent être mobiles. Ainsi, plus que l'entropie, c'est la variation d'entropie résultant d'un changement de topologie qui nous intéresse. Que représente cette variation d'entropie ? C'est la capacité de l'auto-organisation à retrouver un état d'équilibre, *i.e.* à fournir une nouvelle organisation respectant les nouvelles contraintes topologiques. Cette variation d'entropie témoigne de la capacité d'adaptation aux changements d'états. Nous pensons qu'une *bonne* auto-organisation doit posséder une faible entropie (pour montrer sa capacité à bien organiser le réseau) et doit être peu sensible aux changements topologiques (*i.e.* faible variation de l'entropie).

Ainsi, la variation d'entropie est définie par la différence d'entropie entre avant et après un (ou plusieurs) changement(s) d'états :

$$\Delta E = E_{change-} - E_{change+}. \quad (2.2)$$

Nous allons nous intéresser à la variation d'entropie dans deux cas précis : *i)* départ d'un seul nœud puis *ii)* départ de plusieurs nœuds. Notons que le départ et l'arrivée d'un nœud sont symétriques : étudier le départ d'un nœud est équivalent à étudier l'arrivée d'un nouveau nœud, les variations d'entropie sont égales en valeur absolue.

**Départ d'un seul nœud** Nous considérons ici une topologie composée de 200 nœuds déployés aléatoirement sur un carré  $1 \times 1$  et avec une portée radio de 0.16. À la date  $T_0$  tous les nœuds sont

actifs dans le réseau tandis qu'à la date  $T_1$  ( $T_1 > T_0$ ), nous introduisons un changement d'état lié au départ d'un nœud. Plus précisément, nous faisons 200 simulations pour mesurer l'impact du départ de chacun des nœuds. Une fois de plus, nous considérons le réseau non structuré puis soumis à trois auto-organisations : LMST, RNG et CDS localisé.

Les figures 2.7(a)-2.7(d) fournissent la PDF (*Probability Density Function*) de la variation de l'entropie pour le départ d'un seul nœud. Dans un réseau à *plat* (figure 2.7(a)), la variation de l'entropie est toujours positive, distribuée entre 1 et 9 : certains nœuds ont un impact plus important que d'autres suivant leur degré dans la topologie physique (et donc suivant les interactions locales). Une variation positive de l'entropie signifie que l'entropie décroît : ce résultat était attendu puisque l'entropie est corrélée aux interactions locales : supprimer un nœud, c'est diminuer ces interactions. C'est le CDS localisé qui a la plus grande plage de variation d'entropie parmi tous les protocoles d'auto-organisation considérés. L'importance du rôle des nœuds dans un CDS (dominant ou dominé) accentue l'impact de la disparition d'un nœud puisque les règles du CDS doivent être respectées. Notons également que la disparition d'un nœud dans un CDS peut augmenter l'entropie et donc, introduire du désordre.

D'après les figures 2.7(c) et 2.7(d), les stratégies LMST et RNG ont une variation d'entropie assez faible et resserré : cela montre que ces stratégies s'adaptent bien au changement d'état du au départ d'un nœud. Leur capacité d'adaptation est donc supérieure à celle d'un réseau à *plat* et d'un réseau organisé suivant un CDS localisé.

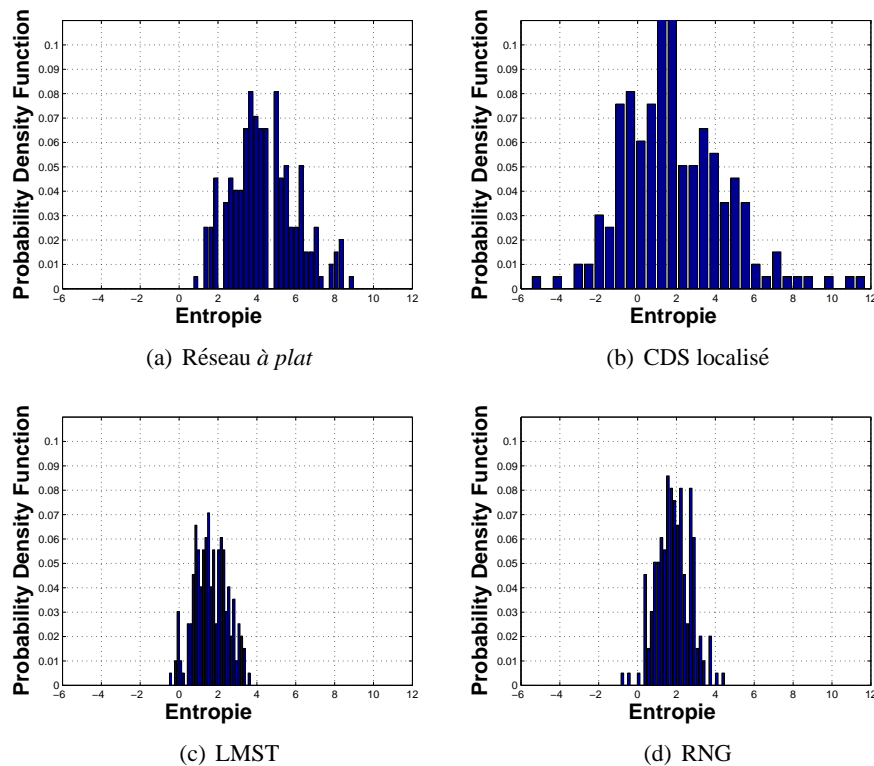


FIG. 2.7 – PDF de la variation de l'entropie - Cas du départ d'un seul nœud

**Départ de plusieurs nœuds** Une fois de plus, le réseau est considéré déployé à la date  $T_0$ . Nous considérons ici des topologies aléatoires de différentes cardinalité (et donc de différentes densités) : 100, 150, 180 et 200 nœuds sont considérés. À la date  $T_1$ , plusieurs nœuds tirés aléatoirement disparaissent du réseau. Attention, nous ne considérons que des topologies toujours connexes après le départ de ce groupe de nœuds. Nous mesurons la variation d'entropie avant/après le départ de ces nœuds. Notons qu'une seconde stratégie de construction de CDS a été évaluée : il s'agit d'un CDS obtenu après la construction d'un MIS puis la connexion des ensembles indépendants (noté MIS-CDS) [8]. Cette construction n'est pas localisée mais progresse par vagues dans le réseau.

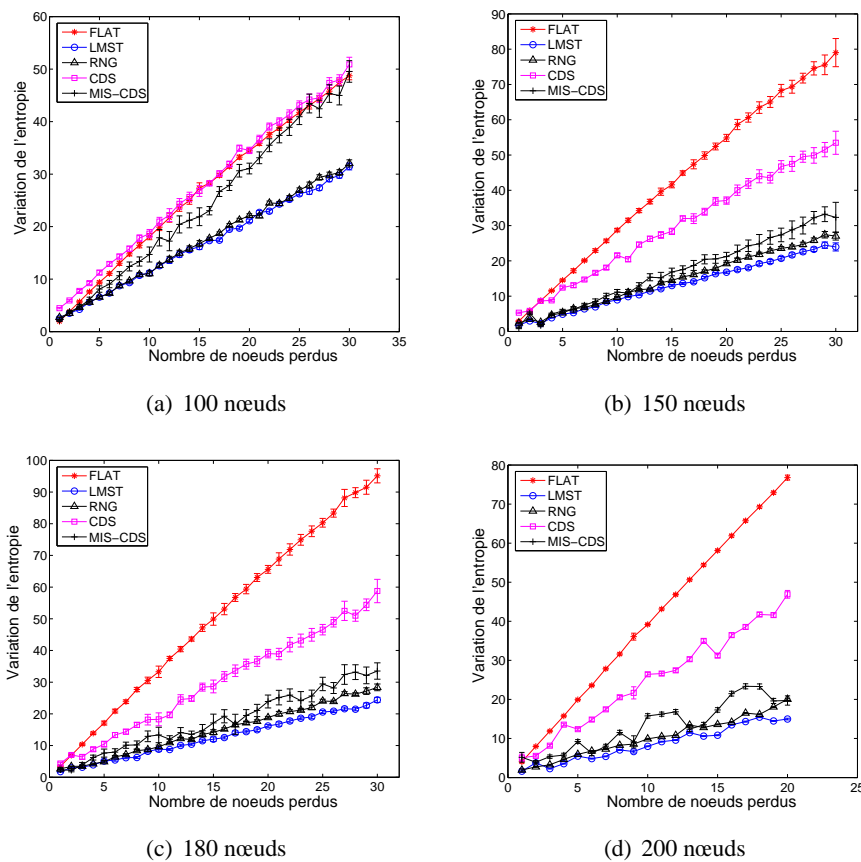


FIG. 2.8 – Comparaison des variations d'entropie

Les figures 2.8(a) à 2.8(d) indiquent la variation de l'entropie pour les quatre protocoles d'auto-organisation et le réseau à *plat* en fonction du départ de plusieurs nœuds (de 1 à 30) et suivant plusieurs cardinalités de réseaux. Clairement, les protocoles LMST et RNG offrent la meilleur adaptation aux changements d'états. Les constructions de CDS (localisé et par vague) s'adaptent mieux que le réseau à *plat* car la topologie logique d'auto-organisation est simplifiée mais moins que les stratégies LMST et RNG : cela est à mettre sur le compte de la nécessité de maintenir les règles d'un CDS (rappel : un nœud dominé est voisin d'au moins un dominant, les dominants forment un ensemble connexe). Ce n'est pas uniquement à cause du nombre de liens logiques sélectionnés. Notons qu'à faible densité le comportement d'un réseau à *plat* est très

proche des auto-organisations considérées. En revanche l'augmentation de la densité accentue l'intérêt pour l'auto-organisation de façon évidente.

### 2.5.5 Remarques et synthèse

Nous avons introduit la notion d'entropie pour le cas particulier de l'auto-organisation de réseaux radio multi-sauts. Pour ce faire nous avons repris la définition bien connue en thermodynamique qui permet de caractériser un comportement global à partir d'interactions locales. Ce travail a permis de souligner la pertinence des approches auto-organisées. En particulier, cela permet de montrer qu'un réseau auto-organisé possède un ordre, une stabilité plus importante qu'un réseau à *plat* et qu'un réseau auto-organisé est plus à même de réagir aux changements d'états dans le voisinage. Cela est principalement dû à la simplification de la topologie que réalisent les protocoles d'auto-organisation. De nombreuses questions se posent, dont nous n'avons pour l'heure pas les réponses : quel est finalement le lien entre entropie et qualité de service (disponibilité des routes, etc.) ? Quelle est l'auto-organisation qui conduit à l'entropie minimale et à une faible variation de l'entropie ?

Dans les chapitres suivants nous allons nous intéresser à l'application des concepts présentés dans la première partie de ce chapitre. Comment structurer un réseau ad hoc ? Comment structurer un réseau de capteurs ? Comment tirer partie de ces organisations pour les protocoles de communications ? Quels sont les impacts sur le réseau d'une auto-organisation par rapport à un réseau à plat ?



# Bibliographie

- [1] J.-L. Lu, Valois F. and M. Dohler. *Quantifying Self-organization in Multi-hop Wireless Networks by Means of Entropy* Research Report RR-xxxx, INRIA, October 2007.
- [2] F. Theoleyre and F. Valois. *Wireless Ad Hoc and Sensor Networks*, chapter Self-organization of ad hoc networks. ISTE Ltd (Hermès Science Publications/Lavoisier Company), July 2007. ISBN : 978 1 905209 86.
- [3] F. Theoleyre and F. Valois. *Réseaux mobiles ad hoc et réseaux de capteurs sans fil*, chapitre Auto-organisation de réseaux ad hoc : concepts et impacts. Hermès Science Publications - Traité IC2, série Réseaux et Télécommunications, Avril 2006. ISBN 2-7462-1292-7.



## Chapitre 3

# Structure virtuelle pour réseaux ad hoc et hybrides

### 3.1 Introduction

Dans ce chapitre nous allons voir comment appliquer les principes de l'auto-organisation préalablement décrit pour adresser les problématiques issus des réseaux ad hoc et des réseaux hybrides<sup>1</sup>. Parmi les points clefs de ces réseaux, nous voulons montrer comment une *bonne* auto-organisation permet *i*) de tirer partie de l'hétérogénéité des nœuds (en termes de mobilité, d'énergie, etc.), *ii*) d'intégrer naturellement un réseau ad hoc dans une architecture réseau, *iii*) améliorer le routage en termes de performances et de scalabilité.

Après avoir motivé le type d'organisation que nous allons utiliser pour répondre aux problématiques de mobilité, d'hétérogénéité ou d'énergie, nous présenterons en détail le protocole de construction de cette organisation. Parce que les nœuds sont mobiles, parce que des nœuds peuvent disparaître, le réseau est dynamique. Ainsi, une organisation peut être valide à un instant mais elle peut aussi être un frein au réseau un peu plus tard. Pour ces raisons, il est impératif de fournir un protocole d'auto-réparation (*self-healing*), de maintenance de cette organisation. Nous étudierons quelques propriétés importantes en termes de cardinalité, de stabilité et de persistance. Basé sur cette auto-organisation, une solution de routage tirant partie de la structure virtuelle sera proposée. Nous la comparerons à quelques protocoles de routage réactifs, pro-actifs et hybrides. Quelques extensions seront proposées comme la mise en simple d'un mécanisme d'énergie ou l'interconnexion d'un réseau ad hoc avec un réseau filaire. Nous essaierons d'insister sur un aspect qui nous semble important : si la construction et la maintenance d'une organisation ont un coût initial et récurrent alors les protocoles de communications tirant partie de cette organisation sont moins coûteux à développer voir *gratuit* comme le mécanisme d'économie d'énergie ou la construction d'une route par défaut pour le routage de paquets IP dans un réseau hybride. Nous y reviendrons.

### 3.2 Quelle structure d'auto-organisation ?

Le travail présenté ici va s'articuler autour de l'introduction de deux niveaux d'organisation. Lors de la première étape, basée sur l'ensemble des nœuds du réseau et de leurs connaissances

---

<sup>1</sup>Par réseaux hybrides, nous désignons les réseaux issus de l'interconnexion de réseaux ad hoc à l'Internet au moyen d'une passerelle d'interconnexion de type point d'accès sans fil.

du voisinage, un arbre va être construit. Lorsque la construction de cet arbre sera stable dans le voisinage, des clusters seront construits en ne considérant que les nœuds membres du backbone (cf. figure 3.1). L'idée de ce *backbone virtuel* est analogue à celle d'un backbone en réseau filaire : il s'agit de fédérer le réseau, de le structurer autour d'une épine dorsale puis d'articuler les protocoles de communication autour de celui-ci. Les clusters peuvent être associées à des zones de services pour, par exemple, développer des mécanismes d'allocations d'adresses ou de services.

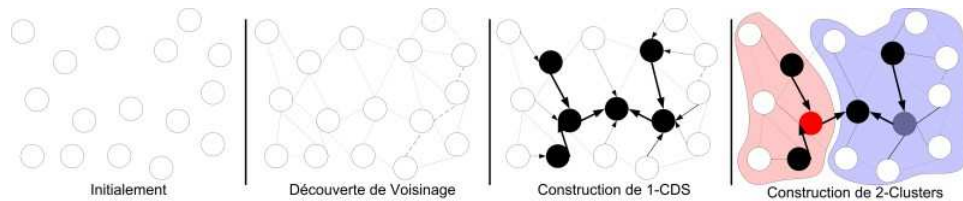


FIG. 3.1 – Construction de la structure virtuelle : découverte du voisinage, construction d'un backbone puis des clusters

Il nous semble important de commencer la construction de l'organisation par le backbone au lieu de construire les clusters puis de les connecter via un arbre. Effectivement, de cette façon, l'optimisation du nombre de nœuds participant à l'organisation peut être maîtrisé et le surcoût de construction des clusters est contenu. Soulignons que les deux processus de construction du backbone et des clusters ne doivent pas être séquentiels : dès qu'une partie du backbone est localement construite, le processus d'élection des leaders de clusters (qui appartiennent à la dorsale) s'initie. Cette décision locale est préférable dans le but de diminuer le temps de construction de la structure virtuelle. Nous avons donc deux processus qui progressent sous forme de deux vagues successives.

Rappelons les objectifs de cette structure :

- Tirer partie de l'hétérogénéité des nœuds en termes de mobilité, d'énergie, de ressources. Cela revient à solliciter davantage les nœuds les plus *forts*. C'est ici que commence la structuration du réseau, son organisation.
- Introduire de la stabilité dans le réseau : cette structure doit masquer une partie de la dynamique des nœuds et plus le niveau hiérarchique est important et plus la structure doit apparaître stable.
- Introduire une vue logique du réseau tel que la vue logique facilite les opérations de routage, adressage, localisation, etc.

Ces objectifs doivent respecter les propriétés énoncées précédemment : robustesse, persistance, construction locale, construction non orientée source, etc.

### 3.3 Structure virtuelle : construction et maintenance

Trois étapes sont nécessaires : acquisition de la connaissance du voisinage, construction du *backbone virtuel* puis des clusters et maintenance de la structure virtuelle.

L'acquisition du voisinage se fait par l'émission périodique de paquets *hello*. Un tel paquet contient l'identité du nœud, son rôle<sup>2</sup> dans la topologie virtuelle, son poids et l'identité de

<sup>2</sup>Voir paragraphe suivant.

son cluster<sup>3</sup>. Afin de supprimer les liens asymétriques, la liste des voisins de chaque nœud est également transmise. Deux paramètres de la topologie virtuelle viennent étendre cette connaissance du voisinage : la construction du backbone requiert la connaissance du voisinage sur  $k_B$  sauts (noté  $k_B$ -voisinage) tandis que la construction des clusters requièrent la connaissance du  $k_C$ -voisinage. Nous privilégions une stratégie à états de liens pour construire une telle connaissance car si une approche vecteur de distance permet de réduire le trafic de contrôle, les solutions à états de liens ont un temps de convergence plus faible.

### 3.3.1 Construction du backbone virtuel

L'algorithme de construction est inspiré de [8] et se déroule en deux phases : la construction d'un ensemble indépendant [13] puis sa connexion pour former un *connected dominating set* (CDS) [13]. Cette construction est initiée par un nœud du réseau : dans le cas d'un réseau hybride un point d'accès sans fil (AP) fait figure de leader naturel. S'il n'existe pas de leader, un mécanisme d'élection distribuée peut être utilisé [55].

Notons que la construction du backbone aurait pu s'appuyer sur tout autre algorithme. Plus précisément, nous allons construire un  $k_B$ -CDS, c'est-à-dire un ensemble connecté dominant tel que tout nœud dominé (non membre du backbone) est à au plus  $k_B$  sauts d'un dominant. La valeur de  $k_B$  peut être ajustée en fonction de la dynamique du réseau et de la cardinalité souhaitée de l'ensemble dominant.

Avant d'énoncer les règles de construction du backbone, précisons les différents états des nœuds. Initialement, tous les nœuds du réseau sont dans l'état *idle*, dans cet état le nœud attend une sollicitation extérieur pour initier une construction locale. Lorsqu'un nœud rentre dans le processus d'élection d'un dominant, il est dans l'état *actif*. Il devient *dominant* s'il est membre du backbone et *dominé* s'il possède au moins un dominant à moins de  $k_B$  sauts.

**Élection de l'ensemble dominant** La littérature abonde de constructions de CDS [5, 9, 18, 21, 80] et d'heuristiques [7, 34, 26] pour construire un CDS de cardinalité minimale (MCDS). Dans nos travaux, nous ne sommes pas focalisés sur la construction d'un CDS de taille minimale. Certes la cardinalité est importante et nous tenterons de la conserver assez faible. Mais, selon nous, ce n'est pas le seul critère qui doit être retenu : la robustesse et le coût protocolaire sont tout aussi important.

Initialement, le leader est le premier dominant du réseau. Il envoie un paquet à ses voisins et, à la réception de ce message, chaque voisin met à jour son état et le diffuse dans un nouveau paquet *hello* selon les règles suivantes :

- un nœud *idle* ou *actif* recevant un message d'un dominant à moins de  $k_B$  sauts devient *dominé* et le choisit comme dominant.
- un nœud *idle* recevant un message d'un dominé à moins de  $k_B+1$  sauts devient *actif* et déclenche un temporisateur pour l'élection.
- un nœud *actif* dont le temporisateur a expiré et qui possède le poids le plus fort au sein de son  $k_B$  voisinage devient *dominant*.

Cet algorithme progresse dans le réseau via une propagation des changements d'états : dès qu'un nœud change d'état, celui-ci est propagé sur  $k_B$  sauts pour maintenir une information cohérente dans son  $k_B$ -voisinage. À la fin de cette première étape, des nœuds sont élus *dominant* tandis

<sup>3</sup>L'identité du cluster correspond à l'adresse du chef du cluster. Initialement cette valeur est à `null`.

que les autres sont dominé. De plus, chaque dominé possède un père, son dominant. Notons que deux dominants sont éloignés d'au plus  $2k_B + 1$  sauts. La figure 3.2 illustre ce mécanisme.

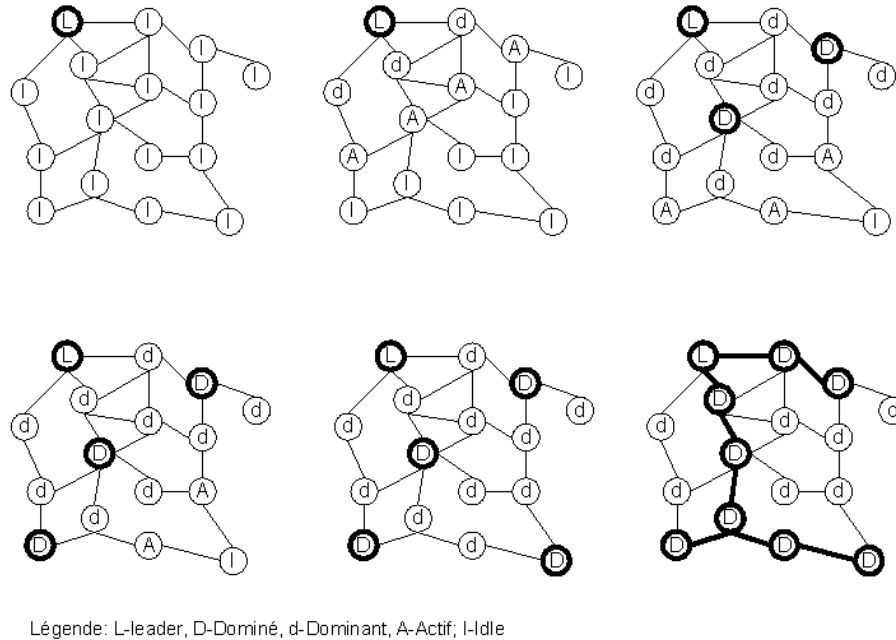


FIG. 3.2 – Construction du backbone virtuel : élection des dominants puis connexion

Sont donc élus *dominants* les nœuds de plus fort poids dans le voisinage à  $k_B$  sauts. Le choix du poids semble un critère déterminant pour la robustesse de la topologie : avoir un poids qui ne reflète pas la capacité d'un nœud à être leader dans son voisinage c'est s'exposer à des successions de changement d'états préjudiciables à l'auto-organisation. En réalité, le poids a assez peu d'influence sur les performances : nous avons comparé plusieurs métriques, plusieurs combinaisons de poids et si des différences apparaissent, elles ne sont pas fondamentales. Seule la prise en compte de l'énergie dans un poids multi-critères permet de favoriser l'extension de la durée de vie des nœuds. Classiquement, des poids de type *LowestId* pour sélectionner le nœud d'identité la plus faible dans le voisinage (ce qui revient à procéder à une sélection aléatoire des dominants) ou de type *HighestDegree* pour sélectionner comme dominants les nœuds ayant le plus de voisins (ce qui semble parfait dans le cas de couches physiques et MAC idéales mais pour des raisons d'interférences et de contention ne semblent pas adapter aux environnements radio réalistes) conduisent à des performances similaires. Nous avons proposé l'introduction d'un poids multi-critères tenant compte à la fois de la mobilité, de l'énergie et du degré tel que :

- l'énergie est un critère discriminant : sous un seuil fixé, un nœud ne peut prétendre à être dominant,
- la mobilité, que nous entendons comme relative, ne doit pas excéder un certain niveau pour éviter la sélection d'un nœud *trop* mobile comme dominant ce qui entraînerait des cassures dans l'auto-organisation,
- le cardinalité du voisinage doit offrir un compromis efficace entre une densité forte entraînant d'intenses contentions pour l'accès au médium et une densité faible préjudiciable à une connectivité de la structure virtuelle.

Bien entendu, le poids est périodiquement mis à jour pour tenir compte des changements topologiques.

**Connexion de l'ensemble dominant** De la même façon que le leader a initié le processus d'élection des dominants, il va initier l'interconnexion de la topologie logique. Pour cela, il va diffuser sur  $2k_B + 1$  sauts un paquet d'invitation de connexion `cds-invite`. Ce message sera relayé par les dominés, chaque dominé relayant au plus  $n$  fois ce type d'invitation afin de faire face aux collisions pouvant se produire. Nous avons montré formellement que cet algorithme était auto-stabilisant [73] sous l'hypothèse que la topologie physique était stable après une série de changements d'états et que le temps entre deux changements d'états (arrivée / départ d'un nœud, etc.) est suffisamment grand pour que les algorithmes convergent. Autrement dit, quelque soit la configuration de la topologie physique, quelque soit les événements se produisant, les algorithmes proposés conduisent à la construction de la topologie virtuelle.

Lorsqu'un dominant non connecté reçoit cette invitation, il envoie un `cds-accept` vers la source de l'invitation en suivant la route inverse. Les dominés relayant ce paquet d'acceptation passent dans l'état dominant. Chaque dominant fixe comme père, le dominant auquel il se connecte pour se rapprocher du leader. Récursivement, tous les dominants se retrouvent donc connectés. Lorsqu'un dominant reçoit plusieurs `cds-invite` dans une fenêtre de temps, plusieurs stratégies peuvent se présenter comme par exemple minimiser la hauteur de l'arbre. Nous avons choisi de maximiser le poids des nœuds intermédiaires du chemin afin d'accentuer la robustesse du backbone construit.

A la fin de cette étape, la construction d'un ensemble dominant connecté en maximisant le poids des nœuds dominant est donc réalisé. Chaque dominant possède un père dans la structure qui représente le prochain nœud vers le leader, racine de l'arbre.

### 3.3.2 Construction des clusters

Nous construisons ensuite des clusters de taille  $k_C$  tel que  $k_C > k_B$ . Dans le but de diminuer le trafic de contrôle, uniquement les dominants participent à ce processus : un dominé sera membre du même cluster que son père. Il est donc nécessaire que le backbone soit construit dans le voisinage pour démarrer ce processus : l'attente de la construction de tout le backbone n'est pas nécessaire ce qui permet de réduire le délai de convergence. Plus précisément, dès qu'un dominant est connecté, *i.e.* tel que tout son  $k_B$ -voisinage est déterminé, il peut initier la construction des clusters.

Les dominants vont initier une découverte de leurs voisins dominants sur le backbone virtuel, ce que nous nommerons le voisinage virtuel. Pour ce faire, chaque dominant émet périodiquement un paquet `cluster-hello` sur  $k_C - k_B$  sauts : ce paquet n'est relayé que par les dominants. Un dominant relaie ce paquet que s'il vient d'un voisin virtuel. Un dominant qui possède le plus fort poids dans le  $k_C - k_B$ -voisinage virtuel devient chef de cluster, `clusterhead`. Pour notifier sa décision, il émet un `cluster-hello` à son  $k_C - k_B$ -voisinage virtuel. Un dominant  $D$  relayant ce message choisit l'émetteur comme `clusterhead` si les conditions suivantes sont remplies :

- $D$  ne possède pas de `clusterhead`,
- l'émetteur du paquet est à au plus  $k_C - k_B$  sauts,
- le nœud précédent a choisi le même `clusterhead`.

Les clusters formés sont donc connexes et chaque dominant possède un clusterhead à au plus  $k_C - k_B$  sauts. De plus comme chaque dominé est à au plus  $k_B$  sauts de son dominant, les clusters formés ont un rayon maximal de  $k_C$ .

Comme les paquets `cluster-hello` ne sont transmis que par les dominants, le coût en terme de contrôle est limité. De plus, ces messages ne sont utilisés que pour la construction et non pas pour la maintenance qui se base uniquement les informations contenues dans les paquets `hello`.

### 3.3.3 Algorithmes de maintenance

Parce que les topologies des réseaux ad hoc et hybrides sont dynamiques, il est impératif de proposer une stratégie de maintenance. Plutôt que de reconstruire l'ensemble backbone virtuel et clusters périodiquement ou dès que survient une cassure, nous privilégions de proposer une maintenance réactive événementielle qui préserve les parties du backbone non touchées par un changement d'état. Dès que les règles de la structure virtuelle sont violées, un mécanisme de reconstruction local est initié.

La maintenance va être basée sur la connaissance du  $k_B$ -voisinage de chaque nœud. La diffusion périodique des paquets `hello` permet d'informer le voisinage des changements de topologie. Chaque nœud diffuse donc périodiquement : son identité, son poids, son état, son père, le nombre de sauts vers son père, l'identité de son clusterhead, la liste de ses voisins ainsi que leur poids et leur état. Chaque dominant a donc la connaissance de tous ses dominés et de tous ses fils *i.e.* les dominants dont il est le père dans la structure virtuelle. Notons également qu'un dominé a également la connaissance de plusieurs pères potentiels en plus de son dominant courant : il peut donc constituer une liste de pères secondaires qui peut être utile en cas de perte de son propre père.

Plusieurs règles doivent être surveillées. Dès leur violation, une procédure ad hoc sera initiée. Chaque nœud doit vérifier et maintenir :

- la propriété de dominance,
- le maintien de la connexité de la structure virtuelle,
- le maintien des clusters.

A ces règles, viennent se greffer deux règles pour l'optimisation de la cardinalité de la structure virtuelle via la suppression de dominants inutiles.

**Propriété de dominance** Chaque dominé  $d$  doit vérifier que son père  $P$  est toujours valide, *i.e.* *i)* que  $P$  est toujours dominant *ii)* que la distance entre  $d$  et  $P$ ,  $k_{dP}$ , est d'au plus  $k_B$  sauts et *iii)* qu'il existe un autre dominé à  $k_{dP} - 1$  sauts ayant également  $P$  comme père afin de forcer la connexité des zones de dominance.

Afin de favoriser la persistance de la structure, un père valide est toujours conservé. En revanche dès que son père n'est plus valide,  $d$  doit chercher un autre dominant valide dans sa table de voisinage. S'il en trouve un, il envoie un nouveau paquet `hello` contenant la nouvelle identité de son père et celui-ci ajoutera  $d$  à sa liste de dominés. S'il n'y a pas de dominant éligible dans sa table de voisinage alors  $d$  passe dans l'état `actif` et une nouvelle élection se produit conformément aux règles énoncées pour l'initialisation.

**Connexité de la structure virtuelle** Il est impératif que la structure virtuelle et donc que le backbone virtuel reste connexe. Pour vérifier la connexité, le leader de la structure virtuelle doit

émettre périodiquement un paquet `ap-hello` contenant un numéro de séquence croissant relayés uniquement par les dominants sur le backbone. Lorsqu'un dominant reçoit un paquet de son père, il le relaie, sinon il fixe l'émetteur comme père secondaire si le numéro de séquence reçu est supérieur au numéro de séquence du dernier paquet `ap-hello` relayé par son père. Nous évitons ainsi de prendre comme père secondaire un descendant. Cette table de pères secondaires permet de créer des liens de secours autorisant une reconnexion sans latence et sans trafic de contrôle la plupart du temps.

Un dominant  $D$  se considère déconnecté si son père n'est plus dominant ou n'est plus son voisin ou s'il n'a pas reçu de paquets `ap-hello` de son père pendant une durée  $\tau$ . Il choisit alors comme nouveau père le dominant de poids le plus fort dans sa liste de pères secondaires. Si cette liste de pères secondaires est vide, un mécanisme de découverte est nécessaire :

1.  $D$  diffuse sur  $2k_B + 1$  sauts un paquet `cds-reconnect` contenant le dernier numéro de séquence connu,
2. Ce paquet est relayé par les dominés du  $2k_B + 1$ -voisinage vers les autres dominants,
3. Les dominants recevant ce paquet et ayant reçu un paquet `ap-hello` contenant un numéro de séquence supérieur répondent avec un paquet `cds-invite` relayé vers  $D$  en prenant la route inverse.

Les autres dominants entendant ce paquet `cds-invite` mettent à jour leur table de pères secondaires en inscrivant la source de ce paquet comme père secondaire. Lorsque  $D$  reçoit ce paquet, il répond explicitement avec l'émission unicast vers son nouveau père d'un `cd-reply` : tous les dominés relayant ce paquet vers le nouveau père deviennent alors dominants.

Si l'émission de plusieurs `cds-reconnect` ne donne aucune réponse, notamment à cause d'un taux de collisions trop important,  $D$  va prendre l'initiative de casser la structure virtuelle dont il est le père. Il va diffuser un `cds-break` vers tous ses fils dans le backbone virtuel et vers tous ses dominés. Un nœud possédant comme père l'initiateur de ce message va le relayer puis va passer dans l'état `idle` dans l'attente d'une sollicitation extérieure afin de redémarrer une phase d'élection. Nécessairement, il se trouve un dominant situé à au plus  $k_B + 1$  sauts de  $D$  : c'est ce nœud qui enverra à  $D$  un `cds-invite` pour afin que  $D$  bascule dans l'état `actif`. La reconstruction locale se propagera par vague comme lors de l'initialisation.

**Optimisation de la cardinalité** Deux règles appliquées au cours la maintenance permettent d'optimiser la cardinalité de la structure virtuelle :

1. Le mécanisme d'élection des dominants entraîne l'élection de dominants inutiles *i.e.* ne possédant pas de fils dominant et ne possédant que des dominés à au plus  $k_B - 1$  sauts. Un tel dominant  $D'$  va diffuser un paquet `cds-useless` à ses dominés afin qu'ils choisissent comme dominant le père de  $D'$ .  $D'$  devient dominé en maintenant son père comme dominant.
2. Chaque dominant peut optimiser la distance vers le leader. Quand un dominant  $D_1$  reçoit un paquet `ap-hello` d'un dominant  $D_2$  possédant un numéro de séquence supérieur au dernier numéro de séquence connu par  $D_1$  alors celui-ci prends  $D_2$  comme père. Cette règle permet de minimiser la hauteur de l'arbre et donc la cardinalité du backbone.

**Maintien des clusters** De la même façon que les dominés ne prennent pas part à la construction des clusters, ils ne participent pas à leur maintenance. Rappelons qu'un dominant diffuse également dans son paquet `hello` la distance vers son clusterhead  $C$  et le prochain saut  $N$  dans le backbone pour l'atteindre.

Un dominant  $D$  considère que son clusterhead n'est plus valide si l'une des conditions suivantes est remplie :

- $N$  n'est plus voisin de  $D$ ,
- le clusterhead de  $N$  est différent de celui de  $D$ ,
- le nombre de sauts entre  $N$  et  $C$  est supérieur à  $k_C - k_B - 1$ .

Si  $N$  annonce un nouveau clusterhead  $C'$  à au plus  $k_C - k_B - 1$  sauts alors  $D$  prend  $C'$  comme clusterhead.

Si  $C$  n'est plus valide,  $D$  va tenter une reconnexion explicite en recherchant dans ses voisins virtuels, un nouveau clusterhead tel que :

- un voisin virtuel  $D^*$  ayant pour clusterhead  $C^*$  annonce une distance vers son clusterhead inférieure à  $k_C - k_B - 1$  sauts ,
- un voisin virtuel  $D^*$  ayant pour clusterhead  $C$  annonce une distance vers son clusterhead inférieure à  $k_C - k_B - 1$  sauts et  $D$  n'est pas le prochain dominant vers  $C$  de  $D^*$ .

Si un dominant ne peut pas se rattacher à un clusterhead, il s'élit lui-même clusterhead.

Dès qu'un dominant change de clusterhead ou d'état, il diffuse un nouveau paquet `hello` mis à jour pour que les dominants ayant choisi  $D$  comme prochain saut vers le clusterhead modifient également leurs informations.

De la même façon qu'il peut y avoir des dominants inutiles, des clusterheads inutiles peuvent exister. Un clusterhead est dit inutile si aucun voisin virtuel ne l'a choisi comme clusterhead. Un tel clusterhead va perdre son rôle et chercher un clusterhead pour s'y rattacher.

### 3.4 Auto-organisation : propriétés et impacts

Avant d'étudier l'apport de l'auto-organisation sur les protocoles de communications et les architectures pour réseaux hybrides, nous allons étudier quelles sont les propriétés clefs de l'auto-organisation proposée. Les résultats sont issus de campagnes de simulation en utilisant OPNET Modeler [4]. Les nœuds utilisent une couche MAC de type 802.11b en mode DCF et sans l'utilisation de RTS/CTS. La portée de communication est de 300 mètres. Pour modéliser la mobilité des nœuds, les résultats présentés ici considèrent un modèle de type *random waypoint* [20]. Notons que d'autres campagnes de simulation, non présentées ici, ont étudiées l'impact d'autres modèles de mobilité comme le *random walk*. Si les résultats diffèrent, il est important de souligner que le comportement global et les propriétés annoncées ici sont toujours les mêmes. Par défaut, nous considérons une topologie de 40 nœuds avec un degré de 10 et une vitesse moyenne de  $5m.s^{-1}$ . L'ensemble des résultats de simulation s'appuient sur un intervalle de confiance d'au moins 95%. Les performances de la topologie virtuelle sont notés CDCL sur les courbes.

Nous avons cherché à souligner les propriétés qui nous semblent clef pour une auto-organisation : cardinalité, robustesse, convergence en fonction des paramètres de l'auto-organisation :  $k_B$  pour le backbone et  $k_C$  pour les clusters. La résistance à la mobilité est également mesurée afin de souligner le comportement de la procédure de maintenance. Enfin, le surcoût protocolaire est analysé.

#### 3.4.1 Évolution de la cardinalité

La cardinalité du backbone est mesurée au cours du temps afin de tenir compte de la dynamique des nœuds et de valider la capacité de l'algorithme de maintenance à réagir au changement topologique. Nous comparons ici avec l'algorithme de [80] qui propose la construction d'un CDS

d'un point de vue localisé. Clairement, plus la distance entre un dominant et un dominé peut être importante (*i.e.*  $k_B$ ) et plus la cardinalité du backbone est faible (figure 3.3(a)). Si des fluctuations concernant la cardinalité sont observées, notons qu'une valeur moyenne stationnaire est observable : même s'il y a de brusques changements de topologie, la cardinalité du backbone n'est pas grandement affectée. Des observations similaires sont réalisables concernant les clusters (figure 3.3(b)).

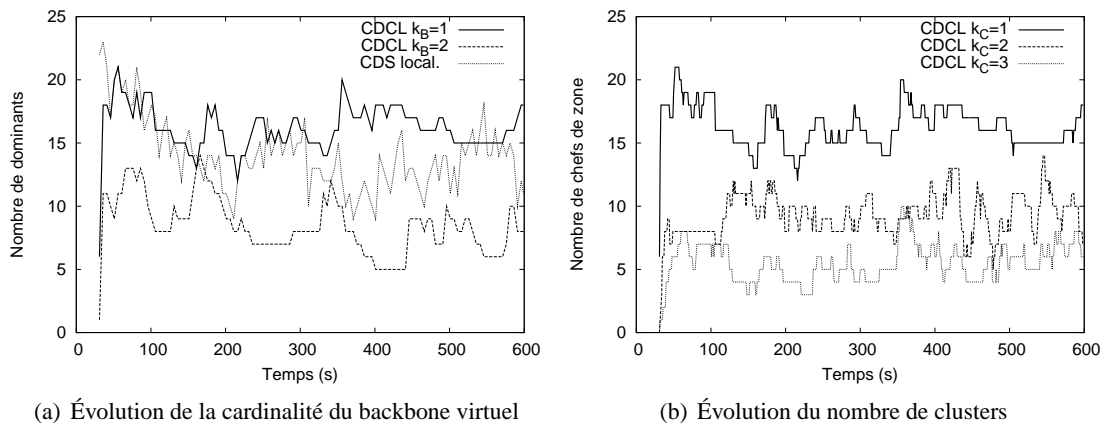


FIG. 3.3 – Variations de la cardinalité au cours du temps.

### 3.4.2 Impact du degré

Les résultats précédant concernent une topologie de 40 nœuds. Une question demeure : comment se comportent les procédures de construction et de maintenance de la structure virtuelle en fonction de la densité. C'est ce qu'indique la figure 3.4(a). Le nombre de dominants décroît logiquement en fonction du degré puisque plus le degré augmente et plus un dominant peut couvrir de nœuds dans son voisinage à 1 saut ou à 2 sauts. Pour des densités supérieures à 6-7, les résultats des différentes stratégies ( $k_B = 1$  ou  $k_B = 2$ ) sont très proches, et on ne note plus de gain significatif.

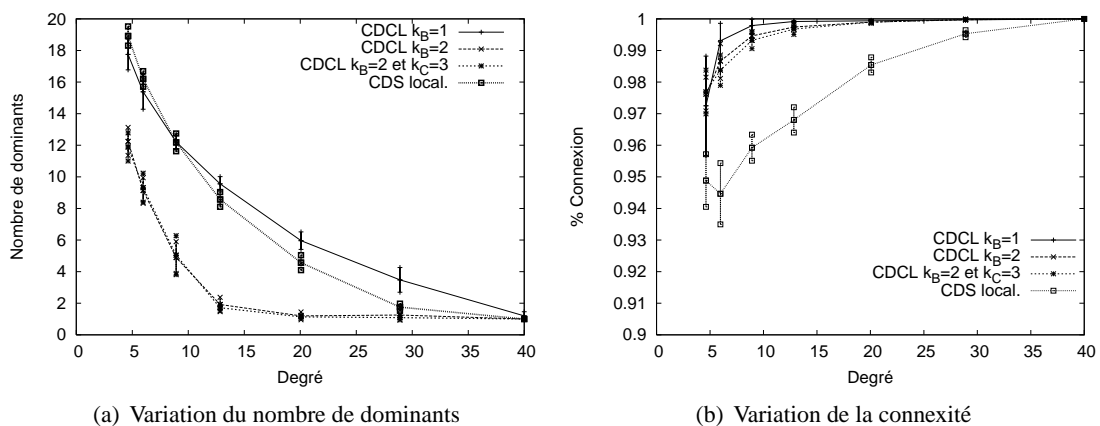


FIG. 3.4 – Influence du degré sur la cardinalité et la connexité.

La figure 3.4(b) illustre une propriété clef de la structure d'organisation que nous avons introduit. Nous mesurons ici la connexité de la structure, autrement dit la proportion de temps pendant laquelle la topologie virtuelle est effectivement disponible. Le premier résultat est que, quelque soit le degré, quelque que soit le paramétrage retenu pour l'organisation ( $k_B = 1$  ou  $k_B = 2$ ), ce pourcentage de connexion est important. Bien entendu, plus le degré augmente et plus le taux de connexion est important : les opportunités de reconnexion en cas de cassure sont plus importantes. L'analyse du comportement du CDS localisé indique que la connexité plus faible obtenue est le résultat d'incohérences dans les tables de voisinage. Des résultats similaires, non présentés ici, sont à mettre au crédit du clusters.

### 3.4.3 Impact de la mobilité

Les protocoles de communications pour les réseaux MANET doivent être capable de faire face à la mobilité des nœuds. La figure 3.5(a) montre que la cardinalité du backbone est peu sensible à l'intensité de la mobilité. La cardinalité s'avère stable grâce aux mécanismes de maintenance proposés. Il en est de même pour la connexité (figure 3.5(b)) : même pour des vitesses élevées dépassant les  $30m.s^{-1}$ , le taux de connexité de la structure virtuelle est supérieur à 98% indépendamment de la valeur de  $k_B$ . Notons que la cardinalité de la structure virtuelle est toujours plus importante pour de fortes vitesses : c'est principalement dû à des incohérences de voisinage résultant de la mobilité. Cette cardinalité supérieure apporte un surplus de robustesse permettant de supporter ces fortes mobilités.

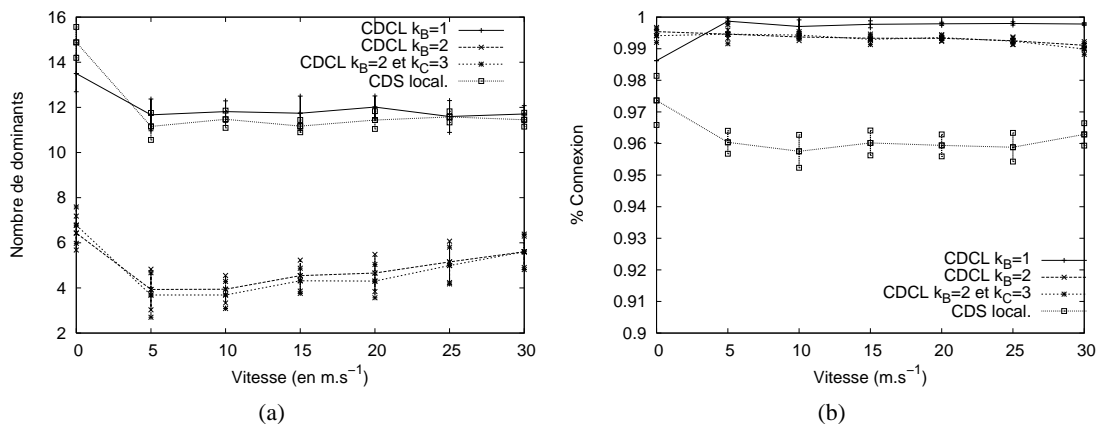


FIG. 3.5 – Impact de la mobilité sur (a) la cardinalité et (b) la connexité du backbone.

Nous avons également mesuré la persistance des dominants dans la structure d'auto-organisation soit en tant que membre du backbone soit en tant que leader de clusters. Nous appelons persistance le temps moyen pendant lequel un nœud exerce son rôle de dominant. Les figures 3.6(a) et 3.6(b) montrent clairement que plus le paramètre  $k_B$  est important et plus la persistance est faible : effectivement, les deux structures d'auto-organisation utilisées sont plus complexes à maintenir car elles nécessitent une information sur plusieurs sauts : plus de changements dans le backbone et les clusters sont observés. Toutefois, la persistance des dominants et des leaders de clusters restent élevés. L'auto-organisation introduit donc de la stabilité en construisant un référentiel la plupart du temps disponible.

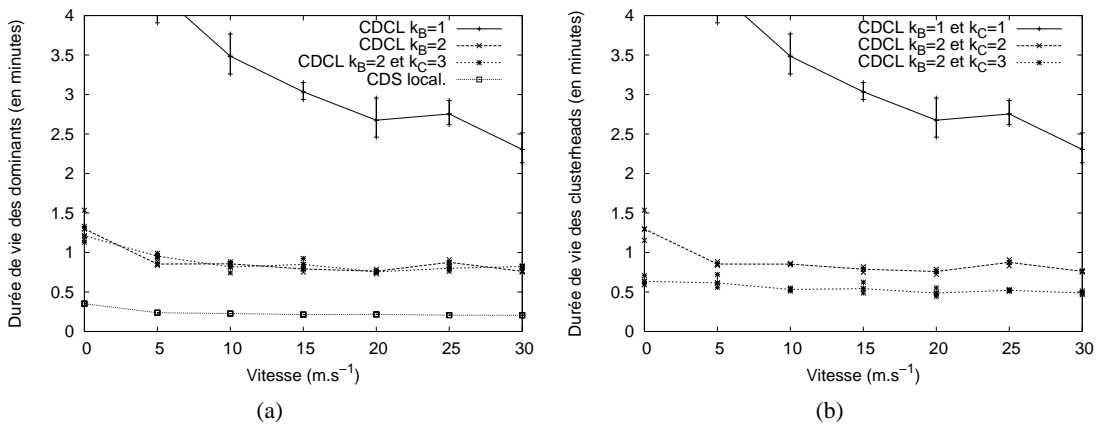


FIG. 3.6 – Impact de la mobilité sur la persistance (a) du backbone et (b) leaders de clusters.

### 3.4.4 Passage à l'échelle

Les précédents résultats de simulation considéraient un réseau composé de 40 nœuds. Nous étudions ici l'impact de la cardinalité du réseau sur les performances de la structure d'auto-organisation (figures 3.7(a) et 3.7(b)). Notons que la connexité est peu sensible à la cardinalité du réseau ce qui montre la faculté du passage à l'échelle des protocoles d'auto-organisation proposés. En revanche la cardinalité est croissante lorsque le nombre de nœuds devient important : le nombre de sauts qui séparent un dominant d'un dominé devenant important, plus de paquets `ap_hello` sont perdus, entraînant de nouvelles reconnections, limitant les opportunités de suppression de redondance et pénalisant la cardinalité de la structure virtuelle. Précisons que les mêmes observations sont faites pour les clusters.

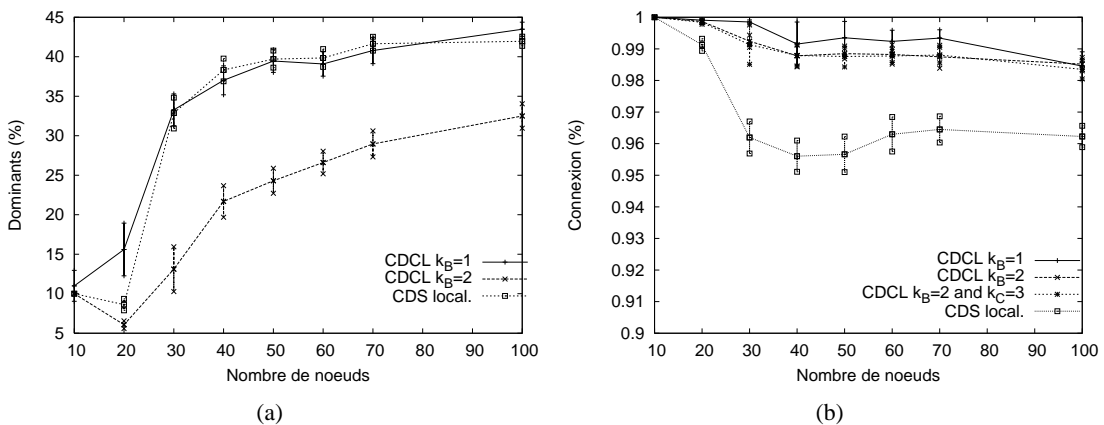


FIG. 3.7 – Impact de la cardinalité du réseau sur (a) le nombre de dominants et (b) la connexité de la structure.

### 3.4.5 Coût du trafic de contrôle

Si les sections précédentes ont montré les *bonnes* propriétés de la structure d'auto-organisation, nous devons nous attarder sur son coût protocolaire de construction et de maintien. Plus précisément, nous avons observé l'évolution du coût protocolaire en fonction de la cardinalité du réseau (figure 3.8). Nous avons défini le coût protocolaire en nombre de paquets de contrôle par nœud et par seconde. L'auto-organisation proposée exhibe un coût protocolaire bien supérieur à des algorithmes classiques comme le CDS localisé pour lequel seul un paquet `hello` périodique est envoyé. Le coût de contrôle provient de la propagation des paquets `hello` sur  $k_B$  sauts mais aussi des procédures de maintien de la connexité du backbone et des clusters. Il est intéressant de souligner que l'auto-organisation supporte bien le passage à l'échelle. Toutefois, eu égard aux performances mesurées précédemment notamment en terme de robustesse et de persistance, ce surcoût semble acceptable.

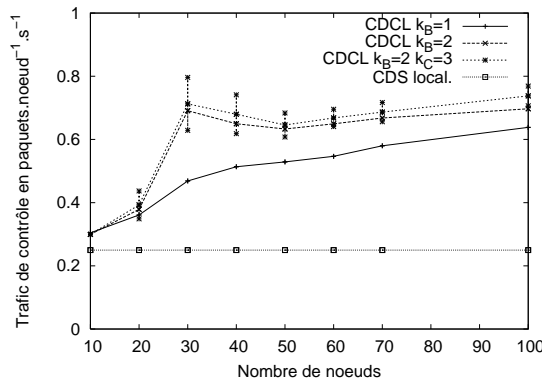


FIG. 3.8 – Analyse du trafic de contrôle

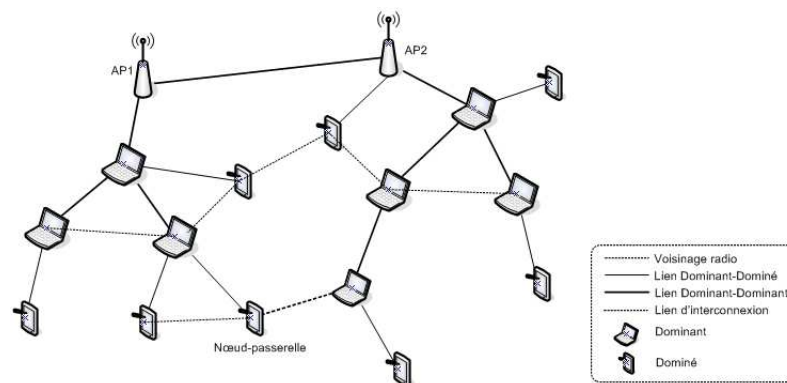
Le tableau 3.1 détaille le surcoût protocolaire pour deux configurations données. Notons que les paquets `hello` représentent autour de 50% du trafic de contrôle : cela signifie que les mécanismes de construction et de maintien ne sont pas si coûteux. Logiquement lorsque  $k_B$  augmente, le trafic de contrôle notamment à cause de la propagation des paquets `hello`. Le coût de construction/maintenance des clusters est négligeable et celui requis pour la construction/maintenance du backbone est acceptable.

Paramétrage	hello	Backbone	ap-hello	Clusters	Total
CDCL ( $k_B = 1, k_C = 2$ )	0.25	0.1	0.16	0	0.51
CDCL ( $k_B = 2, k_C = 23$ )	0.35	0.2	0.09	0.0016	0.54

TAB. 3.1 – Détail du coût de contrôle

## 3.5 Extensions

Les propriétés mise en évidence précédemment soulignent plusieurs *bonnes propriétés* de la structure virtuelle proposée. La persistance, la robustesse, l'adaptabilité de la structure virtuelle aux changements topologiques nous poussent à étudier quel apport peut avoir cette structure virtuelle pour fournir une architecture de communication adaptée aux contraintes des réseaux ad

FIG. 3.9 – Interconnexion de *backbones virtuels*

hoc et des réseaux hybrides. Nous revenons ainsi à notre objectif initial : quel apport l'auto-organisation peut-elle avoir dans les mécanismes réseaux ? Et nous essaierons de répondre à cette question : un réseau basé sur une auto-organisation (donc structuré) est-il plus efficace qu'un réseau considéré à plat ?

A partir de maintenant nous supposons que la structure virtuelle composée du backbone virtuel fédérateur et des clusters est formée. Nous allons montrer successivement comment nous pouvons aisément adresser les problématiques posées par les réseaux maillés (section 3.5.1) puis comment la présence d'une auto-organisation peut conduire à la mise en place de mécanismes simples de gestion d'énergie (section 3.5.2). Enfin, nous montrerons comment une telle structure d'auto-organisation permet de construire une route par défaut immédiate pour la mise en place d'un trafic entre le cœur du réseau ad hoc et l'Internet. À la fin de ce chapitre, nous présenterons un protocole de routage basé sur la topologie virtuelle et nous étudierons ses performances vis à vis des protocoles de routages classiques (proactif, réactif).

### 3.5.1 D'un arbre à un treillis

La structure virtuelle se base sur la construction d'un arbre. Outre la cardinalité, l'intérêt de recourir à une telle structure réside dans une structure sans boucle permettant son utilisation à des fins de routage de façon efficace. Toutefois, cette structure virtuelle peut être aisément adaptée au cas des réseaux maillés [17] offrant ainsi plus de robustesse et une meilleure répartition de la charge de trafic. La construction d'un réseau maillé se fait en deux étapes. Tout d'abord la construction d'un *backbone virtuel* par routeur (AP) du réseau maillé est réalisée. Ensuite, l'interconnexion de deux backbones issus d'AP différents est réalisée via un mécanisme spécifique exécuté par les dominants. Nous tirons ainsi partie de la construction puis de la maintenance efficace de la structure virtuelle sur lesquelles nous greffons des mécanismes d'interconnexion. Chaque *backbone virtuel* est identifié par l'identité de sa racine (AP) qui est diffusée dans les paquets `ap-hello` et `hello`. Bien entendu lors de la maintenance, un dominant recevant plusieurs `ap-hello` issus d'APs différents peut changer de racine, impliquant que tous les nœuds l'ayant comme ancêtre change également de racine.

**Interconnexion des *backbones virtuels*** Nous supposons que tous les APs, routeurs du réseau maillé, sont interconnectés, soit directement, soit via un réseau d'interconnexion. Initialement,

Paramètre du backbone ( $k_B$ )	Taux de livraison	Taux de connexité
1	97,2	99,3
2	97,4	97,5

TAB. 3.2 – Inondation du backbone - Efficacité

Paramètre du backbone ( $k_B$ )	Trafic de contrôle en nombre de paquets par inondation			
	Dominants		Dominés	
	paquets	retransmissions	paquets	retransmissions
1	15	6,9	5,2	0,4
2	8,6	4,2	6,2	0,3

TAB. 3.3 – Inondation du backbone - Trafic de contrôle

chaque AP lance un processus de construction de la structure virtuelle. Ensuite, localement, vient la phase d'interconnexion. Nous allons donc faire une élection de nœuds passerelles en contrôlant la cardinalité des nœuds jouant un rôle dans l'auto-organisation (*i.e.* nœuds passerelles et dominants inclus). Chaque dominé va diffuser dans son paquet `hello` les identités d'APs différentes entendues dans son voisinage. De cette façon les dominants acquièrent une connaissance suffisante pour sélectionner les dominés qui deviennent alors des nœuds passerelles. Pour contrôler la cardinalité des nœuds actifs dans la structure virtuelle, nous imposons que chaque dominant possède au plus un nœud passerelle vers un backbone différent.

**Application** Lors d'une inondation, pour une découverte de route par exemple, les requêtes sont diffusées uniquement par les dominants et peuvent être relayées par les nœuds-passerelles afin de propager l'inondation plus rapidement.

**Performances** Nous nous intéressons ici aux performances d'un réseau maillé sous l'angle de la robustesse. Le treillis considéré est le résultat de l'interconnexion de deux backbone liés à différents AP. Pour mesurer la connexité du treillis ainsi construit, un paquet inonde périodiquement les liens des backbones et les liens d'interconnexion. Le taux de livraison est alors très important : en moyenne plus de 97% des nœuds reçoivent ce paquet (tableau 3.2). Notons que ce trafic de contrôle (tableau 3.3) est essentiellement acheminé par les dominants avec un faible impact sur les dominés. La hiérarchisation du rôle des nœuds dans le réseau permet donc d'éviter d'impliquer les nœuds les plus *faibles* dans l'inondation. Soulignons également que les performances présentées ici ré-affirment un résultat bien connu : l'inondation est plus efficace, moins coûteuse lors de l'utilisation d'un backbone au lieu d'une inondation aveugle : dans le premier cas 18.4 paquets sont requis alors que 40 seraient nécessaire dans le deuxième cas.

### 3.5.2 Mécanisme d'économie d'énergie

Rappelons que seul l'endormissement d'un nœud peut économiser de l'énergie de façon significative. La structure virtuelle proposée est *naturellement* adaptée à la mise en place d'un tel mécanisme. Effectivement, l'auto-organisation structure le réseau ad hoc/hybride en deux catégories de nœuds : les dominants et les dominés. Clairement, les dominants sont impliqués dans la *vie du réseau* à travers le maintien de l'organisation, l'acheminement des paquets d'inon-

dation ou de découvertes de routes tandis que les nœuds dominés ont un rôle passif. Ainsi si un nœud dominé éteint sa radio et/ou se met en veille pour économiser de l'énergie cela aura peu d'impact sur le comportement du réseau : son dominant devra conserver en cache les paquets qui lui sont arrivés pendant son sommeil afin de les délivrer à son réveil. Tous les dominés ne doivent pas s'endormir car une partie d'entre eux est nécessaire aux procédures de maintenance. Il s'agit de trouver le ratio optimal de dominés endormis sans compromis sur les performances.

**Description du mécanisme** Cette procédure d'endormissement ne se déclenche que lorsque la structure virtuelle est localement construite. Un dominant ne possédant pas suffisamment de voisins actifs n'est pas autorisé à s'endormir pour éviter les risques de cassure de la topologie virtuelle. Nous proposons qu'un dominant possède au moins 6 voisins actifs pour pouvoir changer de rôle et s'endormir. Pour ne pas pénaliser le processus de maintenance, nous interdisons à un dominant de s'endormir lors d'une procédure de reconnexion de dorsale. Nous introduisons une pénalité ( $P_e$ ) représentant le nombre de 1-voisin de plus faible poids. La probabilité d'endormissement d'un nœud (dominé ou dominant) est alors :

$$P_{sleep} = \frac{1}{P_e}$$

Les nœuds autorisés à s'endormir sont donc les dominés et les dominants les moins critiques pour la topologie virtuelle. Soulignons que tout autre mécanisme peut être adapté.

**Impact du mécanisme sur les performances** Nous avons simulé un réseau hybride avec 1 seul point d'accès fixe et 30 nœuds mobiles qui se déplacent suivant un modèle de mobilité de type *random waypoint* [20]. Nous avons mesuré l'impact que pouvait avoir le mécanisme d'économie d'énergie sur la connexité de la structure virtuelle mais aussi sur le temps moyen d'endormissement par nœud dans le cas où les nœuds ont initialement le même niveau d'énergie puis dans le cas où un nœud particulier ( $N_1$ ) possède très peu d'énergie. Le tableau 3.4 fournit les performances mesurées.

	Taux de Connexion backbone	Taux de connexion Clusters	Temps moyen d'endormissement (s)	Temps endormi $N_1$
Avec énergie	89.7	95.4	40.1	216
Sans énergie	93.4	96.6	0	0

TAB. 3.4 – Impact du mécanisme d'énergie sur les performances

Nous donnons ici les résultats uniquement en tenant compte du poids que nous avons introduit précédemment. Comme celui-ci tiens compte de l'énergie résiduelle pour sélectionner les nœuds dominants, il conduit à de meilleures performances que les poids indépendant de ce critère qu'ils soient basés sur l'identité la plus faible ou le degré le plus fort. Il faut souligner que l'utilisation d'un mécanisme d'économie d'énergie fait chuter dans des proportions raisonnables la persistance de la structure virtuelle. Ce mécanisme favorise également les nœuds possédant le moins d'énergie résiduelle.

### 3.5.3 Interconnexion à un réseau hybride et routage par défaut

L'interconnexion de réseaux ad hoc avec Internet via des points d'accès sans fil laissent apparaître deux nouvelles problématiques : *i*) tout d'abord il s'agit de construire une route vers un

nœud du réseau ad hoc faisant office de passerelle vers Internet *ii*) puis il s'agit de mettre en place un mécanisme de gestion de la mobilité pour acheminer le trafic entrant et maintenir les échanges de flux de trafic. Une fois de plus, la littérature abonde de contributions scientifiques pertinentes sur le sujet comme MIPMANET [46] ou MEWLANA [31]. L'objectif de cette section est de montrer comment, à partir de l'auto-organisation proposée, le backbone peut apparaître comme une extension du réseau filaire grâce à ses propriétés de persistance notamment. Nous verrons qu'une solution de route par défaut et de localisation peut être aisément mis en place.

**Route par défaut - trafic montant** Il est évident que le leader de la structure d'auto-organisation doit être le point d'accès sans fil faisant également office de passerelle vers Internet. Le leader est alors le routeur par défaut pour l'ensemble des nœuds dominés/dominants du réseau ad hoc. Ainsi dès qu'un nœud du réseau ad hoc souhaite envoyer un paquet vers Internet, le paquet sera transmis à son dominant qui lui même le relaiera vers son père dans le backbone et ainsi de suite jusqu'au point d'accès. Cette construction de cette route par défaut se fait sans latence et sans utilisation de paquets de contrôle supplémentaire. Cette route par défaut apparaît donc gratuite lorsque le réseau a été organisé préalablement.

**Routage du trafic descendant et localisation** Lorsque le trafic est initié par les nœuds du réseau ad hoc, il est possible de maintenir une route inverse depuis le leader vers le destinataire pour permettre la transmission d'un flux de données dont la source ne se trouve pas dans le réseau ad hoc. Une fois de plus, il n'est pas nécessaire d'utiliser un trafic de contrôle et la latence est nulle.

En revanche, lorsqu'une route inverse n'est pas connue, un mécanisme de localisation proactif du destinataire dans le réseau ad hoc doit être proposé. La proposition que nous faisons tire partie de la construction en arbre de l'auto-organisation. Lorsque le leader reçoit un paquet à destination d'un nœud  $D$  qui n'est pas connu dans sa table de routage, il va diffuser un paquet `route_request` à ses fils dominants. Chaque dominant qui reçoit ce paquet vérifie s'il connaît une route vers  $D$ . Si oui, alors le dominant joue le rôle de proxy et réponds avec un `route_reply` à destination du leader. Une route est alors connue. Si le dominant ne connaît pas de route vers  $D$  alors il relaie le `route_request` à ses propres fils dominants. Finalement, seuls les dominants sont utilisés pour construire une route vers  $D$  et vu que la cardinalité de l'ensemble dominant est assez faible (environ 25%, voir résultats précédents), le coût protocolaire de localisation est maîtrisé.

**Performances** Nous avons évalué les performances de ces mécanismes en utilisant l'outil de simulation OPNET Modeler. Les nœuds utilisent une interface radio de type 802.11 avec une portée radio de 300 mètres. Les nœuds se déplacent suivant le modèle de mobilité *random waypoint*. Nous supposons que la vitesse des nœuds est de 5m/s, que le réseau d'un degré de 9 est composé de 40 nœuds. Le trafic de données est modélisé par la transmission d'une rafale de 8 paquets consécutifs espacés de 0.25s. L'émission des rafales suit une loi exponentielle de moyenne 2s. Dans le cas du trafic montant (resp. descendant), la source (resp. destination) est choisie aléatoirement parmi les nœuds du réseau ad hoc. Le paramètre de la topologie virtuelle est  $k_B = 2$ .

Les performances de ce routage (noté cdcl sur les courbes) sont comparés à une solution classique d'interconnexion de réseaux ad hoc vers Internet : MEWLANA [31]. De façon synthétique, MEWLANA offre un routage proactif au sein du réseau ad hoc (DSDV) et une solution proche de

Mobile IP [69] pour le trafic montant et descendant. Tous les nœuds connaissent donc la localisation et une route vers la passerelle d'interconnexion vers Internet.

Les figures 3.10(a) et 3.10(b) comparent la capacité de notre proposition et de MEWLANA à résister à la mobilité. La vitesse des nœuds varie de 0m/s à 30m/s. Nous mesurons le délai et le taux de succès de paquets effectivement acheminés. Notre proposition mixant approche proactive et approche réactive offre ici un compromis pertinent en ayant une latence plus faible qu'un protocole réactif mais une charge de trafic moins élevée qu'une approche réactive. Le délai des deux solutions est proche que ce soit pour le trafic montant et le trafic descendant. En revanche, notre proposition offre un taux de livraison de paquets bien supérieur : cela est à mettre sur le compte de l'auto-organisation et de la persistance associée du backbone.

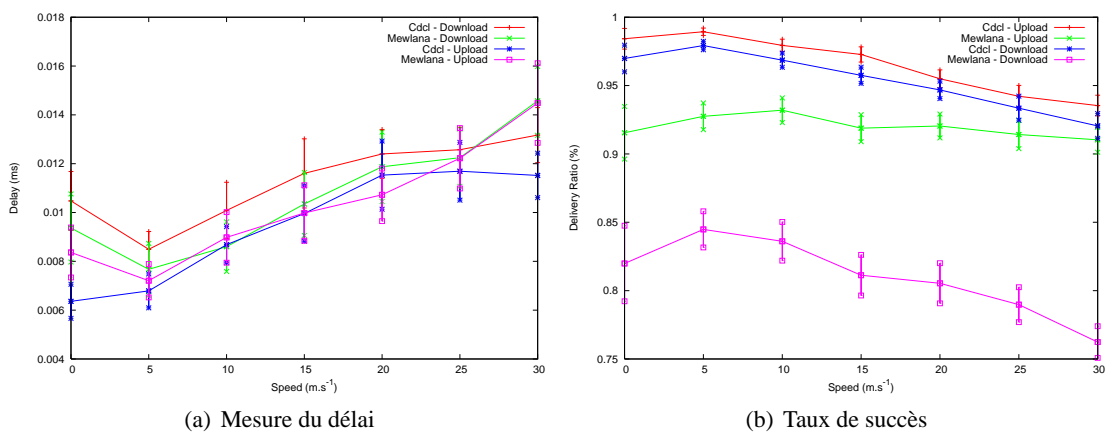


FIG. 3.10 – Impact de la mobilité sur les performances

Les figures 3.11(a) et 3.11(b) nous renseignent sur la capacité de passage à l'échelle de notre proposition et de MEWLANA. Le passage à l'échelle est vu sous deux angles complémentaires : l'impact de la cardinalité du réseau (à degré constant) puis l'impact de la charge de trafic. Lorsque le nombre de nœuds dans le réseau croît, la distance en nombre de sauts entre la passerelle d'interconnexion et un nœud du réseau ad hoc croît également : c'est la raison pour laquelle les délais augmentent. Lorsque la charge (*i.e.* le nombre de connexions simultanées) dans le réseau augmente (figure 3.11(b)), il faut noter la bonne résistance des deux protocoles même si les performances sont plus stables et plus intéressantes avec notre solution basée sur la notion d'auto-organisation. Une fois de plus, l'introduction de l'auto-organisation s'accompagne de l'introduction de davantage de stabilité dans le réseau, conduisant à de meilleures performances.

## 3.6 Routage

Les trois extensions précédentes nous ont permis de montrer que la topologie d'auto-organisation que nous avons conçu sous forme d'arbre pouvait également être un treillis et que des mécanismes simples comme la gestion de l'énergie et l'interconnexion de réseaux ad hoc avec Internet via des passerelles dédiées pouvaient se faire en tirant partie de l'auto-organisation.

Cependant, ces extensions ne sont pas au cœur de la problématique réseau qui demeure le routage. Dans cette section, nous allons étudier comment un routage dans un réseau ad hoc peut tirer partie d'une auto-organisation. Les réseaux ad hoc *à plat* subissent la mobilité et l'hétérogénéité

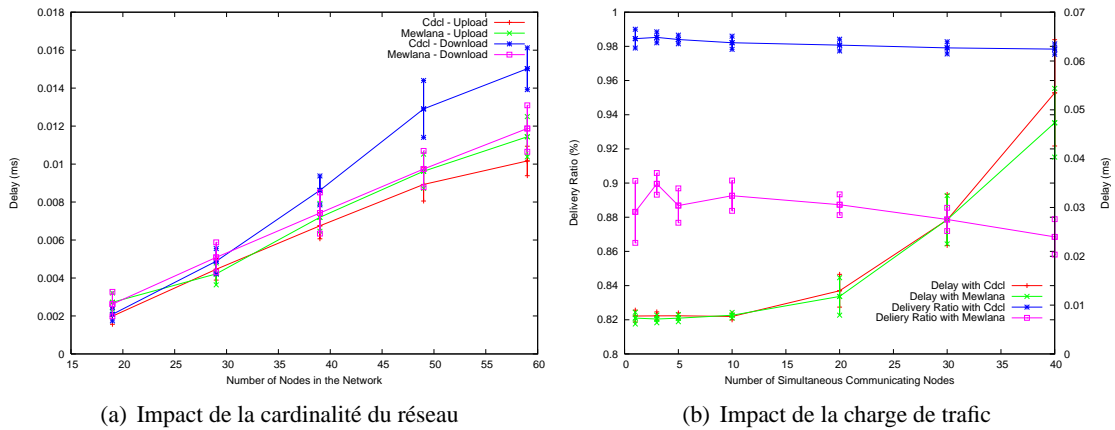


FIG. 3.11 – Passage à l'échelle

alors que l'auto-organisation introduit de la stabilité, de la persistance, de la robustesse. Autant d'éléments qui font qu'un routage basé sur une auto-organisation doit exhiber de meilleures performances. Finalement la question à laquelle nous allons essayer de répondre est : pour router un paquet, est-il préférable de router directement sur le réseau à plat ou est-il préférable de structurer, d'organiser le réseau en premier lieu ?

### 3.6.1 Routage local aux clusters

Le routage (*VSR - Virtual Structure Routing* [75, 74]) proposé doit tirer partie de la structure d'auto-organisation introduite. Quels sont les points clés ? Une structure hiérarchique de clusters interconnectés par un backbone virtuel. Rappelons que pour l'auto-organisation, chaque nœud connaît son voisinage sur  $k_B$  sauts pour construire le backbone et sur  $k_C$  sauts ( $k_C > k_B$ ) pour construire les clusters. Autrement dit nous pouvons introduire, sans créer un surcoût important, un routage pro-actif à l'intérieur des clusters. Ce point nous semble important : certes le coût en paquets de contrôle peut sembler important pour déployer l'auto-organisation mais la mise en place d'un protocole de routage pro-actif au sein des clusters apparaît la plupart du temps comme *gratuit* : chaque nœud peut donc calculer un plus court chemin vers les membres du cluster sans rajouter de trafic de contrôle. Pour le routage inter-clusters, nous proposons d'utiliser le backbone virtuel pour acheminer les paquets de découvertes de routes (*route\_request*) et de confirmation de routes (*route\_reply*). En revanche, comme le voisinage radio est dynamique du à la mobilité des nœuds et au lien radio versatile, nous proposons de construire des routes non pas comme une suite d'identifiants de nœuds mais comme une suite d'identifiants de clusters. De cette façon, nous allons tirer parti de la stabilité et de la persistance des clusters.

### 3.6.2 Routage inter-clusters

Nous proposons l'utilisation d'un protocole de routage réactif entre les clusters : un nœud source construit donc une route à la demande. Comme mentionné précédemment, la route est définie par la topologie en clusters, plus stable. Le mécanisme de découverte de routes utilise le backbone pour optimiser la diffusion de la requête.

**Découverte de la topologie en clusters** La route étant définie sous la forme d'une suite d'identifiants de clusters à suivre, un nœud doit connaître les clusters adjacents et une route pour les atteindre. Nous rajoutons donc cette informations dans les paquets `hello`. En plus de son identifiant, de son poids, de son état et de l'identifiant de son leader de cluster, un nœud va donc diffuser les identifiants des clusters voisins. Ce nœud pourra de servir passerelle vers les clusters annoncés. La route pour atteindre une passerelle est calculée via l'utilisation du protocole pro-actif. Cet accroissement en taille est acceptable : en transmission radio, le nombre de paquets a un impact plus important sur les performances que la taille des paquets.

**Découverte de routes** Plusieurs cas peuvent se produire lorsqu'un nœud  $S$  souhaite envoyer un paquet de données à un nœud  $D$  :

- $D$  est à au plus  $k_B$  sauts de  $S$  ou  $S$  et  $D$  ont le même leader de cluster.  $D$  est donc dans la table de voisinage de  $S$  qui peut exécuter le routage pro-actif introduit précédemment pour atteindre  $D$  ;
- $D$  est dans la table de routage de  $S$ .  $S$  a donc une route de clusters pour atteindre  $D$ . Il exécute l'algorithme de routage inter-cluster ;
- Sinon,  $S$  initie une découverte de routes.

Lors de la découverte de routes c'est le dominant de  $S$  qui propage la demande de construction de routes : il apparaît comme un proxy.  $S$  génère un `route_request` dans lequel il inscrit l'adresse de son leader de cluster dans la liste des leaders de cluster du paquet. Puis il l'envoie aux autres dominants du backbone. Les dominés ne participent donc pas à la découverte de routes. Chaque dominant recevant un `route_request` relaie le paquet s'il ne connaît pas  $D$  et si c'est la première copie reçue du paquet. Avant de le relayer, le dominant inscrit son identifiant de cluster s'il est différent du précédent identifiant de cluster contenu dans l'en-tête du paquet. Si  $D$  est dans la table de voisinage du dominant, alors celui-ci génère un `route_reply` contenant la route des clusters contenue dans le `route_request` en complétant avec l'adresse de son identifiant de cluster et l'identifiant de cluster de  $D$  s'ils n'y sont pas déjà présents. La `route_reply` contient donc finalement la route de clusters à suivre de  $D$  à  $S$ . Le dominant envoie la `route_reply` vers  $S$  en exécutant l'algorithme de routage inter-cluster. Les `route_request` ne sont relayés que par les membres de la dorsale, diminuant le coût de la découverte de routes.

### 3.6.3 Routage : vue d'ensemble

Seuls les paquets de données et les `route_reply` utilisent le routage inter-clusters. La route de clusters à suivre est contenue dans l'en-tête du paquet. Avant de relayer un `route_reply`, un nœud peut mettre en cache la route apprise vers  $S$  et vers  $D$ , afin de réduire le nombre de découvertes de routes générées plus tard. Si la destination finale est dans la table de voisinage, alors le nœud  $N_1$  envoie directement le paquet grâce au routage intra-cluster. Sinon, il cherche le premier cluster connu  $G$ , le plus proche de la destination :

- Un 1-voisin  $N_2$  possède le leader de cluster  $G$ .  $N_2$  est le prochain saut ;
- Un 1-voisin  $N_2$  est passerelle pour  $G$ .  $N_2$  est le prochain saut ;
- Un nœud  $N_2$  est la plus proche passerelle vers  $G$  du cluster de  $N_1$ .  $N_1$  exécute le routage intra-cluster pour atteindre  $N_2$ .  $N_1$  et  $N_2$  sont dans la même cluster, ils ont donc une même vue locale. Ainsi, ils prendront des décisions de routage cohérentes.

Il est important de préciser que ce routage ne crée pas de boucle : le paquet se rapproche à chaque fois d'un saut de la destination. Cependant, il est possible d'avoir des incohérences dans

les tables de voisinages, créées de façon distribuée [82]. Pour éviter les boucles, un paquet avec le même triplet adresses source/destination/identifiant déjà relayé est supprimé. Le calcul de la route pour traverser un cluster est fait dynamiquement. Un nœud choisit toujours de relayer le paquet vers le premier cluster connu. Il doit donc mettre à jour la route du `route_reply` s'il la modifie, de telle manière que la source bénéficie de ce calcul de route dynamique. Pour un paquet de données, une telle modification n'est pas requise puisque la source ne recevra pas la modification. Si cette solution de routage tire partie de l'auto-organisation, il est en revanche clair que la longueur des routes n'est pas optimale. Toutefois, la longueur de la route est limitée en relayant le paquet au cluster connu le plus proche de la destination. Un tel calcul dynamique de route est très robuste : les paquets arrivent à destination, même si de nombreux nœuds individuels se déplacent puisque la mobilité des nœuds est masquée par la stabilité des clusters : il suffit que la route de clusters soit valide pour que le paquet soit acheminé.

**Réparation de routes** Pour optimiser le taux de livraison, un simple mécanisme d'acquittement de paquets est utilisé. Un nœud ayant envoyé un paquet de données et n'ayant reçu aucun acquittement au bout d'un certain temps va retransmettre le paquet. Un acquittement passif en utilisant une coopération avec la couche MAC de IEEE 802.11 est également possible, comme DSR [45]. Dans ce cas là, aucun overhead n'est induit. Si un paquet n'arrive pas à être transmis sur un lien, nous proposons, comme CBRP [44], un mécanisme de réparation locale de route. Un tel nœud ré exécute l'algorithme de routage, en interdisant simplement comme prochain saut le nœud fautif. Une telle reconstruction permet de limiter l'impact du délai de convergence des tables de voisinage, améliorant le taux de livraison, mais pénalisant le délai de bout en bout.

### 3.6.4 Performances du routage basé sur une auto-organisation

Les performances du protocole de routage VSR sont étudiées en utilisant OPNET Modeler. Nous considérons une couche MAC 802.11b et une portée radio de 300. Les nœuds suivent un modèle de mobilité du type *random waypoint*. Par défaut, nous considérons 40 nœuds, un degré de 10, une vitesse de  $5\text{m}\cdot\text{s}^{-1}$  ainsi que 4 flux de données simultanés dans le réseau. Le trafic de données est modélisé par la transmission de 20 paquets de 128 octets en moyenne, envoyés tous les 0.25 secondes. Le temps inter arrivée des flux suit une loi exponentielle de moyenne 5 secondes. Pour la topologie virtuelle, nous avons retenu les configurations suivantes :  $(k_B, k_C) = (1, 2)$  et  $(k_B, k_C) = (2, 3)$ . Comme dans [44], nous avons fixé  $timeout_{RREQ}$  à 0,5s. Nous autorisons au maximum une réparation locale de route par nœud.

Les performances de VSR sont évaluées en fonction de la mobilité, du nombre de nœuds et de la charge du réseau. Nous nous intéressons principalement au délai de bout en bout, au taux de livraison, à la longueur de la route, et l'overhead. Nous comparons les performances de VSR à celles d'un routage pro-actif (OLSR), d'un routage réactif (DSR) et d'un routage basé sur clusters (CBRP).

**Passage à l'échelle** Tout d'abord nous étudions l'impact de la cardinalité du réseau sur le comportement de ces protocoles de routage. Nous travaillons ici à densité constante. La figure 3.12 nous donne, en fonction du nombre de nœuds dans le réseau, le délai, le taux de paquets correctement reçus et la longueur moyenne des routes (en nombre de sauts). Le délai de bout-en-bout d'AODV et CBRP augmente de façon importante lorsque le nombre de nœuds croît dans le

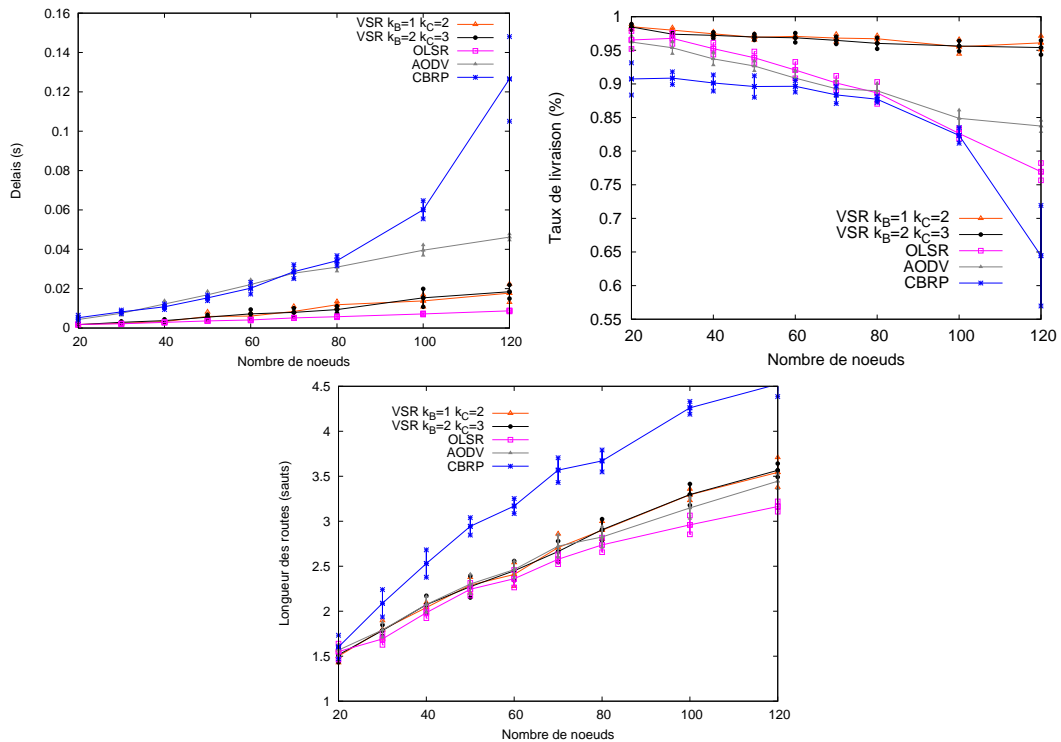


FIG. 3.12 – Passage à l'échelle

réseau : c'est la conséquence d'une approche réactive où le temps de découverte des routes augmente. En revanche, OLSR tire partie de sa connaissance complète de la topologie : les routes sont toutes connues, sans délai. VSR, quelque soit la configuration retenue, présente un délai stable, subissant assez peu l'augmentation de la taille du réseau. L'existence du backbone structure le réseau et optimise la découverte de routes.

Le taux de livraison de paquets souligne la pertinence de l'approche d'un routage basé sur une auto-organisation : la topologie virtuelle étant plus stable que le voisinage radio, les routes sont plus stables également, favorisant un acheminement plus robuste. Ainsi, VSR fournit le meilleur taux de livraison tout en étant moins sensible que les autres protocoles à l'augmentation de la cardinalité du réseau. CBRP souffre de routes non optimales. La construction des routes pour AODV subit trop de collisions au niveau des `route_request` et ne supporte pas l'augmentation de la taille du réseau. OLSR conserve des performances importantes bien que l'inondation des paquets de `topology_control` conduisent à des collisions.

Enfin, en dehors de CBRP, la longueur des routes construites sont similaires. OLSR fournit bien entendu le chemin le plus court. CBRP subit la construction de routes suivant la topologie de clusters mais sans optimisation. Le comportement de VSR évolue peu en fonction des paramètres retenus.

**Influence de la charge** Nous nous intéressons ici au passage à l'échelle sous un autre angle : celui d'une augmentation de la charge de trafic dans le réseau. Pour une topologie physique de 40 nœuds, nous regardons l'évolution du délai, le taux de livraison et la latence de découverte de routes (figure 3.13). OLSR et VSR représentent le délai le plus faible, ainsi que la plus faible

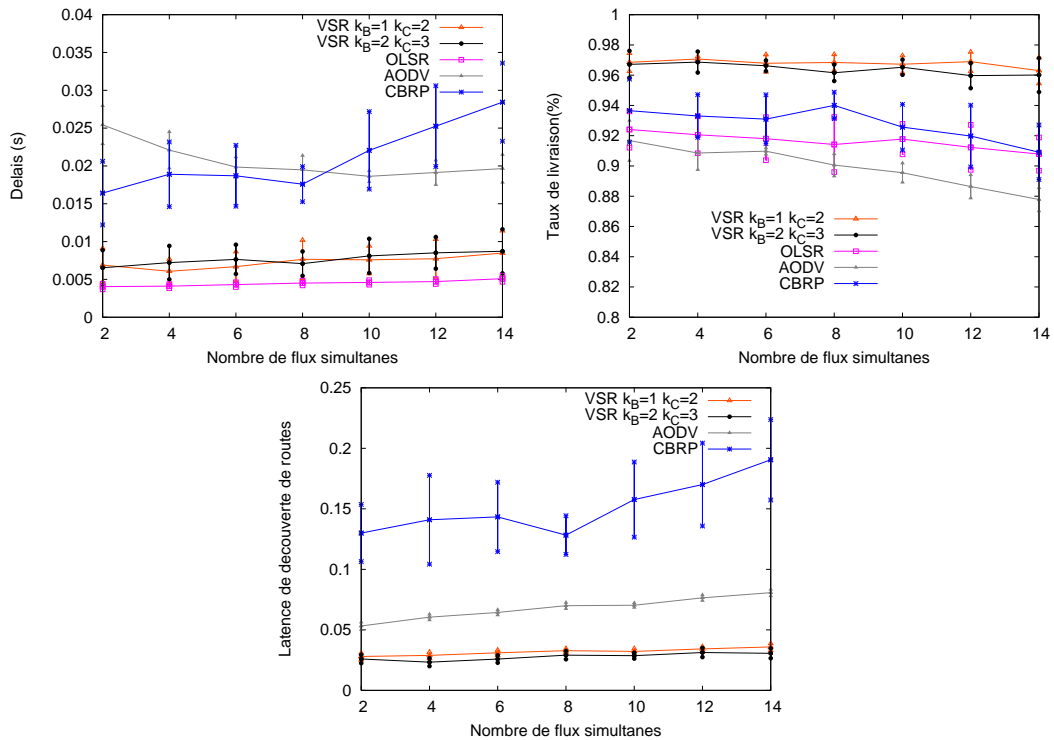


FIG. 3.13 – Augmentation de la charge de trafic

latence, relativement insensible à la charge de trafic. Le délai de VSR est plus faible dans la configuration ( $k_B = 2, k_C = 3$ ) que dans ( $k_B = 1, k_C = 2$ ) : cela provient d'une proportion croissante de routes intra-clusters plus importante et donc de moins de découvertes de route. En revanche, la latence de découverte de routes est plus longue dans ( $k_B = 2, k_C = 3$ ) que ( $k_B = 1, k_C = 2$ ). AODV et CBRP souffrent de la montée en charge car plus de découvertes de routes sont initiées. Le taux de succès souligne l'intérêt de l'approche routage basé sur une auto-organisation : les routes étant plus stables, le taux de succès demeure stable également et insensible à la charge. Notons, même si les résultats ne sont pas présentés ici, que les performances de VSR dans le cas de forte mobilité sont également plus intéressantes que celles obtenus avec les protocoles de routage classique car l'organisation en zones de dominances puis clusters masque une partie de la dynamique des nœuds et du réseau.

**Surcoût protocolaire** La figure 3.14 fournit l'impact de la charge sur le coût de contrôle des différents protocoles de routage. La charge est vue sous l'angle complémentaire d'une augmentation du nombre de flux dans le réseau et d'une augmentation de la cardinalité du réseau. Dans le cas d'OLSR, ce surcoût protocolaire est associé aux paquets `hello` et `topology_control` ; dans le cas d'AODV ce surcoût est lié aux paquets `route_request` et `route_reply` ; dans le cas CBRP ce surcoût est lié au maintiens des clusters et des routes. Dans VSR, ce coût est lié à la construction et au maintiens de la topologie virtuelle ainsi qu'au contrôle associé au routage. Dans le cas d'une augmentation de la charge de trafic, le surcoût protocolaire d'OLSR et VSR reste stable alors que CBRP et AODV subissent cette augmentation de trafic qui exigent plus de découvertes de routes. Dans le cas d'une augmentation de la cardinalité du réseau, AODV ap-

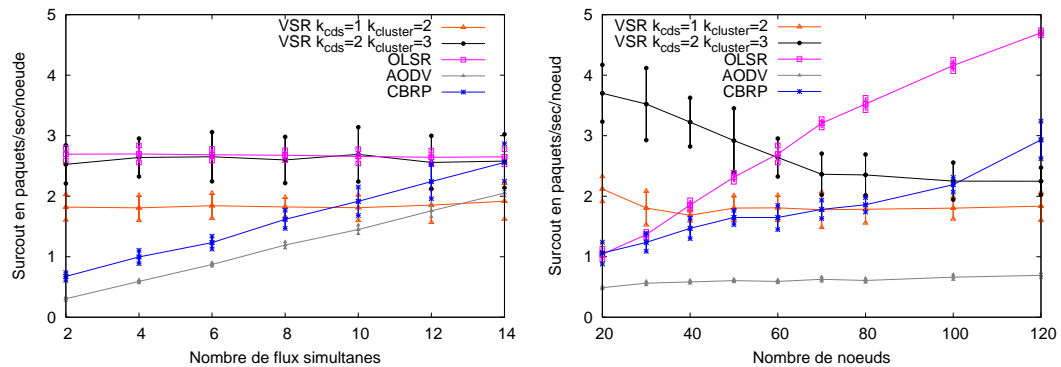


FIG. 3.14 – Surcoût protocolaire des protocoles de routage

paraît comme très résistant au passage à l'échelle tandis que CBRP subit toujours le processus de découverte de routes basé sur les clusters. VSR offre le meilleur passage à l'échelle alors que le coût de contrôle d'OLSR croît de façon importante à cause de la diffusion de davantage de paquets hello et `topology_control`.

### 3.7 Synthèse

Ce chapitre est notre première contribution à l'auto-organisation de réseaux radio multi-sauts. L'étude des réseaux mobiles ad hoc offrent un ensemble de challenges clefs : routage, localisation, interconnexion avec un réseau sans fil le tout avec des contraintes de topologie dynamique due à la mobilité des nœuds et d'hétérogénéité. Au lieu de répondre à ces challenges en proposant trois protocoles différents, nous avons choisi de mettre en place une auto-organisation dont le but est de structurer, fédérer le réseau tout en *masquant* sa dynamique à l'aide d'une topologie logique. Nous avons proposé l'utilisation d'un backbone virtuel sur lequel des clusters sont déployés. Le backbone apparaît comme une ressource fédératrice pour les nœuds du réseau et comme une extension naturelle à réseau filaire. Les clusters sont associés à un voisinage radio pour masquer une partie de la mobilité des nœuds. Nous avons donc proposé des protocoles distribués de construction et de maintenance. Ces protocoles ont été évalués dans le cadre d'hypothèses de couches physique et MAC réalistes. Des preuves de cardinalité et d'auto-stabilisation ont également été faites pour justifier leurs adéquations aux objectifs des contraintes ad hoc (les preuves n'ont pas été fournies dans ce manuscrit). Les propriétés de cette topologie virtuelle sont nombreuses : robustesse, persistance, résistance à la mobilité, passage à l'échelle. Nous nous sommes alors focalisés sur un objectif clef : que peut apporter une telle approche d'auto-organisation pour une problématique de protocoles de niveau 3 dans un contexte ad hoc ? Après la présentation d'un simple, mais efficace, mécanisme d'économie d'énergie visant à augmenter la durée de vie du réseau, nous avons étudié le passage d'un réseau ad hoc à un réseau hybride. Nous avons vu que la construction d'une route par défaut vers Internet mais aussi que la localisation des nœuds mobiles pouvaient se faire de façon pertinente grâce à la topologie virtuelle. Enfin, nous nous sommes intéressés au routage en proposant un protocole de routage basé sur cette auto-organisation. Ce routage, baptisé VSR, tire partie du backbone pour acheminer efficacement le trafic de contrôle et tire partie des clusters pour introduire de la stabilité dans les routes. Les performances de VSR vis-à-vis des approches classiques (proactif, réactif, etc.) montrent que le routage proposé apparaît comme un compromis

proactif/réactif efficace.

# Bibliographie

- [1] F. Theoleyre and F. Valois. VSR : a routing protocol based on a structure of self-organization. *Studia Informatica - Special Issue on Wireless Ad Hoc and Sensor Networks*, 2007. In press.
- [2] F. Theoleyre and F. Valois. A self-organization structure for hybrid networks. *Ad hoc journal, Elsevier Ed.*, 2007. In press.
- [3] F. Theoleyre and F. Valois. Structure virtuelle pour une auto-organisation dans les réseaux ad hoc et hybrides. *Annales des Télécommunications*, volume 62, n. 1-2, January 2007.
- [4] F. Theoleyre and F. Valois. Indoor experiments of self-organization and localization protocols for hybrid networks. In *2nd IEEE Workshop on advanced EXPERIMENTAL activities ON WIRELESS networks & systems (ExponWireless)*, Helsinki, Finland, June 2007.
- [5] R. Thout, F. Theoleyre, and F. Valois. New metrics to evaluate mobility models properties. In *4th International Symposium on Wireless Pervasive Computing (ISWPC)*, San Juan, Porto Rico, February 2007.
- [6] H. Rivano, F. Theoleyre, and Valois F. Capacity evaluation framework and validation of self-organized routing schemes. In *3th International Workshop on Wireless Ad Hoc and Sensor Networks (IWVAN)*, New-York, USA, June 2006. IEEE.
- [7] F. Theoleyre and Valois F. Performance evaluation of the routing protocols in manet : Classical versus self-organized approaches. In *5th International Conferences on Networking (Networking)*, Coimbra, Portugal, May 2006. IFIP, Vol. LNCS 3976/2006.
- [8] F. Theoleyre and Valois F. About the self-stabilization of a virtual topology for self-organization in ad hoc networks. In *7th International Symposium on Self-Stabilizing Systems (SSS)*, Barcelona, Spain, October 2005.
- [9] F. Theoleyre and Valois F. Mobility management in multihops wireless access networks. In *10th International Conference on Personal Wireless Communications (PWC)*, Colmar, France, August 2005. IFIP.
- [10] F. Theoleyre and Valois F. Virtual structure routing in ad hoc networks. In *International Conference on Communication (ICC)*, Seoul, Korea, May 2005. IEEE.
- [11] F. Theoleyre and F. Valois. A virtual structure for hybrid network. In *IEEE Wireless Communications and Networking Conference (WCNC)*, Atlanta, USA, March 2004. IEEE.
- [12] F. Theoleyre and F. Valois. Robustness and reliability for virtual topologies in wireless multihops access networks. In *Third Mediterranean Ad Hoc Networking Workshop (Med-HocNet'04)*, Bordum, Turkey, June 2004. IFIP.
- [13] H. Rivano, F. Theoleyre, and F. Valois. Influence de l'auto-organisation sur la capacité des réseaux ad hoc. In *AlgoTel 2005 : 7ème Rencontres Françaises sur les Aspects Algorithmiques des Télécommunications*, Prequ'île de Giens, France, May 2005.

- [14] F. Theoleyre and F. Valois. Routage hybride sur structure virtuelle dans les réseaux mobiles ad-hoc. In *CFIP'2005 : Colloque francophone sur l'ingénierie des protocoles*, Bordeaux, France, March 2005.
- [15] F. Theoleyre and F. Valois. Topologie virtuelle pour une organisation des réseaux hybrides multi-sauts. In *6ème Journées Doctorales Informatique et Réseau (JDIR 04)*, Lannion, France, November 2004.
- [16] H. Rivano, F. Theoleyre, and Valois F. About the capacity of flat and self-organized ad hoc and hybrid networks. Research Report RR-5977, INRIA, October 2006.
- [17] F. Theoleyre and F. Valois. About the self-stabilization of a virtual topology for self-organization in ad hoc networks. Research Report RR-55650, INRIA, August 2005.
- [18] F. Theoleyre and F. Valois. Localization and routing in multihops wireless access networks. Research Report RR-5461, INRIA, January 2005.
- [19] F. Theoleyre and F. Valois. Topologie virtuelle pour réseaux hybrides. Research Report RR-5035, INRIA, November 2003.

## Chapitre 4

# Auto-organisation de réseaux de capteurs

### 4.1 Motivations, objectifs

Le travail présenté précédemment avait pour objectif de montrer comment auto-organiser un réseau mobile ad hoc et comment la notion d'auto-organisation pouvait impacter le comportement des protocoles de communications. Les idées et concepts développés peuvent être appliqués au cas particulier des réseaux de capteurs (WSN<sup>1</sup>). Toutefois, les contraintes et les propriétés inhérentes aux réseaux de capteurs font que cette solution n'est pas applicable :

- les motifs de communications entre réseaux ad hoc et réseaux de capteurs sont différents. Alors que dans un réseau MANET les communications sont typiquement  $any \rightarrow all$  ou  $any \rightarrow any$ , les communications dans un réseau WSN sont plutôt du type  $1(sink) \rightarrow any$  ou  $1(sink) \rightarrow all$  ou  $any \rightarrow 1(sink)$ . Si les protocoles de routage, d'inondation, de dissémination doivent en tenir compte, il en va de même pour l'auto-organisation ;
- la cardinalité d'un réseau WSN est bien plus importante qu'un réseau ad hoc, l'exigence de *scalabilité* n'est pas une vaine exigence et ne saurait se limiter à considérer des topologies de moins de cent nœuds ;
- la dynamique de la topologie d'un réseau de capteurs ne résulte pas (ou peu) de la mobilité des nœuds mais de mécanismes d'économie d'énergie qui éteignent des nœuds pour augmenter leur durée de vie ainsi que de processus de *densification* du réseau et de mort des nœuds après expiration de leur batterie ;
- un capteur est un nœud communicant en radio et à faible capacité (mémoire, CPU, énergie). La complexité des algorithmes utilisée doit donc être faible.

Ainsi, l'utilisation périodique de paquets `hello` doit être proscrite afin d'économiser l'utilisation du médium radio, principal consommateur d'énergie lors d'émission et de réception [67].

Dans ce chapitre, nous allons étudier comment développer un protocole d'auto-organisation adapté aux contraintes des réseaux de capteurs puis nous envisagerons son apport dans les protocoles de communications et le domaine applicatif. Auparavant, et parce que les réseaux de capteurs ont la particularité d'être orientés application, nous faisons les hypothèses suivantes :

- tous les capteurs ont les mêmes capacités en termes de mémoire, puissance de calcul, communication et d'énergie. Seul le puits peut avoir des capacités différentes mais nous ne

---

<sup>1</sup>Wireless Sensor Networks [6]

tenons pas compte de ce nœud particulier dans l'élaboration de notre proposition afin de garantir plus de flexibilité ;

- l'optimisation de l'énergie doit être présente systématiquement pour améliorer la durée de vie du réseau<sup>2</sup> ;
- aucune information de localisation géographique est connue de la part des nœuds et aucune information globale n'est requise.

Au niveau des nœuds et de la topologie du réseau, nous supposons :

- les nœuds sont déployés aléatoirement sur une zone de surveillance 2D ;
- les nœuds se réveillent aléatoirement au cours du déploiement. Ces deux dernières hypothèses impliquent que le réseau se déploie progressivement au cours du temps ;
- les liens radios sont symétriques : si  $v$  est un voisin de  $u$  alors  $u$  est un voisin de  $v$ . De plus les portées radios sont homogènes ;
- il n'y a aucune *mobilité* considérée. Une fois le réseau déployé, les seules variations au sein de la topologie sont le résultat de la mort de certains nœuds, de processus d'endormissement et du re-déploiement de nœuds.

Nous allons débiter par l'introduction de la structure d'auto-organisation adaptée aux contraintes énoncées précédemment (section 4.2). Puis dans la section 4.3 nous nous intéresserons à une solution d'auto-configuration d'identifiants de nœuds dans le cas où les capteurs ne sont pas identifiés initialement. Basé sur cette architecture de communications pour réseaux de capteurs, nous nous intéresserons aux applications en étudiant successivement : comment prendre en charge les puits multiples et la mobilité des puits (section 4.4) ? Comment disséminer des données efficacement et toujours contraint en énergie (section 4.4) ? Comment proposer une solution d'agrégation spatiale et temporelle (4.5) ?

## 4.2 Auto-organisation pour réseaux de capteurs

Si l'objectif d'une auto-organisation appliquée aux réseaux de capteurs reste d'organiser, de structurer le réseau, il est ici nécessaire de tenir compte du déploiement progressif du réseau. L'auto-organisation proposée va tenir compte de l'existence d'un voisinage déjà organisé pour diminuer la latence de mise en place de l'auto-organisation et améliorer la cardinalité de la structure logique. Étant donné que nous considérons l'auto-organisation comme une brique fondamentale pour des protocoles de communications efficaces, nous avons nommé ce protocole : LEGOS (*Low-Energy self-organisation scheme for wireless sensor networks*).

### 4.2.1 Préambule

L'objectif est de fournir un protocole d'auto-organisation simple, efficace en énergie, adapté aux topologies large échelle permettant de structurer le réseau et qui permet de s'adapter aux changements dans le voisinage. Les exigences que nous avons sur cette structure sont :

- seules des interactions locales doivent être utilisées pour faire émerger un comportement global et réagir aux changements dans le voisinage ;
- dans le but d'économiser de l'énergie, l'émission de paquets périodiques `hello` doit être évité ;

<sup>2</sup>Pour la notion de *durée de vie du réseau*, nous considérons la durée de vie du premier nœud qui meurt dans le réseau.

- le protocole proposé doit être efficace en terme de complexité et de puissance de calcul requise pour être adapté aux ressources limitées des capteurs.

Notons que nous n'avons ici aucune hypothèse sur la distribution des nœuds sur la zone de surveillance : les nœuds sont déployés aléatoirement. De plus, nous n'avons aucune hypothèse en terme de synchronisation : les nœuds exécutent LEGOS indépendamment sans être nécessairement synchronisés.

#### 4.2.2 LEGOS : protocole d'auto-organisation pour réseaux de capteurs

LEGOS est un protocole d'auto-organisation asynchrone orienté événement : en fonction des modifications du voisinage, des règles locales sont appliquées. La méthode de découverte du voisinage est fonction de l'existence ou non d'une auto-organisation déjà présente. Seul un sous-ensemble des nœuds du réseau émet périodiquement un paquet de type `hello` pour annoncer un voisinage organisé ; nous verrons que la cardinalité de cet ensemble est assez faible.

Une des différences principales avec les autres protocoles que l'on pourrait associer à l'auto-organisation qu'ils soient de type backbone (RNG [77], LMST [53], CDS [80], ...) ou de type clusters [12] est que LEGOS ne nécessite ni une reconstruction périodique ni une construction par vague : le gain en robustesse, latence, efficacité est alors important. Dans LEGOS, les nœuds peuvent posséder trois rôles : `leader`, `gateway` ou `member`. Un `member` est un nœud passif sans fonction particulière en dehors du comportement applicatif et qui est rattaché à un seul `leader` nécessairement présent dans son voisinage à 1 saut. Un `leader` est en charge des communications à un saut. Un `gateway` permet l'interconnexion de plusieurs `leaders`. L'ensemble des `leaders` et des `gateways` forment un ensemble connecté dominant (voir les figures 4.1(a) et 4.1(b)). Seuls les `leaders` émettent périodiquement des paquets `Leader_Broadcast_Msg` dans le voisinage à 1 saut pour informer de l'existence d'une auto-organisation stable.

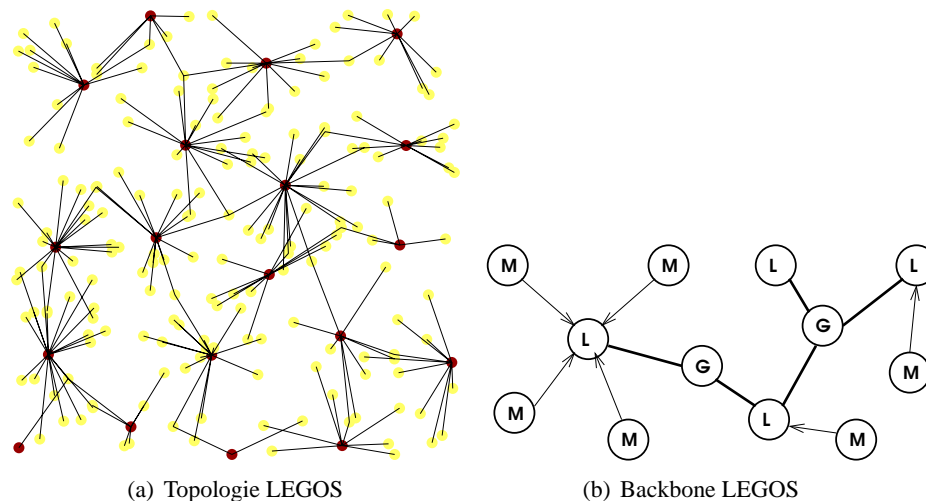


FIG. 4.1 – Topologie d'auto-organisation générée par LEGOS

Quatre événements locaux relatifs à la construction et à la maintenance de LEGOS sont identifiés : l'arrivée de nouveaux nœuds, le départ de nœuds, la gestion de partitions et une procédure de ré-organisation locale. Les deux premiers événements correspondent au processus de déploiement du réseau, au processus de mort et/ou mobilité des nœuds. À cause du caractère

aléatoire de l'arrivée des nœuds, plusieurs partitions peuvent apparaître : il est donc nécessaire de proposer des mécanismes de fusion ainsi que des mécanismes de scission dans le cas de la mort des nœuds. Enfin, et parce que l'énergie est au cœur de notre problématique, nous devons optimiser sa consommation. En particulier dès qu'un nœud devient *leader* il prend en charge son voisinage, dépensant une grande partie de son énergie : il est donc nécessaire de faire tourner le rôle de *leader* pour augmenter la durée de vie du réseau. C'est le but du quatrième événement.

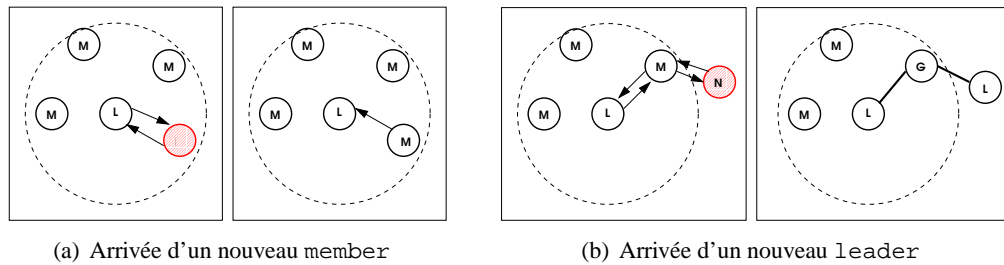


FIG. 4.2 – Illustration des procédures d'arrivées de nouveaux nœuds

**Arrivées de nœuds** Un nouveau nœud  $N$  arrivant dans le réseau suivra les trois étapes suivantes pour intégrer la structure d'auto-organisation (voir figures 4.2(a) et 4.2(b)) :

- $N$  écoute le médium pour détecter un `Leader_Broadcast_Msg`. Si un *leader* est détecté grâce à ce message,  $N$  devient *member* et se rattache à ce *leader* (procédure d'auto-organisation à 1-saut).
- S'il n'y a pas de *leader* dans le voisinage,  $N$  initie une découverte de voisinage en émettant un `Member_Sollicitation_Msg`. Si un nœud  $M$  de l'auto-organisation entend ce message, ce ne peut être qu'un *member*. Il répond alors avec `Member_Advertisement_Msg` :  $N$  devient alors *leader* et  $M$  devient *gateway* (procédure d'auto-organisation à 2-sauts).
- Si les deux premières procédures sont infructueuses,  $N$  va se considérer comme le premier nœud d'une partition : il va se déclarer *leader*, générer un numéro aléatoire de partition et émettre périodiquement un paquet `Leader_Broadcast_Msg` qui inclut cet identifiant de partition.

Notons qu'un nouveau nœud tire pleinement profit du voisinage déjà organisé : nous verrons que la cardinalité de la topologie logique LEGOS est alors insensible à une augmentation de la densité.

**Départs de nœuds** Le départ d'un *leader* n'a pas le même impact que le départ d'un *gateway* ou d'un *member*. Nous proposons les procédures suivantes :

- le départ d'un *member* n'a pas besoin d'être détecté. Étant donné que ce type de nœud ne joue qu'un rôle minimal dans l'auto-organisation, cette procédure est tolérable. Cependant tous les paquets applicatifs qui devaient être transmis vers ce *member* seront perdus.
- Le départ d'un *gateway* doit être détecté : soit il informe explicitement chacun des *leaders* qu'il connecte de son prochain départ, soit -en l'absence de transmission de données- un mécanisme de détection explicite des *gateways* par les *leaders* est nécessaire. Pour faciliter la reconstruction locale, chaque *leader* doit connaître le(s) identifiant(s) des *leader(s)* connectés via les *gateways*. Lors du départ d'un *gateway*,



**Gestion de partitions** À cause du déploiement progressif du réseau et de l'instant de démarrage aléatoire des nœuds, les mécanismes précédents entraînent l'apparition de plusieurs partitions dans le réseau : des mécanismes de détection et de fusion de partitions est donc requises. De plus, comme les nœuds du réseau ont une durée de vie limitée, il faut proposer un mécanisme de scission de partition.

- *Scission de partition* : Le point clef dans la scission de partitions est de conserver les règles du backbone LEGOS et de maintenir la structure virtuelle de communication dans chaque partition. Si un `member` perd son `leader`, il procédera aux mêmes actions que s'il était un nouveau nœud pour se rattacher à un autre `leader`. Les nœuds qui étaient `leaders` avant scission restent `leaders`. Les seuls nœuds qui demandent une ré-organisation locale sont les `gateways` qui peuvent re-basculer dans l'état `member` s'ils ne sont plus connectés qu'à un seul `leader`. Après une scission, si un `leader` a perdu tous ses `gateways`, il génère un nouvel identifiant unique de partition et informe tous ses voisins du nouvel identifiant utilisé. Ce nouvel identifiant est nécessaire pour identifier de façon unique les partitions.
- *Détection de partitions* : Lors du déploiement du réseau, beaucoup de partitions peuvent être générées. A chaque fois qu'un nouveau nœud se retrouve isolé, il devient `leader` et génère un nouvel identifiant de partition. Plus le réseau se densifie et plus des partitions sont amenées à fusionner. Nous laissons l'initiative de la détection de plusieurs partitions aux nouveaux nœuds arrivant en se basant sur l'écoute dans le voisinage à 1 saut des paquets `Leader_Broadcast_Msg` émis par les `leaders` qui véhiculent l'identité des partitions ainsi que le nombre de `leaders` présent dans la partition.
- *Fusion de partitions* : Un nouveau nœud qui détecte plusieurs partitions informe chacun des `leaders` de son voisinage. Ce nouveau nœud s'intègre dans la partition ayant le plus de `leaders` puis devient `gateway`. Il informe ensuite les autres partitions du nouvel identifiant résultant de la fusion. Une inondation est alors nécessaire. Le coût de cette inondation est contrôlée par la faible cardinalité du backbone LEGOS comme nous le montrerons.

**Ré-organisation locale** Dans le but d'optimiser la durée de vie du réseau, nous proposons de faire tourner les rôles de `leader` et `gateway` suivant les propriétés de la topologie locale et une métrique locale à chacun des nœuds. Le but est de répartir la charge de communication entre les nœuds [71]. La métrique que nous utilisons est basée sur l'énergie résiduelle et sur les propriétés du voisinage. Cette métrique  $M$  n'est appliquée que sur les `leaders` et `gateways`.  $M$  est défini par :

- $M(u) = \frac{E_{residual}}{E_{max}} \cdot \frac{N_{member}+5}{N_{member}} \cdot \frac{N_{gateway}+1}{N_{gateway}}$  pour les `leaders`.
- $M(u) = \frac{E_{residual}}{E_{max}} \cdot \frac{N_{leader}}{N_{leader}+1}$  pour les `gateways`.

où  $E_{residual}$  représente l'énergie résiduelle pour le nœud,  $E_{max}$  l'énergie maximum disponible et  $N_x$  le nombre de nœuds de type  $x$  dans le voisinage. L'idée de cette métrique est de tenir compte à la fois de la capacité en énergie qu'un `leader` ou `gateway` a tenir son rôle mais aussi de la capacité à trouver un autre `leader` ou `gateway` dans le voisinage (densité locale de `leader` et/ou de `gateway`). Dès que  $M(u)$  atteint un seuil, le nœud (qu'il soit `leader` ou `gateway`) bascule dans l'état `member` après avoir diffusé, localement, une notification de ré-organisation locale. A la réception de cette notification, l'ensemble du voisinage à 1 saut de l'ancien `leader` bascule à nouveau dans l'état `member` pour une phase de découverte de voisinage, identification et/ou recherche de `leaders`. Tous les nœuds qui n'ont pu se rattacher à un

leader informe l'ancien leader qui sélectionne son (ses) successeur(s) à partir des informations locales qu'il possède.

### 4.2.3 Analyse et complexité

Avant de décrire les performances de LEGOS sous des hypothèses de couches radio et MAC réalistes, nous allons nous intéresser à quelques propriétés théoriques de LEGOS, essentiellement en termes de borne théorique supérieure de la cardinalité et complexité protocolaire. Nous supposons ici qu'un réseau WSN peut se modéliser sous la forme d'un graphe  $G$  en faisant abstraction des problématiques d'interférences, de délai de propagation et de contention. La portée radio est supposée égale à  $R$ .

**Cardinalité du backbone** L'objectif est ici de montrer que la cardinalité du backbone est bornée pour n'importe quel réseau  $G$ . Tout d'abord nous allons montrer que le nombre de leaders à deux sauts dans une configuration aléatoire est borné. Puis nous allons paver une zone de surveillance avec des motifs réguliers pour montrer que le nombre de motifs sur cette zone est également borné. Nous en déduisons l'existence d'une borne supérieure sur le nombre de leaders dans le réseau.

**Lemme 1** *L'ensemble des leaders forme un ensemble indépendant du graphe  $G$ .*

**Preuve 1** *Lorsqu'un nouveau nœud arrive dans le réseau, il devient leader si et seulement si il n'y pas de leaders dans son voisinage à 1 saut. Les leaders sont ainsi espacés de deux sauts. L'ensemble des leaders forment donc bien un ensemble indépendant de  $G$ .*

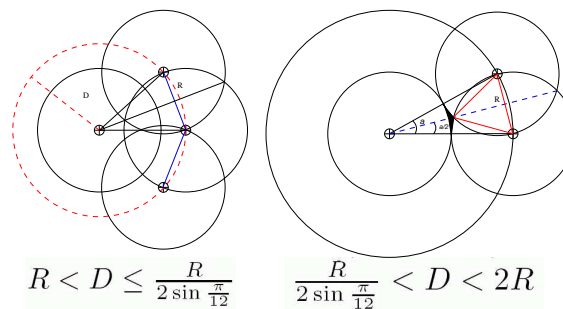


FIG. 4.5 – Le nombre de nœuds de  $S$  dans le voisinage de  $u$

**Lemme 2** *Soit  $S$  un ensemble indépendant de  $G$ , soit  $u$  un nœud quelconque de  $S$  et soit  $D$  une distance euclidienne comprise entre  $R$  et  $2R$ . Le nombre de nœuds (noté  $N_{local}$ ) de  $S$  qui sont à exactement deux sauts de  $u$  et à une distance  $d$ 'au plus  $D$  est :*

1. au plus  $\left\lceil \frac{\pi}{\arcsin(R/2D)} \right\rceil \quad \forall D \leq \frac{R}{2 \sin \pi/12}$ .
2. au plus 18 pour  $D > \frac{R}{2 \sin \pi/12}$ .

**Preuve 2** Les nœuds de  $S$  sont éloignés d'au moins  $R$ . D'après la figure 4.5, l'angle  $\alpha$  entre deux nœuds adjacents et voisins de  $u$  est au moins  $2 \arcsin(R/2D)$ . Le nombre maximum de nœuds de  $S$  qu'il est possible de placer sur un cercle de rayon  $D$  centré sur  $u$  est  $\lceil \frac{2\pi}{\alpha} \rceil$  pour  $D \leq \frac{R}{2 \sin \pi/12}$  puisque la distance entre l'intersection de deux cercles adjacents et  $u$  est nécessairement plus petite que  $R$ . La première partie du Lemme 2 est vraie.

Pour toutes les distances  $D$  plus grandes que  $\frac{R}{2 \sin \pi/12}$ , l'angle  $\alpha$  est plus grand que  $\pi/6$ . La figure 4.5 montre clairement que l'intersection entre les deux cercles de rayon  $R$  centré sur deux nœuds adjacents est hors de portée radio de  $u$ . Pour  $\frac{R}{\sin \pi/12} < D < 2R$ , le nombre maximum de nœuds qui peuvent être placés sur le cercle de rayon  $D$  est 12. La distance entre chaque zone d'ombre est alors inférieure à  $R$ . De plus, il est possible de placer 6 nœuds parmi les 12 zones d'ombre de telles façons que ces nœuds soit distant de deux sauts. Ainsi 18 nœuds de  $S$  peuvent être placés. Ce qui montre la deuxième partie du Lemme 2.

**Lemme 3** La cardinalité d'un ensemble indépendant couvrant une zone donnée est bornée.

**Preuve 3** Le lemme précédent montre que le nombre de nœuds indépendants dans un voisinage local à deux sauts est borné, ce qui est finalement proche du résultat de [7]. Une question demeure : quel est le pavage qui maximise le nombre de nœuds indépendants sur une zone donnée ? Le motif ayant le plus de nœuds indépendants (18 nœuds, dodécagone) ne conduit pas nécessairement à la borne supérieure. Nous devons chercher un pavage tel que le nombre de nœuds indépendants est maximal sur une zone de dimension  $a \times a$  avec la contrainte que la distance entre deux nœuds indépendants au sein du motif utilisé pour le pavage est supérieur à  $R$ . Pour un pavage donné, il est alors possible de calculer le nombre maximal de motifs nécessaires pour couvrir la zone en fonction de  $a$ ,  $D$  et  $R$ . Cette valeur multipliée par  $N_{local}$  (nombre maximal de nœuds indépendants dans un motif) permet d'exprimer  $N_{IS}$ , le nombre de nœuds indépendants sur une zone de service.  $N_{IS}$  est donné par :

$$\begin{cases} = \left\lceil \frac{a+2R}{(2D+R \cdot \cos \alpha)} \cdot \frac{a+2R}{2D+R \sin(\alpha/2)} \right\rceil \cdot \left\lceil \frac{2\pi}{\alpha} + 1 \right\rceil, R < D \leq \frac{R}{2 \sin \frac{\pi}{12}} \\ = \left\lceil \frac{a+2R}{(2D+R \cdot \cos \alpha)} \cdot \frac{a+2R}{2DR \sin(\alpha/2)} \right\rceil \cdot (18 + 1), \frac{R}{2 \sin \frac{\pi}{12}} < D < 2R \end{cases} \quad (4.1)$$

avec  $\alpha = 2 \arcsin(R/2D)$  et  $f(x) = \lceil x \rceil$  étant la fonction qui retourne l'entier le plus proche de  $x$ .

Pour la première expression, la dérivée étant toujours négative, la valeur de  $N_{IS}$  est maximale lorsque  $D = R$ . Pour la seconde expression, la dérivée est également toujours négative et la valeur de  $N_{IS}$  est maximale pour  $D = \frac{R}{2 \sin \frac{\pi}{12}}$ . La cardinalité de l'ensemble indépendant couvrant une zone donnée est donc bornée par  $\text{Max}(N_{IS}(D = R), N_{IS}(D = \frac{R}{2 \sin \frac{\pi}{12}}))$ . Finalement,  $N_{IS}(D = R)$  est la valeur maximale du nombre de nœuds indépendants. Elle est égale à :

$$7 \cdot \left\lceil \frac{a+2R}{(2+\sqrt{3}/2)R} \cdot \frac{a+2R}{2.5R} \right\rceil$$

**Lemme 4** Le nombre de leaders de LEGOS présents dans une zone de service fixée est bornée.

**Preuve 4** À partir des lemmes 1 et 3, la preuve de ce lemme est immédiate.

**Lemme 5** La cardinalité du backbone généré par LEGOS est bornée pour une zone de service donnée.

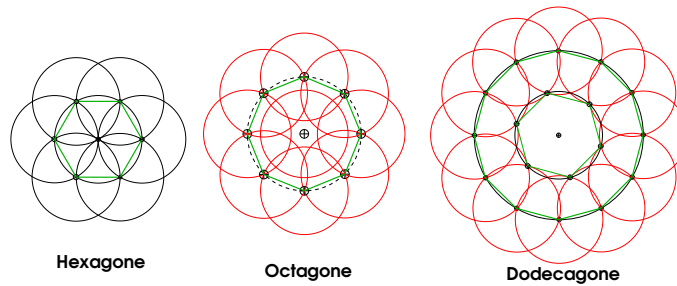


FIG. 4.6 – Motifs : hexagone, octogone, dodécagone

**Preuve 5** Notons que dans LEGOS, il y a au plus une gateway entre deux leaders adjacents. D'après le lemme 4, le nombre de leaders est borné. Il est en donc de même pour les gateways qui forment également un ensemble indépendants. Le backbone de LEGOS étant formé uniquement de leaders et de gateways, sa cardinalité est donc bornée.

**Complexité en messages** Durant l'exécution de LEGOS, le coût en message pour qu'un nœud rejoigne l'auto-organisation ou lors de la ré-organisation locale est  $O(1)$  car le nombre de messages échangés est constant dans ces procédures. Comme LEGOS ne se base pas sur l'émission périodique de paquets `hello` mais s'appuie uniquement sur des diffusions locales de paquets `Leader_Broadcast_Msg` généré par les leaders, la complexité en messages de cette diffusion locale est  $O(1)$ . Finalement la complexité de LEGOS est  $O(n)$ . Le tableau 4.1 fournit une synthèse des complexités des principaux protocoles d'auto-organisation que l'on peut considérer afin de situer LEGOS.

TAB. 4.1 – Complexité de protocoles d'auto-organisation

Algorithme	Type de messages	Complexité algorithmique	Complexité en messages
CDS [81]	Hello/1-hop	$O(\Delta^2)^3$	$O(n)$
MPR-CDS[5]	Hello/1-hop	$O(\Delta^2)$	$O(n)$
LMST[22]	Hello/1-hop+pos.	$O(\Delta)$	$O(n)$
RNG[22]	Hello/1-hop+pos.	$O(\Delta^2)$	$O(n)$
MIS[51]	Hello+control/1-hop	$O(\log(\Delta) \cdot \log(n))$	$O(n)$
LEGOS	LDBR/local	$O(\Delta)$	$O(n)(< n)$

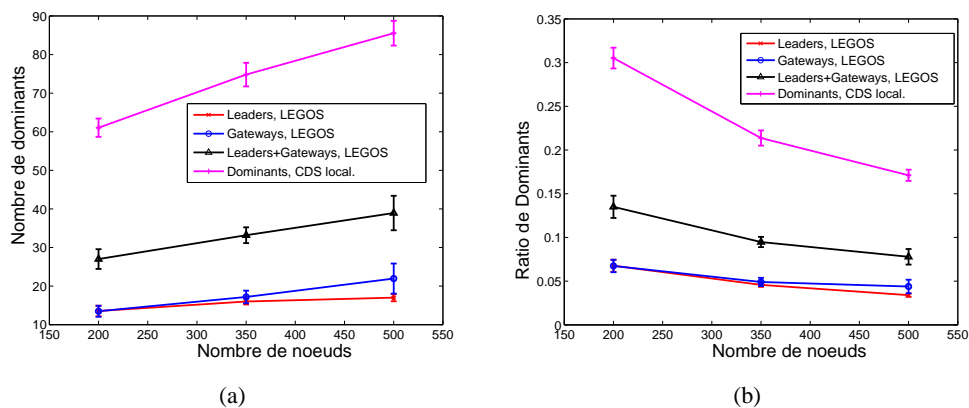
#### 4.2.4 Évaluation de performances

Les résultats présentés ici se basent sur des campagnes de simulation de JiST/SWANS (*Scalable Wireless Ad hoc Network Simulator* [10]). Chaque résultat est fourni avec un intervalle de confiance supérieur à 99%. Les paramètres de simulation sont présentés dans le tableau 4.2. Chaque nœud simulé correspond à un capteur 802.15.4 [59, 66] pour lequel deux couches MAC sont étudiées : CSMA/CA (802.11) [58] et BMAC [63] afin de souligner le gain en énergie de LEGOS.

TAB. 4.2 – Paramètres de simulation

Paramètres	Données
Interface radio	802.15.4, 2.4GHz
Débit	250kbps
Puissance de transmission	0 dBm
Sensibilité en réception	-92 dBm
Couche MAC	BMAC
Périodicité des paquets LEGOS LDBR	1 s
Périodicité des paquets hello (CDS, LMST)	1 s
Modélisation couche physique	Modèle à seuil
Temps inter-arrivées des nœuds (s)	Uniformément distribué dans [673, 1000]

**Propriétés du backbone LEGOS** Les figures 4.7(a) et 4.7(b) illustrent la cardinalité du nombre de dominants dans LEGOS (`leader` et `gateway` inclus) et dans un ensemble dominant connecté construit en utilisant l'algorithme de construction d'un CDS localisé [80]. Le ratio du nombre de dominants dans LEGOS diminue en fonction du nombre de nœuds dans le réseau et en fonction de la densité (figure 4.7(b)). Ce résultat montre bien que la construction du backbone tire parti de l'existence de `leaders` dans le voisinage : un nœud qui arrive plus tard lors du déploiement du réseau a plus de chance de devenir `member` minimisant ainsi son impact sur le backbone. Accessoirement la cardinalité de l'ensemble `leaders` et `gateways` croît moins vite que  $O(n)$ . Soulignons aussi la faible cardinalité de LEGOS par rapport au CDS localisé (figure 4.7(a)) : son utilisation à des fins de diffusion, d'inondation sera nécessairement plus efficace en énergie.

FIG. 4.7 – Cardinalité des nœuds actifs (`leader`, `gateway`) dans LEGOS

**Propriétés locales de LEGOS** La figure 4.8 permet de comprendre les propriétés locales du backbone LEGOS, *i.e.* la densité de nœuds `gateways` et `members` dans le voisinage. Notons que le nombre de `members` par `leader` croît en  $O(n)$ . Cette observation est à corrélérer à la croissance des `leaders` dans le réseau.

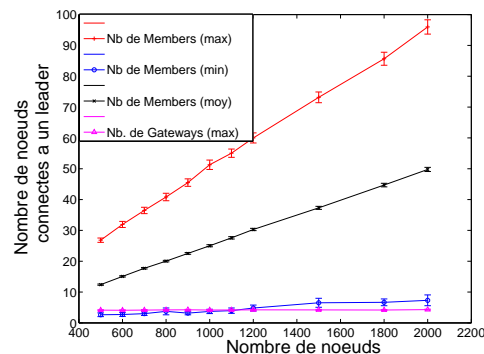


FIG. 4.8 – Propriétés locales dans le voisinage d'un leader

**Latence d'auto-organisation** Aucune synchronisation n'est requise pour LEGOS et les nœuds arrivent et s'allument sporadiquement sur la zone de surveillance. Nous considérons ici que l'auto-organisation débute dès l'arrivée du premier nœud et se termine dès que le dernier nœud arrivé est inséré dans l'auto-organisation LEGOS. La figure 4.9 indique le temps moyen nécessaire d'insertion dans LEGOS pour un nouveau nœud : clairement, cette latence est décroissante avec le nombre de nœuds car plus la densité du réseau est importante est plus le nombre d'auto-organisation à 1 saut l'est également (un nœud arrivant tardivement a alors plus de chance de devenir member).

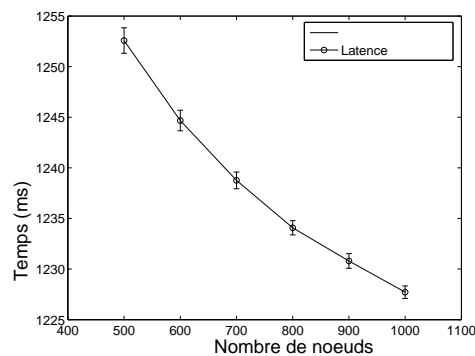


FIG. 4.9 – Latence d'auto-organisation

**Coût protocolaire** Les figures 4.10(a) et 4.10(b) indiquent le surcoût protocolaire pour la construction de LEGOS en émission et réception unicast. Le nombre de paquets unicast par nœud nécessaire à la construction (figure 4.10(a)) décroît légèrement en fonction de la cardinalité et de la densité du réseau : on retrouve empiriquement la complexité en message de LEGOS égale à  $O(1)$ . Une observation similaire peut être faite pour le nombre moyen de messages nécessaires pour les members. Le coût de mise en place de l'auto-organisation est inférieur à celle du CDS localisé (figure 4.10(b)) ; ce coût en messages est également stable lorsque le nombre de nœuds croît dans le réseau puisque le nombre de leaders est également stable.

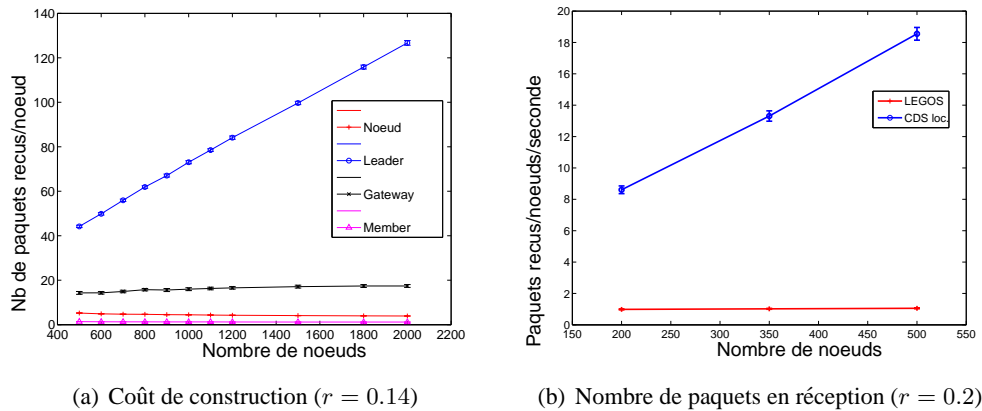


FIG. 4.10 – Coût protocolaire

**Évolution du nombre de partitions** Au lieu de supposer que le réseau était complètement déployé lors de la mise en place de l'auto-organisation, nous avons supposé que les nœuds étaient déployés aléatoirement dans l'espace et sporadiquement dans le temps. La conséquence de cette hypothèse est l'apparition de plusieurs partitions disjointes dans le réseau. Ces partitions nécessitent, par la suite, d'être fusionnées. La figure 4.11 illustre la variation du nombre de partitions entre le premier nœud déployé et le dernier (400<sup>ème</sup>). Un pic apparaît lorsqu'une vingtaine de nœuds est déployé : le réseau est alors peu dense. En revanche, et fort logiquement, dès que le nombre de nœuds déployé croît, la couverture de la zone de surveillance par les dominants devient plus importante et le nombre de partitions décroît. À la fin du déploiement, une seule partition demeure.

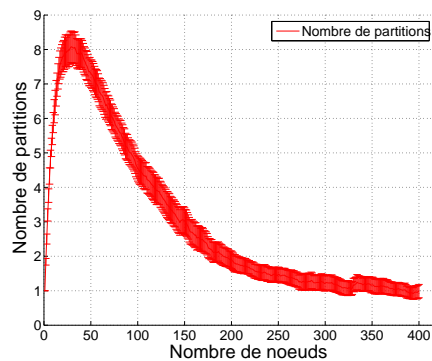


FIG. 4.11 – Évolution du nombre de partitions au cours du déploiement

**Impact de la ré-organisation locale sur la durée de vie du réseau** La figure 4.12 souligne l'intérêt qu'il faut porter aux mécanismes d'ordonnancement distribués pour partager la fonction de leader. La durée de vie du réseau est représentée ici en fonction de l'énergie initialement disponible dans chaque nœud. Rappelons que nous définissons la durée de vie du réseau comme la durée de vie associée au premier nœud qui meurt dans le réseau. L'énergie initiale est la même

pour tous les nœuds. L'utilisation de ce mécanisme de ré-organisation locale en fonction d'une métrique tenant compte de l'énergie résiduelle et de la densité de nœuds *leaders* et *gateways* dans le voisinage entraîne un gain significatif de la durée de vie du réseau, quelque soit le niveau d'énergie initial.

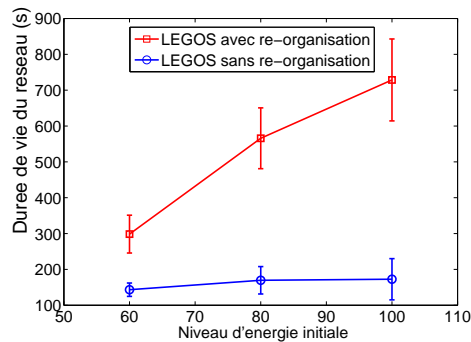


FIG. 4.12 – Ré-organisation locale : efficacité en énergie (300 nœuds, radius  $r = 0.2$ )

**Gestion de l'énergie** La faible consommation en énergie de LEGOS est un des objectifs clés déjà souligné par la faible complexité en messages et la faible cardinalité des nœuds actifs (*leaders*, *gateways*) dans le réseau. Toutefois, la seule stratégie pour significativement économiser de l'énergie est l'utilisation de phases de sommeil pour les nœuds afin de couper les processus d'émission et réception radio [67]. Pour ce faire, il est nécessaire d'utiliser une couche MAC permettant l'endormissement des nœuds. Nous utilisons ici BMAC [63] une couche MAC de type CSMA/CA autorisant un ordonnancement des phases de réveil et de sommeil des nœuds. Nous ne nous intéressons pas au gain fourni par BMAC mais à la valeur ajoutée proposée par LEGOS. Pour la consommation en énergie des capteurs, nous nous sommes basés sur le composant 802.15.4 MC13192 [66]. Les figures 4.13(a) et 4.13(b) indiquent clairement que le temps d'activité des nœuds dans LEGOS comparé à une autre stratégie d'auto-organisation (LMST [53] ici) est très faible et est insensible à la densité du réseau. Cela est la conséquence de la non-utilisation de paquets *hello* transmis périodiquement réduisant significativement le temps d'écoute du voisinage. Les protocoles basés sur l'utilisation de paquets *hello* périodiques ont une consommation en énergie qui s'accroît avec la densité du réseau (que ce soit un LMST comme ici ou un CDS, etc.).

#### 4.2.5 LEGOS : d'un arbre à un treillis

La topologie de LEGOS présentée jusqu'à présent correspondait à un arbre. La construction d'une structure logique maillée peut avoir plusieurs avantages dans des cas particuliers. Ainsi, un réseau maillé fournira un plus court chemin entre un capteur et le puits comparé à une structure en arbre. Un réseau maillé permet également une meilleure répartition de la charge de trafic sur plusieurs chemins du réseau. Enfin, ce type de topologie apparaît comme plus robuste. Toutefois, la cardinalité de la topologie sera plus importante que dans un arbre qui contient peu de nœuds *leaders* et *gateways*. Or comme les nœuds du backbone consomment plus d'énergie que les nœuds *members*, une structure en forme d'arbre dépensera également moins d'énergie.

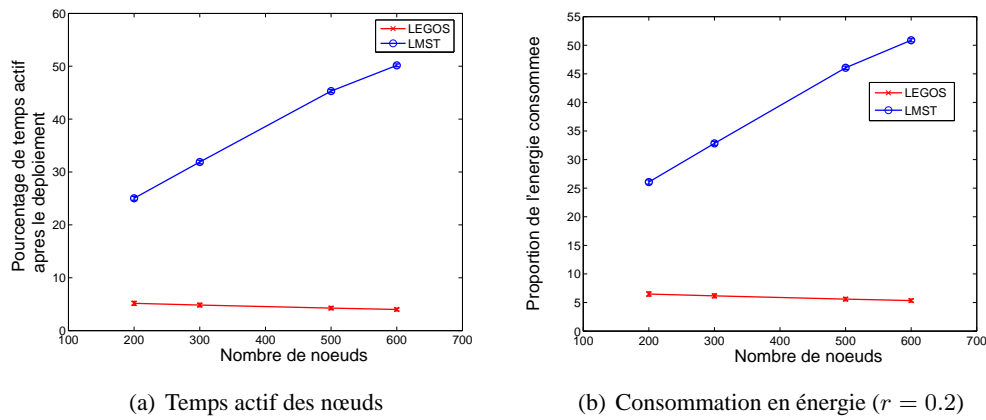


FIG. 4.13 – Dépenses en énergie de LEGOS (couplé à l'utilisation de BMAC)

Nous étudions ici la possibilité de construire un réseau logique maillé. Pour ce faire, nous rajoutons à la construction en arbre de LEGOS un mécanisme local de construction de liens redondant pour tendre vers un treillis. Finalement, ce mécanisme va accroître le nombre de *gateways* dans le réseau. Un *member* va se déclarer *gateway* s'il entend au moins  $p$  *leaders* dans son voisinage,  $p$  étant un paramètre de la construction.

Les figures 4.14(a)-4.14(d) illustrent les différentes topologies construites en fonction de la valeur de  $p$ . Lorsque  $p = 1$ , la topologie reste proche de l'arbre initial en proposant assez peu de chemins disjoints. Logiquement, lorsque  $p$  augmente, le nombre de *gateways* augmente significativement (représenté par des nœuds orange sur les figures). De fortes valeurs de  $p$  conduisent logiquement à un nombre plus important de nœuds dans la structure LEGOS et donc par une multiplicité de chemins entre deux nœuds. Rappelons qu'une augmentation de cette cardinalité entraîne une consommation d'énergie plus importante : un compromis sur la valeur de  $p$  doit être trouvé en fonction de l'application et des besoins. L

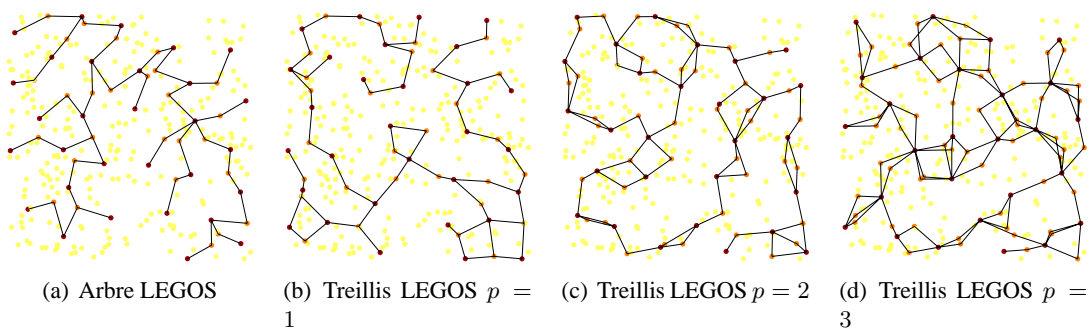


FIG. 4.14 – Topologie LEGOS : arbre vs treillis

Les figures 4.15(a) et 4.15(b) donnent l'évolution de la cardinalité des treillis LEGOS pour différentes topologies et différentes  $p$  variant de 1 à 3. La référence à la cardinalité de l'arbre LEGOS est également fourni. Les figures nous fournissent des informations sur la topologie du treillis LEGOS en fonction de la cardinalité du réseau. Une zone de surveillance de  $1 \times 1$  est considéré ainsi qu'une portée de communication  $r = 0.18$ . Augmenter la valeur de  $p$  fait croître

la cardinalité de l'ensemble des gateways mais fait décroître dans le même temps le diamètre du réseau (*i.e.* la longueur du plus long chemin). Toutefois, gardons en mémoire qu'une cardinalité importante de la logique conduit à une consommation en énergie importante : en fonction des besoins de l'application,  $p$  devra être ajusté.

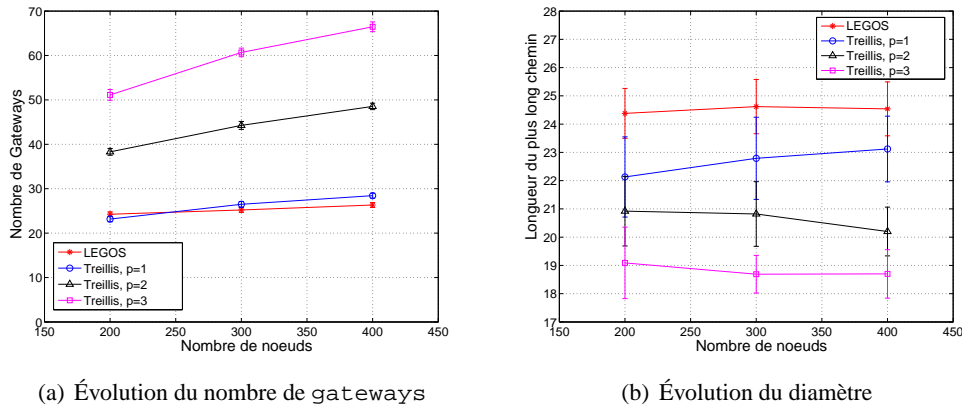


FIG. 4.15 – Analyse du treillis LEGOS ( $r = 0.18$ )

Comme dans le chapitre précédent, nous allons montrer comment et pourquoi un réseau de capteurs structuré est plus efficace qu'un réseau considéré *à plat*. Pour cela, nous allons passer en revue un certain nombre de mécanismes clefs tels que : l'auto-configuration d'adresses de capteurs (section 4.3, la gestion de puits mobiles ainsi que la dissémination de données (section 4.5) avant de terminer par les mécanismes d'agrégation temporelles et spatiales (section 4.5).

### 4.3 Support pour l'auto-configuration d'adresses

La problématique de l'auto-configuration d'adresses est une problématique abondamment traitée dans la littérature avec essentiellement deux stratégies : DHCP distribué où chaque nœud se retrouve en position d'allouer une adresse [57, 85] et les solutions basées sur la détection d'adresses en conflit soit localement [78] soit globalement [43]. Il s'agit alors de fournir un identifiant *unique* à un nœud du réseau pour ensuite mettre en place un routage, une localisation, etc. Bien entendu la difficulté réside dans l'*unicité* de l'adresse. Quelques soient les solutions envisagées, le coût de vérification est souvent trop important et entraîne une sur-consommation en énergie, préjudiciable à la vie du réseau. De plus, l'exigence d'un marché de masse conduit à une production de capteurs sans identifiant matériel [1] pour baisser les coûts de production : plus que jamais les solutions d'auto-configuration à faible coût en énergie et en complexité doivent être proposées.

La solution (LEADS - *Low Energy Address Allocation Scheme for WSN*) proposée ici exploite l'existence du backbone LEGOS stable pour diffuser des informations d'allocation d'adresses lorsque c'est nécessaire et tire bénéfice de la faible cardinalité de l'ensemble des nœuds *leaders*. La solution que nous proposons peut être perçue comme un DHCP distribué mais uniquement sur les *leaders*. Nous allons chercher à :

- ne pas avoir de conflits d'adresses au sein d'une même partition, ne pas avoir de conflits après fusion de plusieurs partitions ou lors de la scission d'une partition,

- ne pas diffuser dans tout le réseau des informations d'états sur les allocations d'adresses pour économiser de l'énergie.

En réalité, plus qu'une solution d'auto-configuration basée sur une auto-organisation, nous proposons la mise en place d'un processus simultané : l'auto-organisation se fait conjointement avec l'auto-configuration, *i.e.* nous utiliserons les mêmes paquets décrits dans la section précédente pour la mise en place de l'organisation du réseau et la configuration des nœuds. Le gain en latence, en complexité, en messages utilisés et, finalement, en énergie sera alors appréciable. Nous supposons que la taille de la zone de surveillance est connue ainsi que le nombre de capteurs utilisés.

### 4.3.1 Allocation d'adresses à deux niveaux

En nous basant sur LEGOS, chaque leader fait office de DHCP local. L'espace d'adressage nécessaire est divisé en pool d'adresses (voir figure 4.16) et chaque leader possède au moins un pool ainsi qu'un vecteur d'état décrivant les pools de libre. Comme nous avons montré précédemment que le nombre de leaders était borné pour un réseau de taille fixé, il est possible de diviser l'espace d'adresses en  $T$  pools tel que :  $T > 7 \cdot \left[ \frac{a+2R}{(2+\sqrt{3}/2)R} \cdot \frac{a+2R}{2.5R} \right]$ . Un nouveau nœud qui arrive dans le voisinage à 1 saut d'un leader devient member et obtiens une adresse nécessairement unique (allocation à 1 saut). Un nouveau qui arrive à 2 sauts d'un leader devient leader également et obtiens une adresse ainsi qu'un pool d'adresses pour allouer des adresses à son voisinage à 1 saut (allocation à 2 sauts). Lors de l'allocation d'un pool d'adresses à un nouveau leader une inondation sur le backbone de LEGOS est requise pour que chaque leader maintienne une vue cohérente des pools utilisés. Lorsqu'un leader a alloué toutes les adresses de son pool d'adresses, il utilise le premier pool disponible après avoir informé les autres leaders du réseau.

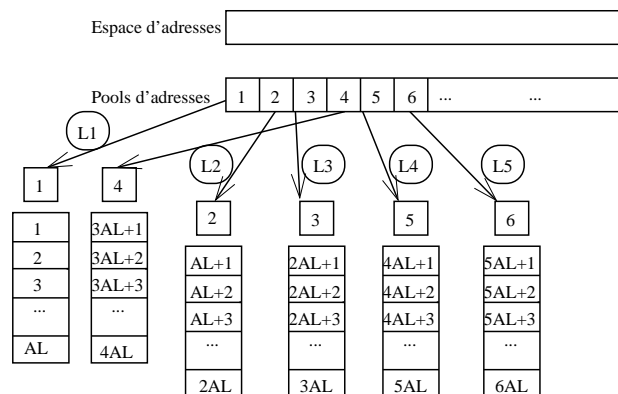


FIG. 4.16 – Espace d'adressage et pools d'adresses

Lors de départs de nœuds, deux stratégies sont envisageables. Tout d'abord, les nœuds peuvent informer leur voisinage local de leur départ : l'adresse utilisée est à nouveau disponible. Dans le cas du départ d'un leader le pool qui était alloué revient au nœud qui devient leader (voir section précédente). En revanche, si les nœuds ne peuvent pas informer de leur départ, un mécanisme de vérification des nœuds voisins doit être utilisé : en l'absence de trafic de données permettant de vérifier le voisinage, nous pouvons utiliser celui proposé pour l'auto-organisation. Dans chaque

cas, l'adresse du nœud disparu est restituée et dans le cas d'un leader le pool est repris par le nouveau leader dans le voisinage.

Le cas de la scission d'une partition ne pose pas de problème clef : puisqu'il n'y a pas de conflit dans la partition d'origine, il ne peut pas y en avoir dans les deux nouvelles partitions. Dans le cas d'une détection de la scission d'une partition, rappelons qu'un nouvel identifiant de partition est tiré aléatoirement pour éviter des conflits. Ce nouvel identifiant est propagé sur l'arbre LEGOS ce qui permet, au passage, de faire le point sur les pools d'adresses utilisés dans les nouvelles partitions et de rendre à nouveau disponible les pools non utilisés. Une inondation supplémentaire est alors requise. Le cas de la fusion est un peu plus coûteux. Rappelons que la fusion de partitions ne peut se produire que sur l'initiative d'un nouveau nœud (member ou gateway) arrivant et que la partition de plus faible cardinalité fusionne dans celle de plus forte cardinalité. Pour éviter des conflits entre les deux espaces d'adresses utilisées, les pools d'adresses de la plus petite partition sont décalés jusqu'à au premier pool libre dans la plus grande partition : ce mécanisme nécessite une inondation (contrôlée) de la plus petite partition et une re-numérotation des adresses.

### 4.3.2 Évaluation de performances

Les hypothèses de simulation sont les mêmes que lors de l'évaluation de performances de LEGOS dans la section précédente.

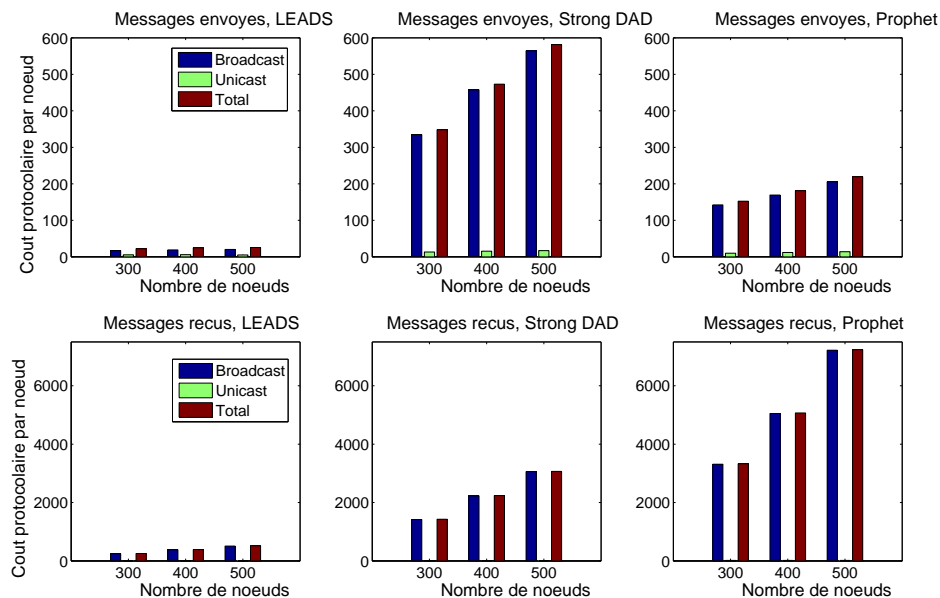


FIG. 4.17 – Comparaison des coûts protocolaires pour l'auto-configuration

**Coût protocolaire** La figure 4.17 montrent le nombre moyen de message par nœud pour la phase d'auto-organisation et d'auto-configuration pour la solution introduite précédemment. Ce coût est comparé à *StrongDAD*<sup>4</sup> [43] et *Prophet*<sup>5</sup> [85]. Nous avons séparé les coûts en réception

<sup>4</sup> Allocation d'adresses localement puis détection des conflits dans le réseau.

<sup>5</sup> Allocation d'adresses suivant un DHCP distribué.

des coûts en émission en regardant séparément les paquets unicast et les paquets broadcast. Sous les mêmes hypothèses, notre solution s'avère nettement plus efficace que *StrongDAD* et *Prophet* avec un très faible coût en message notamment grâce à l'absence de l'utilisation de paquets périodiques que ce soit dans les phases d'auto-organisation ou d'auto-configuration. Ce faible coût s'explique aussi par la faible cardinalité de l'ensemble des *leaders* et par l'utilisation d'un backbone pour l'inondation des mise à jour de pools d'adresses.

**Consommation en énergie** Pour la consommation en énergie, nous supposons l'utilisation d'une couche MAC autorisant la mise en veille de l'interface radio et la mise en place de phases de sommeil comme BMAC [63]. Une fois de plus nous mesurons le pourcentage de temps où les nœuds sont actifs (figure 4.18(a)) et la consommation moyenne en énergie (figure 4.18(b)). La consommation de LEADS est beaucoup plus faible que pour les autres stratégies : ce gain est à mettre sur la non-utilisation de paquets périodiques, sur la faible cardinalité de l'ensemble des *leaders* et sur la participation des *leaders* uniquement dans les phases de configuration. Nous retrouvons un résultat déjà connu pour l'auto-organisation LEGOS : l'insensibilité de la consommation en énergie en fonction de la densité du réseau alors que *StrongDAD* et *Prophet* voient leur consommation augmentée avec la densité. Fort logiquement la consommation en énergie des *leaders* est plus importante que celle des *gateways* qui est elle même bien supérieure à celle des *members*.

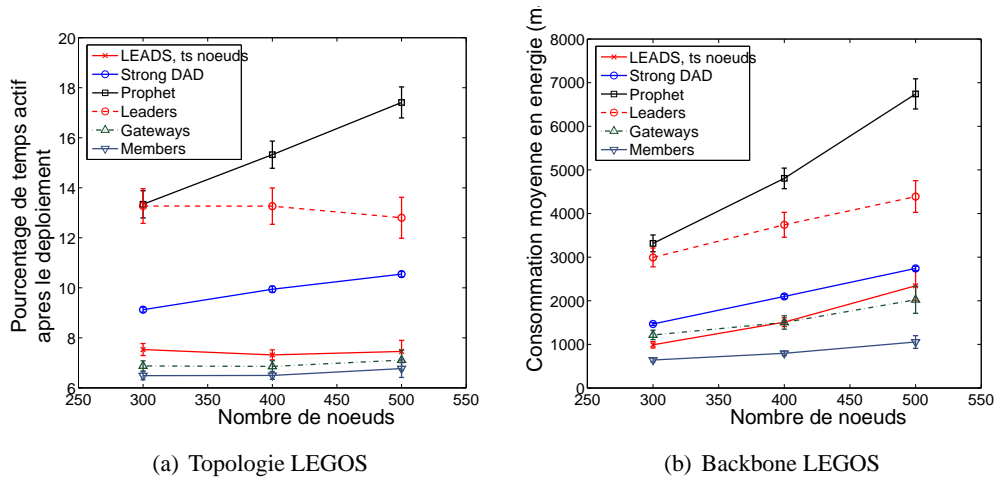


FIG. 4.18 – Temps d'activités et consommation en énergie (portée radio  $r = 0.20$ )

#### 4.4 Gestion de puits mobiles et acheminement de données

Le déploiement d'un réseau de capteurs est conditionné par les applications utilisées et par la localisation du puits dans le réseau. Actuellement et dans les propositions faites dans ce chapitre, la localisation du puits n'a jamais été considérée : le déploiement du réseau et la mise en place de l'auto-organisation en sont indépendants pour plus de flexibilité. Plus qu'un seul puits statique, force est de constater de la nécessité de considérer plusieurs puits éventuellement mobiles. L'introduction de puits multiples/mobiles dans les réseaux de capteurs permet de réduire le temps de

réponse de l'application, de répartir la charge de trafic sur les nœuds du réseau, de diminuer fortement les goulots d'étranglements, de router sur de plus courts chemins, etc. Bref, d'économiser une consommation d'énergie quelquefois excessive. De plus, différents puits peuvent avoir des intérêts de collecte de données différents.

Classiquement, on retrouve deux types de stratégies dans la littérature : basée sur des rendez-vous et orientée source. Dans les stratégies de type *rendez-vous*, un ou plusieurs points de rendez-vous sont sélectionnés pour collecter les données [68, 83], néanmoins cela nécessite la connaissance d'informations de localisation. Dans les stratégies orientées sources, le puits, lorsqu'il se déplace, diffuse une information de localisation : les nœuds peuvent alors utiliser un routage géographique ou par gradient pour acheminer leurs données [40, 16, 56]. Toutefois, maintenir l'information de localisation du puits a un coût en messages important, et donc, entraîne une consommation énergétique forte.

Une fois de plus, la question que nous adressons est la suivante : comment l'utilisation d'un paradigme d'auto-organisation peut-il amener à apporter des réponses à ce problème ? Plus précisément : comment, en se basant sur l'auto-organisation LEGOS, apporter une réponse simple et efficace à la problématique des puits multiples et mobiles ?

Lors de la phase de déploiement du réseau, l'auto-organisation LEGOS se met en place, qu'elle soit du type arbre ou treillis. Cette auto-organisation, indépendante de toute notion de localisation, indépendante du (des) puits, apparaît comme une structure non orientée. C'est à travers cette topologie logique que la prise en compte des puits va être faite : le support des requêtes, l'acheminement des informations depuis le puits vers les nœuds et depuis les nœuds vers le puits ainsi que la conservation des données remontées par les capteurs si aucun puits n'est connu.

La solution proposée (*BBDD- Backbone Data Dissemination*) apparaît comme une stratégie à deux niveaux : un niveau global de traitement des données entre les *leaders* et le puits, et un niveau local entre les capteurs qui surveillent l'environnement et leur *leader* en charge de remonter les informations perçues.

#### 4.4.1 Prise en compte des requêtes

Après que la topologie logique LEGOS soit déployée, les puits vont s'annoncer dans le réseau. L'annonce d'un puits correspond à l'inondation d'un tuple (*InterestID*, *SinkID*) permettant d'identifier l'identité du puits et le type d'information recherchée par lui. Lorsqu'un puits arrive dans le réseau, il diffuse dans son voisinage ce tuple. Chaque *leader* (ou un *member* ou un *gateway* s'il n'y a pas de *leader* dans le voisinage) recevant une copie de ce message va alors inonder cette information dans le réseau en utilisant uniquement comme relais les *leaders* et *gateways* : le coût de diffusion de l'information est alors contrôlé et comme les *members* sont tous dans le voisinage d'au moins un *leader*, ils connaîtront tous l'identité du puits. Cette propagation du tuple émis par un puits peut être perçue comme une orientation dynamique des arêtes de LEGOS. Lorsqu'un nœud reçoit plusieurs copies du même tuple, venant de différents *leaders*, il privilégie le plus court chemin vers le puits. La figure 4.19 illustre cette orientation dynamique.

Ainsi, cette stratégie permet de gérer simplement et efficacement plusieurs puits ayant des objectifs (*InterestID*) différents ou un puits ayant plusieurs objectifs. En cas de mobilité d'un puits, un nouveau tuple (*InterestID*, *SinkID*) est diffusé dans le réseau, ce qui permet de mettre à jour le chemin depuis les capteurs vers le puits. Pendant le déplacement d'un puits, ce

sont les *leaders* qui conservent l'information faisant office de points de rendez-vous distribués.

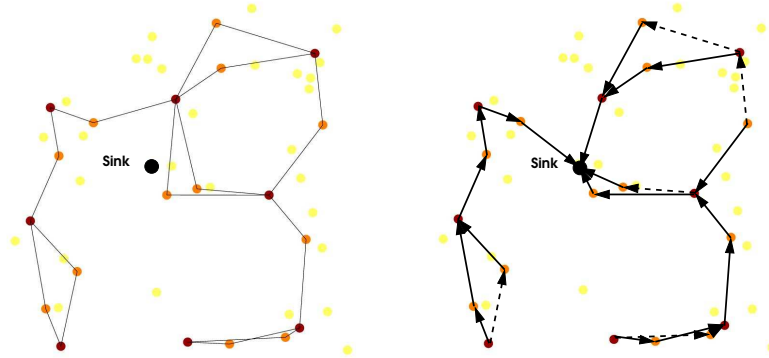


FIG. 4.19 – Propagation d'un requête InterestID, SinkID sur LEGOS

#### 4.4.2 Acheminement des données

Lorsqu'un capteur a une donnée à remonter vers le puits correspondant à l'InterestID, l'information est transmise via la topologie logique LEGOS en suivant l'orientation des arêtes qui a été déterminée lors de la propagation du tuple (InterestID, SinkID). Le capteur transmet alors l'information à son *leader* qui se charge de remonter les données vers le puits. La figure 4.20 illustre ce mécanisme.

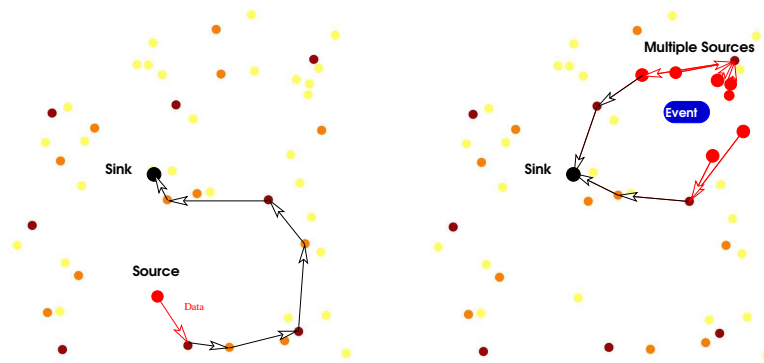


FIG. 4.20 – Remontée des données depuis un capteur vers un puits

#### 4.4.3 Stockage des données et points de rendez-vous

Si un puits n'a pas annoncé sa présence (expiration d'un temporisateur ou non mise à jour de localisation), les capteurs du réseau ne peuvent plus remonter les données collectées. LEGOS apparaît alors comme un système de rendez-vous distribué et dynamique : une information perçue par un capteur sera conservé par son *leader* en attendant qu'un puits intéressé par cette information se manifeste. Cette solution est plus flexible que [83] ou [68] car les points de rendez-vous de LEGOS (*i.e.* les *leaders*) ne sont pas connus grâce à une information de localisation et sont répartis dans le réseau dynamiquement en fonction des interactions locales.

#### 4.4.4 Analyse du coût protocolaire

**Impact sur la longueur du chemin** Comme le coût protocolaire de BBDD est basé sur la longueur des chemins de LEGOS, nous devons auparavant faire une analyse de l'évolution de la longueur des chemins : nous utiliserons alors une borne supérieure. Nous supposons le déploiement aléatoire de  $N$  nœuds sur un carré  $1 \times 1$  avec une intensité  $\lambda$  avec un portée radio uniforme de  $r$ . Nous définissons la fonction  $H_{\lambda,r}(d(x,y)) = \xi \frac{d(x,y)}{r}$  représentant le nombre de sauts entre les nœuds  $x$  et  $y$  où  $d(x,y)$  représente la distance euclidienne entre  $x$  et  $y$ , et  $\xi$  un facteur de pondération.

Dans la construction d'une topologie logique maillée, basée sur LEGOS, le paramètre  $p$  a un impact déterminant sur la topologie logique. Nous notons  $\lambda_p$  l'intensité des nœuds de la topologie logique. Lorsque  $p$  croît,  $\lambda_p$  croît également et  $H_{\lambda,r}(d(x,y))$  décroît. La longueur du chemin le plus long est bornée par  $H_{\lambda,r}(d(x,y))$ . Si  $N_B$  est le nombre de nœuds présents dans la topologie logique alors  $N_B \approx \frac{\lambda_p}{\lambda} N$ .

Nous allons comparer le coût de contrôle de BBDD avec TTDD [83] et RailRoad [68]. Dans le premier cas, une grille virtuelle est construite par nœud et chaque sommet de la grille correspond à un point de rendez-vous utilisé pour un routage géographique. Dans le deuxième cas, les points de rendez-vous correspondent aux points présents sur la trajectoire d'un train chargé de collecter les données.

**Complexité en messages** Trois types de messages sont utilisés dans BBDD : diffusion de la requête et du tuple, notification d'un message de données et acheminement de ce message. Nous supposons que la taille de ces messages est respectivement :  $p_q$ ,  $p_e$  et  $p_d$ . Ces messages sont également utilisés dans TTDD et RailRoad. En revanche, dans TTDD, il faut rajouter un coût de contrôle supplémentaire associé à l'intérieur d'une cellule de la grille virtuelle de taille  $p_c$ . Nous supposons qu'il y a  $m$  puits mobiles et  $n$  sources de données dans le réseau. Chaque puits génère  $\bar{q}$  requêtes et chaque source génère  $\bar{e}$  données en moyenne. Ainsi, le nombre de requêtes et de données transitant dans le réseau sont  $m\bar{q}$  et  $n\bar{e}$  respectivement.

Le coût de contrôle de ces deux protocoles a été analysé dans [68] en supposant un routage glouton du type [48] pour l'acheminement des requêtes et des données. Néanmoins le coût de ce protocole de routage, ainsi que le coût associé à l'émission périodique de paquets hello n'a pas été considéré. Nous complétons le résultat de [68] en rajoutant ces coûts de contrôle pour avoir une vision d'ensemble. Nous supposons que l'intensité d'émission des paquets hello est la même que l'annonce des requêtes par le puits, noté  $\bar{q}$ . Le coût du protocole de routage est alors égal à  $N\bar{q}$ . Ainsi, les coûts protocolaires de TTDD et de RailRoad sont :

$$C_{TTDD} = m\bar{q} [N_C + \lambda/r \cdot 2] p_q + \lambda/r \cdot n\bar{e} [2 + (\sqrt{2}/(2\alpha))] p_d + (n \frac{4N}{\lambda r \cdot (1/\alpha)} + N\bar{q}) p_c; \quad (4.2)$$

$$C_{Railroad} = n\bar{e} \lambda/r \cdot (\sqrt{2}/4) p_e + \lambda/r [(n\bar{e} + m\bar{q}) \cdot (\sqrt{2}/4) + m\bar{q} \cdot (2\sqrt{2})] p_q + 4n\bar{e} \lambda/r \cdot (\sqrt{2}/4) p_d + N\bar{q} p_c; \quad (4.3)$$

Où :  $\alpha$  représente la taille d'une cellule dans TTDD and  $N_C$  est le nombre moyen de nœuds par cellule. Dans cette analyse,  $N_C = N\alpha^2$  car nous considérons une zone de déploiement  $1 \times 1$ .

Concernant notre proposition, BBDD, le coût protocolaire est fonction de la diffusion de la requête et du tuple, de la notification d'un message de donnée et de l'acheminement de ce message

auquel il faut rajouter le coût de construction de la topologie d'auto-organisation LEGOS. Ce coût s'écrit donc :  $C_{BBDD} = C_{DN} + C_{QF} + C_{DF} + C_{LEGOS}$ . Puisque uniquement les nœuds leaders et gateway de LEGOS diffusent des paquets de contrôle de type `hello`, le coût de LEGOS est :  $C_{LEGOS} = N_B \cdot \bar{q}p_c$  où  $N_B$  est le nombre de nœuds du backbone LEGOS. Dans BBDD, tous les nœuds du réseau sont à 1 saut d'un membre de LEGOS : quand une source veut notifier la présence d'une donnée, la notification ne se déroule que dans le voisinage. Le coût protocolaire est alors :  $C_{DN} = n\bar{e}p_e$ . La diffusion des requêtes et des tuples par les puits se fait uniquement via le backbone de LEGOS : le coût protocolaire est lié à la taille de celui-ci. Dans le pire cas, nous avons alors :  $C_{QF} = m\bar{q}(N_B + N_G)p_q$  où  $N_G$  est le nombre de gateways dans LEGOS. Le coût d'acheminement des données est lié à la longueur du chemin via le backbone, taille qui est bornée par  $H_{\lambda_p, r}(\sqrt{2})$ . Dans le pire cas :  $C_{DF} = n\bar{e}p_d H_{\lambda_p, r}(\sqrt{2})$ . Finalement, nous obtenons :

$$C_{BBDD} = n\bar{e}p_e + n\bar{e}\lambda_p/r \cdot (\sqrt{2})p_d + N \frac{\lambda}{\lambda_p} m\bar{q}p_q + N \frac{\lambda}{\lambda_p} \bar{q}p_c; \quad (4.4)$$

Les coûts protocolaires sont représentés sur la figure 4.21. Une des faiblesses que l'on pourrait attendre de BBDD basé sur LEGOS est le surcoût protocolaire : effectivement avant de proposer une solution de gestion de mobilité des puits et de mise en place de points de rendez-vous, nous imposons la construction d'une structure logique d'auto-organisation dont le coût n'est pas négligeable. Néanmoins et comme la figure le présente, le surcoût de BBDD demeure bien plus faible que le coût d'autres solutions bien connues. Ce résultat vient appuyer un de nos arguments en faveur de l'auto-organisation : avant de mettre en place un protocole dans un réseau, structurons-le et alors l'ensemble des protocoles de communications seront plus efficaces. Nous le voyons ici sous l'angle du coût de contrôle.

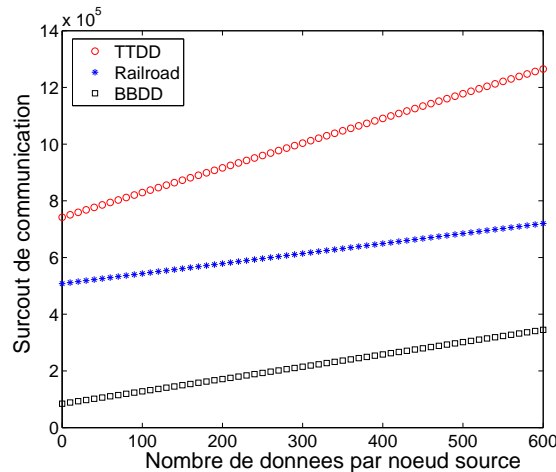


FIG. 4.21 – Surcoût protocolaire en fonction du nombre de données ( $\bar{e}$ ) générées ( $m = 5$  puits mobiles,  $n = 10$  nœuds sources,  $N = 10000$  nœuds, portée radio  $r = 0.1$ ,  $\bar{q} = 50$  requêtes par les puits. Pour TTDD :  $\alpha = 0.3$ )

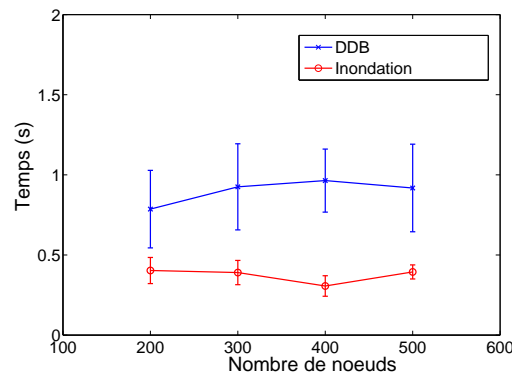


FIG. 4.22 – Latence de construction de BBDD

#### 4.4.5 Évaluation de performances

Nous avons simulé avec JiST/SWANS le comportement de LEGOS et de la construction de BBDD. Les hypothèses retenues sont identiques à celles posées dans la section 4.2.4 (évaluation de performances de LEGOS).

**Latence de construction de BBDD** Nous avons mesuré la latence de l'orientation de LEGOS pour fournir BBDD. Ce temps est mesuré entre l'émission de la requête et du tuple par un puits jusqu'au dernier instant de réception. Nous comparons ce temps avec une inondation classique depuis le puits. L'inondation classique est proche d'une diffusion basée sur un NES (*Neighborhood Elimination Scheme* [70]). Le temps de l'inondation avec LEGOS est plus important : cela est dû à la construction non optimale du backbone en terme de diamètre à cause de l'arrivée aléatoire des nœuds dans le réseau alors que le diamètre du graphe des nœuds qui sont sélectionnés dans l'inondation est bien plus faible. Néanmoins, comme nous avons montré que la cardinalité de backbone LEGOS était bornée, cela signifie que, quelque soit la topologie considérée, le temps d'orientation de LEGOS pour fournir BBDD l'est aussi.

**Dissémination de données sur BBDD** Bien entendu, la topologie LEGOS peut être utilisée pour la dissémination de données dans un réseau de capteurs pour diffuser une information depuis un puits vers tous les nœuds du réseau. La dissémination suit alors un protocole similaire à la diffusion d'une requête et d'un tuple : seuls les nœuds *leaders* et *gateways* relaient l'information. L'objectif d'une dissémination est de diminuer le nombre de d'émissions mais surtout de réceptions multiples du même message. La figure 4.23 fournit une comparaison entre une l'inondation classique et BBDD en termes de nombre de paquets émis et reçus (en absolu et en moyenne). Concernant BBDD, tous les paquets sont pris en compte, notamment ceux construction de LEGOS. Le gain important en nombre de paquets émis et surtout reçu vient de l'utilisation d'un CDS pour la diffusion d'une information. Plus que LEGOS, c'est la construction d'un arbre de faible cardinalité qui mène à ces bons résultats. Nécessairement le gain en énergie sera important.

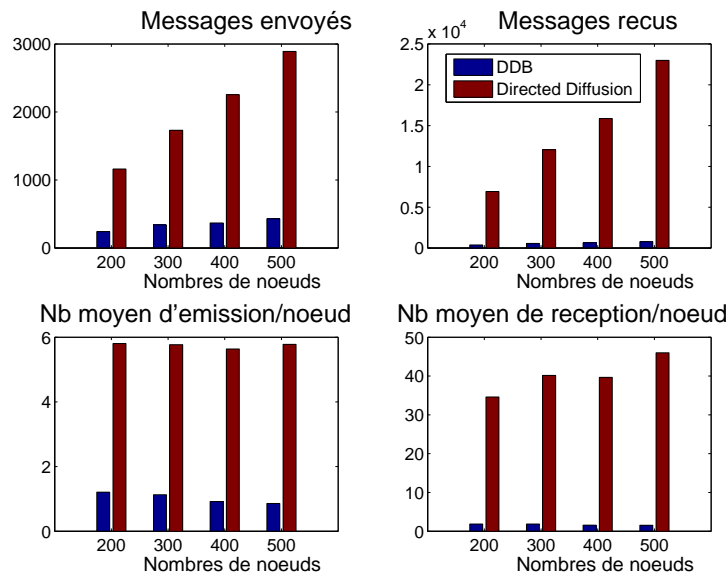


FIG. 4.23 – Latence de construction de BBDD

## 4.5 Agrégation de données

L'agrégation est un mécanisme clef pour les réseaux de capteurs [47]. L'objectif est de réduire la charge de trafic acheminé dans le réseau ainsi que la consommation en énergie à travers une factorisation de l'information transmise par les capteurs. Deux types d'agrégations sont à considérer : l'agrégation spatiale où l'on cherche à combiner des informations venant de plusieurs sources et l'agrégation temporelle venant de différentes périodes de temps. L'agrégation se fait sur le chemin allant du capteur vers son puits. Classiquement les mécanismes d'agrégation reposent sur un arbre d'agrégation où les nœuds de l'arbre sont responsables de l'agrégation : la construction est initiée par le puits au travers un message d'initialisation qui est inondé dans le réseau. L'utilisation de paquets `hello` est également requise pour maintenir cette topologie sauf à reconstruire périodiquement cette structure. De toute évidence, ces mécanismes ne sont pas efficaces en énergie.

LEGOS fournit une topologie en arbre initialement construite lors du déploiement du réseau. L'arbre LEGOS, déjà utilisé pour la gestion des puits mobiles et pour la dissémination de données, peut être utilisé pour développer un nouveau mécanisme d'agrégation temporelle de données. L'agrégation proposée se base sur une prédiction des données collectées en utilisant un modèle ARMA (*AutoRegression Moving Average* [23]) : au lieu de remonter les nouvelles mesures collectées par les capteurs, seuls les coefficients de la série prédictive sont remontés lorsque cela est nécessaire. L'utilisation du médium radio est alors minimisée au profit d'un calcul local des coefficients du modèle ARMA.

### 4.5.1 Modèle ARMA et prédiction de séries temporelles

L'objectif d'un modèle ARMA est de développer une prédiction du futur à partir d'un échantillon de données historiques. La plupart des données physiques mesurées par des capteurs peuvent

se modéliser ainsi (détection de température, mesure de pollution, etc.). Un modèle ARMA est composé de deux entités :

- la partie AR (*AutoRegression*) est une auto-régression linéaire représentant le comportement quasi-déterministe d'une série temporelle. Elle permet de prédire la valeur de  $X_t$  à partir des  $p$  valeurs précédentes :

$$X_t = \phi_0 + \phi_1 \times X_{t-1} + \dots + \phi_p \times X_{t-p},$$

où  $\phi_i$  sont les coefficients des termes AR. Un modèle  $AR(p)$  est caractérisé par  $p + 1$  coefficients.

- la partie MA (*Moving Average*) capture le comportement aléatoire qui est indépendant du processus AR. Un modèle MA consiste en l'insertion de comportements aléatoires sur  $q$  éléments précédents :

$$X_t = \theta_1 \times \epsilon_{t-1} + \dots + \theta_q \times \epsilon_{t-q},$$

où  $\theta_i$  sont les coefficients de la série MA. Un modèle  $MA(q)$  est caractérisé par  $q$  coefficients.

Un modèle  $ARMA(p, q)$  permet alors de modéliser des séries temporelles sous l'hypothèse de stationnarité et d'inversibilité. Nous reviendrons, dans la section 4.5.6, sur cette hypothèse et son impact notamment en termes de précisions de la prédiction en cas de situation anormale. Si l'on considère des séries de données temporelles non stationnaires, l'emploi de série ARIMA (*AutoRegression Intergrated Moving Average*) est requis. Les contextes des travaux présentés ici se focalisant sur la surveillance de la température, l'emploi de séries  $ARMA(p, q)$  est alors justifié. Les données collectées et traitées proviennent de la Cité Interdite (Pékin, Chine).

#### 4.5.2 Agrégation temporelle locale

Basiquement, les capteurs remontent les informations de l'environnement surveillé vers le puits. Cette remontée peut être soit périodique soit sur changement de valeurs soit sur dépassement de seuil. Toutefois, chaque remontée entraîne une charge de communication et un coût en énergie non négligeable. Nous proposons ici que chaque nœud, en plus de son activité de surveillance de l'environnement, calcule un modèle ARMA basé sur la série de  $W$  échantillons consécutifs afin d'exploiter la corrélation temporelle des données. Ce sont les paramètres de la série ARMA qui sont transmis au puits pour permettre de reproduire les données perçues par le capteur. Localement, chaque nœud vérifie les valeurs prédites par le modèle et leur pertinence vis-à-vis des mesures effectivement prélevées. Tant que la précision du modèle est satisfaisante, aucun trafic n'est généré dans le réseau. En revanche, dès que le modèle ARMA s'éloigne des valeurs de mesures, de nouveaux paramètres ARMA sont calculés localement et transmis vers le puits pour permettre de reproduire les valeurs des données tout en maintenant la précision souhaitée.

Plusieurs points doivent être étudiés pour valider cette approche : *i*) complexité du calcul au regard de la puissance des capteurs, *ii*) précision du modèle vis-à-vis des données observées et *iii*) coût du transfert des coefficients dans le réseau.

#### 4.5.3 Complexité du calcul local

Les capteurs sont limités tant par leur capacité de calcul que par leur capacité de stockage. Seuls des algorithmes à faible complexité peuvent être utilisés pour le calcul local des coefficients

ARMA. [29] indique que la complexité de calcul des coefficients ARMA est en  $O(m^3W)$  avec  $m$  étant l'ordre du modèle et  $W$  est le nombre d'échantillons requis. Dans notre contexte, il nous faut donc un ordre  $m$  faible ainsi qu'une fenêtre d'observation  $W$  relativement petite. Notons que des valeurs élevées de  $m$  et  $W$  conduisent à une très bonne précision du modèle et qu'un ordre  $m$  faible ne pourrait être pertinent pour faire de la prédiction de série temporelle. Finalement, un compromis doit être trouvé car une faible complexité et une précision très élevée sont des objectifs antinomiques.

Dans le but de trouver ce compromis, nous proposons d'évaluer régulièrement la pertinence du modèle ARMA à travers une fenêtre glissante et d'adapter les coefficients des séries temporelles. Les paramètres cet *Adaptive-ARMA* sont :

- $W$ , la taille de la fenêtre glissante : les paramètres du modèle ARMA seront calculés sur les  $W$  derniers échantillons ;
- $th_{err}$  est le seuil d'erreur tolérable basé sur l'erreur moyenne (RMS - *root mean square error*) entre les données estimées et les données perçues.  $th_{err}$  représente la précision souhaitée du modèle ;
- $S$  est le pas de progression. Lorsqu'un capteur possède  $S$  échantillons, il calcule l'erreur moyenne entre ces échantillons et la prédiction du modèle. Si l'erreur RMS est trop importante (supérieure à  $th_{err}$ ) alors le nœud recalcule localement les nouveaux paramètres du modèle ARMA.

Les étapes de l'estimation adaptative des paramètres ARMA sont alors (se reporter également à la figure 4.24) :

1. Dès qu'un nœud a collecté  $W$  échantillons, il calcule localement le modèle  $ARMA(p, q)$  :  $p + 1$  paramètres sont calculés pour la partie AR et  $q$  paramètres pour la partie MA. Les paramètres du modèle sont transmis ainsi que les  $p$  premiers échantillons.
2. Régulièrement, le capteur mesure l'environnement. Dès qu'il a collecté  $S$  nouveaux échantillons, le capteur mesure l'erreur RMS entre ces nouveaux échantillons et les  $S$  valeurs prédites à partir du modèle précédent. Si l'erreur est inférieure au seuil toléré  $th_{err}$ , le modèle précédent est conservé et aucune transmission d'information n'est réalisée.
3. Si l'erreur dépasse le seuil  $th_{err}$ , alors le nœud recalcule les coefficients ARMA à partir des  $W$  échantillons les plus récents. Seuls ces nouveaux paramètres sont transmis vers le puits via la structure LEGOS et non pas les données physiquement mesurées.

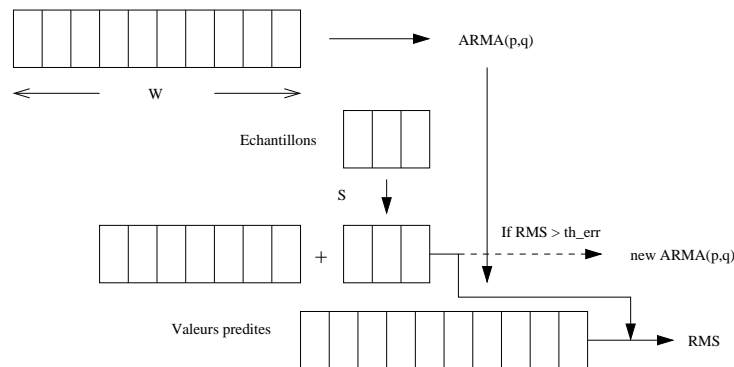


FIG. 4.24 – Adaptive ARMA

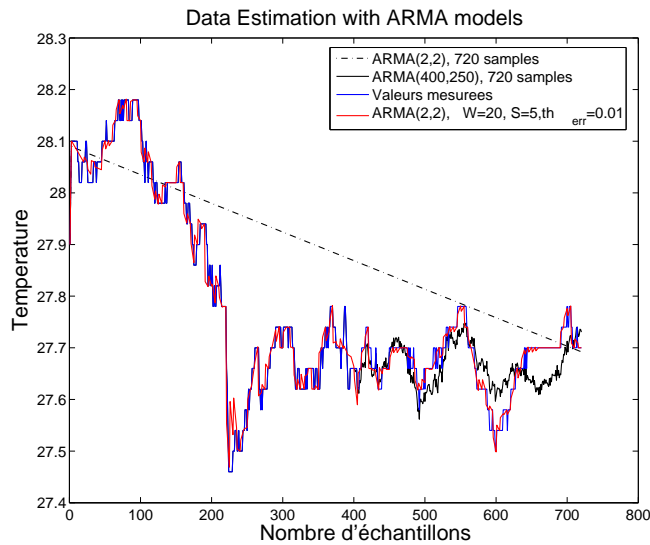


FIG. 4.25 – Précision du modèle vs données réellement mesurées (variation de température intérieure)

#### 4.5.4 Précision du modèle

La figure 4.25 fournit la précision du modèle pour plusieurs stratégies ARMA. Trois stratégies sont considérées : un modèle  $ARMA(2, 2)$ , un modèle  $ARMA(400, 250)$  et un modèle *adaptive-ARMA(2,2)* avec  $W = 20$ ,  $S = 5$  et  $th_{err} = 0.01$ . Les 720 données utilisées pour valider cette approche proviennent d'une campagne de mesure de température dans un environnement intérieur fait par ICT CAS EASINet [2]. De toute évidence, le modèle  $ARMA(2, 2)$  ne convient pas aux échantillons en notre possession, le résultat est inexploitable. En revanche le modèle  $ARMA(400, 250)$  est proche des échantillons prélevés mais la complexité du calcul est trop importante pour être exploité dans le cadre de réseaux de capteurs. Enfin, le modèle *adaptive-ARMA(2,2)* est très proche des données mesurées tout en offrant une complexité faible. Cette approche apparaît alors comme un compromis acceptable entre précision de la prévision et complexité de calcul.

#### 4.5.5 Analyse du coût de communication

Nous allons regarder combien d'informations sont échangées depuis les nœuds vers le puits en fonction du type de stratégie considérée. Nous excluons de la discussion la stratégie  $ARMA(2, 2)$  car sa précision est trop faible pour être retenue. Nous supposons que le transport des données observées par les capteurs requièrent autant de bits que le transport des paramètres des modèles ARMA. Dans le cas basique où tous les échantillons sont remontés vers le puits sans agrégation, le coût de communication est toujours à égal à 720 unités (720 données sont collectées). Dans le cas de du modèle  $ARMA(400, 750)$ , le coût de communication est égal au nombre de paramètres du modèles *i.e.*  $400 + 250 + 1 = 651$  unités. Dans le cas de la stratégie *adaptive-ARMA(2,2)*, le coût de communication est égal au nombre de mise-à-jour du modèle multiplié par le nombre de paramètres du modèle. Dans le pire cas, une mise à jour du modèle doit être envoyé tous les

$S = 720/5$  échantillons. Comme le modèle nécessite  $2 + 2 + 1 = 5$  paramètres, le coût total dans le pire cas est de 720 unités. En moyenne, le coût sera donc nécessairement plus faible : moins il y aura de mise à jour et plus le coût protocolaire sera faible. La quantité de données transmises sera alors fortement réduite.

La figure 4.26(a) fournit l'erreur RMS entre les données collectées et les prédictions en utilisant le modèle *adaptive-ARMA*. L'erreur RMS est donnée en fonction du pas de progression ( $S$ ) et du seuil  $th_{err}$ . Le compromis intéressant se situe dans la zone  $5 < S < 20$  et  $0.0 < th_{err} < 0.02$ . La figure 4.26(b) indique le nombre de mise-à-jour requise pour le modèle *adaptive-ARMA* en fonction du pas de progression ( $S$ ) et du seuil  $th_{err}$ .

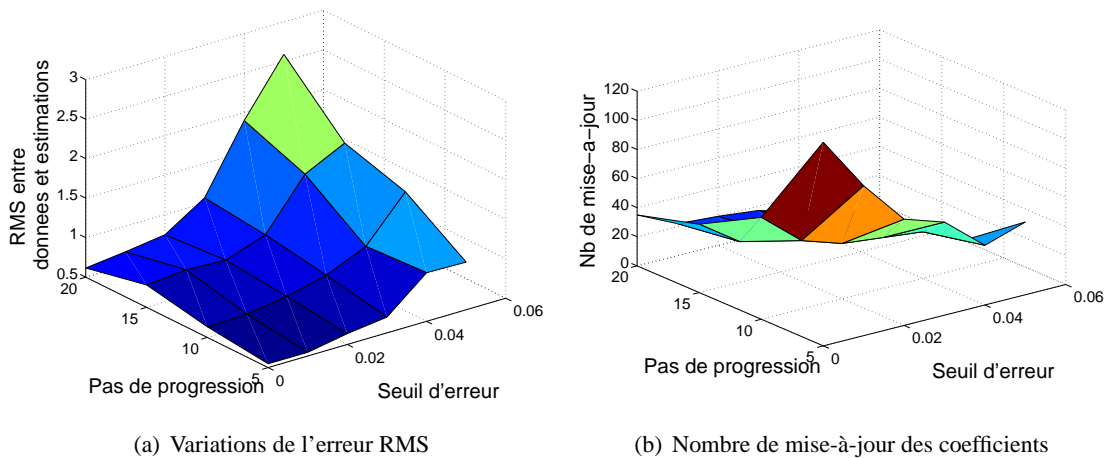
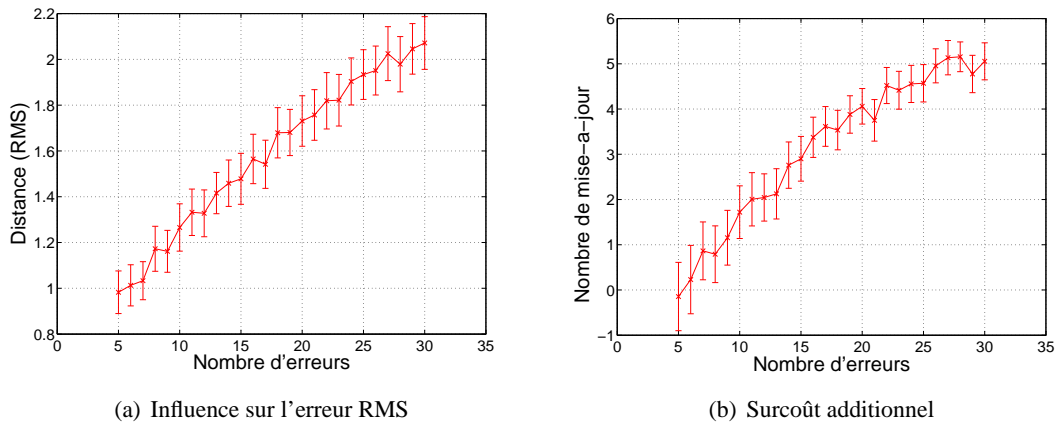


FIG. 4.26 – *adaptive-ARMA* : erreurs et mise-à-jour de coefficients

#### 4.5.6 Impact de mesures erronées

L'environnement dans lequel se situe les capteurs n'est pas forcément stable : des erreurs peuvent apparaître sur les mesures collectées. Cela revient à introduire des variations aléatoires sur les séries de données. Nous étudions ici l'impact de mesures erronées sur les performances d'*adaptive-ARMA* en termes de précisions et surcoût de communication. Nous avons étudié plusieurs types d'erreurs : *i*) des erreurs sporadiques et indépendantes se produisant sur les échantillons mesurés *ii*) des rafales d'erreurs de taille variant de 5 à 30 données erronées consécutives. Finalement quelque soit le type d'erreurs, les résultats exhibent un comportement similaire. Nous ne présentons ici que les performances d'*adaptive-ARMA* lors d'erreurs se produisant de façon indépendantes. Les figures 4.27(a) et 4.27(b) fournissent l'évolution des erreurs et le surcoût additionnel respectivement. Les erreurs se produisant de façon aléatoire, plus elles sont nombreuses et plus l'erreur est importante, générant un surcoût protocolaire associé. Notons que le surcoût protocolaire croît plus rapidement que l'erreur : plus les erreurs sont nombreuses et plus les mises-à-jour doivent être importantes. Néanmoins ce surcoût est tolérable. L'impact d'erreurs consécutives est légèrement différents : si les erreurs consécutives sont peu nombreuses, alors le surcoût est négligeable. Les valeurs erronées sont lissées par l'ensemble des échantillons de  $W$  et  $S$ . En revanche, s'il y a beaucoup d'erreurs consécutives, le surcoût devient alors important pour conserver une prédiction fiable.

FIG. 4.27 – Performances d'*adaptive-ARMA(2,2)* en présence d'erreurs

## 4.6 Synthèse

Dans ce chapitre, nous nous sommes intéressés au cas particulier de l'auto-organisation de réseaux de capteurs. Ici, pas de nœuds mobiles mais une contrainte forte sur le passage à l'échelle, la gestion de l'énergie et un contexte embarqué conduisant à une exigence de faible complexité algorithmique. Pour cela, nous avons banni l'utilisation d'informations de localisation (GPS ou assimilé), nous avons restreint l'utilisation de paquets périodiques du type `hello` et nous n'avons eu aucune exigence au regard de la synchronisation des nœuds. Enfin, nous avons considéré le réseau depuis son déploiement initial avec l'arrivée sporadique des nœuds jusqu'à sa stabilisation avec toute la topologie de déployée.

Nous avons introduit un protocole d'auto-organisation localisé asynchrone orienté événement : LEGOS. L'atout de LEGOS est de tirer partie d'un voisinage déjà organisé pour *i)* réduire la latence de l'auto-organisation et *ii)* maintenir une faible cardinalité des nœuds actifs dans le réseau. Deux versions de LEGOS sont proposés : sous forme d'arbre ou de treillis pour s'adapter aux exigences des applications. Basé sur cette auto-organisation, nous avons étudié plusieurs mécanismes critiques pour les réseaux de capteurs : allocation d'adresses, dissémination de données, gestion de puits multiples et mobiles, agrégation de données. Pourquoi critiques ? Car ces mécanismes exigent la plupart un lourd processus de communication, de connaissance du voisinage, etc. conduisant à une consommation importante des ressources, en particulier énergétique. A chaque fois, nous avons pu tirer parti de l'auto-organisation LEGOS pour réduire le nombre de messages, contrôler la dépense en énergie. Nous pensons que l'approche proposée ici s'avère pertinente dans ce cadre particulier des réseaux de capteurs.



# Bibliographie

- [1] K. Heurtefeux and Valois F. Topology control algorithms : a qualitative study during the sensor networks life. In *3rd International Workshop on Localized Communication and Topology Protocols for Ad hoc Networks (LOCAN'07)*, in conjunction with MASS, Pisa, Italy, October 2007.
- [2] J.-L. Lu and F. Valois. On the data dissemination in wsns. In *3rd International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, New-York, USA, October 2007.
- [3] J.-L. Lu, F. Valois, D. Barhel, and M. Dohler. Low-energy address allocation scheme for wireless sensor networks. In *IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, Athens, Greece, September 2007.
- [4] K. Heurtefeux and Valois F. Self-organisation protocols : Behavior during the sensor network life. In *IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, Athens, Greece, September 2007.
- [5] M. Dohler, D. Barthel, S. Aubert, C. Dugas, F. Maraninchi, L. Mounier, A. Buhrig, F. Paugnat, M. Renaudin, A. Duda, M. Heusse, and F. Valois. The areza project : Facilitating research, development and commercialization of wsns. In *4th Annual IEEE Communications Society Conference on Sensor, Mesh, and Ad Hoc Communications and Networks (SECON)*, San Fransisco, USA, June 2007.
- [6] M. Dohler, T. Watteyne, D. Barthel, F. Valois, and J.-L. Lu. Kumar, zipf and other laws : How to structure an optimum large-scale wireless (sensor) network ? In *13th European Wireless Conference*, Paris, France, April 2007.
- [7] J.-L. Lu, F. Valois, D. Barthel, and M. Dohler. A fully integrated scheme of self-configuration and self-organization for wsn. In *IEEE Wireless Communications and Networking Conference (WCNC)*, Hong-kong, March 2007.
- [8] J.-L. Lu, F. Valois, and D. Barhel. Low-energy self-organization scheme for wireless ad hoc sensor network. In *4th Annual Conference on Wireless On demand Network Systems and Services (WONS)*, Obergurgl, Austria, January 2007.
- [9] J.-L. Lu, F. Valois, and D. Barthel. Range adjustment for broadcast protocols with a realistic radio transceiver energy model in short-range wireless networks. In *International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, Wuhan, China, December 2005. Vol. LNCS 3794/2005.



## Chapitre 5

# Modélisation et évaluation de performances de couches MAC

### 5.1 Introduction et motivations

L'étude des protocoles de niveau 3 dans le domaine des réseaux de capteurs et des réseaux ad hoc conduit à ce constat : les protocoles de niveau 3 peuvent être aussi performants que possible, si la couche MAC permettant l'accès au canal n'affiche pas de *bonnes* propriétés, les performances finalement obtenues seront médiocres. Qu'entendons-nous par *bonnes* propriétés ? Dans le domaine quantitatif, il est important de maximiser le taux d'utilisation du médium, le débit utile et de minimiser les collisions. Dans le domaine qualitatif, la mesure clef est l'équité : garantir à chaque station le *même* accès au canal. Il nous faut différencier l'équité à court terme de l'équité à long terme, les deux n'étant pas nécessairement liées. A court terme, cela signifie qu'une station voisine peut monopoliser le médium pour transmettre une rafale de paquets tandis qu'à long terme cela signifie qu'une station accède, en moyenne, plus souvent au canal qu'une autre. De nombreux travaux se sont penchés sur l'amélioration du protocole de gestion de collisions de 802.11 (BEB - *Binary Exponential Backoff*) dans le cas multi-sauts en proposant une solution parfois performante, parfois équitable. A chaque fois un compromis doit être trouvé.

Dans le domaine ad hoc, la couche MAC de référence est celle de IEEE 802.11 [58] : *Carrier Sense Multiple Access / Collision Avoidance* (CSMA/CA). propose l'évaluation de performances de référence mais adaptée au cas infrastructure uniquement et non extensible vers des cas multi-sauts. [35] s'est longuement intéressée à l'(in)-équité des réseaux ad hoc : plusieurs topologies de référence ont été mis en avant comme le cas des 3 paires (figure 5.1(a)), les nœuds cachés (figure 5.1(b)), les nœuds transitifs<sup>1</sup> (figure 5.1(c)). Ces topologies reflètent les situations emblématiques d'iniquité dans les réseaux radio multi-sauts. Les travaux menés couvrent alors le domaine de l'expérimentation, de la simulation et de la modélisation analytique en utilisant une classique chaîne de Markov. Toutefois, la modélisation proposée [25] ne permet d'adresser qu'une seule topologie pour une taille de paquets donnée et pour un débit fixé. La question qui se pose alors est : comment étudier ces différentes topologies en tenant compte des différents débits possibles, de différentes tailles de paquets, de différentes stratégies de gestion de collisions, etc. ? En observant attentivement la littérature dans ce domaine, on s'aperçoit, qu'une fois de plus, la simulation de réseaux semble être la panacée. Toutefois, gardons à l'esprit l'ensemble des griefs qui peuvent

<sup>1</sup>Parfois nommé la face cachée des nœuds cachés par l'auteur.

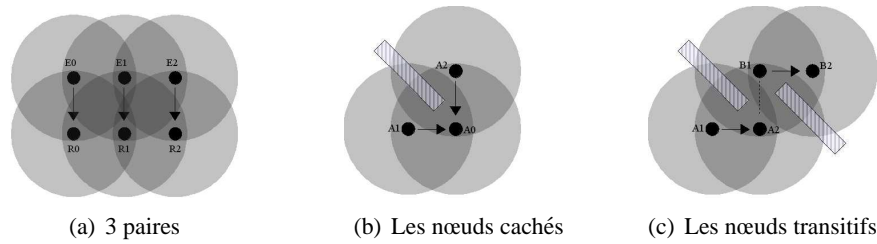


FIG. 5.1 – Les principales topologies étudiées.

être formulés [41] en se souvenant que le temps de calcul est particulièrement long : pour les topologies traitées un peu plus loin dans ce manuscrit quelques heures de simulation ont parfois été nécessaires. Cela nous motive d'autant plus à nous tourner vers les outils de modélisation et d'évaluation de performances analytiques. La question du choix du paradigme de modélisation est alors déterminante.

## 5.2 PEPA, un algèbre de processus stochastique

### 5.2.1 Pourquoi utiliser PEPA ?

Le choix d'un outil de modélisation doit être dicté par ce que l'on veut mesurer, ce que l'on attend et les propriétés du système étudié. Ici, l'objectif est clairement affiché : basé sur un modèle analytique, nous devons être capable de mesurer l'utilisation du médium en termes du taux de transmission effectivement réussie, du taux de collisions sans oublier la mesure de l'équité à court terme. À l'aide de ce modèle, nous devons être capable d'étudier plusieurs topologies<sup>2</sup>, plusieurs débits<sup>3</sup>, plusieurs tailles de paquets<sup>4</sup>, plusieurs algorithmes de gestion de collisions<sup>5</sup>. Mais aussi de considérer plusieurs modèles de canal pour étudier l'influence du médium sur les performances<sup>6</sup> voire plusieurs techniques d'accès au médium<sup>7</sup>.

Il nous semble que le choix de PEPA (*Performance Evaluation Process Algebra*) [38], un algèbre de processus stochastique dédié à l'évaluation de performances de systèmes et de réseaux, s'impose. Pourquoi PEPA et pas un autre formalisme de modélisation ? L'utilisation de chaînes de Markov ne permet pas d'appréhender la généralité du modèle souhaité car une nouvelle chaîne doit être construite pour chaque nouvelle topologie. A notre connaissance, deux autres formalismes auraient pu être éligibles : les réseaux de Petri stochastiques et les réseaux d'automates stochastiques (SAN). Dans chaque cas une approche compositionnelle, générique pouvait être envisagée. Ainsi à l'aide de SAN, un automate stochastique aurait pu être associé à chaque élément du réseau (algorithme de backoff, modèle de trafic, etc.) et à l'aide des opérateurs de Kronecker, la modélisation du système global aurait pu être envisagée. Toutefois, la difficulté ne réside pas nécessairement dans la modélisation mais dans la résolution du modèle. Or, quelque soit l'approche utilisée, elle nous conduit à une explosion de l'espace d'états que seule une technique

<sup>2</sup>Les 3 paires, les nœuds cachés, les nœuds transitifs, la chaîne de transmission, le mode infrastructure, etc.

<sup>3</sup>1, 2, 5.5 et 11Mbit/s

<sup>4</sup>de 100 à 1500 octets.

<sup>5</sup>Le classique BEB [58] mais aussi ses variantes DIDD [24], MILD [14], etc.

<sup>6</sup>Modèle d'erreurs à seuil, modèle de Rayleigh, modèle d'erreurs corrélés.

<sup>7</sup>Le classique CSMA mais aussi TDMA.

comme les bornes stochastiques ou les techniques d'agrégation permettraient de lever. La difficulté dans l'utilisation des bornes stochastiques n'est pas de construire un majorant et un minorant mais de maîtriser la qualité des bornes. Les techniques d'agrégation ont pour objectif de grouper en macro-états des états stochastiquement équivalents : dans des chaînes de plusieurs centaines de milliers d'états, identifier de tels états devient alors le principal problème.

Toutes ces raisons nous amènent à penser que PEPA est le formalisme répondant au mieux à notre problématique. PEPA, comme tout algèbre, est basé sur une approche compositionnelle liée aux opérateurs du langage que nous décrirons. Ce qui fait la force de PEPA c'est l'utilisation d'une méthode d'agrégation *automatique*, i.e. l'existence de l'opérateur modulo permet d'identifier les états équivalents à un modulo près. Ainsi, l'utilisation de PEPA nous permet-elle de construire un modèle générique en tirant partie de son approche compositionnelle et l'explosion de l'espace d'états est solutionnée par les techniques d'agrégation *automatique*.

### 5.2.2 Introduction au formalisme

Une introduction à PEPA et à sa technique d'agrégation peut être trouvée dans [79] qui incluent quelques exemples de files classiques. Nous allons nous concentrer ici sur les opérateurs du langage qui nous seront utiles. PEPA étend les classiques algèbres de processus en associant à chaque action une variable aléatoire distribuée représentant la durée de l'action. Comme cette variable aléatoire suit une distribution exponentielle, cela permet d'établir un lien évident avec les processus markoviens à temps continu. Le lecteur déjà familier avec PEPA peut passer cette section et aller directement dans la section 5.3.

**Notions de base** Les éléments de base de PEPA sont les *composantes* et les *activités* correspondant aux états et aux transitions d'une CMTC. Nous supposons qu'il existe un ensemble dénombrable de composantes possibles  $\mathcal{C}$ . Chaque composante  $P$  possède un ensemble  $Act(P)$  d'activités. Chaque activité possède un *type d'action* appartenant à l'ensemble dénombrable  $\mathcal{A}$  contenant tous les types. Lorsque le type de l'action est inconnu, il sera représenté par un type d'action particulier,  $\tau$ , vu comme un type non spécifié. Ces activités sont privées à la composante dans laquelle elles apparaissent.

La durée de chaque activité est représentée par un paramètre exponentiellement distribué : le *taux* de l'activité. Ce paramètre est un nombre réel positif, ou le symbole particulier  $\top$  si la valeur n'est pas spécifiée (dans le cas de synchronisations). Ainsi, chaque activité  $a$  est décrite par une paire  $(\alpha, r)$  où  $\alpha \in \mathcal{A}$  est le type de l'action et  $r$  est le taux de l'activité. Un système est donc modélisé par un ensemble de *composantes* qui interagissent entre elles. Elles s'engagent alors dans des *activités* individuellement ou à plusieurs.

**Les opérateurs de PEPA** PEPA fournit un ensemble réduit d'opérateurs. Ils permettent de construire des expressions définissant le comportement des composantes, via les activités et les interactions considérées. Nous allons donc présenter ces opérateurs et leur interprétation. Pour les études qui nous intéressent, nous avons :

$$S \stackrel{\text{def}}{=} P \mid (\alpha, r).P \mid P + Q \mid P \boxtimes_L Q$$

**Constante :**  $S \stackrel{\text{def}}{=} P$  On affecte à la composante  $S$  le comportement de la composante  $P$ . C'est grâce à cette composante que nous pouvons définir ou affecter le comportement d'une composante ou d'un système.

**Préfixe :**  $S \stackrel{\text{def}}{=} (\alpha, r).P$  C'est le mécanisme de base par lequel les comportements des composantes sont spécifiés. La composante s'engage dans l'activité  $(\alpha, r)$  puis se comporte comme la composante  $P$ , habilitant toutes les activités dans  $\text{Act}(P)$ . Soulignons qu'il existe toujours une ressource implicite qui facilite les activités de la composante et qui n'est pas modélisée explicitement.

**Choix :**  $S \stackrel{\text{def}}{=} P + Q$  Le système  $S$  se comportera soit comme la composante  $P$  soit comme la composante  $Q$ . De ce fait, toutes les activités courantes de  $P$  et  $Q$  sont habilitées pour  $S$ . Notons que le choix est toujours *vrai* : si  $P$  et  $Q$  exécutent la même activité, on distinguera alors les instances de cette activité. De plus, la nature continue des distributions de probabilités garantit que la probabilité que  $P$  et  $Q$  exécutent en même temps une activité est nulle. Il est intéressant de voir ici que le choix représente une compétition entre les composantes :  $P$  et  $Q$  concourent pour la même ressource implicite.

**Coopération :**  $S \stackrel{\text{def}}{=} P \underset{L}{\bowtie} Q$  L'ensemble  $L$  ( $L \subseteq \mathcal{A}$ ) est l'union des types d'action sur lesquels  $P$  et  $Q$  doivent se synchroniser pour coopérer. Une nouvelle activité partagée est alors formée par la coopération de  $P$  et  $Q$ , remplaçant les activités individuelles par des activités ayant le même type d'action, mais ayant un taux correspondant au temps de la composante la moins rapide. Notons que chaque composante en coopération a sa propre ressource implicite. De plus, l'ensemble  $L$  pourra être utilisé pour identifier les verrous mortels. Si une activité est de taux  $\top$  alors la composante est passive et ne contribue pas à la coopération, même si elle est nécessaire. Enfin, lorsque  $L = \emptyset$ ,  $\underset{L}{\bowtie}$  a un effet de composition parallèle permettant aux composantes d'agir en concurrence sans interaction entre elles :

$$P \underset{L=\emptyset}{\bowtie} Q \equiv P||Q$$

Tous ces opérateurs ne possédant pas la même priorité d'exécution, la règle de précedence suivante est appliquée :

— Préfixe — Coopération — Choix

### 5.2.3 Graphe de transitions et propriétés

#### Graphe de transitions

Pour représenter les différents comportements possibles d'une composante dans PEPA, il est possible de construire un diagramme de transitions. Chaque nœud traduit un comportement de la composante et chaque transition a pour étiquette l'activité qu'elle représente. La chaîne de Markov sous-jacente est alors évidente.

### Propriétés

Si nous envisageons un graphe dans lequel les nœuds et les arcs représentent les transitions possibles entre les objets du langage, alors les règles définissant le comportement d'un système régissent le modèle du graphe. Comme PEPA distingue les différentes instanciations de la même activité, alors le graphe considéré est un multigraphe. Le graphe obtenu est alors le *graphe de dérivation* décrivant les possibles comportements de toutes les composantes PEPA. Dans la suite, nous définissons les notions de dérivée, d'ensemble de dérivée et de graphe de dérivation.

**Définition 1** Si  $P \xrightarrow{(\alpha,r)} P'$ , alors  $P'$  est une dérivée de  $P$ . Plus généralement, si  $P \xrightarrow{(\alpha_1,r_1)} \dots \xrightarrow{(\alpha_n,r_n)} P'$ , alors  $P'$  est une dérivée de  $P$ .

Les dérivées sont les labels des états du graphe de transitions du système. Pour chaque composante PEPA, l'ensemble des dérivées peut être obtenu récursivement depuis la composante initiale.

**Définition 2** L'ensemble des dérivées d'une composante  $C$  est notée  $ds(C)$  et est définie comme le plus petit ensemble des composantes tel que :

- Si  $C \stackrel{\text{def}}{=} C_0$  alors  $C_0 \in ds(C)$  ;
- Si  $C_i \in ds(C)$  et il existe  $a \in Act(C_i)$  telle que si  $C_i \xrightarrow{a} C_j$ , alors  $C_j \in ds(C)$ .

L'ensemble des dérivées reflète tous les états atteignables du système. Nous avons vu précédemment que le graphe de transition permet de visualiser les états possibles du système et les relations entre eux. En termes d'ensemble de dérivées du système, nous obtenons alors le *graphe de dérivation*.

**Définition 3** Étant donné une composante  $C$  et son ensemble de dérivées  $ds(C)$ , le graphe de dérivation  $D(C)$  est un multigraphe orienté dont :

- l'ensemble des nœuds est  $ds(C)$ ,
- l'ensemble des arcs  $M$  est composé des éléments pris dans l'ensemble  $ds(C) \times ds(C) \times Act$ .

#### 5.2.4 Le modèle stochastique

A partir du *graphe de dérivation*, il est alors possible de générer le processus Markovien représentant le modèle. Le *graphe de dérivation* est le multigraphe dont chaque composante dérivée est un nœud du graphe obtenu depuis le graphe de départ. L'état initial est la composante définissant le système. Les arcs entre deux nœuds portent comme étiquette le type de l'action et le taux de l'activité :  $(\alpha, r)$ . Ce graphe de dérivation décrit le comportement possible de chaque composante PEPA et fournit un support pour étudier le modèle. Il est alors aisé d'associer un processus stochastique tel que :

- à chaque nœud du graphe on associe un état,
- les *transitions* entre états sont définies par les arcs du graphe,
- le *taux de transition* est égal à la somme des taux d'activités sur les arcs reliant les nœuds correspondants dans le graphe.

Nous notons  $ds(C)$  l'ensemble de dérivation de la composante  $C$  dans PEPA : c'est l'ensemble de toutes les composantes que l'on peut atteindre à partir de  $C$  (témoignant donc du comportement de  $C$ ). Enfin un théorème important permet de valider le processus de Markov (d'après [38]) :

**Théorème 1** Pour n'importe quel modèle (fini)  $C \stackrel{\text{def}}{=} C_0$  de PEPA, si nous définissons le processus stochastique  $X(t)$  tel que  $X(t) = C_i$  indique que le système se comporte comme la composante  $C_i$  à l'instant  $t$ , alors  $X(t)$  est un processus de Markov à temps continu.

Le taux de transition entre deux composantes  $C_i$  et  $C_j$ , noté  $q(C_i, C_j)$ , est le taux faisant passer le système de la composante  $C_i$  à la composante  $C_j$ . Ce taux correspond aux éléments non-diagonaux du générateur infinitésimal du processus de Markov. Nous avons alors :

$$q(C_i, C_j) = \sum_{a \in \text{Act}(C_i|C_j)} r_a$$

où  $\text{Act}(C_i|C_j) = \{a \in \text{Act}(C_i) / C_i \xrightarrow{a} C_j\}$

Le temps moyen de séjour du système dans l'état associé à la composante  $C$  est donné par :

$$\left( \sum_{a \in \text{Act}(C)} r_a \right)^{-1}$$

Et enfin, le taux de sortie de l'état du système associé à la composante  $C$  est :

$$q(C) = \sum_{a \in \text{Act}(C)} r_a$$

### 5.2.5 Résolution et calcul de performances

Après avoir modélisé un réseau à l'aide de composantes et d'activités traduisant le comportement stochastique du système, nous pouvons alors générer le graphe de dérivation permettant d'étudier son caractère dynamique. Bien entendu, une telle modélisation a comme objectif premier de calculer des critères de performances, appelés *récompenses*. Pour ce faire, il faudra dans un premier temps construire le générateur infinitésimal  $Q$  du processus de Markov associé. Ensuite, en notant que  $\Pi(C_i)$  est la probabilité que le système se comporte comme la composante dérivée  $C_i$ , la distribution stationnaire  $\Pi(\cdot)$  sera solution du système suivant :

$$\left\{ \begin{array}{l} \Pi Q = 0 \\ \sum_{C_i \in ds(C_0)} \Pi(C_i) = 1 \end{array} \right.$$

Le calcul des récompenses se fait par rapport aux états du processus ou aux transitions entre ces états. Puisque le comportement dans PEPA est associé aux activités et non pas aux états, nous devons donc définir une structure de récompense basée sur les activités du modèle. La mesure de récompenses associée à une composante, donc à un état correspondant, est alors la somme des récompenses attachées aux activités que cette composante habilite. Si nous notons  $\rho_i$  la récompense associée à la composante  $C_i$ , et  $\rho(a)$  la récompense associée à l'activité  $a$ , alors :

$$\rho_i = \sum_{a \in \text{Act}(C_i)} \rho_a$$

La structure des récompenses dans PEPA a donc été développée au niveau du modèle et non pas au niveau du processus markovien sous-jacent. Remarquons que la structure des récompenses est similaire à ce qui est fait dans les réseaux d'automates stochastiques.

### 5.2.6 L'agrégation exacte

Un avantage de PEPA est de proposer une agrégation automatique basée sur la notion d'équivalence forte. Ainsi PEPA permet, pour un modèle donné agrégeable, de réduire la taille de son espace d'états. La difficulté des méthodes d'agrégation est de trouver une partition juste, vérifiant les conditions d'agrégation. PEPA construit ainsi, et grâce à l'équivalence forte, une telle partition. Bien entendu, la partition obtenue n'est pas forcément la *meilleure* au sens qu'elle n'est pas obligatoirement celle qui engendre l'espace d'états le plus faible.

#### Notion d'équivalence forte

La notion d'équivalence forte est basée sur le taux de transition conditionnel total défini par :

$$q[C_i, C_j, \alpha] = \sum_{C_j \in T} q(C_i, C_j, \alpha)$$

Nous pouvons maintenant donner la définition d'une équivalence forte :

**Définition 4** Une relation d'équivalence sur les composantes, notée  $\mathfrak{R} \subseteq \mathcal{C} \times \mathcal{C}$  est une équivalence forte si  $\exists (P, Q) \in \mathfrak{R}$  alors  $\forall \alpha \in A$  et  $\forall S \in \mathcal{C}/\mathfrak{R}$  :

$$q[P, S, \alpha] = q[Q, S, \alpha]$$

où  $\mathcal{C}/\mathfrak{R}$  est l'ensemble des classes d'équivalence induites par  $\mathfrak{R}$  sur  $\mathcal{C}$ .

Étant donné que la manipulation de l'outil d'agrégation automatique peut s'avérer lourde et complexe, nous donnons ici les définitions essentielles à sa compréhension et à sa bonne manipulation. Elles seront suivies de la présentation de la méthode d'agrégation compositionnelle.

#### Définitions et propriétés

**Définition 5**  $P$  et  $Q$  sont fortement équivalentes, notée  $P \cong Q$ , si  $(P, Q) \in \mathfrak{R}$ , pour une relation d'équivalence forte donnée, i.e. :

$$\cong = \bigcup \{ \mathfrak{R} \mid \mathfrak{R} \text{ est une équivalence forte} \}$$

**Proposition 1**  $\cong$  est l'équivalence forte la plus large.

La relation d'équivalence étant une congruence, nous avons alors les propriétés suivantes :

- si  $P_1 \cong P_2$  alors
  - $a.P_1 \cong a.P_2$
  - $P_1 + Q \cong P_2 + Q$
  - $P_1 \bowtie Q \cong P_2 \bowtie Q$
  - $P_1/\tilde{L} \cong P_2/\tilde{L}$
- Soient  $E$  et  $F$ , deux expressions contenant les variables  $\tilde{X}$ , si  $E \cong F$  alors pour tout ensemble de composantes  $\tilde{P}$ , nous avons :

$$E\{\tilde{P}/\tilde{X}\} \cong F\{\tilde{P}/\tilde{X}\}$$

Les propriétés de l'équivalence forte peuvent être associées aux composantes PEPA d'un système. Soient  $P$  et  $Q$  deux composantes systèmes. Alors, si  $P \cong Q$  :

- $\mathcal{A}(P) = \mathcal{A}(Q)$  : même ensemble de types d'actions,
- $q(P) = q(Q)$  : même taux de sortie,
- $q(P, \alpha) = q(Q, \alpha)$  : même taux de sortie conditionnel,
- $p[P, S, \alpha] = P[Q, S, \alpha]$  et  $p[P, S] = P[Q, S]$ .

Si  $P$  et  $Q$  sont fortement équivalentes ( $P \cong Q$ ), alors ces composantes sont capables d'exhiber les mêmes schémas de comportement.

PEPA utilise donc cette notion d'équivalence forte appliquée aux composantes pour proposer une agrégation *automatique*, dans le sens où nous n'intervenons pas pour chercher une partition vérifiant le théorème de Kemeny-Snell. Ainsi, elle tire partie :

- des symétries et des schémas de comportement répétitifs de l'ensemble de dérivation  $ds$ ,
- du fait que l'équivalence forte est une congruence.

Considérant que  $P$  est une composante PEPA, nous définissons alors :

- $ds(P)/\cong$  : ensemble des classes d'équivalence sur l'ensemble de dérivation de  $P$ ,
- $T$  : classe d'équivalence de cet ensemble ( $T \in ds(P)/\cong$ ).

Le graphe de dérivation introduit précédemment, que l'on note  $D_{\cong}(P)$ , orienté et étiqueté, a l'ensemble de ses nœuds dans  $ds(P)/\cong$  et l'ensemble des arcs  $A_{\cong}$  est défini comme suit :

- les éléments de  $A_{\cong}$  sont pris dans :  $ds(P)/\cong \times ds(P)/\cong \times \vec{Act}_{\cong}(S)$
- l'arc  $(T_i, T_j, (\alpha, q')) \in A_{\cong}$  si  $(\alpha, q') \in Act_{\cong}(T_i)$  et  $q' = q[Q_i, T_j, \alpha], \forall Q_i \in T_i$
- le nœud  $T_0$ , où  $P \in T_0$ , est le nœud initial du graphe.

### Agrégation compositionnelle

Puisque PEPA est basé sur une description compositionnelle d'un système, il en est de même pour la méthode d'agrégation. L'idée est de remplacer les composantes qui coopèrent par des composantes fortement équivalentes, réduisant ainsi l'espace d'états. La méthode d'agrégation automatique repose sur les six étapes de l'algorithme 1.

## 5.3 Modélisation et évaluation de performances de couches MAC

Reformulons l'objectif de départ : mesurer les performances et l'équité d'algorithmes de backoff dans des topologies multi-sauts et en mode infrastructure soumis à différentes hypothèses de trafic, différentes hypothèse de canal radio, etc. L'objectif est de comprendre comment évolue le compromis performances-équité voire d'émettre des recommandations pour la conception de couches MAC pour réseaux multi-sauts. Le cœur du travail est d'étudier différents algorithmes de gestion de collisions (backoff), de comparer les différentes stratégies existantes. Deux contributions nous paraissent essentielles ici : l'approche de modélisation générique tirant partie de PEPA et les résultats tirés de ces modèles.

### 5.3.1 Méthodologie de modélisation

La modélisation d'un réseau sans fil multi-sauts en utilisant PEPA va nous conduire à la définition de plusieurs composantes qui interagissent entre elles pour modéliser le comportement du réseau. L'identification des composantes et synchronisations nécessaires pour permettre une

**Algorithm 1** Étapes de l'agrégation automatique

1. Construire le modèle PEPA du système  $S$  à modéliser.
2. Identifier les composantes atomiques du modèle qui coopèrent,  $X$  et  $Y$ . Appliquer l'agrégation sur la coopération de ces composantes,  $P = X \bowtie_L Y$ , pour former  $\hat{P}$  :
  - (a) Former l'ensemble de dérivation de la composante  $P$ ,  $ds(P)$ .
  - (b) Former les classes d'équivalence à partir de  $ds(P)$  et former l'ensemble de dérivation au sens *lumpability*  $ds(P)/\cong$ .
  - (c) Former l'ensemble des activités au sens *lumpability* pour chaque  $T \in ds(P)/\cong$  et construire le graphe de dérivation au sens *lumpability*  $D_{\cong}(P)$ .
  - (d) Construire la composante  $\hat{P}$ , basée sur  $D_{\cong}(P)$  et remplacer  $P$  par  $\hat{P}$ .
 Répéter avec toutes les paires de composantes atomiques.
3. Considérer le prochain niveau du modèle, i.e.  $Q \bowtie_K \hat{P}$ . Répéter les étapes 2(a) - 2(d) sur  $Q \bowtie_K \hat{P}$ . Continuer jusqu'à construction complète de  $\hat{S}$ .
4. A partir du graphe de dérivation de  $\hat{S}$  (graphe de dérivation de  $S$  construit précédemment), construire le processus de Markov agrégé.
5. Affecter les récompenses aux états du processus.
6. Résoudre le processus de Markov agrégé et calculer les récompenses.

démarche générique<sup>8</sup> est donc une étape clef. Le nombre et le rôle de chacune des composantes est issu de l'observation attentive d'un réseau sans fil. Finalement, la question que nous devons nous poser est : qu'est-ce qui identifie clairement un réseau sans fil ? De toute évidence un réseau est défini par :

- la topologie du réseau, le nombre de nœuds, les interactions entre les nœuds,
- la définition de l'environnement radio : sommes-nous en présence d'une couche physique idéale ? Les erreurs survenant sont-elles indépendantes ou corrélées ?
- l'architecture de communication de chaque nœud qui peut elle-même se diviser :
  - le générateur de trafic,
  - une file d'attente pour stocker les paquets dans l'attente que le médium soit libre,
  - la méthode d'accès au médium
  - l'algorithme de gestion de collisions.

**Composantes associées à un réseau sans fil**

Nous associerons une composante pour chacun des 6 éléments précédemment listés (figure 5.2).

**Composante *Topology* - Topologie du réseau** Un réseau est défini par les interactions entre les nœuds au travers du médium. Suivant la topologie considérée, les performances mesurées sont différentes et l'efficacité de l'algorithme de gestion de collisions aussi. Avoir une composante

<sup>8</sup>i.e. une démarche dans laquelle une composante peut en remplacer une autre pour étudier par exemple une nouvelle topologie ou un nouvel algorithme de gestion de collisions.

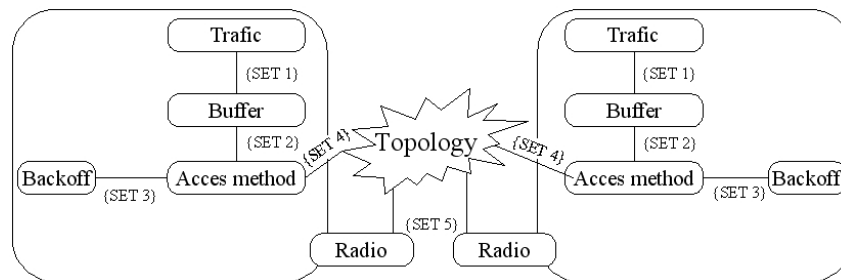


FIG. 5.2 – Modélisation générique d'un réseau radio : ensemble des composantes et de leurs interactions

pour décrire les interactions entre les nœuds, c'est autoriser l'étude de plusieurs topologies tant que l'on respecte les ensembles de synchronisation définis plus loin. Ainsi il nous sera possible d'étudier les nœuds cachés, le mode infrastructure, etc. sans toucher aux autres composantes. Plus précisément, cette composante traduira le partage du médium entre les nœuds.

**Composante *Radio* - L'environnement radio** Classiquement, les études analytiques supposent un modèle de réception à seuil, sans interférence, sans erreur corrélée : bref, une couche physique idéale. De toute évidence dans un environnement radio réaliste, il nous faut tenir compte des interférences, de l'affaiblissement, de l'effet de masque, etc. Grâce à l'introduction d'une composante décrivant les propriétés du médium radio en termes d'erreurs (en rafales, corrélées, etc.), d'affaiblissement, etc. il nous sera possible de répondre à la question : comment influe l'environnement radio sur l'équité mais aussi sur les performances du réseau ?

**Composante *Trafic* - Générateur de trafic** La plupart des études analytiques fonctionnent en régime saturé. Ainsi la plupart des résultats connus le sont dans le cas de nœuds ayant toujours un paquet à transmettre. Ce cas est donc le pire cas mais la question reste ouverte : comment se comporte l'équité dans le cas de sources de trafic non saturées avec des arrivées de paquets de type ON/OFF par exemple ? Cette composante permettra, pour chaque nœuds, de décrire le modèle de trafic associé. La taille des paquets générés correspond à des paquets de données avec les en-têtes MAC / IP / UDP.

**Composante *Buffer* - Buffer** Si le médium est occupé alors les paquets générés doivent être mis en attente avant transmission. Si le buffer arrive à saturation, les nouveaux paquets générés seront donc perdus. En fait, plus qu'un buffer classique, cette composante modélise le buffer présent dans la couche LLC du niveau 2, avant transmission par la couche MAC.

**Composante *Access Method* - Méthode d'accès au médium** Une fois le paquet généré, il doit être transmis sur l'interface radio. Dans les solutions de type 802, la technique d'accès est toujours de type CSMA. Toutefois, l'étude de couche MAC de type TDMA (*Time Division Multiple Access*) nous permettra de nous intéresser à un système avec un ordonnancement parfait entre les nœuds, *i.e.* équitable.

**Composante *Backoff* - Algorithme de gestion de collisions** <sup>9</sup> Cette composante est au cœur du modèle. C'est à travers cette composante que BEB sera modélisé et que nous pourrons étudier plusieurs stratégies de gestion de collisions. C'est clairement la composante qui aura la plus grande influence sur les performances obtenues car nous décrirons à la fois la stratégie suivie dans le cas d'une transmission réussie mais aussi dans le cas de collision.

### Synchronisations requises

La figure 5.2 décrit l'organisation des différentes composantes précédemment décrites ainsi que les interactions (synchronisations) pouvant se produire. Une composante pourra être aisément remplacée par une autre ou être supprimée tant que les contraintes de synchronisations énoncées ci-après sont respectées.

- {SET 1} décrit les interactions entre la composante TRAFIC et la composante BUFFER : c'est le processus de stockage des paquets dans la file d'attente.
- {SET 2} décrit les interactions entre la composante BUFFER et la composante ACCESS METHOD : c'est le début de la transmission d'un paquet qui était en attente.
- {SET 3} : deux actions sont définies pour synchroniser les composantes BACKOFF et ACCESS METHOD dans le but de faire évoluer la fenêtre de contention en cas de transmission réussie ou de collision.
- {SET 4} : pour synchroniser la composante ACCESS METHOD et la composante TOPOLOGY, trois actions sont définies : *i*) une première action pour écouter le canal avant émission *ii*) une action de la part du médium pour indiquer si la transmission a réussi *iii*) une action de la part du médium pour indiquer si la transmission a échoué.
- {SET 5} est utilisé pour générer des erreurs paquets lors des transmissions suivant des lois d'arrivées connues.

Notons qu'aucun récepteur n'a été modélisé : c'est le médium, à travers la composante TOPOLOGY qui reçoit tous les paquets et qui est donc capable de spécifier une transmission réussie ou une collision. Le médium est donc un point de synchronisation de tous les nœuds de la topologie qui sont en interaction. Il est également possible de modéliser plusieurs médiums, en particulier dans le cas d'études de topologies sur plusieurs sauts consécutifs. L'étude présentée ici n'aborde pas ce point particulier.

### 5.3.2 Le modèle de base

Pour bien comprendre l'aspect générique du modèle, nous allons commencer par modéliser un réseau simple. Puis nous chercherons à modifier ce modèle pour étudier un réseau multi-sauts, à nous intéresser aux effets d'erreurs paquets corrélées et à faire tomber l'hypothèse de régime saturé, etc.

### 5.3.3 Description du modèle de base

La topologie étudiée correspond au fonctionnement de 802.11 en mode infrastructure avec deux nœuds (figure 5.3) ainsi qu'une couche physique idéale et nous supposerons l'existence d'un régime saturé. Ainsi, uniquement trois composantes sont nécessaires pour décrire la topologie, la méthode d'accès au médium (CSMA) et l'algorithme de gestion de collisions (BEB).

<sup>9</sup>Également noté algorithme de BACKOFF.

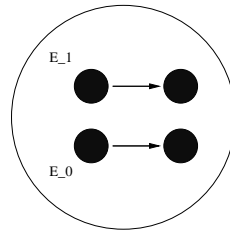


FIG. 5.3 – Mode infrastructure de 802.11

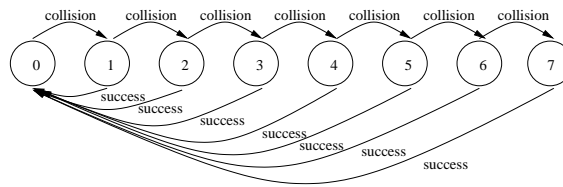
### 5.3.4 Description du modèle PEPA associé

#### La méthode d'accès

Il s'agit ici de modéliser la détection du canal libre, le tirage aléatoire du backoff, la transmission et d'attendre une notification de transmission réussie ou de collision. La détection du canal libre et la transmission se feront via une synchronisation avec la composante TOPOLOGY tandis que le tirage aléatoire du backoff et son incrément (resp. décrémentation) dans le cas d'une collision (resp. transmission réussie) se fera avec la composante BACKOFF. La même composante est utilisée pour les deux nœuds du réseau. Rappelons que seul l'émetteur a été modélisé.  $E_{i,x}, i \in \{0,1\}$  représente la composante modélisant le nœud  $i$  dans l'état  $x$  :

$E_{i,000}$	$\stackrel{def}{=}$	$(free, \mu_{trans}).E_{i,001};$
$E_{i,001}$	$\stackrel{def}{=}$	$(difs, \mu_{difs}).E_{i,002};$
$E_{i,002}$	$\stackrel{def}{=}$	$(free, \mu_{trans}).E_{i,003} + (occ, \mu_{data}).E_{i,000};$
$E_{i,003}$	$\stackrel{def}{=}$	$(db_i, \top).E_{i,004};$
$E_{i,004}$	$\stackrel{def}{=}$	$(free, \mu_{trans}).E_{i,005} + (occ, \mu_{data}).E_{i,000};$
$E_{i,005}$	$\stackrel{def}{=}$	$(transmit, \mu_{trans}).E_{i,006};$
$E_{i,006}$	$\stackrel{def}{=}$	$(ack_i, \top).E_{i,007} + (collision, \top).E_{i,008};$
$E_{i,007}$	$\stackrel{def}{=}$	$(succ_i, \mu_{trans}).E_{i,009};$
$E_{i,008}$	$\stackrel{def}{=}$	$(coll_i, \mu_{trans}).E_{i,000};$
$E_{i,009}$	$\stackrel{def}{=}$	$(sync_i, \top).E_{i,010};$
$E_{i,010}$	$\stackrel{def}{=}$	$(out_i, \mu_{trans}).E_{i,000};$

La modélisation de la transmission d'un paquet nécessite l'utilisation de dix sous-composantes. Avant l'émission d'un paquet, le nœud doit vérifier que le canal est libre, *i.e.* si la composante TOPOLOGY est dans l'état *idle* : l'action  $(free, \mu_{trans})$  permet la synchronisation avec cette composante. Si le canal n'est pas libre alors le nœud reste dans l'état  $E_{i,000}$ . Le temps  $\mu_{trans}$  modélise une durée très petite équivalent au temps de détection de porteuse. Si le médium est libre, l'émetteur passe dans la composante  $E_{i,001}$  et décrémente le compteur DIFS grâce à l'activité  $(difs, \mu_{difs})$ . Notons que PEPA étant un paradigme de modélisation stochastique, le taux de l'action DIFS est distribué suivant une loi exponentielle alors que, dans 802.11, la durée DIFS est statique et égale à  $50\mu s$ . De façon plus générale dans ce travail, l'ensemble des durées statiques utilisées dans 802.11 sont approximées par des durées exponentielles de moyenne le temps spécifié dans la norme. Après la décrémentation de DIFS, l'état du médium est à nouveau testé et, si celui-ci n'est plus libre, la composante se comporte comme  $E_{i,000}$  après avoir attendu la fin de la transmission en cours. Si le médium est encore libre, nous passons dans la composante  $E_{i,003}$  qui va permettre de modéliser le tirage puis la décrémentation du backoff : ces opérations sont

FIG. 5.4 – Représentation de l’algorithme BEB - *Binary Exponential Backoff*

alors réalisées dans la composante BACKOFF grâce à l’activité partagée  $(db_i, \top)$ . Cette activité est dite passive : son taux n’est pas spécifié mais il le sera lors de la synchronisation avec la composante BACKOFF. A la fin de la décrémentation du backoff, le médium est à nouveau testé : dans le cas où celui-ci est libre, le nœud transmet son paquet et attend pour une notification de transmission réussie ou de collision. Cette information provient de la composante TOPOLOGY qui a la connaissance de la topologie et du partage du médium entre les deux nœuds : les deux activités  $(ack_i, \top)$  et  $(collision, \top)$  sont donc partagées avec cette composante. L’activité  $(succ_i, \mu_{trans})$  (resp.  $(coll_i, \mu_{trans})$ ) fournit une synchronisation avec la composante BACKOFF pour notifier une transmission réussie et permettre de revenir dans la fenêtre initiale du backoff (resp. notifier une collision et permettre de doubler la fenêtre de contention). La composante  $E_{i,009}$  ne modélise aucun comportement de 802.11, elle est présente à des fins de mesure de performances uniquement. La composante  $E_{i,010}$  est utilisée dans le cas où la composante BUFFER existe pour dépiler le paquet suivant à envoyer.

### L’algorithme de gestion de collisions

L’algorithme de gestion de collisions correspond à l’algorithme de gestion de la fenêtre de contention. Dans le cas du *Binary Exponential Backoff* (BEB) utilisé dans 802.11, en cas de transmission réussie la fenêtre de contention est ré-initialisée à sa taille d’origine ( $CW_0 = [0, 2^5 - 1]$ ) tandis que dans le cas d’une collision, la taille de la fenêtre de contention est multipliée par deux jusqu’à la taille maximale ( $CW_0 = [0, 2^{10} - 1]$ ). La figure 5.4 illustre ce mécanisme<sup>10</sup>. La modélisation en utilisant PEPA est fort simple : la composante  $BO_{i,x}$  décrit le backoff du nœud  $i$ , dans la fenêtre de contention  $x$  avec un maximum de 7 retransmissions possibles. Trois actions sont présentes : l’action locale  $db_i$  permet de modéliser le tirage du backoff et le temps d’attente associé tandis que les actions  $succ_i$  et  $coll_i$  sont des actions partagées avec la composante précédente afin d’augmenter la fenêtre de contention ou de la ré-initialiser dans son état initial. Ces deux dernières actions n’ont donc pas de taux de spécifié : la détection d’une transmission réussie ou d’une collision se fait dans la composante  $Med_x$  (voir après). En revanche, le taux de l’activité  $db_i$ , noté  $f_{j,j \in \{0,7\}}$ <sup>11</sup>, dépend du nombre de collisions successives que la transmission en cours a subi.  $f_j$  représente la durée moyenne de la fenêtre de contention après  $j$  collisions successives. Ainsi,  $f_0$  est associé à la fenêtre de contention initiale comprenant  $2^5 - 1$  slots d’une durée de  $20\mu s$  : la durée moyenne associée est donc égale à  $20\mu s \times (2^5 - 1)/2$ . Plus généralement,  $\forall i \in \{0..5\}$ ,  $f_i$  est la durée moyenne du backoff dans la fenêtre de contention  $[0..2^{5+i}]$ . Alors que dans la norme 802.11, le temps d’attente est le résultat d’un tirage aléatoire dans la fenêtre  $[0..2^{5+i}]$ , nous approximations ces tirages aléatoires par un comportement moyen : en *moyenne* lorsque nous sommes dans la fenêtre  $[0..2^{5+i}]$ , la durée d’attente avant transmission est de  $20\mu s \times (2^5 - 1)/2$ . La

<sup>10</sup>Le nombre indiqué représente la  $n^{eme}$  fenêtre de contention

<sup>11</sup>Tel que  $f_7 = f_6 = f_5 = 20\mu s \times (2^{10} - 1)/2$ .

raison de cette approximation vient de la complexité engendré par une modélisation plus réaliste du backoff : en modélisant finement le tirage du backoff, les techniques d'agrégation utilisées ne suffisent pas à réduire suffisamment l'espace d'états. En revanche, la modélisation que nous faisons impacte les performances d'environ 15% <sup>12</sup>.

$BO_{i,0}$	$\stackrel{def}{=}$	$(db_i, f_0).BO_{i,0} + (succ_i, \top).BO_{i,0} + (coll_i, \top).BO_{i,1};$
...	$\stackrel{def}{=}$	...
$BO_{i,j}$	$\stackrel{def}{=}$	$(db_i, f_1).BO_{i,j} + (succ_i, \top).BO_{i,0} + (coll_i, \top).BO_{i-(j+1)}, \forall j \in [1..6];$
...	$\stackrel{def}{=}$	...
$BO_{i,7}$	$\stackrel{def}{=}$	$(db_i, f_5).BO_{i,7} + (succ_i, \top).BO_{i,0} + (coll_i, \top).BO_{i,0};$

Il est trivial de comprendre que la modélisation d'autres algorithmes de gestion de collisions (backoff) n'est pas difficile tant que nous respectons les contraintes de synchronisations des actions  $succ_i$  et  $coll_i$ .

Notons également que cette approche permet d'associer un algorithme de backoff à un nœud et un autre algorithme de backoff à un autre nœud. A notre connaissance, aucun autre modèle analytique ne permet une telle souplesse. Des études sur le comportement de backoff hétérogènes peuvent donc être menées.

### La topologie

La composante  $Med_x$  modélise l'état du médium résultant de la topologie représentée figure 5.3. Les activités de détection de porteuse ( $free, \top$ ) et de transmission ( $transmit, \top$ ) sont partagées avec la composante  $E_{i,x}$  modélise la méthode d'accès au canal. Les autres activités sont locales à cette composante et modélisent l'utilisation du médium.

Initialement le médium est libre ( $Med_{00,00}$ ). Lorsqu'un nœud teste le médium pour connaître son état, la composante reste dans l'état  $Med_{00,00}$ . En revanche, si un nœud prend possession du médium pour transmettre (action ( $transmit, \top$ )), la composante suit le comportement de la composante  $Med_{00,01}$ . Dans cet état le médium est encore libre pour une durée  $\mu_{slot}$  : si une autre transmission débute pendant cette durée, il y a collision et le médium passe dans l'état  $Med_{00,05}$ . S'il n'y a pas de transmission simultanée, le médium apparaît libre et peut être testé par un autre nœud. En revanche, à expiration de cette durée  $\mu_{slot}$ , le médium passe dans l'état occupé par une transmission et un acquittement est renvoyé. Les actions  $sync_i$  ne sont utilisées qu'à des fins de mesure de performance pour calculer la probabilité qu'une transmission est correctement réussie. Les composantes  $Med_{00,05}$  et  $Med_{00,06}$  représentent les états de collisions : deux composantes sont nécessaires pour se synchroniser avec chacun des deux émetteurs rentrés en collision.

$Med_{00,00}$	$\stackrel{def}{=}$	$(free, \top).Med_{00,00} + (transmit, \top).Med_{00,01};$
$Med_{00,01}$	$\stackrel{def}{=}$	$(free, \top).Med_{00,01} + (go_{next}, \mu_{slot}).Med_{00,02} + (transmit, \top).Med_{00,05};$
$Med_{00,02}$	$\stackrel{def}{=}$	$(ack_n, \mu_{trans}).Med_{00,03} + (ack_m, \mu_{trans}).Med_{00,04};$
$Med_{00,03}$	$\stackrel{def}{=}$	$(sync_n, \mu_{data}).Med_{00,00};$
$Med_{00,04}$	$\stackrel{def}{=}$	$(sync_m, \mu_{data}).Med_{00,00};$
$Med_{00,05}$	$\stackrel{def}{=}$	$(collision, \mu_{trans}).Med_{00,06};$
$Med_{00,06}$	$\stackrel{def}{=}$	$(collision, \mu_{data}).Med_{00,00};$

<sup>12</sup>TBC : tableaux des espaces d'états, mesures de performances...

### Le modèle du scénario

La composante *Basic*, modèle du scénario étudié (figure 5.3), résulte de la coopération des composantes précédentes :

$$Basic \stackrel{def}{=} ((E_{0.000} \boxtimes_K BO_{0.0}) || (E_{1.000} \boxtimes_L BO_{1.0})) \boxtimes_M Med_{00.00};$$

Les ensembles de synchronisations sont définis par :  $K = \{db_0, succ_0, coll_0\}$ ,  $L = \{db_1, succ_1, coll_1\}$  et  $M = \{free, transmit, collision, ack_0, ack_1, sync_0, sync_1\}$ .

Dans la section suivante, nous allons décrire comment, à partir de ce cas simple, nous pouvons étudier différentes topologies, différents algorithmes de gestion de fenêtre de contention, étudier différentes sources de trafic et s'intéresser à des hypothèses plus réalistes concernant le médium radio. Dans la section 5.5, nous discuterons de la partie validation du modèle puis des résultats d'évaluation de performances en termes de performances (taux de collisions, débit utile) et en termes d'équité à court et long terme.

## 5.4 Extension vers d'autres études

### 5.4.1 Vers un autre mécanisme d'accès au médium

Le but de ce travail est d'étudier le compromis performances / équité dans les réseaux radio multi-sauts. En termes de performances, on cherche à maximiser le taux d'utilisation du canal pour des transmissions effectivement réussies. En termes d'équité à long terme, cela signifie qu'en moyenne, toutes les stations ont transmis la même quantité d'information sur une fenêtre d'observation assez longue. Mais quid de l'équité à court terme ? Pour bien se représenter l'équité à court terme, il faut faire référence à un système où le temps de parole des nœuds est organisé à tour de rôle. Ce système, proche d'un TDMA couplé à une allocation de ressources équitable, nous permet de construire une mesure de référence sur l'équité.

La modélisation de TDMA, à partir de la composante ACCESS METHOD  $E_x$ , se fait en rajoutant quelques éléments de synchronisation pour modéliser l'ordonnement entre les nœuds pour l'accès au médium. L'utilisation de TDMA simplifie également le modèle car il n'est plus nécessaire de recourir à un algorithme de gestion de collisions ni d'écouter le médium avant de transmettre un paquet. Nous considérons toujours un réseau composé de deux nœuds dans la même zone de transmission, avec une hypothèse de canal physique idéal et un régime saturé. La modélisation du nœud  $E_i$  utilisant TDMA est ainsi définie :

$$\begin{array}{l} E_{i.000} \stackrel{def}{=} (e_i, \mu_{trans}).E_{i.001}; \\ E_{i.001} \stackrel{def}{=} (transmit, \mu_{trans}).E_{i.002}; \\ E_{i.002} \stackrel{def}{=} (ack_i, \top).E_{i.003} + (collision, \top).E_{i.004}; \\ E_{i.003} \stackrel{def}{=} (succ_i, \mu_{trans}).E_{i.005}; \\ E_{i.004} \stackrel{def}{=} (coll_i, \mu_{trans}).E_{i.006}; \\ E_{i.005} \stackrel{def}{=} (sync_i, \top).E_{i.006}; \\ E_{i.006} \stackrel{def}{=} (e_j, \top).E_{i.000}; \end{array}$$

Deux nouvelles actions sont rajoutées :  $e_i$  et  $e_j$ . Elles sont rendues nécessaires pour créer l'ordonnancement : lorsque le nœud  $i$  termine sa transmission, il se synchronise avec le nœud  $j$  et attends une nouvelle synchronisation pour transmettre un paquet. Il est donc nécessaire de rajouter une synchronisation entre les deux nœuds. En revanche, la composante BACKOFF n'est pas utilisée de part l'absence de collision. Il en va de même avec l'action *collision* dont l'usage n'est plus motivé. La composante TOPOLOGY n'est pas modifiée.

La composante *Sync* modélisant ce scénario s'écrit ainsi :

$$Sync \stackrel{def}{=} (E_{0.000} \bowtie_K E_{1.000}) \bowtie_M Med_{00.00};$$

Les ensembles de synchronisations sont définis par :  $K = \{e_0, e_1\}$  et  $M = \{transmit, collision, ack_0, ack_1, sync_0, sync_1\}$ .

### 5.4.2 Modélisation des variantes de BEB

Le cœur de notre travail est d'étudier les algorithmes de gestion de la fenêtre de contention et en particulier d'étudier BEB ainsi que quelques-unes de ses variantes dans un contexte multi-sauts. La modélisation d'un algorithme de gestion de BACKOFF se fait aisément tant que les synchronisations  $succ_i$  et  $coll_i$  sont respectées. Les algorithmes proposant une augmentation (resp. diminution) de la fenêtre de backoff de façon linéaire ou exponentielle sont donc adressables avec des modifications mineures.

#### DIDD

Le comportement de BEB apparaissant trop agressif à cause de la réduction de la fenêtre de contention vers la fenêtre initiale dès la première transmission réussie peut être évité en adoptant un comportement exponentiel : en cas de transmission réussie, la fenêtre de contention est divisée par deux. C'est ce que propose DIDD (*Double Increase Double Decrease* (cf. figure 5.5) [24].

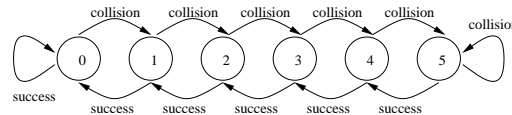


FIG. 5.5 – Principe de l'algorithme DIDD - *Double Increase Double Decrease*

La composante  $DIDD_{i,x}$  associée à cet algorithme utilisé par le nœud  $i$  est ainsi modélisée :

$$\begin{array}{l} DIDD_{i,0} \stackrel{def}{=} (db_i, f_0).DIDD_{i,0} + (succ_i, \top).DIDD_{i,0} + (coll_i, \top).DIDD_{i,1}; \\ \dots \stackrel{def}{=} \dots \\ DIDD_{i,j} \stackrel{def}{=} (db_i, f_j).DIDD_{i,j} + (succ_i, \top).DIDD_{i,j-1} + (coll_i, \top).DIDD_{i-j+1} \forall j \in [1, 4]; \\ \dots \stackrel{def}{=} \dots \\ DIDD_{i,5} \stackrel{def}{=} (db_i, f_5).DIDD_{i,5} + (succ_i, \top).DIDD_{i,4} + (coll_i, \top).DIDD_{i,5}; \end{array}$$

Une fois de plus les composantes modélisant la topologie et la technique d'accès au médium ne sont pas affectées. La composante modélisant le réseau 802.11 en mode infrastructure composé de deux nœuds utilisant CSMA et sous les hypothèses de canal parfait et de régime saturé est :

$$Basic \stackrel{def}{=} ((E_{0.000} \boxtimes_K DIDD_{0.0}) || (E_{1.000} \boxtimes_L DIDD_{1.0})) \boxtimes_M Med_{00.00};$$

Les ensembles de synchronisations sont définis par :  $K = \{db_0, succ_0, coll_0\}$ ,  $L = \{db_1, succ_1, coll_1\}$  et  $M = \{free, transmit, collision, ack_0, ack_1, sync_0, sync_1, collision\}$ .

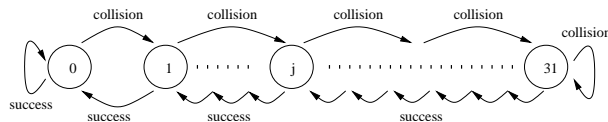


FIG. 5.6 – Principe de l’algorithme MILD - *Multiplicative Increase Linear Decrease*

De la même façon que nous avons modélisé DIDD, MILD (*Multiplicative Increase Linear Decrease* [14] peut être modélisé (cf. figure 5.6) tout aussi facilement :

$$\begin{array}{ccc} \dots & \stackrel{def}{=} & \dots \\ MILD_{i-j} & \stackrel{def}{=} & (db_i, f_j).MILD_{i-j} + (succ_i, \top).MILD_{i-(j-1)} + (coll_i, \top).MILD_{i-(2 \times j + 1)} \\ \dots & \stackrel{def}{=} & \dots \end{array}$$

### 5.4.3 Investigation de topologies multi-sauts

L’exemple donné jusqu’à présent s’intéresse à un cas trivial des réseaux 802.11 : un mode infrastructure composé uniquement de 2 nœuds. En introduction de ce chapitre, nous avons précisé vouloir étudier plusieurs topologies multi-sauts tels que : les 3 paires (figure 5.1(a)), les nœuds cachés (figure 5.1(b)) et les nœuds transitifs (figure 5.1(c)). Une fois de plus, nous allons mettre en avant tout l’intérêt que nous pouvons tirer de l’approche compositionnelle de PEPA. En modifiant uniquement la composante TOPOLOGY, nous pourrions proposer de nouveaux modèles pour étudier ces trois topologies. Pour chacun des trois scénarii suivants, nous supposons une couche MAC idéale, une source de trafic saturé et l’utilisation de BEB.

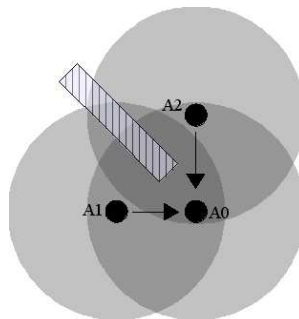


FIG. 5.7 – Le cas des nœuds cachés

### Le cas des nœuds cachés

Dans cette topologie, nous considérons deux transmetteurs et un récepteur unique tel que les deux transmetteurs ne s'entendent pas (figure 5.7). Comme le médium apparaît toujours libre, les deux émetteurs peuvent accéder au canal et transmettre leurs paquets : le taux de collisions est alors important. Comme le médium est toujours libre, la phase de détection de porteuse n'est pas nécessaire dans le modèle : pour cette raison l'action *free* est supprimée des composantes TOPOLOGY et ACCESS METHOD. Le modèle de ce scénario est alors donné par :

$Med_{00.00}$	$\stackrel{def}{=}$	$(transmit, \top).Med_{00.01};$
$Med_{00.01}$	$\stackrel{def}{=}$	$(frag, \mu_{data10}).Med_{00.02} + (transmit, \top).Med_{00.14};$
...	$\stackrel{def}{=}$	...
$Med_{00.j}$	$\stackrel{def}{=}$	$(frag, \mu_{data10}).Med_{00.(j+1)} + (transmit, \top).Med_{00.14}, \forall j \in [1..9];$
...	$\stackrel{def}{=}$	...
$Med_{00.10}$	$\stackrel{def}{=}$	$(ack_m, \mu_{trans}).Med_{00.11} + (ack_n, \mu_{trans}).Med_{00.12};$
$Med_{00.11}$	$\stackrel{def}{=}$	$(sync_m, \mu_{data10}).Med_{00.00};$
$Med_{00.12}$	$\stackrel{def}{=}$	$(sync_n, \mu_{data10}).Med_{00.00};$
$Med_{00.13}$	$\stackrel{def}{=}$	$(collision, \mu_{data}).Med_{00.00} + (transmit, \top).Med_{00.14};$
$Med_{00.14}$	$\stackrel{def}{=}$	$(collision, \mu_{data10}).Med_{00.13};$

L'action *frag* présente dans les composantes  $Med_{00.j}, \forall j \in [1..10]$  est associée à la transmission d'un paquet. Au cours de la transmission, une autre transmission peut être lancée, conduisant à une collision contrairement au mode infrastructure où les collisions ne peuvent se produire que lorsqu'elles démarrent simultanément. Il nous faut donc modéliser le fait que la transmission puisse être interrompue. Dans un algèbre de processus, les actions ont des comportements atomiques : si on veut modéliser une interruption, il est nécessaire de les décomposer en plusieurs actions consécutives. C'est ce que nous faisons ici avec l'action *frag*. La durée de cette action ( $\mu_{data10}$ ) correspond à la durée de transmission d'un fragment, dix fois plus petit qu'un paquet. Cette activité est toujours en concurrence avec l'activité *transmit* modélisant la transmission d'un paquet par l'autre nœud. Un acquittement est renvoyé vers l'émetteur dès la transmission du dernier fragment. En cas de collision, la durée de celle-ci est égale à la durée de transmission d'un paquet.

La composante *Hidden* modélisant le scénario étudié résulte de la coopération des composantes :

$$Hidden \stackrel{def}{=} ((E_{0.000} \boxtimes_K BO_{0.0}) || (E_{1.000} \boxtimes_L BO_{1.0})) \boxtimes_M Med_{00.00};$$

Les ensembles de synchronisations sont définis par :  $K = \{db_0, succ_0, coll_0\}$ ,  $L = \{db_1, succ_1, coll_1\}$  et  $M = \{send, ack_0, ack_1, sync_0, sync_1, collision\}$ .

### Le scénario des trois paires

Ici, nous considérons trois paires d'émetteurs/récepteurs en cours de communication tel que la paire centrale partage le médium avec les deux paires extérieures tandis que les deux paires extérieures sont indépendantes (figure 5.8). La modélisation se focalise sur la transmission d'un flux plus que sur la relation émetteur-récepteur. Ainsi l'action *free* est remplacée par trois actions

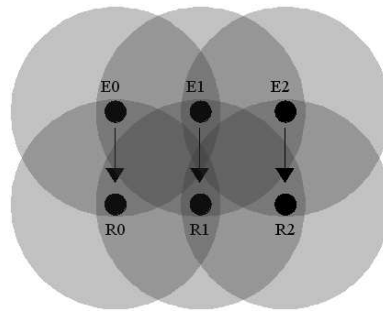


FIG. 5.8 – Le scénario des 3 paires

$free_0$ ,  $free_1$  et  $free_2$  pour modéliser l'utilisation du médium par les trois paires<sup>13</sup>. Il en va de même pour l'action  $transmit$ . Le modèle suivant fait apparaître le partage du médium autour de la paire centrale. Une autre modélisation est possible en ayant recours à plusieurs médiums.

$Med_{00.00}$	$\stackrel{def}{=}$	$(free_0, \top).Med_{00.00} + (free_1, \top).Med_{00.00} + (free_2, \top).Med_{00.00}$ $+ (transmit_0, \top).Med_{00.02} + (transmit_1, \top).Med_{00.01} + (transmit_2, \top).Med_{00.09}$ ;
$Med_{00.01}$	$\stackrel{def}{=}$	$(ack_1, \top).Med_{00.01}$ ;
$Med_{00.02}$	$\stackrel{def}{=}$	$(frag, \mu_{data25}).Med_{00.03} + (free_2, \mu_{trans}).Med_{00.02}$ ;
$Med_{00.03}$	$\stackrel{def}{=}$	$(frag, \mu_{data25}).Med_{00.04} + (free_2, \mu_{trans}).Med_{00.03} + (transmit_2, \top).Med_{00.06}$ ;
$Med_{00.04}$	$\stackrel{def}{=}$	$(frag, \mu_{data25}).Med_{00.05} + (free_2, \mu_{trans}).Med_{00.04} + (transmit_2, \top).Med_{00.07}$ ;
$Med_{00.05}$	$\stackrel{def}{=}$	$(ack_0, \mu_{data25}).Med_{00.00} + (free_2, \mu_{trans}).Med_{00.05} + (transmit_2, \top).Med_{00.08}$ ;
$Med_{00.06}$	$\stackrel{def}{=}$	$(ack_0, \mu_{data75}).Med_{00.11}$ ;
$Med_{00.07}$	$\stackrel{def}{=}$	$(ack_0, \mu_{data50}).Med_{00.10}$ ;
$Med_{00.08}$	$\stackrel{def}{=}$	$(ack_0, \mu_{data25}).Med_{00.09}$ ;
$Med_{00.09}$	$\stackrel{def}{=}$	$(frag, \mu_{data25}).Med_{00.10} + (free_0, \mu_{trans}).Med_{00.09}$ ;
$Med_{00.10}$	$\stackrel{def}{=}$	$(frag, \mu_{data25}).Med_{00.11} + (free_0, \mu_{trans}).Med_{00.10} + (transmit_0, \top).Med_{00.13}$ ;
$Med_{00.11}$	$\stackrel{def}{=}$	$(frag, \mu_{data25}).Med_{00.12} + (free_0, \mu_{trans}).Med_{00.11} + (transmit_0, \top).Med_{00.14}$ ;
$Med_{00.12}$	$\stackrel{def}{=}$	$(ack_2, \mu_{data25}).Med_{00.00} + (free_0, \mu_{trans}).Med_{00.12} + (transmit_0, \top).Med_{00.15}$ ;
$Med_{00.13}$	$\stackrel{def}{=}$	$(ack_2, \mu_{data75}).Med_{00.04}$ ;
$Med_{00.14}$	$\stackrel{def}{=}$	$(ack_2, \mu_{data50}).Med_{00.03}$ ;
$Med_{00.15}$	$\stackrel{def}{=}$	$(ack_2, \mu_{data25}).Med_{00.02}$ ;

La composante  $Med_{00.00}$  représente l'état initial du médium inoccupé. On retrouve le même principe que la modélisation de la composante TOPOLOGY dans le modèle *basic*. Avant de transmettre, les nœuds testent le médium via l'activité  $free_{x,x \in [1..3]}$ . Grâce à l'activité  $transmit_{x,x \in [1..3]}$ , les nœuds prennent possession du médium. Si la paire centrale prend le médium alors les deux paires extérieures ne peuvent y accéder. En revanche, si l'une des paires extérieures transmet, la paire centrale ne peut accéder au médium tandis que l'autre paire extérieure peut transmettre simultanément. Les composantes  $Med_{00.02}$  à  $Med_{00.08}$  font référence à une transmission démarrée par la paire extérieure 0 alors que les composantes  $Med_{00.09}$  à  $Med_{00.15}$  font référence à la paire extérieure 2. Comme souligné dans le modèle précédent (section 5.4.3), le comportement des activités PEPA étant atomiques, si nous voulons autoriser la transmission d'une paire extérieure

<sup>13</sup>Les indices 0, 1 et 2 font référence aux paires extérieures (0, 2) et à la paire centrale (1).

pendant que l'autre transmet, nous devons fragmenter la transmission d'un paquet en plusieurs petits paquets successifs. C'est ce que nous faisons ici avec les activités *frag*.

Le modèle final de cette topologie fait rentrer en interaction 3 paires qui se partagent le médium. Le même modèle de transmetteur que le modèle *basic* est utilisé. Ce scénario est modélisé par :

$$3p \stackrel{def}{=} ((E_{0,000} \bowtie_J BO_{0,0}) || (E_{1,000} \bowtie_K BO_{1,0}) (E_{2,000} \bowtie_L BO_{2,0})) \bowtie_M Med_{00,00};$$

Les ensembles de synchronisations sont définis par :  $J = \{db_0, succ_0, coll_0\}$ ,  $K = \{db_1, succ_1, coll_1\}$ ,  $L = \{db_2, succ_2, coll_2\}$  et  $M = \{free_0, free_1, free_2, transmit_0, transmit_1, transmit_2, collision, ack_0, ack_1, ack_2\}$ .

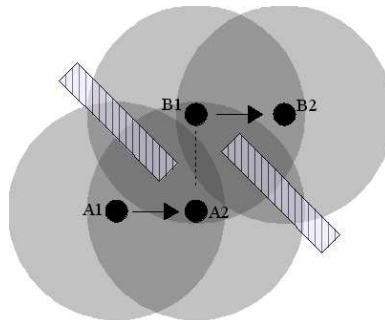


FIG. 5.9 – La topologie des nœuds transitifs

### La topologie des nœuds transitifs

Ce scénario peut être interprété comme le cas de nœuds cachés asymétriques. Pour transmettre la paire  $A_1A_2$  doit accéder au canal lors d'une période de silence de la paire  $B_1B_2$ , conduisant à une forte probabilité de collision. Le modèle se focalise sur le partage du médium vu du récepteur  $A_2$  de la paire  $A_1A_2$  qui reçoit les trafics des deux émetteurs. La notation  $A$  fait référence à la paire  $A_1A_2$  tandis que la notation  $B$  fait référence à la paire  $B_1B_2$ .

$Med_{00.00}$	$\stackrel{def}{=}$	$(transmit_B, \top).Med_{00.01} + (transmit_A, \top).Med_{00.05};$
...	$\stackrel{def}{=}$	...
$Med_{00.(i)}$	$\stackrel{def}{=}$	$(frag, \mu_{data25}).Med_{00.(i+1)} + (transmit_A, \top).Med_{00.(i+8)}, \forall i \in [1 \dots 3];$
...	$\stackrel{def}{=}$	...
$Med_{00.04}$	$\stackrel{def}{=}$	$(ack_B, \mu_{data25}).Med_{00.00} + (transmit_A, \top).Med_{00.12};$
...	$\stackrel{def}{=}$	...
$Med_{00.(i)}$	$\stackrel{def}{=}$	$(frag, \mu_{data25}).Med_{00.(i+1)} + (transmit_B, \top).Med_{00.(i+8)}, \forall i \in [5 \dots 7];$
...	$\stackrel{def}{=}$	...
$Med_{00.08}$	$\stackrel{def}{=}$	$(ack_A, \mu_{data25}).Med_{00.00} + (transmit_B, \top).Med_{00.16};$
$Med_{00.09}$	$\stackrel{def}{=}$	$(ack_B, \mu_{data100}).Med_{00.20};$
$Med_{00.10}$	$\stackrel{def}{=}$	$(ack_B, \mu_{data75}).Med_{00.19};$
$Med_{00.11}$	$\stackrel{def}{=}$	$(ack_B, \mu_{data50}).Med_{00.18};$
$Med_{00.12}$	$\stackrel{def}{=}$	$(ack_B, \mu_{data25}).Med_{00.17};$
$Med_{00.13}$	$\stackrel{def}{=}$	$(collision_A, \mu_{collision100}).Med_{00.04};$
$Med_{00.14}$	$\stackrel{def}{=}$	$(collision_A, \mu_{collision75}).Med_{00.03};$
$Med_{00.15}$	$\stackrel{def}{=}$	$(collision_A, \mu_{collision50}).Med_{00.02};$
$Med_{00.16}$	$\stackrel{def}{=}$	$(collision_A, \mu_{collision25}).Med_{00.01};$
$Med_{00.(i)}$	$\stackrel{def}{=}$	$(frag, \mu_{collision25}).Med_{00.(i+1)} + (transmit_B, \top).Med_{00.(i-4)}, \forall i \in [17 \dots 19];$
$Med_{00.20}$	$\stackrel{def}{=}$	$(collision_A, \mu_{collision25}).Med_{00.00} + (transmit_B, \top).Med_{00.16};$

Dans ce scénario, la paire  $B_1B_2$  peut toujours transmettre tandis que la paire  $A_1A_2$  subit des collisions. Notons que pour chacun des émetteurs, le médium apparaît libre, ce qui nous autorise à éliminer l'action *free*. Une fois de plus, pour autoriser la transmission par un émetteur alors qu'un autre est déjà en cours de transmission, nous introduisons plusieurs composantes conduisant à fragmenter la transmission d'un paquet. Dans ce modèle, lorsque la paire  $B_1B_2$  accède au médium ( $Med_{00.3}$ ), la paire  $A_1A_2$  peut également transmettre, conduisant à passer dans la composante  $Med_{00.11}$ . Dans cet état, le temps de transmission du paquet émis par  $B_1$  s'écoule puis le modèle atteint la composante  $Med_{00.18}$  permettant de modéliser la collision.

Le modèle général de ce scénario est décrit par :

$$Transitifs \stackrel{def}{=} ((E_{A.000} \underset{K}{\bowtie} BO_{A.0}) || (E_{B.000} \underset{L}{\bowtie} BO_{B.0})) \underset{M}{\bowtie} Med_{00.00};$$

Les ensembles de synchronisations sont définis par :  $K = \{db_B, succ_B, coll_B\}$ ,  $L = \{db_A, succ_A, coll_A\}$  et  $M = \{transmit_A, transmit_B, collision_A, ack_A, ack_B\}$ .

#### 5.4.4 Modélisation de sources de trafic non saturé

La plupart des études en termes de performances et d'équité ne se sont intéressées qu'au régime saturé. La modélisation de sources trafic non saturés introduit une nouvelle complexité, souvent liée à la taille de l'espace d'états. À l'aide du modèle PEPA construit, nous pouvons aisément introduire une source de trafic non saturé associée à une file d'attente permettant de stocker les paquets en attente de transmissions : deux nouvelles composantes sont donc introduites : la composante TRAFIC et la composante BUFFER.

La composante TRAFIC suivante modélise des arrivées qui suivent une loi de Poisson d'intensité  $\lambda$  pour la station  $i$  :

$$Gen_i \stackrel{def}{=} (arrival_i, \lambda).Poisson_i;$$

Alors que la composante suivante modélise un flux On-Off (Interrupted Poisson Process) :

$$\begin{aligned} Gen_{i,0} &\stackrel{def}{=} (arrival_i, \lambda).Gen_{i,1}; \\ Gen_{i,1} &\stackrel{def}{=} (onon, p).Gen_{i,0} + (onoff, (1-p)).Gen_{i,2}; \\ Gen_{i,2} &\stackrel{def}{=} (silence_i, \beta).Gen_{i,3}; \\ Gen_{i,3} &\stackrel{def}{=} (offoff, q).Gen_{i,2} + (offon, (1-q)).Gen_{i,0}; \end{aligned}$$

Cette composante doit se synchroniser avec la composante BUFFER grâce à l'action  $arrival_i$ . Lorsque cette file d'attente utilise une discipline FIFO et qu'elle est de taille  $n$ , elle se modélise ainsi :

$$\begin{aligned} Queue_{i,0} &\stackrel{def}{=} (arrival_i, \top).Queue_{i,001}; \\ \dots & \\ Queue_{i,j} &\stackrel{def}{=} (arrival_i, \top).Queue_{i,(j+1)} + (out_i, \top).Queue_{i,(j-1)} \quad \forall j \in [1..n-1]; \\ \dots & \\ Queue_{i,n} &\stackrel{def}{=} (arrival_i, \top).Queue_{i,n} + (out_i, \top).Queue_{i,(n-1)}; \end{aligned}$$

Il est alors nécessaire d'introduire une première synchronisation entre la composante TRAFIC et la composante BUFFER via l'activité  $out_i$ . Si l'on considère un réseau en mode infrastructure composé de 2 nœuds utilisant BEB et en supposant un canal radio idéal et des arrivées de paquets suivant une loi de Poisson, le modèle PEPA est le suivant :

$$\begin{array}{c} \text{Arrival} \stackrel{def}{=} \\ ((Gen_{0,0} \boxtimes_A Queue_{0,0} \boxtimes_B E_{0,0} \boxtimes_C BO_{0,0}) || (Gen_{1,0} \boxtimes_J Queue_{1,0} \boxtimes_K E_{1,0} \boxtimes_L BO_{1,0})) \boxtimes_M Med_{0,0} \end{array}$$

Les ensembles de synchronisations sont définis par :  $M = \{free, transmit, collision, ack_0, ack_1, sync_0, sync_1\}$ ,  $A = \{arrival_0\}$ ,  $B = \{out_0\}$ ,  $C = \{db_0, succ_0, coll_0\}$ ,  $J = \{arrival_1\}$  et  $K = \{out_1\}$ ,  $L = \{db_1, succ_1, coll_1\}$ .

### 5.4.5 Tenir compte des propriétés du canal radio

#### Discussions autour de la modélisation du canal radio

Une des difficultés de la modélisation analytique est de rester fidèle au système étudié et de conserver des hypothèses réalistes. Autant dans le domaine de l'évaluation de performances de couches MAC en réseaux filaires, il est relativement aisé d'approcher le système réel, autant dans le domaine des réseaux sans fil, les propriétés du canal radio posent de nouveaux challenges pour l'évaluation de performances. Aujourd'hui, la plupart des études se font sous l'hypothèse restrictive de canal radio idéal, c'est-à-dire sans tenir compte des interférences, de l'effet des chemins multiples, du phénomène d'évanouissement et celui d'effet de masque. Ces conditions supposées de canal idéal s'appuient donc sur des hypothèses erronées [49]. Nous proposons d'étendre les modèles précédant en ajoutant une nouvelle composante venant moduler les propriétés du canal radio au cours du temps. Ce que nous cherchons à montrer c'est, sous des hypothèses plus réalistes du canal radio, l'évolution du compromis performance - équité en environnement non idéal.

Classiquement, trois hypothèses sont formulées pour tenir compte des phénomènes physiques [84] :

- la liaison entre deux nœuds d'un réseau radio est supposée être soit parfaite soit inexistante. C'est le principe de la liaison à seuil. Relâcher cette hypothèse consiste à introduire sur le lien entre deux nœuds une probabilité de transmission.
- L'existence d'une liaison entre deux nœuds dépend uniquement de la distance qui les sépare. C'est le principe d'isotropie. Cette hypothèse combinée à la précédente conduit au modèle à disque. Relâcher cette seconde hypothèse, c'est établir une nouvelle fonction de liaison qui ne dépend plus uniquement de la distance de telle façon que cette fonction soit non-isotrope. Le choix d'une telle fonction est lié à l'environnement radio.
- La qualité de chaque liaison radio est indépendante des autres. C'est le principe d'indépendance. C'est cette hypothèse qui permet d'éviter les problèmes d'interférences. L'introduction de ces interférences peut se faire via l'introduction d'un deuxième voisinage (zone de brouillage) ou encore par l'assimilation des interférences à un bruit gaussien moyen dépendant du taux de transmission sur l'ensemble du réseau.

Le travail mené ici a pour but de regarder l'impact de la première hypothèse. La deuxième hypothèse n'a pas d'intérêt ici car nous décrivons la topologie étudiée à partir des liaisons en puissance, sans notion de position géographique. La troisième hypothèse est prise en compte dans la gestion de l'accès au canal.

Étudier l'impact de la première hypothèse, c'est introduire un taux d'erreur paquet. Ce taux d'erreur paquet peut être corrélé ou non. Dans un canal idéal les erreurs sont liées uniquement au bruit électronique du récepteur rendant la probabilité d'erreur indépendante pour chaque paquet. Dans un environnement plus réaliste, en tenant compte notamment du phénomène d'évanouissement, la puissance du signal varie au cours du temps, ce qui entraîne une variation du rapport signal/bruit et donc une variation de la probabilité d'erreurs. Elle devient alors un processus corrélé engendrant des erreurs en rafale quand la vitesse d'émission des paquets est plus rapide que la vitesse d'évolution du canal. Cette corrélation peut être représentée par un processus markovien de type Gilbert-Elliot comme nous le faisons ici. C'est ce processus markovien qui va être intégré dans la nouvelle composante RADIO.

### Modélisation du canal radio

L'introduction d'un taux d'erreurs paquet basé sur des erreurs non corrélées se fait via l'introduction de la composante :

$$gen_0 \stackrel{def}{=} (error, p).gen_0 + (no\_error, (1 - p)).gen_0;$$

Dans ce modèle, le taux d'erreur bit est introduit pour calculer le taux d'erreur paquets :  $p = PER = 1 - (1 - BER)^{bit}$ , où *bit* représente la taille des paquets en bit.

Le modèle de Gilbert-Elliot permet l'introduction d'erreurs corrélées contrairement à la composante précédente où les erreurs sont indépendantes. Une modélisation simple mais réaliste d'erreurs corrélées peut se faire via l'introduction d'erreurs générées suivant un processus IPP (*Interrupted Poisson Process*). Nous proposons ainsi la composante suivante où  $gen_{00}$  modélise une rafale d'erreurs et  $gen_{01}$  modélise un comportement sans erreur :

$$\begin{array}{l} gen_{00} \stackrel{def}{=} (suc, \mu_{test}).gen_{00} + (move, p).gen_{01} + (move, 1 - p).gen_{00}; \\ gen_{01} \stackrel{def}{=} (col, \mu_{test}).gen_{01} + (move, 1 - q).gen_{00} + (move, q).gen_{01}; \end{array}$$

Le réseau en mode infrastructure composé de deux nœuds utilisant CSMA comme technique d'accès au médium, BEB comme algorithme de gestion de la fenêtre de contention, supposant l'existence d'un régime saturé et soumis à des rafales d'erreur est défini par :

$$Error \stackrel{def}{=} ((E_{0.000} \boxtimes_K BO_{0.0}) || (E_{1.000} \boxtimes_L BO_{1.0})) \boxtimes_M Med_{00.00} \boxtimes_N gen_{01};$$

Les ensembles de synchronisations sont définis par :  $K = \{db_0, succ_0, coll_0\}$ ,  $L = \{db_1, succ_1, coll_1\}$ ,  $M = \{free, transmit, collision, ack_0, ack_1, sync_0, sync_1\}$  et  $N = \{suc, col\}$

## 5.5 Évaluation de performances

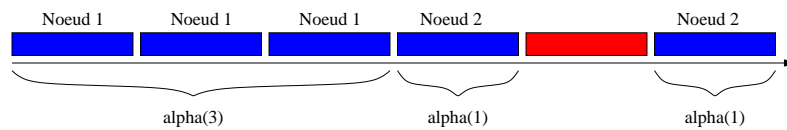
### 5.5.1 Mesures de performances

Basé sur la composition des composantes préalablement décrites, nous sommes en mesure de construire une chaîne de Markov agrégée, de calculer son état stationnaire et de dériver ainsi plusieurs métriques de performances. Nous nous intéressons ici à l'état du médium radio (libre, transmission, collision) et à son partage entre les nœuds de la topologie considérée. Aucun résultat lié au mode infrastructure n'est présenté pour mieux se concentrer sur trois topologies multi-sauts : les nœuds cachés, les nœuds transitifs et le scénario des trois paires. Des résultats supplémentaires peuvent être trouvés dans les références bibliographiques.

**Utilisation du médium** La mesure du taux d'utilisation du médium pour une transmission effectivement réussie ou une collision se fait en mesurant la probabilité, à l'état stationnaire, d'être dans les états du processus Markovien correspondant aux composantes témoignant d'une transmission effectivement réussie ou d'une collision. Si nous reprenons l'exemple du mode infrastructure composé de deux nœuds, la probabilité que le médium soit occupé par une collision est liée aux composantes  $Med_{00.05}$  et  $Med_{00.06}$  tandis que la probabilité qu'une transmission soit réussie est liée aux composantes  $Med_{00.02}$ ,  $Med_{00.03}$  et  $Med_{00.04}$ .

**Mesure de l'équité** Plusieurs métriques ont été proposées pour mesurer l'équité. En liaison avec notre modélisation, nous avons introduit une métrique témoignant de l'équité à court terme. Cette métrique permet de mettre en avant la possible monopolisation du médium pour plusieurs transmissions réussies et successives. Nous définissons  $\alpha_i$  la probabilité que la  $i^{eme}$  transmission soit réussie sachant que les  $(i - 1)$  précédentes transmissions se sont faites avec succès et sans interruption par un autre nœud. La figure 5.10 fournit une illustration de cette métrique. Une valeur strictement croissante de  $\alpha_i$  témoigne d'un comportement tout à fait inéquitable pouvant se traduire par : plus un nœud transmet et plus il a de chance de continuer à transmettre. Une valeur strictement décroissante de cette métrique laisse penser à une situation non équitable mais moins critique que précédemment : ici, plus un nœud transmet des paquets successifs et avec succès et moins il a de chance de monopoliser le médium : en quelque sorte il est pénalisé de son comportement glouton. Une valeur constante égale à  $1/n$  où  $n$  est le nombre de nœuds est probablement le comportement le plus équitable possible.

Pour construire cette mesure d'équité, il faut être capable pour chaque nœud de comptabiliser le nombre de paquets transmis successivement avec succès. Cela est rendu possible par la duplication de la composante *Medium* en autant de sous-composantes que de nombre de paquets transmis successivement et avec succès.

FIG. 5.10 – Signification de la métrique d'équité  $\alpha_i$ 

### 5.5.2 Validation du modèle

La validation d'un tel modèle peut se faire soit par comparaison avec des mesures expérimentales soit par d'intensives campagnes de simulation. Nous avons simulé plusieurs des topologies présentées ici en utilisant NS-2 [3]. Le but est de valider la démarche de modélisation proposée, de valider la *qualité* des différents modèles. Nous présentons ici trois types de validation.

Dans le premier cas, nous considérons la situation des nœuds cachés où les deux nœuds utilisent BEB sous les hypothèses de couche MAC idéale et de trafic saturé. Nous mesurons le débit utile par nœud en fonction de la taille des paquets dans le cas d'une vitesse d'émission de 11 Mbit/s. Les résultats obtenus (figure 5.11(a)) montrent clairement la proximité des résultats de simulation et des résultats théoriques.

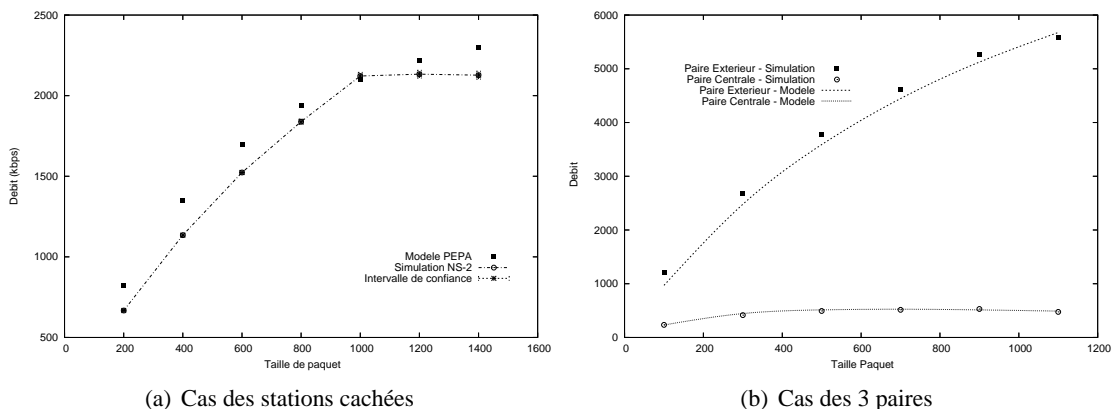


FIG. 5.11 – Validation du modèle : mesure du débit

Dans un deuxième temps, nous considérons la topologie des 3 paires utilisant BEB et sous les hypothèses classiques de couche MAC idéale et de régime saturé. Nous mesurons le débit des paires extérieures et de la paire centrale, résultat mettant en avant un phénomène d'iniquité à long terme. Nous pouvons noter la proximité des résultats entre le modèle proposé et les simulations conduites sous NS-2.

Enfin, nous nous sommes intéressés à l'évaluation de la métrique d'équité  $\alpha_i$  : la figure 5.12(a) témoigne de la qualité du modèle dans le cas du mode infrastructure composé de deux nœuds tandis que la figure 5.12(b) illustre la métrique d'équité dans le cas des stations cachées.

### 5.5.3 Étude des nœuds cachés

Nous étudions ici le comportement des algorithmes de gestion de fenêtre de contention BEB et DIDD dans le cas d'un couche physique idéale et d'un trafic saturé. Nous supposons une taille de paquets fixé à 1000 octets ainsi qu'un débit de 11Mbit/s. Nous considérons à la fois le mécanisme

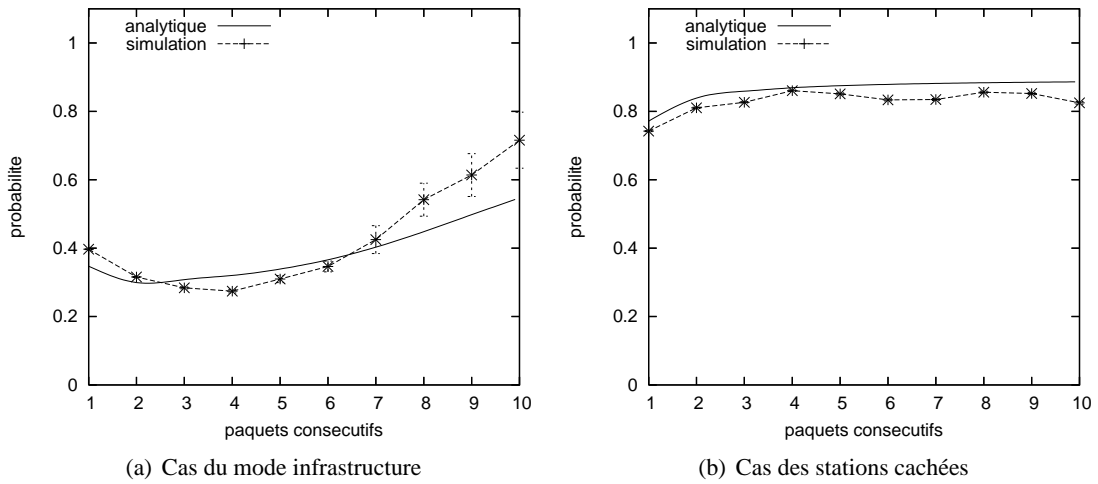


FIG. 5.12 – Validation de la métrique d'équité  $\alpha_i$

d'accès classique de 802.11 et le mécanisme RTS/CTS conçu pour lutter contre les collisions dans ce scénario bien précis.

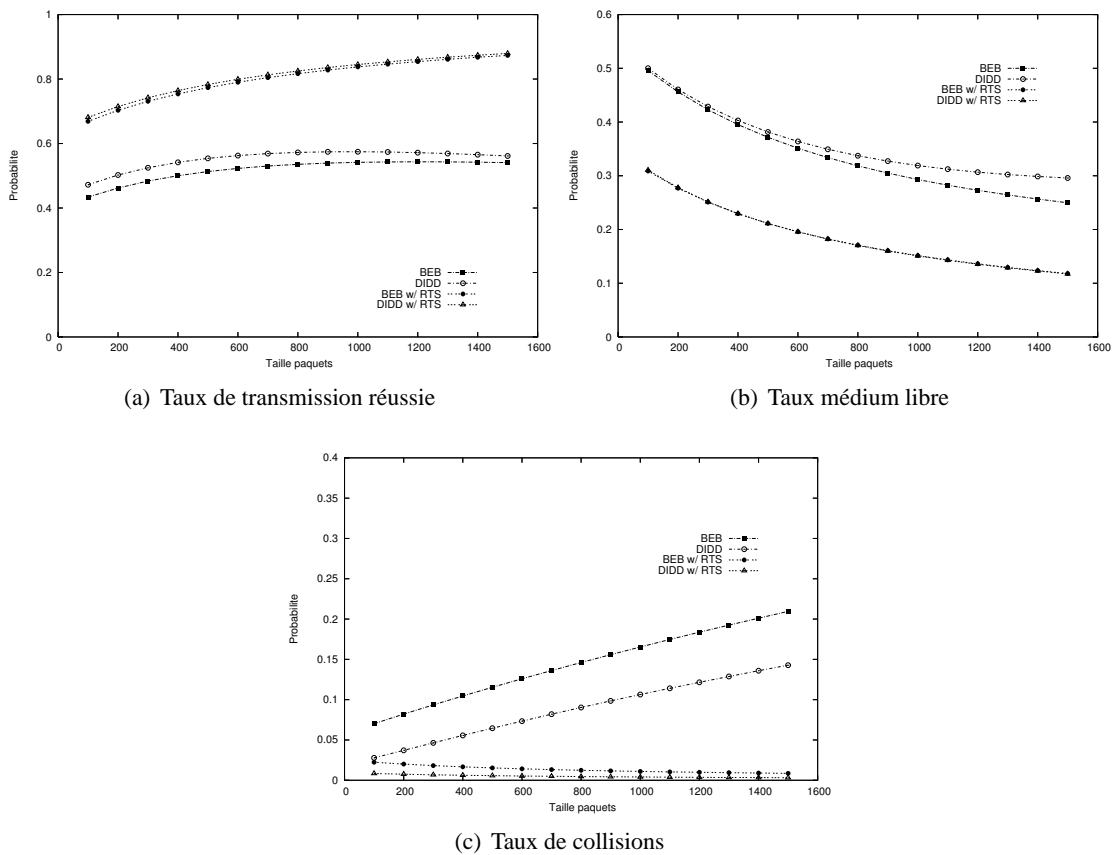


FIG. 5.13 – Mesures de performances - nœuds cachés

Les figures 5.13(a), 5.13(b) et 5.13(c) indiquent l'utilisation du médium pour les algorithmes BEB et DIDD en considérant le mécanisme classique de 802.11 puis le mécanisme RTS/CTS. Notons que, logiquement, le taux d'inutilisation du canal décroît lorsque la taille des paquets augmente ou que le débit diminue. L'utilisation du mécanisme RTS/CTS conduit à une nette amélioration des performances grâce à un taux de collisions bien plus faible. Soulignons que, quelque soit l'algorithme de gestion de collisions utilisé, l'utilisation du mécanisme RTS/CTS semble rendre le taux de transmission réussie assez insensible à la taille des paquets : les écarts de performances observés sont faibles. Soulignons que l'utilisation de RTS/CTS semble masquer l'efficacité des mécanismes de gestion de la fenêtre de contention : les performances obtenues en utilisant RTS/CTS sont toujours très similaires. Globalement, l'utilisation de DIDD conduit à moins de collisions et à un taux de transmission réussie bien plus important.

La figure 5.14(a) nous renseigne sur la distribution des fenêtres de backoff. Dans le cas de DIDD, on note que la fenêtre de contention est soit dans sa taille minimale, soit dans sa taille maximale la plupart du temps (à plus de 90% dans le cas de l'accès standard, à plus de 60% lorsque RTS/CTS est utilisé). En fait, en dehors de la fenêtre initiale, la distribution des fenêtres de backoff semble assez insensible aux algorithmes de gestion de fenêtre utilisés.

L'équité à court terme ( $\alpha_i$ ) est représentée sur la figure 5.14(b). Quelque soit l'algorithme utilisé (BEB ou DIDD), quelque soit le mécanisme de transmission utilisé (mécanisme standard ou RTS/CTS), le comportement inéquitable est flagrant. Dans chaque cas, la probabilité de réussir la première transmission est faible : autour des 20%. En revanche, dès la première transmission réussie, la probabilité de transmettre un second paquet avec succès et sans être interrompu se situe autour des 70% pour se stabiliser à plus de 80% de taux de réussite de transmission d'un paquet dès le 3<sup>ème</sup> paquet transmis avec succès et sans interruption. L'utilisation du mécanisme RTS/CTS accentue l'iniquité soulignée. DIDD apparaît également moins équitable que BEB qui tire profit de son comportement agressif pour revenir dans sa fenêtre initiale dès la première transmission réussie.

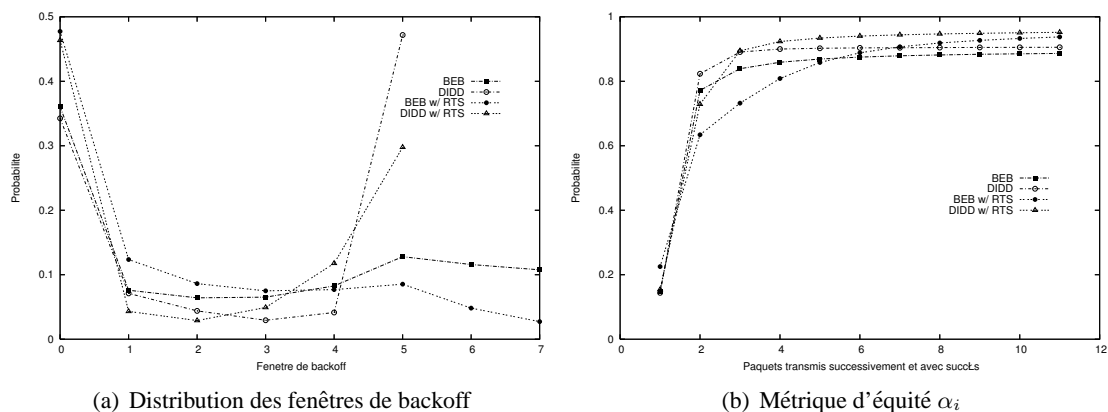


FIG. 5.14 – Mesures de performances - nœuds cachés

### 5.5.4 Étude du scénario des trois paires

Nous nous intéressons au scénario des trois paires dans le cas où les paires extérieures sont hors de portée et où la paire centrale partage le médium avec les deux paires extérieures. Nous

allons regarder deux sous cas : *i*) tout d'abord lorsque le partage du médium signifie que les nœuds sont dans la même zone de transmission puis *ii*) lorsque le partage du médium signifie que la paire centrale est dans la même zone de transmission qu'une paire extérieure mais dans la zone de détection porteuse de l'autre paire. Dans ce cas, la norme impose l'utilisation d'EIFS ( $360\mu s$ ) à la place de DIFS ( $50\mu s$ ). Nous considérons quatre algorithmes de backoff : BEB, DIDD, MILD et BEB inversé. Ce dernier affiche un comportement symétrique à BEB en fixant la taille de fenêtre initiale à celle de la fenêtre maximale. Le but est d'obtenir plus d'équité que BEB.

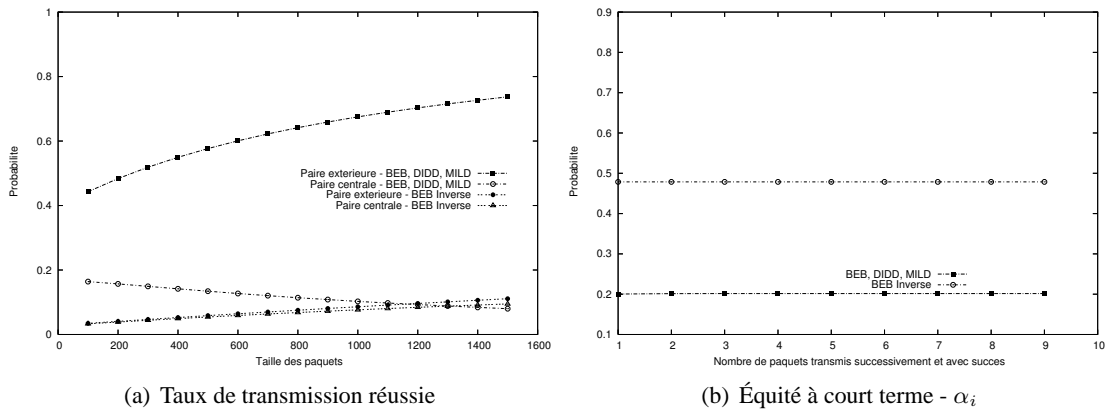


FIG. 5.15 – Scénario des 3 paires - DIFS

Les figures 5.15(a) et 5.15(b) représentent, respectivement, les performances et l'équité dans le cas *i*). Notons, que dans ce scénario, les collisions étant négligeables, le taux de transmission réussie est le même quelque soit l'algorithme de gestion de fenêtre de contention, excepté pour BEB inversé. Logiquement, lorsque la taille des paquets augmente, le taux d'utilisation du médium augmente également. L'iniquité à long terme de BEB, DIDD et MILD est importante : partageant le médium avec deux voisins, la paire centrale transmet peu de paquets (de quelques pourcent à moins de 20%). Cela est principalement dû à la petite taille de fenêtre de contention. Cette iniquité augmente avec la taille des paquets ou lorsque le débit diminue. En revanche, à long terme, BEB inversé apparaît équitable : le taux de transmission réussie pour la paire centrale est proche de celui des paires extérieures. L'équité à court terme est également moins forte avec BEB inversé qu'avec BEB, DIDD ou MILD (figure 5.15(b)). Soulignons que d'après la mesure de  $\alpha_i$  obtenue ici pour BEB inversé ( $1/2$ ), cet algorithme de gestion de fenêtre de contention apparaît même très équitable :  $1/2$  correspond à ce qui serait obtenue avec une allocation max-min. Bien que BEB, DIDD et MILD soient inéquitable, il faut souligner que, contrairement au scénario des 3 paires,  $\alpha_i$  reste constant en fonction de  $i$ .

Si une des deux paires extérieures partage le médium avec la paire centrale non plus sous l'angle du partage de la même zone de transmission mais du partage de la même zone de détection de porteuse (cas *ii*)), nous pouvons noter que l'iniquité à long terme est plus forte tandis que l'équité à court terme est améliorée pour BEB, DIDD et MILD (voir figures 5.16(a) et 5.16(b)).

### 5.5.5 Étude des nœuds transitifs

Les nœuds transitifs introduisent une dissymétrie dans le scénario bien connu des nœuds cachés. Ici, un seul nœud subit des collisions : effectivement dans ce cas là, le médium apparaît

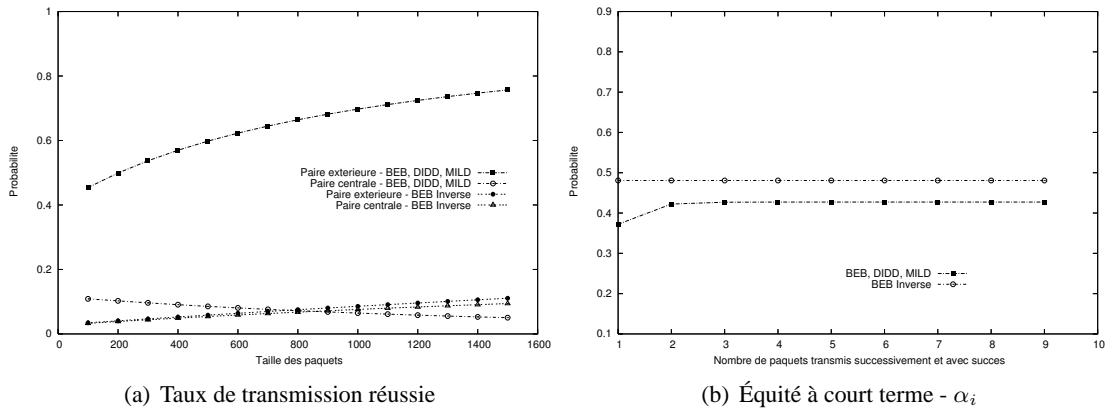


FIG. 5.16 – Scénario des 3 paires - EIFS

toujours libre pour les deux émetteurs mais l'un des récepteurs étant exposé aux deux transmissions, un taux de collisions important pour l'une des paires communicantes est observé (voir la figure 5.9). Nous considérons ici les algorithmes BEB, DIDD, MILD et BEB inversé.

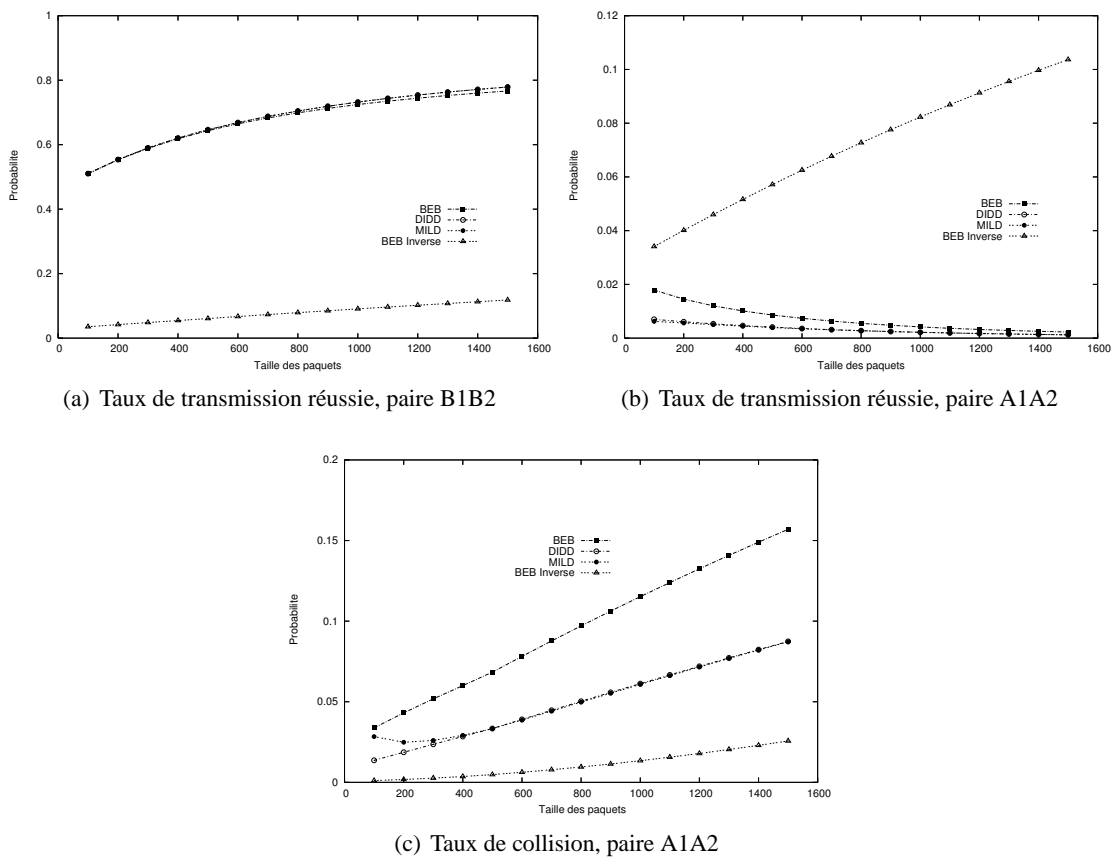
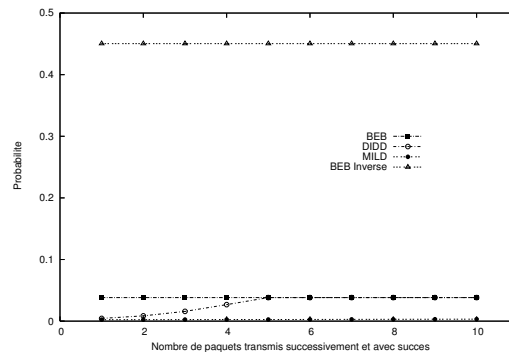


FIG. 5.17 – Scénario des nœuds transitifs - performances

FIG. 5.18 – Scénario des nœuds transitifs - Équité  $\alpha_i$ 

Le taux de transmission réussie pour les paires  $A_1A_2$  et  $B_1B_2$  est reporté sur les figures 5.17(a) et 5.17(b) respectivement. On peut observer l'iniquité à long terme résultant de cette configuration. Le taux de transmission réussie pour la paire  $B_1B_2$  est le même lors de l'utilisation de BEB, DIDD et MILD : dans chaque cas l'émetteur reste dans la fenêtre de contention initiale car l'émetteur  $B_1$  ne subit pas de collision. L'utilisation de BEB inversé, dont la fenêtre de contention initiale correspond à la taille maximale de la fenêtre de contention des algorithmes BEB, DIDD et MILD, conduit à une efficacité beaucoup plus faible. Une fois de plus, l'augmentation de la taille des paquets ou la diminution du débit de transmission améliore les performances. En revanche, les résultats pour la paire  $A_1A_2$  sont totalement différents à cause du nombre important de collisions rencontrées (figure 5.17(c)) : ici BEB inversé affiche les meilleures performances alors que BEB, DIDD et MILD conduisent à un taux de succès très faible. L'émetteur  $B_1$  étant dans une situation sans collision, il est toujours dans sa fenêtre de contention initiale qui, dans le cas de BEB inversé, correspond à la fenêtre maximale : ainsi l'émetteur  $A_1$ , subissant des collisions, se trouve dans une fenêtre de taille plus petite et a plus de chances de transmettre un paquet avec succès. Les meilleures performances affichées par BEB par rapport à MILD et DIDD sont à mettre sur le compte de son comportement agressif.

La métrique d'équité, vue de la paire  $A_1A_2$  est reportée figure 5.18. BEB inversé affiche ici la meilleure équité alors que l'utilisation des algorithmes BEB, DIDD et MILD conduit à un comportement tout à fait inéquitable. BEB semble un peu plus équitable que DIDD et MILD car la métrique  $\alpha_i$  reste indépendante du nombre de transmissions successives.

### 5.5.6 Influence du médium sur les performances et l'équité

Finalement les résultats obtenus précédemment le sont sous l'hypothèse d'un canal radio idéal. Dans le paragraphe 5.4.5 nous avons rappelé quelques points fondamentaux pour la modélisation du canal radio et comment en tenir compte dans un modèle analytique. La question à laquelle nous souhaitons répondre est : dans un cadre réaliste, en tenant compte d'une réception probabiliste due à l'environnement radio, comment évoluent les performances et l'équité ? Nous nous intéressons ici à deux résultats : l'impact d'erreurs indépendantes et l'impact d'erreurs corrélées. Nous considérons une seule type de topologie : celle des nœuds cachés utilisant CSMA/CA.

**Erreurs liées aux PER/BER** Nous supposons ici un taux d'erreur bit (BER) constant tel que rencontré dans le cas du bruit blanc additif gaussien. Dans le modèle PEPA, la probabilité d'erreur est introduite sous la forme d'un taux d'erreur paquet ( $PER = 1 - (1 - BER)^{bit}$ ), où  $bit$  est la longueur du paquet en bit : le PER augmente avec la taille de paquet pour une valeur de BER fixée.

La probabilité de succès d'une transmission en fonction de la taille des paquets est présentée figure 5.19(a). Pour un BER élevé ( $BER \geq 10e^{-4}$ ), les transmissions réussies sont négligeables. Pour un BER plus faible ( $BER = 10e^{-5}$ ), le taux de transmission réussie diminue en fonction de la taille des paquets : les transmissions erronées proviennent plus des erreurs de transmission que des collisions. En revanche, pour des taux d'erreurs beaucoup plus faible ( $BER \leq 10e^{-6}$ ) les courbes présentent un maximum qui met en évidence l'existence d'un compromis entre erreurs de transmission (plus importantes pour les grands paquets) et collision (plus importante pour les petits paquets). Ces premiers résultats soulignent l'importance de l'erreur de transmission et l'intérêt de notre modèle pour en rendre compte.

Nous représentons la métrique d'équité  $\alpha_i$  en fonction de la taille de paquets et pour plusieurs BER sur la figure 5.19(b). Notons que l'équité à court terme augmente lorsque le BER augmente car la probabilité qu'un nœud monopolise le médium diminue. En effet, avec un  $TEB = 10e^{-5}$  et des paquets de 1000 octets, la couche MAC se comporte de manière équitable à court et long terme car la valeur de  $\alpha_i$  est proche de 1/2 quelque soit  $i$ . Cela signifie qu'après chaque transmission les nœuds ont la même probabilité (1/2) d'envoyer correctement un paquet. Ce résultat montre que l'iniquité bien connue de 802.11 est fortement réduite lorsque l'on tient compte du BER.

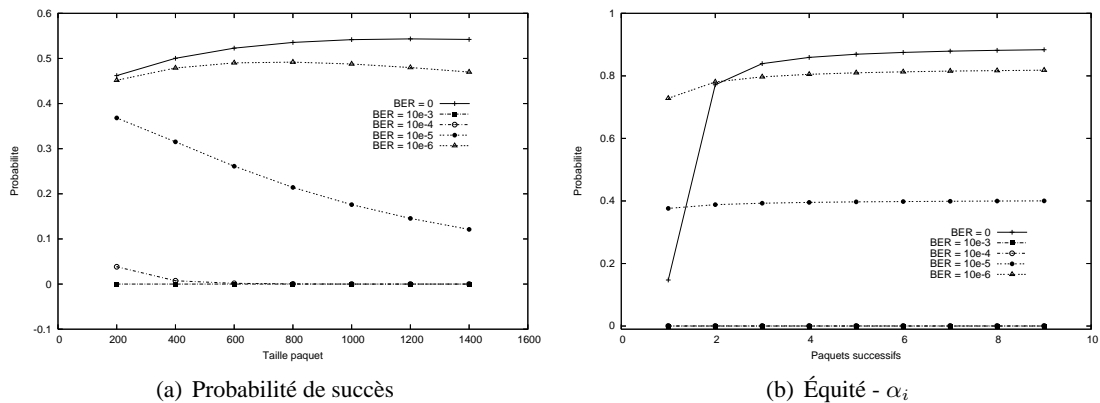


FIG. 5.19 – Influence du TEB sur les performances et l'équité

**Influence d'erreurs corrélées** On s'intéresse maintenant à l'influence d'erreurs corrélées conformément au modèle de Gilbert-Elliot. Ce modèle permet l'introduction de rafales d'erreurs modélisant, par exemple, un évanouissement dû au phénomène de multi-chemins. Nous avons considéré plusieurs types de rafales d'erreurs que l'on peut présenter schématiquement par :  $(i, j)$  tel que  $i$  représente le nombre de paquets correctement transmis consécutivement ( $i \in [3, 5, 10, 15]$ ) et  $j$  le nombre de paquets erronés consécutivement (ici  $j = 5$ ).

La figure 5.20(a) illustre l'impact des erreurs corrélées sur l'efficacité : fort logiquement, plus la période sans erreur est courte et plus l'efficacité augmente. La figure 5.20(a) montre également que si l'on considère un ratio équivalent de nombres paquets en erreurs et de nombre de paquets

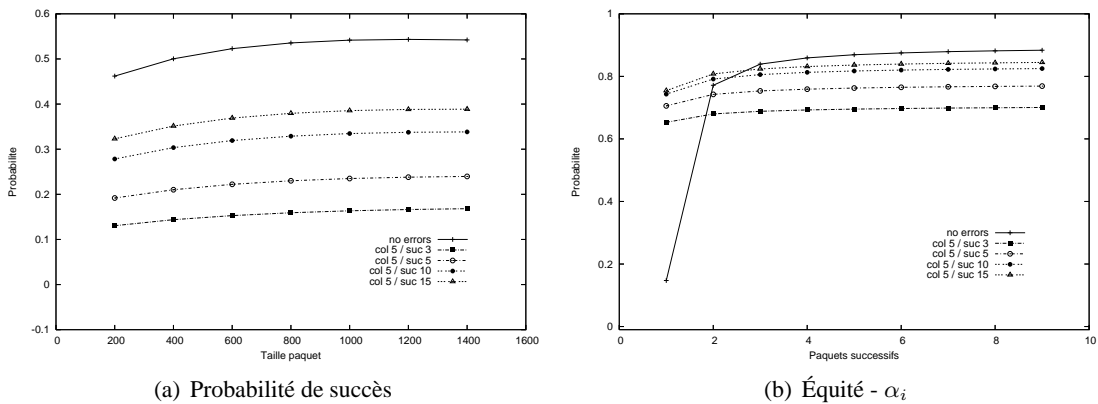


FIG. 5.20 – Influence des erreurs corrélées sur les performances et l'équité

corrects, la probabilité d'occupation du médium est divisée par deux. Cela souligne que BEB gère correctement les erreurs corrélées : un mécanisme de gestion de collisions moins agressif n'affichera pas d'aussi bonnes performances.

La figure 5.20(b) montre l'effet d'erreurs corrélées sur l'équité. Clairement, l'introduction d'erreurs améliore l'équité dans le cas des stations cachées : alors qu'une transmission subit des erreurs, la transmission courante peut accéder au canal et réussir à transmettre ses paquets. En revanche, l'introduction d'erreurs corrélées n'affecte pas de manière significative l'équité dans le réseau : effectivement EBE considère que le médium est dans un état correct dès qu'une seule transmission a réussi. Autrement dit, BEB supporte très bien ce type d'erreurs.

## 5.6 Synthèse et conclusions

Que retenir du travail présenté dans cette section ? De notre point de vue, deux résultats importants sont à souligner :

- une modélisation générique des réseaux multi-sauts permettant l'étude de plusieurs topologies de réseaux, de plusieurs algorithmes de gestion de fenêtre de contention, de plusieurs hypothèses de trafic, de différents types de canaux radio, etc. Cette modélisation tire parti de l'approche compositionnelle d'un algèbre de processus stochastique (PEPA) et de la méthode d'agrégation proposée pour réduire significativement la taille de l'espace d'états permettant d'envisager des modèles complexes. Ce modèle nous permet d'obtenir des indicateurs précis de performances : taux de collisions, taux d'utilisation du médium, taux de transmission et nous permet également de souligner les phénomènes d'équité à court terme et à long terme.
- sous l'angle de l'évaluation de performances, nous avons étudié 3 topologies multi-sauts et le mode infrastructure (résultats non présentés ici) en nous intéressant au compromis performance / équité suivant les stratégies de fenêtre de contention utilisée (BEB, MILD, DIDD, BEB inversé). Plusieurs débits et plusieurs tailles de paquets ont été considérés. Quels enseignement tiré de l'ensemble de ces résultats ?
  - dans le cas de topologies à fort taux de collisions (ex : nœuds cachés), l'utilisation du mécanisme RTS/CTS permet de réduire le taux de collisions et d'améliorer le taux de



FIG. 5.21 – Chaîne de transmission multi-sauts

transmission réussie. Toutefois, dès que l'utilisation du RTS/CTS est faite, la stratégie de gestion de fenêtre de contention importe peu tant que la fenêtre de contention initiale correspond à la fenêtre minimale. Les performances obtenues par BEB, DIDD et MILD sont alors similaires. En revanche, l'utilisation du RTS/CTS accentue le phénomène d'iniquité à court terme.

- dans le cas de topologies sans collision (ex : 3 paires) ou avec un taux de collisions faibles, tous les algorithmes sont finalement équivalents tant que la fenêtre de contention initiale correspond à la fenêtre minimale.
- l'utilisation d'un algorithme de gestion de fenêtre de contention dont la fenêtre initiale est dimensionnée à la fenêtre maximale (comme BEB inversé) favorisera toujours le comportement équitable au détriment des performances.
- l'augmentation de la taille des paquets ou la diminution du débit d'émission a toujours pour effet d'accroître l'iniquité observée.
- enfin, la modélisation d'un médium réaliste semble conduire à une diminution de l'efficacité et une amélioration de l'équité. Si cette conclusion est issue du scénario des nœuds cachés et ne saurait être généralisée, elle invite à reconsidérer un certain nombre de résultats jusqu'ici connus.

Quelle(s) suite(s), quelle(s) perspective(s) donner à ces travaux ? D'un côté, il convient de poursuivre les études amorcées : il en est ainsi pour l'investigation sur les sources de trafic non saturées mais aussi, et surtout, pour souligner l'influence du médium radio sur les performances et l'équité. Bien sûr, plusieurs topologies peuvent être considérées comme celle de la chaîne de transmission (figure 5.21). En revanche, lorsque l'on observe les principales stratégies de gestion de fenêtre de contention, on observe uniquement les méthodes de gestion de fenêtre suivantes :

- en cas de transmission réussie : retour immédiat vers la fenêtre initiale (BEB) ou diminution exponentielle de la fenêtre de contention (DIDD) ou diminution linéaire (MILD)
- en cas de collisions : transition immédiate vers la fenêtre de contention maximale ou accroissement linéaire de la fenêtre de contention ou accroissement exponentiel.

Soit neuf combinaisons possibles d'algorithmes de backoff qu'il nous semble important d'étudier pour souligner quelle stratégie est préférable compte tenu de la topologie, des hypothèses de trafic, etc.

Un autre challenge retient notre attention : c'est l'évaluation de performances de topologies aléatoires. Si une topologie est définie par une matrice d'interactions décrivant les nœuds partageant la même zone de transmission et les nœuds au sein de même zone de détection de porteuse, alors est-il possible de construire avec l'algèbre de processus PEPA un modèle où les synchronisations des composantes seraient issues de la matrice d'interactions ?



# Bibliographie

- [1] J.-M. Fourneau, L. Kloul, and F. Valois. Performance modelling of hierarchical cellular networks using PEPA. *Performance Evaluation, Elsevier Ed.*, 50(2-3) :83–99, October 2002.
- [2] T. Razafindralambo, J.-M. Gorce, and F. Valois. How realistic medium assumptions increase fairness? In *First IRAMUS Workshop on radio interfaces for WSN and MANET networks*, Val Thorens, France, January 2007.
- [3] T. Razafindralambo and F. Valois. Performance evaluation of backoff algorithms in 802.11 ad-hoc networks. In *3d ACM Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN)*, Malaga, Spain, October 2006. ACM.
- [4] T. Razafindralambo and F. Valois. Stochastic behavior study of backoff algorithms in case of hidden terminals. In *17th International Symposium on Personal, Indoor and Mobile Radio Communication (PIMRC)*, Helsinki, Finland, September 2006. IEEE.
- [5] T. Razafindralambo and F. Valois. Modeling methodology for wireless LANs performance evaluation. In *4th International Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks (Het-Nets)*, Ilkley, West Yorkshire, U.K., September 2006.
- [6] L. Kloul and Valois F. Investigating unfairness scenarios in manet using 802.11b. In *2nd Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN)*, Montréal, Canada, October 2005. ACM.
- [7] T. Razafindralambo, J.-M. Gorce, and F. Valois. Influence du médium radio sur le phénomène d'équité dans le cas des stations cachées. In *Rencontres Francophones sur l'Algorithmique pour les Télécommunications (Algotel)*, Ile d'Oléron, France, May 2007.
- [8] T. Razafindralambo and F. Valois. Modélisation et évaluation de performances de stratégies de backoff. In *CFIP'2006 : Colloque francophone sur l'ingénierie des protocoles*, Tozeur, Tunisie, November 2006.



## Chapitre 6

# Conclusion, perspectives

Les travaux présentés dans ce manuscrit ici se situent dans le contexte des réseaux radio multi-sauts qu'ils soient soumis à la mobilité des nœuds (type réseaux ad hoc - MANET) ou contraints en énergie et cardinalité de réseau (type réseaux de capteurs - WSN). Nous nous sommes intéressés à deux points clefs :

- l'auto-organisation ou comment structurer un réseau pour que les protocoles de communications soient plus efficaces,
- la modélisation et évaluation de performances de topologies multi-sauts.

### 6.1 Synthèse sur l'auto-organisation

L'idée répandue lorsque l'on s'intéresse à la problématique de l'auto-organisation c'est que cette notion est liée au comportement *autonome* des protocoles distribués ou localisés comme le routage : lors de changements topologiques, les protocoles s'adaptent à l'évolution du voisinage de façon autonome, sans contrôle extérieur. En quelque sorte, les nœuds s'organisent pour intégrer les nouveaux arrivant ou pour s'adapter aux départs. Ce n'est pas l'idée que nous avons défendu ici. Nous avons défini l'auto-organisation comme étant liée à la notion topologie logique du réseau. L'idée clef sur laquelle nous avons travaillé au cours des chapitres 2, 3 et 4 c'est que, avant de router, avant de configurer un nœud, avant de disséminer une information, il faut structurer, organiser le réseau. Le but recherché est d'introduire de la stabilité, de la persistance, de la robustesse et de favoriser le passage à l'échelle. *In fine*, l'objectif n'est pas d'organiser pour organiser mais de développer des protocoles de communication sur cette vue logique tel qu'ils soient plus efficaces que si l'on avait considéré le réseau à *plat*, comme c'est essentiellement le cas aujourd'hui. Basiquement, la construction d'une vue logique requiert l'utilisation d'une fonction  $f$  permettant de construire une relation entre un espace physique et un espace logique. Les principales stratégies de construction se basent sur des clusters, des backbones et des DHT. Nous avons rappelés et discutés plusieurs propriétés fondamentales qui sont requises :

- se baser sur des interactions locales uniquement,
- faire émerger une structure globale à partir d'informations locales,
- être réactif aux changements locaux et être robuste,,
- proposer une structure d'auto-organisation non orientée source,
- faciliter le passage à l'échelle.

Dans les travaux présentés, nous nous sommes concentrés sur la construction de topologies logiques sous forme d'arbres, de treillis et de clusters. Nous avons appliqué ces concepts du domaine des réseaux ad hoc puis des réseaux de capteurs. Dans chaque cas, nous avons proposé une auto-organisation tenant compte des spécificités de chacun des réseaux considérés :

- dans le cas des réseaux ad hoc, les principales contraintes viennent de la mobilité, de la dynamique des nœuds, de leur hétérogénéité et de l'évolution du voisinage,
- dans le cas des réseaux de capteurs, le déploiement progressif de la topologie physique, la technologie embarquée utilisée limitant les opérations et le peu d'énergie exploitable sont autant de points durs à adresser de façon globale.

Notre démarche de travail a toujours été la même : proposer des algorithmes distribués ou localisés pour la construction de ces topologies logiques. Parce que la topologie physique est dynamique, nous avons également proposé des algorithmes de maintenance de ces topologies, en favorisant des modifications locales uniquement. Ensuite, basé sur ces topologies, nous avons étudié plusieurs problématiques clefs de ces réseaux pour montrer comment une auto-organisation pouvait en améliorer le comportement et les performances.

Ces auto-organisations ont été caractérisées sous les angles complémentaires de la complexité algorithmique, de la cardinalité de la structure logique, de la robustesse, du coût protocolaire et de la capacité en termes de flots. Les principaux enseignements que nous en avons tiré sont que l'auto-organisation introduit une importante stabilité en *masquant* une partie de la dynamique de la topologie physique et qu'elle la simplifie. Nous avons également montré que ces topologies logiques sont persistantes, fournissant un référentiel commun pour les nœuds du réseau. Ces propriétés sont des atouts clefs pour le développement de protocoles de communication efficaces.

Pour montrer qu'il est préférable d'organiser un réseau au lieu de le considérer à plat, nous avons étudié et développé plusieurs cas précis d'exploitation des ces topologies logiques d'auto-organisation. Le routage, la localisation, l'interconnexion de réseaux ad hoc avec Internet via un point d'accès, l'inondation, l'agrégation de données, la gestion de puits mobiles sont autant des challenges que nous relevés sous l'angle exclusif de réseau structuré. A chaque fois, et dans la mesure du possible, nous avons proposé une évaluation de performances dans le but de quantifier le gain dû à l'auto-organisation. Généralement, nous avons constaté que l'auto-organisation présente le meilleur compromis performances/surcoût protocolaire. Mais surtout que l'auto-organisation offre un support efficace pour le déploiement d'une famille de protocoles de communications en facilitant le déploiement de certains d'entre eux.

Nous avons également souligné qu'aucune des métriques classiques ne permettaient de refléter le comportement intrinsèque d'une auto-organisation : quel est le protocole qui organise *le mieux* ? Pour répondre à cette question, nous avons appliqué la notion d'entropie statistique au cas particulier des réseaux radio multi-sauts. Pour mesurer l'entropie globale du réseau, nous nous sommes basés sur les interactions locales entre les nœuds. Ces interactions évoluent en fonction de la présence des nœuds, de la qualité du lien radio et de la simplification de la topologie réalisée par les protocoles d'auto-organisation. Les résultats montrent clairement qu'un réseau structuré est toujours plus efficace qu'un réseau à plat et que les protocoles d'auto-organisations basés sur l'élection de dominants sont les protocoles d'auto-organisation les moins efficaces.

Finalement ce travail permet d'avoir des éléments de réponses à la question suivante : devons-nous structurer un réseau radio multi-sauts ? Au vu des résultats présentés dans ce manuscrit, nous répondons par l'affirmative.

## 6.2 Synthèse sur l'évaluation de performances

Les performances mesurées dans la section précédente, et de façon plus générale dans toutes topologies radio multi-sauts, montrent l'existence de topologies particulières entraînant des situations d'inefficacité et/ou d'iniquité. La plupart des études utilisent l'outil de simulation pour mesurer ces critères de performances en fonction du type de la topologie, du type de trafic, de l'algorithme de gestion de collisions, etc. Dans la dernière partie de ce manuscrit, nous avons proposé un modèle analytique basé sur l'utilisation d'un algèbre de processus stochastique (PEPA). L'objectif était double :

- proposer un modèle *générique* tirant parti de l'approche compositionnelle de l'algèbre de processus utilisé et de sa méthode d'agrégation stochastique pour fournir une modélisation fine des éléments clés d'une topologie d'un réseau radio.
- mesurer les performances et l'équité de plusieurs protocoles de gestion de collisions sous des hypothèses différentes de trafics, de topologies, de propriétés du canal radio, etc.

Cette démarche nous a permis de construire un modèle exploitable dans le but de mesurer les performances (débit, taux de collisions, taux du canal libre) et l'équité (à court terme, à long terme) de topologies décrites par les six composantes PEPA suivantes : intensité de la source de trafic, propriétés du canal radio, algorithmes de gestion de collisions, méthodes d'accès au canal radio, files d'attente en émission, matrices d'interactions entre les nœuds. Plusieurs résultats ont été montrés dans le cas d'un accès au médium radio de type CSMA/CA :

- tout d'abord le compromis performance/équité est une constante quelque soit la topologie considérée, quelque soit l'algorithme de gestion de collisions, etc. ;
- une stratégie efficace pour favoriser l'équité au détriment de l'efficacité est de dimensionner la taille de la fenêtre de backoff à sa taille maximale ;
- l'augmentation de la taille des paquets ou la diminution du débit en émission conduit toujours à accentuer l'iniquité observée ;
- nous avons également souligné qu'en tenant compte d'une modélisation fine du canal radio, l'équité s'améliorait et que les performances se dégradent.

## 6.3 Perspectives

Comme ce manuscrit apporte des contributions sur deux sujets, deux familles de perspectives vont être mises en avant.

Dans le domaine de l'auto-organisation, d'autres stratégies méritent d'avoir notre attention comme les DHT. Mais de façon plus générale, l'abstraction d'une structure logique d'auto-organisation à une fonction  $f$  autorise probablement une approche plus formelle : qu'est-ce qu'une *bonne* auto-organisation ? Quelles sont les propriétés que doit avoir  $f$  ? Les travaux menés ici, s'ils répondent aux exigences des réseaux ad hoc et des réseaux de capteurs, n'en sont pas moins spécifiques. Ils montrent comment une auto-organisation favorise l'efficacité des protocoles de communication. Ainsi, d'autres domaines d'applications, d'autres familles de réseaux méritent notre attention comme les réseaux radio maillés.

L'autre chemin à poursuivre est lié à l'évolution des réseaux ces dernières années et notamment la production en masse de capteurs communicants, fortement contraint, pour lesquels aucune identification matérielle ne sera disponible. De nouvelles problématiques apparaissent alors : comment router sans adresse ? comment organiser sans adresse ? Les solutions sont certainement

à chercher vers le routage par contenu. Dans ce domaine, la description d'un protocole de routage par un langage récursif devrait permettre d'adresser ces points durs.

Dans le domaine de l'évaluation de performances, il va sans dire que l'étude de topologies plus générales doit être envisagée. Notamment, comment à partir d'une matrice d'interactions des nœuds d'un réseau, construire un modèle PEPA issu de l'approche générique que nous avons eu ? Si une topologie est définie par une matrice d'interactions décrivant les nœuds partageant la même zone de transmission et les nœuds au sein de même zone de détection de porteuse, alors est-il possible de construire avec l'algèbre de processus PEPA un modèle où les synchronisations des composantes seraient issues de la matrice d'interactions ?

# Bibliographie

- [1] Conversation privée avec Dominique Barthel à l'occasion du lancement du CRE 46130157, France Télécom R&D, Grenoble, Avril 2005.
- [2] Easinet research group. <http://www.easinet.cn>.
- [3] ns-2 : The Network Simulator, 2004.
- [4] OPNET Modeler, Scalable Network Simulation, <http://www.opnet.com/>, 2004.
- [5] Cedric Adjih, Philippe Jacquet, and Laurent Viennot. Computing connected dominating sets with multipoint relays. *Ad hoc and Sensor Wireless Networks*, 1(1-2), January 2005.
- [6] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks : a survey. *Computer Networks, Elsevier Ed.*, 28(2-3) :393–422, 2002.
- [7] K. M. Alzoubi, P.-J. Wan, and O. Frieder. Distributed heuristics for connected dominating set in wireless ad hoc networks. *IEEE ComSoc/KICS Journal of Communications and Networks, Special Issue on Innovations in Ad Hoc Mobile Pervasive Networks*, 4(1) :22–29, march 2002.
- [8] S. Bannerjee and S. Khuller. A clustering scheme for hierarchical control in wireless networks. In *INFOCOM*, pages 1028–1037, Anchorage, USA, April 2001.
- [9] Lichun Bao and J.J. Garcia Luna Aceves. Topology management in ad hoc networks. In *International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, Anapolis, USA, June 2003. ACM.
- [10] R. Barr. *Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad hoc Wireless, and Peer-to-Peer Networks.*, chapter Scalable Wireless Ad hoc Network Simulation, pages 297–311. CRC Press, 2005.
- [11] S. Basagni. Distributed clustering for ad hoc networks. *Proc of the International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN)*, 1999.
- [12] S. Basagni, M. Mastrogiovanni, A. Panconesi, and C. Petrioli. Localized protocols for ad hoc clustering and backbone formation : A performance comparison. *IEEE Transactions on Parallel and Distributed Systems, Special Issue on Localized Communication and Topology Protocols for Ad Hoc Networks (S. Olariu, D. Simplot-Ryl, and I. Stojmenovic, editors)*, 17(4) :292–306, April 2006.
- [13] C. Berge. *Theory of Graphs*. Dover Publications, 2001. 256 p.
- [14] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. Fair medium access in 802.11 based wireless ad-hoc networks. In *ACM MOBIHOC*, Boston, Massachusetts, USA, August 2000.

- [15] L. Boltzmann. Über die Beziehung zwischen dem zweiten Hauptsatz der mechanischen Wärmetheorie und der Wahrscheinlichkeitsrechnung respektive den Sätzen über das Wärme-gleichgewicht. *Wien. Ber.*76, 373-435, *Band II*, pages 164–223, 1877.
- [16] D. Braginsky and D. Estrin. Rumor routing algorithm for sensor networks. *Proc. of 1st ACM international workshop on Wireless sensor networks and applications, Atlanta, USA*, pages 22–31, 2002.
- [17] Raffaele Bruno, Marco Conti, and Enrico Gregori. Mesh networks : Commodity multihop ad hoc networks. *IEEE Communications Magazine*, 43(3) :123–131, March 2005.
- [18] S. Butenko, X. Cheng, D.-Z. Du, and P. M. Pardalos. On the construction of virtual backbone for ad hoc wireless networks. In *Cooperative Control : Models, Applications and Algorithms*, volume 1 of *Cooperative Systems*, chapter 3, pages 43–54. Kluwer Academic Publishers, January 2003.
- [19] S. Calomme and G. Leduc. Conception d’un protocole de contrôle de topologie pour les overlays construits sur des réseaux ad hoc. In *Colloque Francophone sur l’Ingénierie des Protocoles*, Tozeur, Tunisie, November 2006.
- [20] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing (WCMC) : Special issue on Mobile Ad Hoc Networking : Research, Trends and Applications*, 2(5) :483–502, August 2002.
- [21] M. Cardei, X. Cheng, X. Cheng, and D.-Z. Du. Connected domination in ad hoc wireless networks. In *International Conference on Computer Science and Informatics (CSI)*, North Carolina, USA, March 2002.
- [22] J. Cartigny, F. Ingelrest, D. Simplot-Ryl, and I. Stojmenovic. Localized MST and RNG based minimum energy broadcast protocols in ad hoc networks. *Ad Hoc Networks*, 3(1) :1–16, January 2005.
- [23] C. Chatfield. *The Analysis of Time Series*. CRC Press, 2004.
- [24] P. Chatzimisios, A.C. Boucouvalas, V. Vitsas, A. Vafiadis, A. Oikonomidis, and P. Huang. A simple and effective backoff scheme for the IEEE 802.11 MAC protocol. In *CITSA*, Orlando, Florida, USA, July 2005.
- [25] C. Chaudet, I. Guérin Lassous, E. Thierry, and B. Gaujal. Study of the impact of asymmetry and carrier sense mechanism in iee 802.11 multi-hops networks through a basic case. In *Proceedings of the 1st ACM Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, Venice, Italy, October 2004. IFIP, Vol. LNCS 3976/2006.
- [26] Yuanzhu Peter Chen and Arthur L. Liestman. Approximating minimum size weakly-connected dominating sets for clustering mobile ad hoc networks. In *International Symposium on Mobile Ad Hoc Networking and Computing (MOBICOM)*, pages 165–172, Lausanne, Switzerland, June 2002. ACM.
- [27] Imrich Chlamtac, Marco Conti, and Jennifer J. N. Liu. Mobile ad hoc networking : imperatives and challenges. *Ad Hoc Networks*, 1(1) :13–64, July 2003.
- [28] T. Clausen and P. Jacquet. Optimized link state routing protocol (OLSR). RFC 3626, IETF, October 2003.
- [29] K. Deng, A. Moore, M. Proc. IEEE Int. Symp. on Computational Intelligence in Robotics Nechyba, and pp. 246-50 1997 Automation, vol. 1. Learning to recognize time series :

- combining ARMA models with memory-based learning. *Proc. of IEEE Int Symposium on Computational Intelligence in Robotics and Automation*, 1 :246–250, 1997.
- [30] F. Dressler. Self-organization in ad hoc networks : Overview and classification. Research report, Autonomic Networking Group, Dept. of Computer Science, University of Erlangen, Germany, February 2006.
- [31] M. Ergen and A. Puri. Mewlana-mobile ip enriched wireless local area network architecture. In *Vehicular Technology Conference (VTC Fall)*, Vancouver, Canada, September 2002. IEEE.
- [32] P. Felber, K. Ross, E. Biersack, L. Garces-Erice, and G. Urvoy-Keller. Self-organization in spontaneous networks : the approach of dht-based routing protocols. *Ad Hoc Networks Journal, Elsevier Ed.*, 3(5), September 2005.
- [33] P. Felber, K. Ross, E. Biersack, L. Garces-Erice, and G. Urvoy-Keller. Structured peer-to-peer networks : faster, closer, smarter. *IEEE Data Engineering Bulletin, Special Issue on In-Network Query Processing*, 28(1) :55–65, March 2005.
- [34] Sudipto Guha and Samir Khuller. Approximation algorithms for connected dominating sets. In *European Symposium on Algorithms*, pages 179–193, Barcelona, Spain, Septembre 1996.
- [35] I. Guérin-Lassous. Autour de la notion de bande passante dans les réseaux radio multisautes, habilitation à diriger des recherches de l'université de Lyon et de l'insa Lyon, décembre 2005.
- [36] Z. Haas, J. Deng, B. Liang, P. Papadimitatos, and S. Sajama. *Wireless Ad Hoc and Sensor Networks*, chapter Wireless ad hoc networks. John Proakis, editor, Encyclopedia of Telecommunications. John Wiley, December 2002.
- [37] Z. J. Hass and M. R. Pearlman. Zrp : a hybrid framework for routing in ad hoc networks. pages 221–253, 2001.
- [38] J. Hillston. A compositional approach to performance modelling. *PhD. Thesis, The University of Edinburgh.*, 1994.
- [39] F. Ingelrest. *Localized Broadcast Protocols and Energy Efficiency in Ad Hoc and Sensor Networks*. PhD thesis, Université des Sciences et Technologies de Lille, June 2006.
- [40] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion : a scalable and robust communication paradigm for sensor networks. *IEEE/ACM Transactions on Networking (TON)*, 11 :2–16, 2000.
- [41] R. Jain. *The Art of Computer Systems Performance Analysis Techniques for Experimental Design, Measurement, Simulation, and Modeling*. John Wiley & Sons, Ltd, 1991. 720 p.
- [42] Jerzy Jaromczyk and Goedfried T. Toussaint. Relative neighborhood graphs and their relatives. In *Proceedings of the IEEE*, volume 80 of 9, pages 1502–1517, September 1992.
- [43] J. Jeong and J. Park. Autoconfiguration technology for IPv6-based mobile ad-hoc network. *IEEE ICON, Singapore*, August 2002.
- [44] M. Jiang, J. Li, and Y. C. Tay. Cluster based routing protocol (CBRP). Internet draft version 01, IETF, July 1999.
- [45] D. B. Johnson, D. A. Maltz, and Y.-C. Hu. The dynamic source routing protocol for mobile ad hoc networks (DSR). Internet draft version 09, IETF, April 2003.

- [46] F. Jonsson, U. and Alriksson, T. Larsson, P. Johansson, and G. Q. Maguire. Mipmanet - mobile ip for mobile ad hoc networks. In *Iterational Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pages 75–85, Boston, USA, May 2000. ACM, IEEE Press.
- [47] H. Karl and A. Willig. *Protocols and Architectures for Wireless Sensor Networks*. John Wiley & Sons, Ltd, 2006. 497 p.
- [48] B. Karp and H. Kung. GPSR : Greedy perimeter stateless routing for wireless networks. *Proc. of the Sixth ACM/IEEE Intl Conference on Mobile Computing and Networking (MobiCom '00), Boston, USA, 2003*.
- [49] David Kotz, Calvin Newport, and Chip Elliott. The mistaken axioms of wireless-network research. Technical report, Dartmouth College Computer Science, July 2003.
- [50] P. Krishna, N. Vaidya, M. Chatterjee, and D. Pradhan. A cluster-based approach for routing in dynamic networks. In *SIGCOMM*, pages 49–65, Cannes, France, April 1997. ACM.
- [51] F. Kuhn, T. Moscibroda, T. Nieberg, and R. Wattenhofer. Fast deterministic distributed maximal independent set computation on Growth-Bounded graphs. *19th International Symposium on Distributed Computing (DISC), Cracow, Poland, September 2005*.
- [52] T.H. Labella and F. Dressler. A bio-inspired architecture for division of labour in SANETs. *Proc. of 1st IEEE/ACM International Conference on Bio-Inspired Models of Network, Information and Computing Systems (IEEE/ACM BIONETICS 2006), Cavalese, Italy, 2006*.
- [53] N. Li, J.C. Hou, and L. Sha. Design and analysis of an MST-based topology control algorithm. In *INFOCOM*, pages 1702–1712, San Fransisco, USA, April 2003.
- [54] C. R. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. *IEEE Journal of Selected Areas in Communications*, 15(7) :1265–1275, 1997.
- [55] N. Malpani, J.-L. Welch, and N. Vaidya. Leader election algorithms for mobile ad hoc networks. In *Proceedings of the 4th international workshop on Discrete algorithms and methods for mobile computing and communications (DIALM)*, pages 96–103, Boston, USA, August 2000.
- [56] A. Marcucci, M. Nati, C. Petrioli, and A. Vitaletti. Directed diffusion light : low overhead data dissemination in wireless sensor networks. *IEEE 61st Vehicular Technology Conference, VTC 2005-Spring, Stockholm, Sweden, 4 :2538– 2545, 2005*.
- [57] S. Nesargi and R. Prakash. MANETconf : configuration of hosts in a mobile ad hoc network. *IEEE INFOCOM, New York, USA, June 2002*.
- [58] LAN MAN Standards Committee of the IEEE Computer Society. IEEE Std 802.11, 1999 Edition Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. *IEEE-SA Standards Board, 1999*.
- [59] LAN MAN Standards Committee of the IEEE Computer Society. IEEE Std802.15.4 Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low-Rate Wireless Personal Area Network (LR-WPANs). *IEEE-SA Standards Board, 2003*.
- [60] V. Parunak and S. Brueckner. Entropy and self-organization in multi-agent systems. *Proc. of the fifth international conference on Autonomous agents*, pages 124–130, 2001.

- [61] C. E. Perkins. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *SIGCOMM*, pages 234–244, London, United Kingdom, August 1994. ACM.
- [62] C. E. Perkins. *Ad hoc networking*. Addison-Wesley, 2001. 370 p.
- [63] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. *ACM SenSys, Baltimore, Maryland, USA*, 2004.
- [64] C. Prehofer and C. Bettstetter. Self-organization in communication networks : principles and design paradigms. *IEEE Communications Magazine*, 43(7) :78(85), 2005.
- [65] M. Schneider. Self-stabilization. *ACM Computing Surveys*, 25(1) :45–67, March 1993.
- [66] Freescale Semiconductor. MC13192 / MC13193 2.4 GHz Low Power Transceiver for the IEEE 802.15.4 Standard : Technical Data, April 2005.
- [67] Z. Shelby, C. Pomalaza-Rez, and Haapola J. Energy optimization in multihop wireless embedded and sensor networks. *IEEE 15th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Barcelone, Spain*, 2004.
- [68] J. Shin, J. Kim, K. Park, and D. Park. Railroad : virtual infrastructure for data dissemination in wireless sensor networks. *Proc. of the 2nd ACM Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor and Ubiquitous Networks (PE-WASUN), Montreal, Canada*, pages 168–174, 2005.
- [69] J. D. Solomon. *Mobile IP : the Internet unplugged*. Prentice-Hall, Inc., 1997. 350 p.
- [70] I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 13(1), January 2003.
- [71] I. Stojmenovic and J. Wu. *Mobile Ad Hoc Networking*, chapter Broadcasting and activity scheduling, pages 205–229. IEEE/Wiley, 2004.
- [72] F. Theoleyre. *Une auto-organisation et ses applications pour les réseaux ad hoc et hybrides*. PhD thesis, Laboratoire CITI, INSA Lyon, September 2006.
- [73] F. Theoleyre and Valois F. About the self-stabilization of a virtual topology for self-organization in ad hoc networks. In *7th International Symposium on Self-Stabilizing Systems (SSS)*, Barcelona, Spain, October 2005.
- [74] F. Theoleyre and Valois F. Virtual structure routing in ad hoc networks. In *International Conference on Communication (ICC)*, Seoul, Korea, May 2005. IEEE.
- [75] F. Theoleyre and Valois F. Performance evaluation of the routing protocols in manet : Classical versus self-organized approaches. In *5th International Conferences on Networking (Networking)*, Coimbra, Portugal, May 2006. IFIP, Vol. LNCS 3976/2006.
- [76] F. Theoleyre and F. Valois. *Réseaux mobiles ad hoc et réseaux de capteurs sans fil*, chapter Auto-organisation de réseaux ad hoc : concepts et impacts. Hermes Science Publications - Traité IC2, série Réseaux et Télécommunications, Avril 2006. ISBN 2-7462-1292-7.
- [77] G. Toussaint. The relative neighbourhood graph of a finite planar set. *Pattern Recognition*, 12 :261–268, 1980.
- [78] N. Vaidya. Weak duplicate address detection in mobile ad hoc networks. *ACM MobiHoc, Lausanne, Switzerland*, June 2002.

- [79] F. Valois. *Modélisation et évaluation de performances de réseaux cellulaires hiérarchiques*. PhD thesis, Laboratoire PRiSM, Université de Versailles, January 2000.
- [80] J. Wu and F. Dai. Distributed dominant pruning in ad hoc wireless networks. In *Proceedings of International Conference on Communications*, Anchorage, AK, USA, may 2003.
- [81] J. Wu and H. Li. On calculating Connected Dominating Set for efficient routing in ad hoc wireless networks. *Proc. of the Third Int' Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, Seattle, USA*, pages 7–14, August 1999.
- [82] J. Wu and W. Lou. Extended multipoint relays to determine connected dominating sets in manets. In *Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, Santa Clara, USA, October 2004. IEEE.
- [83] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang. A two-tier data dissemination model for large-scale wireless sensor networks. *ACM Journal of Mobile Networks and Applications (MONET)*, 2003.
- [84] G. Zhou, T. HE, S. Krishnamurthy, and J. Stankovic. Models and solutions for radio irregularity in wireless sensor networks. *ACM Transactions on Sensor Networks*, 2(2) :221–262, May 2006.
- [85] H. Zhou, L. Ni, and M. Mutka. Prophet address allocation for large scale MANETs. *IEEE INFOCOM, San Francisco, USA*, April 2003.