Une auto-organisation et ses applications dans les réseaux ad hoc et hybrides

Fabrice Theoleyre

N^o d'ordre: 2006-ISAL-0061 Année 2006

Thèse

Une auto-organisation et ses applications pour les réseaux ad hoc et hybrides

présentée devant

l'INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE LYON

pour l'obtention du

GRADE DE DOCTEUR

Ecole Doctorale Informatique et Informations pour la Société

par

Fabrice THEOLEYRE
(Ingénieur de l'INSA de Lyon)

soutenue le 29 septembre 2006

Après avis de :

Andrzej Duda
Professeur à l'ENSIMAG (Grenoble)
Thomas Noël
Professeur à l'université de Strasbourg
Ivan Stojmenovic
Professeur à l'université d'Ottawa (Canada)

Devant la comission d'examen :

Andrzej Duda (rapporteur)
Professeur à l'ENSIMAG (Grenoble)
Serge Fdida (examinateur)
Professeur à l'université Pierre et Marie Curie (Paris 6)
Éric Fleury (co-directeur de thèse)
Professeur à l'INSA de Lyon
Thomas Noël (rapporteur)
Professeur à l'université de Strasbourg
David Simplot-Ryl (examinateur)
Professeur à l'université de Lille
Fabrice Valois (co-directeur de thèse)
Maître de Conférences à l'INSA de Lyon

Cette thèse a été préparée au Centre d'Innovation en Télécommunications et Intégration de Services (CITI), INSA de Lyon - INRIA Rhône-Alpes

A Céline,

Remerciements

Ce travail a été effectué au sein du Centre d'Innovation en Télécommunications et Intégration de Services (CITI), à l'INSA de Lyon, et dans le projet ARES de l'INRIA Rhône-Alpes.

Je remercie d'abord Fabrice Valois qui m'a encadré durant ces trois ans de thèse, m'a insufflé le goût de la recherche et m'a donné les outils pour réussir dans ce domaine. Je le remercie d'avoir réussi à me supporter pendant ce temps qui a du lui paraître, j'imagine, interminable. Je remercie également Éric Fleury qui m'a co-encadré durant cette thèse et a su partager son expertise scientifique lorsque j'en avais besoin.

Je remercie tout particulièrement Monsieur Ivan Stojmenovic, Professeur à l'université d'Ottawa, Monsieur Andrzej Duda, Professeur à l'ENSIMAG, et Monsieur Thomas Noël, Professeur à l'Université de Strasbourg qui ont accepté de juger ces trois années de travail de thèse et d'en être les rapporteurs.

Je remercie également Monsieur Serge Fdida, Professeur à l'université Paris 6 et Monsieur David Simplot-Ryl, Professeur à l'université Lille 1 d'avoir accepté de faire partie de ce jury de thèse et de s'être intéressés à mon travail.

Je remercie également Monsieur Hervé Rivano qui m'a ouvert les portes du domaine de l'optimisation et avec qui j'ai travaillé avec plaisir, tant pour son apport scientifique que personnel et gastronomique.

Je remercie également Madame Catherine Rosenberg avec qui j'ai travaillé durant mon séjour à l'université de Waterloo, et qui a changé radicalement mon approche de la recherche. J'espère pouvoir appliquer avec rigueur ce qu'elle m'a appris durant ce séjour.

Je tiens à remercier Nathalie Mitton, avec qui j'ai partagé beaucoup durant ces 3 ans, Jia-Liang Lu, Thomas Watteyne qui ont su me supporter dans leur bureau. Je remercie également Tahiry Razafindralambo, Yvan Royon pour les discussions scientifiques ou personnelles que nous avons eu. Je remercie tout autant les chercheurs du laboratoire CITI, les enseignants du départements Télécommunications (pêle-mêle Isabelle Augé-Blum, Stéphane Ubéda, Jean-Marie Gorce, Stéphane Coulondre, Stéphane Frénot, Guillaume Chélius, Isabelle Guérin-Lassous, Yu Chen et tant d'autres que je ne nommerai pas par manque de place et de peur de ne pas être exhaustif).

Enfin, je tiens à remercier Céline, sans qui cette thèse n'aurait, je pense, jamais vu le jour, et qui a su m'accompagner dans cette grande expérience scientifique mais surtout personnelle qu'est une thèse.

Table des matières

Chapitre 1

Introduction

Nous pouvons commencer par une première question, fondamentale : qu'est ce qu'un réseau et à quoi sert-il ? Nous pourrions définir un réseau informatique comme une infrastructure permettant le partage d'information de façon électronique et transparente pour l'utilisateur final. Les réseaux informatiques ont connu un tel développement depuis quelques dizaines d'années qu'ils se sont immiscés dans notre vie quotidienne, entraînant une très forte mutation de nos habitudes, donnant naissance à la *société de l'information* [8]. Nous sommes en train de vivre un deuxième bouleversement, créant un réseau informatique ubiquitaire : le réseau nous entoure, permettant un accès à l'information de tout endroit à tout moment.

1.1 L'évolution des réseaux radio

Les réseaux sans-fil actuels connaissent un engouement saisissant. Leur flexibilité d'utilisation a fait que de tels réseaux ont été rapidement adoptés tant par les particuliers que par les entreprises. Ainsi, le Bluetooth [2] permet de déployer un réseau personnel interconnectant tous les périphériques communicant, de l'imprimante à l'appareil photo en passant pas le téléphone portable. Cependant, IEEE 802.11 [1] représente la technologie sans-fil ayant connu le plus grand développement : une connexion Internet est possible dans les universités, les aéroports, les entreprises... Les normes 802.11 a/b/g se sont développés permettant d'augmenter les débits jusqu'à maintenant un débit théorique de 54 Mbps. De même, les réseaux téléphoniques mobiles se sont développés, passant du GSM à l'UMTS en passant par le GPRS. De nouvelles utilisations ont vu le jour : la vidéo à la demande, la consultation d'informations sont devenues maintenant courantes sur de nombreux téléphones portables. L'utilisation de terminaux sans-fil permet de développer une utilisation itinérante ou même mobile des réseaux qu'ils soient voix ou données. Cependant, un véritable Internet ubiquitaire est encore utopique : les infrastructures sont trop coûteuses et possèdent une portée de couverture réduite.

La deuxième étape est donc d'étendre cette notion de réseaux sans-fil pour lesquels les infrastructures sont trop coûteuses pour créer un réseau où chaque terminal est également acteur et moteur du réseau. Le réseau fonctionne donc de façon autonome, transparente pour l'utilisateur final. Les réseaux ad-hoc constituent un réseau ubiquitaire, dont les potentialités sont infinies. Nous nous attacherons ici au concept de l'auto-organisation de ces réseaux pour en améliorer leur efficacité.

1.2 Réseaux ad-hoc et hybrides : une définition

Le premier réseau ne reposant sur aucune liaison filaire (PRNet) a été construit par la DARPA en 1987 [9]. Naturellement, les protocoles radio de l'époque étant limités, la bande passante disponible était faible. Cependant, il constitue l'une des base des réseaux ad-hoc actuels.

Les réseaux ad-hoc [12] représentent l'extension des réseaux radio classiques : ils représentent littéralement des réseaux *prêts à l'emploi*. Les réseaux ad-hoc doivent fonctionner auto-organisés, de façon autonome, sans configuration, sans intervention humaine et sans infrastructure dédiée. Ainsi, tout naturellement, les terminaux utilisent des liaisons sans-fil, ne requérant aucun câblage. Cependant, les réseaux classiques voient leur utilisation entièrement muter : alors qu'auparavant toutes les fonctions *intelligentes* remplies par le réseau étaient concentrées dans des équipements dédiés, les *routeurs*, un réseau ad-hoc doit collaborer et se répartir les tâches afin de fournir un service réseau autonome et performant, de façon totalement distribuée. Ainsi, l'envoi d'un paquet n'est plus pris en charge par des équipements tels que des routeurs dédiés ou des commutateurs : chaque terminal du réseau doit collaborer afin de trouver la localisation de la destination recherchée et lui acheminer les paquets. De plus, le réseau doit être dynamique : l'arrivée d'un terminal ou au contraire son départ ne doit pas perturber les autres terminaux. Si nous poussons un tel concept de dynamicité jusqu'à son paroxysme, les terminaux peuvent également se mouvoir, de façon indépendante les uns des autres. Le réseau doit trouver une solution afin de s'adapater continuellement aux changements de topologie. De tels réseaux ad-hoc sont souvent appelés Mobile Ad Hoc Networks (MANET). Par conséquence, toute la philosophie des réseaux classiques doit être repensée afin de s'adapter à de telles exigences.

Les réseaux ad-hoc peuvent également être connectés au monde filaire par l'intermédiaire de une ou plusieurs passerelles, que nous appellerons, en référence au monde cellulaire IP, des points d'accès (AP). De tels réseaux sont communément appelés *réseaux hybrides*. Chaque terminal du réseau ad-hoc, s'il possède une double interface filaire et sans-fil peut donc agir en tant que passerelle pour les autres clients de la bulle ad-hoc. Les réseaux hybrides constituent les prémices de l'Internet ubiquitaire de demain.

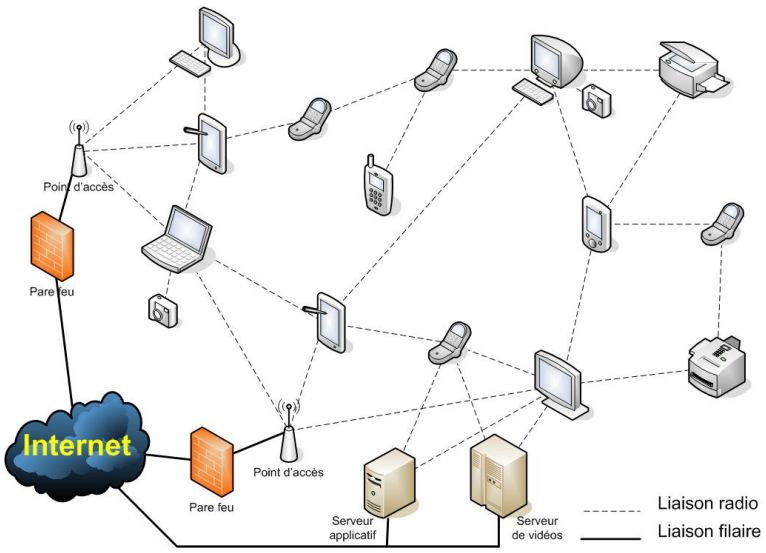


FIG. 1.1 – Un réseau ad-hoc

1.3 Scénarios d'usage

Nous pensons que les réseaux ad-hoc et hybrides sont promis à un large spectre d'utilisation. Les réseaux ad-hoc sont plus particulièrement utiles lors de l'organisation d'évènements tels que des conférences, des salons afin de proposer un réseau de partage de l'information. Ils peuvent également être utilisés lorsqu'une catastrophe naturelle a détruit les infrastructures de télécommunications, et qu'une liaison satellite pour chaque terminal en communication représente un coût trop élevé. Les réseaux ad-hoc trouvent également un champ d'application direct dans les environnements militaires. Ainsi, les communications entre fantassins, véhicules et engins aéro-

portés deviennent autonomes et spontanées. Cependant, de nombreux problèmes restent encore à résoudre avant une application concrète. A titre d'exemple, le ministère de la défense américain lance toujours de nombreux projets afin de promouvoir la recherche dans ce domaine, et notamment de permettre la proposition de protocoles, antennes et équipements aptes à remplir de tels défis.

Les réseaux hybrides constituent pour nous la principale application civile des réseaux ad-hoc. Ils étendent le concept des réseaux sans-fil classiques en proposant la création de réseaux d'accès cellulaires multisauts. La couverture radio est étendue, proposant un Internet ubiquitaire autorisant tant les connexions vers Internet qu'entre pairs clients. Nous pensons que les franges de l'Internet vont progressivement se transformer en intégrant les réseaux personnels sans-fil (Personal Area Network, PAN). De même, le domaine de la domotique constitue une application privilégiée des réseaux ad-hoc [6] : un équipement spécialisé agit en tant que passerelle vers les fournisseurs d'accès et vers Internet, offrant une connexion multisauts à tous les équipements électroménagers ou high-tech de la maison.

1.4 Dynamique dans la communauté scientifique

Les MANET ont concentré depuis déjà plusieurs années les efforts de nombreux scientifiques, tant du domaine de l'algorithmie, des protocoles que de la modélisation. Cependant, tous les défis sont encore loin d'être résolus. Plusieurs groupes scientifiques se focalisent sur la thématique des réseaux ad-hoc. Ainsi, l'Internet Research Task Force [4] (IRTF) possède un groupe de travail, l'Ad hoc Network System (ANS) chargé de développer les aspects de collaboration inter-couches protocolaires, d'auto-configuration, de routage et de qualité de service. De même, l'organisme pendant, l'Internet Engineering Task Force [3] (IETF) possède le groupe MANET, très actif dans la communauté. Ce groupe se focalise actuellement sur le problème du routage, trois protocoles ayant déjà été standardisés (OLSR, AODV et TBRPF). Par ailleurs, la communauté scientifique, très active propose de nombreuses conférences et journaux dédiés aux réseaux ad-hoc.

1.5 Contraintes et Défis

De par leur définition même, les réseaux ad-hoc requièrent une approche très différente de celle empruntée jusqu'à maintenant. Nous notons en particulier les contraintes fortes suivantes, spécifiques aux MANET et hybrides :

- La radio : les terminaux communiquent majoritairement via des liens radio. Ainsi, la diffusion d'information est rendue malaisée à cause de l'instabilité et du manque de fiabilité des liens radio. De même, les interférences radio créent un médium de communication partagé : la capacité en est réduite, requérant de minimiser le trafic de contrôle nécessaire au bon fonctionnement du réseau.
- La dynamicité : chaque terminal peut se déplacer librement et indépendamment des autres. Ainsi, la topologie du réseau est en constante évolution. Un protocole doit donc s'adapter continuellement et rapidement à ces changements afin d'offrir des performances optimales sur la durée. De même, un nœud peut librement s'intégrer au réseau ou au contraire s'en détacher.
- La limitation de ressources : un réseau ad-hoc peut comporter des terminaux embarqués. Or, l'autonomie en énergie et les capacités de traitement d'un PDA par exemple sont limitées. Le concepteur de protocole doit donc garder une telle contrainte à l'esprit.
- L'hétérogénéité : un réseau ad-hoc est constitué des terminaux qui viennent naturellement y participer. Ainsi, un tel réseau est foncièrement hétérogène. Les capacités en terme de mémoire, de calcul, d'énergie présentent une grande variété : un protocole pour les réseaux ad-hoc doit donc prendre en compte une telle hétérogénéité et en tirer parti.

- Localisation des décisions : un réseau ad-hoc est d'une part constitué potentiellement d'un grand nombre de terminaux, et est d'autre part multisauts : une information doit être quelquefois relayée par d'autres clients pour être acheminée de la source à la destination. Ainsi, il est inconcevable de faire participer activement tous les nœuds du réseau à la décision d'un seul nœud : tout processus de décision doit être localisé.
- Auto-configuration : aucune intervention humaine n'étant requise, la prise en charge d'un terminal doit être transparente. Le nœud et le réseau doivent s'auto-configurer.
- Passage à l'échelle : le réseau peut comprendre un grand nombre de terminaux. Cependant, les performances ne doivent pas pour autant chuter de façon drastique lorsque le nombre de participants augmente.

1.6 Domaines à développer

Ces contraintes étant spécifiques aux réseaux ad-hoc, les protocoles réseaux classiques demandent à être repensés. Nous distinguons notamment les domaines de recherche suivant qui nous semblent incontournables :

- Routage : concevoir un protocole performant en terme de délai, de pertes de paquets, de stabilité des routes... De plus, il doit passer à l'échelle, et présenter un trafic de contrôle négligeable afin de ne pas perturber le trafic de données [14].
- Intégration avec Internet : IP représente de facto un standard. Le routage interne à la bulle ad-hoc n'est donc pas auto-suffisant, une intégration symbiotique des réseaux ad-hoc et de l'Internet doit donc abordée. Une architecture complète doit notamment gérer la gestion de la micro et macro-mobilité, l'attribution de préfixes routables, ...
- Auto-configuration : le routage, l'intégration avec Internet doivent être accomplis de façon transparente : ils doivent s'auto-configurer pour s'adapter à l'environnement. De même, des protocoles robustes de configuration d'adresses [15], d'apprentissage de paramètres tels que le préfixe réseau ou l'adresse du serveur DNS, doivent être proposés. Ils doivent présenter une grande robustesse aux fautes et minimiser les incohérences
- Qualité de service : des protocoles tant au niveau MAC que réseau doivent pouvoir établir des priorités entre les flux, limiter les pertes de paquets vitaux pour la gestion du réseau, ou du moins en restreindre l'impact[5].
- Couche MAC : IEEE 802.11 présente des dysfonctionnements importants dans un réseau multi-sauts [7]. Par ailleurs, le débit et la portée offerts par Bluetooth ne permettent pas un réseau ad-hoc étendu multimédia et multi-utilisateurs. Une nouvelle couche MAC doit donc être conçue afin de proposer un débit acceptable dans les réseaux ad-hoc.
- Cross-layer : les modèles classiques de réseaux pré-supposent une indépendance des couches protocolaires afin d'optimiser l'indépendance des couches, et donc leur flexibilité : routage et fonctions MAC sont par exemple indépendants. Cependant, une telle indépendance est communément considérée comme trop coûteuse pour les réseaux sans-fil. Conséquemment, une coopération achevant un compromis entre performances et flexibilité doit être proposée [13].
- Diffusion de l'information : de nombreux protocoles proposés pour remplir les rôles précédents utilisent massivement la diffusion d'information dans le réseau entier. Conséquemment, un protocole efficace de diffusion limitant les collisions et la consommation de ressources radio doit être proposé. Dans une inondation aveugle, un terminal qui reçoit un paquet à diffuser le relaie s'il ne l'a pas déjà reçu. Ainsi, l'intégralité des nœuds devraient le recevoir. Cependant, [11] montre qu'un tel mécanisme présente des problèmes de fiabilité (à cause des collisions) et de redondance (beaucoup de transmissions sont inutiles). Ce problème est connu sous le nom de *tempête de broadcast*.
- Sécurité : des mécanismes d'authentification, de confidentialité, d'intégrité doivent être mis en place au sein d'une communauté d'utilisateurs [16]. Cependant, ce schéma de sécurité

doit être assez flexible pour rendre les communautés semi-perméables, en fonction du niveau de confiance requis pour les échanges. La sécurité représente un domaine transverse à tous les points cités auparavant.

- Terminaux : au niveau des contraintes matérielles, des terminaux embarqués à forte autonomie en énergie et ergonomiques doivent apparaître pour autoriser le développement des réseaux ad-hoc.

1.7 Motivations et Objectifs

Beaucoup de propositions répondant aux contraintes et objectifs précédents s'appuient sur une vue à plat du réseau ad hoc ou hybride : tous les terminaux sont considérés comme égaux et doivent contribuer de façon solidaire à la gestion du réseau pour accomplir toutes les tâches décrites auparavant. L'approche de l'IETF est justement de considérer indistinctement l'ensemble de ces terminaux et de concevoir un protocole de routage, d'attribution d'adresses fonctionnant directement dans un tel réseau. Ainsi, tout terminal doit être interchangeable.

Nous pensons au contraire que le réseau doit être hiérarchisé, *auto-organisé*, avant de concevoir toute solution exploitant un réseau ad hoc. Ainsi, l'auto-organisation ne constitue pas une fin en soi mais représente au contraire une solution devant servir à terme de socle commun à d'autres protocoles remplissant les objectifs décrits dans la section précédente. Une auto-organisation pourrait par exemple créer une hiérarchie reflétant le réseau : les coordinateurs seront par exemple choisis parmi les nœuds les plus faiblement mobiles, avec une autonomie en énergie élevée, de grandes capacités de stockage et de mémoire. Une auto-organisation pourrait donc refléter l'hétérogénéité naturelle des réseaux ad hoc et hybrides et constituer une vue logique plus simple à exploiter au niveau 3 du modèle OSI.

Le concept d'auto-* connaît actuellement un fort engouement, tant dans les réseaux sans-fil que les réseaux filaires : tout doit fonctionner de façon autonome, s'auto-configurant, gérant de façon transparente les erreurs, tout en gérant un grand nombre de nœuds. L'auto-organisation est née de ce concept : elle constitue une couche intermédiaire d'auto-organisation au dessous des protocoles réseau, permettant de leur offrir une vue logique du réseau, hiérarchique, plus facilement exploitable, disjointe de la topologie radio réelle.

Nous pensons qu'une auto-organisation permet de simplifier le développement de protocoles pour réseaux hybrides et ad hoc, et surtout d'optimiser leurs performances. Par exemple, la propriété de passage à l'échelle n'est pour nous atteignable qu'après l'introduction préalable d'une hiérarchie dans le réseau. Par ailleurs, il est souhaitable que cette hiérarchie soit commune à tous les protocoles pour réseaux ad-hoc, amenant à une solution complète d'auto-organisation. Nous pourrions voir une structure d'auto-organisation comme une couche servant de glu aux protocoles des couches supérieures, partageant avec eux un certain nombre d'informations communes, cachant une partie des changements de topologie, fournissant une vue hiérarchique tenant compte des capacités physiques des terminaux...

Le concept d'auto-organisation présente donc un vaste domaine : toute solution de hiérarchie flexible est candidate. Naturellement, nous pensons à la dorsale des réseaux filaires, n'existant pas dans les réseaux radio multisauts. Cependant, des algorithmes pourraient être proposés pour construire une vue logique permettant de ré-introduire le concept de dorsale dans les réseaux ad-hoc. De même, le découpage en sous-réseaux des réseaux IP pourrait être greffé aux réseaux radio. Nous nous proposons donc d'étudier et concevoir une solution d'auto-organisation introduisant de telles fonctionnalités.

Par contre, nous attirons l'attention du lecteur sur le fait qu'une auto-organisation ne constitue aucunement une fin en soi. C'est la raison pour laquelle nous ne mentionnons pas cette problématique dans la liste des domaines à développer. Pour nous, elle ne sert que d'intermédiaire pour les réels défis scientifiques. En conséquence, des protocoles de localisation, de routage tirant parti de cette auto-organisation doivent être proposés. Nous verrons comment parvenir à un tel

objectif.

1.8 Canevas de cette étude

Dans le premier chapitre, nous définirons précisément ce qu'est pour nous une structure d'auto-organisation. Nous en donnerons les principes fondamentaux, les propriétés que doivent suivre de telles structures. L'auto-organisation servant de fil conducteur à cette thèse, nous présenterons également un panorama des différentes solutions proposées auparavant dans les réseaux ad-hoc, et nous expliquerons quelles en sont les carences.

Le chapitre 3 présentera la structure d'auto-organisation que nous proposons. Elle est constituée d'un dorsale, servant de collecteur de trafic et de premier élément de hiérarchie dans le réseau. Sur cette dorsale viennent se greffer des grappes, servant de zones de services regroupant les nœuds proches géographiques. Une telle structure permet d'introduire un deuxième niveau de hiérarchie. Nous présentons des algorithmes de construction mais aussi de maintenance, chaque utilisateur étant libre de se mouvoir. Par ailleurs, une étude de ces performances générales au travers de simulations viendra compléter cette présentation.

Le chapitre 4 détaillera les propriétés présentées par cette structure. La robustesse et la stabilité sont des propriétés fondamentales pour une structure d'auto-organisation. Or, nous verrons que la structure proposée est auto-stabilisante. Des résultats de simulations apporteront une corroboration quantitative de ces propriétés. Nous présenterons également un poids de stabilité et une solution d'économie d'énergie tirant parti de la hiérarchie introduite dans le réseau via l'auto-organisation.

Cependant, une telle structure d'auto-organisation n'est aucunement une fin en soi. Ainsi, le chapitre 5 présentera un protocole de routage, Virtual Structure Routing (VSR) tirant parti de l'auto-organisation créée dans le réseau. La dorsale créée permet de collecter le trafic de contrôle, permettant de limiter l'impact d'une inondation dans le réseau. Les grappes permettent elles d'introduire une route hiérarchique dans le réseau, optimisant la stabilité. De plus, la hiérarchie permet de déployer différents protocoles de routage adaptés à la zone dans laquelle ils se situent, tout en leur permettant de collaborer afin d'autoriser la transmission de paquets jusqu'aux confins du réseau. Des simulations permettront de vérifier les performances de VSR en terme de délai, de pertes de paquets, de passage à l'échelle, de stabilité des routes, surpassant les performances des protocoles de routage classiques.

Par ailleurs, si nous nous focalisons sur un réseau ad-hoc connecté à Internet, i.e. un réseau hybride, il serait intéressant de proposer un protocole de localisation permettant d'interconnecter la bulle ad-hoc à Internet. Le chapitre 6 présente un protocole, Self-Organized Mobility Management (SOMOM) remplissant un tel objectif et s'appuyant sur la dorsale proposée dans le chapitre 3. La stabilité de la structure virtuelle permet d'optimiser la longévité des routes. Par ailleurs, la dorsale permet, comme nous le verrons, de réduire la trafic de contrôle.

La notion de capacité, i.e. le débit maximum atteignable par le réseau, est un élément clé de l'évaluation d'un protocole de routage. En effet, un réseau radio présente une faible bande passante, constituant une contrainte forte que le concepteur doit garder à l'esprit. Il vient donc à l'esprit une question naturelle : l'introduction d'une organisation, d'une hiérarchie nuit-elle à la capacité d'un réseau ? Le chapitre 7 introduit une solution générique d'évaluation de la capacité d'un réseau suivant un protocole de routage et une topologie donnés. Nous décrivons notamment des modèles de partage des ressources radio, et différents modèles d'équité dans un tel partage. Nous verrons qu'un schéma d'auto-organisation présente une capacité légèrement inférieure dans certains cas, mais cette différence nous parait négligeable vis à vis de ses atouts.

L'évaluation de performances de protocoles par le biais des simulations est courante dans les réseaux ad-hoc. Cependant, il nous semble que les expérimentations réelles forment un complément indispensable. Il est nécessaire de valider un protocole dans un environnement réaliste, sans les problèmes de simplification inhérents à toute modélisation. Les interférences radio sont

notamment largement sous-estimées dans les modèles de simulation. Le chapitre 8 présente le testbed que nous avons déployé afin de tester les performances de la structure d'auto-organisation et du protocole de localisation que nous avons proposés. Nous verrons que le testbed fournit des performances encourageantes, malgré les faiblesses bien connu de IEEE 802.11 que nous utilisons comme couche MAC.

Enfin, le chapitre 9 conclura cette thèse. Nous exposerons également quelques perspectives de ce travail. Cette thèse a été réalisée au sein du laboratoire CITI à l'INSA de Lyon, dans le projet ARES de l'INRIA Rhône-Alpes, sous la direction de Fabrice Valois et Éric Fleury.

Bibliographie

[1] IEEE 802.11, local and metropolitan area networks - specific requirements part 11 : Wireless lan medium access control (mac) and physical layer (phy) specifications, 1999.

[2] IEEE 802.15.1, personal area networks - specific requirements, 2002.

[3] IETF mobile ad-hoc networks (manet), http://www.ietf.org/html.charters/manet-charter.html.

[4] IRTF RRG ad hoc network systems research subgroup, http://www.flarion.com/ans-research/.

[5] T. Bheemarjuna Reddy, I. Karthigeyan, B. Manoj, and C. Siva Ram Murthy. Quality of service provisioning in ad hoc wireless networks : a survey of issues and solutions. *Ad Hoc Networks*, 4(1) :83–124, January 2006.

[6] R. Bruno, M. Conti, and E. Gregori. Mesh networks : Commodity multihop ad hoc networks. *IEEE Communications Magazine*, 43(3) :123–131, March 2005.

[7] C. Chaudet, D. Dhoutaut, and I. Guérin Lassous. Performance issues with IEEE 802.11 in ad hoc networking. *IEEE Communications Magazine*, 43(7) :110–116, July 2005.

[8] S. Ghernaouti-Hélie and A. Dufour. *De l'ordinateur à la société d'information*. Presses Universitaires de France - PUF, 2001.

[9] J. Jubin and J. D. Tornow. The darpa packet radio network protocols. In *Proceedings of the IEEE*, volume 75, January 1987.

[10] J. Latvakoski, D. Pakkala, and P. Pääkkönen. A communication architecture for spontaneous systems. *IEEE Wireless Communications*, 11(3) :36–42, June 204.

[11] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. In *International Conference on Mobile Computing and Networking (MOBICOM)*, pages 151–162, Seattle, USA, August 1999. ACM.

[12] C. E. Perkins. *Ad hoc networking*. Addison-Wesley, 2001.

[13] V. Raisinghani and S. Iyer. Cross-layer feedback architecture for mobile device protocol stacks. *IEEE Communications Magazine*, 44(1) :85–92, January 2006.

[14] R. Rajaraman. Topology control and routing in ad hoc networks : a survey. *ACM SIGACT News*, 33(2) :60–73, June 2002.

[15] K. Weniger. Pacman : Passive autoconfiguration for mobile ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 23(3), March 2005.

[16] H. Yang, H. Luo, F. Ye, S. Lu, and L. Zhang. Security in mobile ad hoc networks : Challenges and solutions. *IEEE Wireless Communications*, 11(1) :38–47, February 2004.

Chapitre 2

L'auto-organisation dans les réseaux ad-hoc

2.1 Introduction

Les réseaux classiques sont structurellement organisés pour remplir leur rôle. Par exemple, un réseau GSM [39] est constitué d'une série d'équipements hiérarchiquement organisés formant une arborescence : un mobile est desservi par une des stations de base (BTS), formant le premier niveau hiérarchique. Les zones appelées BSC forment le deuxième niveau et les zones MSC forment le troisième et dernier niveau. De la même façon, les réseaux de données IP introduisent une hiérarchie afin de faciliter les communications, et l'agrégation d'informations, comme dans BGP par exemple [43]. A contrario, un réseau ad-hoc est, par sa nature, sans organisation et hiérarchie préalables : un ensemble de terminaux mobiles doivent spontanément établir des règles afin d'acheminer l'information. Cependant, ces règles ne sont aucunement données par une hiérarchie naturelle, une collaboration systématique étant obligatoire. L'IETF défend une vision très proche de l'architecture physique des réseaux ad hoc : parce qu'un réseau est constitué de terminaux non organisés, un protocole doit directement s'exécuter sur cet environnement. En conséquence, il reste à proposer une famille de protocoles offrant des fonctions de routage, d'attribution d'adresses ou de sécurité. Le problème des réseaux ad hoc n'est jamais considéré comme un tout mais plutôt comme un ensemble de questions auxquelles répondre individuellement. Nous pensons qu'au contraire, le réseau doit être au préalable auto-organisé, et cette auto-organisation doit servir de liant pour tous les protocoles s'exécutant pour les réseaux ad hoc. La réponse à de nombreuses problématiques peut être commune.

2.1.1 Les solutions apportées par une structure virtuelle d'auto-organisation

Nous pensons que l'utilisation efficace d'un réseau ad-hoc est conditionnée par son auto-organisation : un réseau doit être structuré avant d'être utilisé. Un MANET est classiquement considéré *à plat* : chaque terminal intervient de façon égalitaire dans le réseau, la charge devant donc être répartie de façon uniforme dans le réseau pour ne désavantager aucun nœud. Ainsi, le véritable défi inhérent aux MANET résidait dans la proposition de protocoles distribués permettant de réaliser les fonctions réseaux classiques (telles que le routage, la découverte de services, la réservation de bande passante) directement sur un grand nombre de nœuds non hiérarchisés. Proposer des protocoles fonctionnant dans des réseaux destructurés constituait une des questions clé. Nous pensons au contraire qu'un réseau ad-hoc doit fondamentalement être organisé afin de rendre les protocoles plus efficaces. Pour nous, organiser un réseau représente l'introduction d'un comportement global visant à la création d'une vue logique structurée, organisée, basée sur un ensemble de terminaux qui, matériellement, étaient au départ désorganisés. Une hiérarchie doit donc être introduite dans le dessein de :

- Tirer parti de l'hétérogénéité : les terminaux présentent des capacités disparates. Une organisation reflétant cette hétérogénéité doit donc être construite, permettant de faire participer plus activement les nœuds *forts*. Nous choisissons donc de partager inéquitablement les rôles, pour tenir compte des capacités différentes de certains groupes de nœuds.
- Économiser l'énergie : des nœuds *forts* sont élus de façon localisée, les autres nœuds participant moins peuvent réduire leur participation au réseau et économiser leur énergie.
- Optimiser la diffusion d'information : seuls certains nœuds sont habilités à relayer une information afin qu'elle soit diffusée dans tout le réseau. Le nombre de transmissions est donc réduit, et la fiabilité accrue.
- Cacher certains des changements de topologie : l'auto-organisation offre une vue virtuelle plus stable de la topologie. L'impact de la mobilité sur la topologie est donc optimisé par ce biais.
- Offrir un réseau plus facilement exploitable : une vue hiérarchique du réseau est créée afin de rendre le déploiement de protocoles plus rapide et aisé. En conséquence, un protocole de routage présentant de meilleures performances et plus flexible peut être proposé (cf. chapitre 5).

2.1.2 Plan du chapitre

Nous allons dans un premier temps dans la section 2.2 définir ce que selon nous l'auto-organisation signifie. Ensuite, la partie 2.3 exposera les propriétés fondamentales que doit présenter une structure d'auto-organisation pour être utilisable dans les réseaux ad-hoc. La section 2.4 détaillera la modélisation que nous utilisons pour représenter les réseaux ad-hoc et exposera les notations que nous utiliserons dans le reste de ce document. Nous présenterons dans la section 2.5 un panorama des différentes structures d'auto-organisation existant dans les réseaux ad-hoc.

2.2 Structures virtuelles d'auto-organisation : définition et motivations

L'auto-organisation est un terme couramment utilisé en physique pour définir une réorganisation de la matière dans un état plus stable, i.e. réduisant l'entropie globale du système. Wikipédia définit ainsi l'auto-organisation [56] :

> L'auto-organisation est un phénomène de mise en ordre croissant, et allant en sens inverse de l'augmentation de l'entropie ; au prix bien entendu d'une dissipation d'énergie qui servira à maintenir cette structure.

L'encyclopédie Universalis [19] présente une définition similaire de l'auto-organisation appliquée à la physique. Ainsi, l'auto-organisation dans le cas qui nous intéresse serait la construction d'une vue plus stable de la topologie, réduisant les changements de topologies visibles par les nœuds. Si nous poursuivons l'analogie, la dissipation d'énergie pourrait être assimilée au trafic de contrôle nécessaire à l'établissement et la maintenance de cette auto-organisation.

L'encyclopédie Universalis [19] définit ainsi l'auto-organisation au sens large :

> Ce qui caractérise une auto-organisation au sens fort est l'absence de but défini à l'avance et l'**émergence** de ce qui apparaît, après coup, comme un **comportement fonctionnel**, c'est-à-dire ayant un sens.

Une telle définition s'applique parfaitement à notre vision de l'auto-organisation : c'est la structuration d'un réseau afin de rendre son exploitation plus aisée. Ainsi, des règles locales d'organisation sont appliquées formant globalement une auto-organisation émergente. Cette auto-organisation transparaît à travers une structure virtuelle permettant de déployer plus facilement et surtout efficacement des fonctions réseau telles que le routage ou la découverte de services.

Pour nous, une structure virtuelle d'auto-organisation doit absolument être flexible. Plus explicitement, une même structure d'auto-organisation doit servir d'organisation tant pour le routage que pour la diffusion d'information ou la réservation de bande-passante : son coût de construction et de maintenance est mutualisé par tous les services nécessaires au bon fonctionnement d'un réseau.
Au final, nous proposons la définition suivante de l'auto-organisation :

> **Definition** : L'auto-organisation d'un réseau ad hoc le structure : elle crée une vue différente de la topologie radio, introduisant un ou plusieurs niveaux de hiérarchie et facilitant par ce biais l'établissement des services nécessaires au fonctionnement attendu du réseau.

Par extension, certains utilisent le terme d'auto-organisation dans le sens *autonome*. Ainsi, un protocole de routage dans les réseaux ad-hoc peut être dit auto-organisé car les nœuds collaboreront de façon autonome pour établir une fonction de routage, sans intervention extérieure, sans paramétrage manuel. Ainsi, tout protocole conçu pour les réseaux ad-hoc est auto-organisé par nature puisqu'il doit fonctionner librement, et s'adapter aux insertions ou suppressions de nœuds. Nous nous attacherons ici à la précédente notion d'auto-organisation, celle de structure émergente, plus restrictive.

2.3 Les propriétés fondamentales d'une structure virtuelle

Les réseaux ad-hoc présentent des contraintes fortes, explicitées en introduction (partie 1.5). Pour réagir de façon appropriée, il est donc nécessaire que tout algorithme présente les propriétés que nous décrivons dans cette partie. La première série de propriétés (localisation, collaboration, auto-stabilisation, passage à l'échelle et dynamicité) est commune à tout algorithme conçu pour les réseaux ad-hoc. Les trois dernières propriétés (persistance, adaptation et flexibilité d'utilisation) sont cependant spécifiques aux algorithmes d'auto-organisation. Naturellement, les propriétés décrites dans la section 2.1 relatives à l'objectif même d'une structure virtuelle doivent également être remplies. [41] présente des paradigmes proches de ceux exposés ici.

2.3.1 Localisation des décisions

Commençons par définir la distinction entre algorithmes centralisés, distribués et localisés. Un algorithme est dit centralisé s'il possède a priori une connaissance complète. Par exemple, un protocole de routage centralisé possède une vue complète de la topologie et des informations sur toutes les arêtes et nœuds. Il peut donc prendre une décision optimale. Les algorithmes distribués permettent de prendre une décision en distribuant la charge, en échangeant des messages pour partager l'information requise. De tels algorithmes sont par définition beaucoup mieux adaptés aux réseaux ad hoc, possédant un grand nombre de nœuds devant chacun prendre des décisions cohérentes. Enfin, un algorithme localisé est un algorithme distribué n'échangeant des informations qu'avec des nœuds à une distance bornée de lui, et éventuellement avec un nombre borné de nœuds quelconques du réseau. En conséquence, un algorithme localisé conduit rarement à un objectif optimal. Cependant, l'information sur le réseau étant partielle, moins de trafic de contrôle est requis, remplissant la contrainte en terme de bande passante, limitée par l'utilisation de liens radio. De même, les informations nécessaires à une décision étant locales, la réactivité du protocole est accélérée. Ainsi, un algorithme conçu pour les réseaux ad hoc devrait être de préférence localisé.

2.3.2 Coopération

Un ensemble de décisions localisées doit amener à un comportement global émergent. Ainsi, si nous reprenons une définition empruntée à la cybernétique : *le tout est plus que la somme des*

parties [24]. Chaque nœud va appliquer une règle sur l'ensemble des informations desquelles il dispose afin d'en tirer une décision. A l'échelle du réseau, nous pouvons dire que l'algorithme prendra une décision globale permettant de faire émerger une structure d'auto-organisation.

2.3.3 Auto-stabilisation

Un algorithme est dit auto-stabilisant si, partant d'un état quelconque, il converge vers un état légal en un temps fini. Une telle propriété d'auto-stabilisation est requise dans les réseaux ad-hoc afin que l'algorithme converge vers un espace d'états légaux même en cas de faute permanente ou byzantine [1]. Ainsi, si un algorithme auto-stabilisant est exécuté sur un système lui aussi auto-stabilisant, le concepteur peut être certain que l'ensemble des nœuds converge vers un état stable en un temps borné. Le but est naturellement de minimiser ce temps de convergence.

2.3.4 Passage à l'échelle

Tout algorithme pour être efficace ne doit pas voir ses performances baisser de façon drastique si le nombre de nœuds constituant le réseau augmente. Soit λ_i un paramètre (la mobilité, le nombre de nœuds...), un protocole P, et $\mathcal{P}(\lambda_1)$ les performances associées à la valeur λ_1 du paramètre λ_i. Alors, le facteur de passage à l'échelle de P par rapport à λ_i est [47] :

$$\Psi_{\lambda_i} \leq lim_{\lambda_i \to \infty} \frac{log\ \mathcal{P}(\lambda_i, \lambda_2, \dots)}{log\ \lambda_i} \tag{2.1}$$

On dit d'un protocole qu'il passe à l'échelle si $\Psi_{\lambda_i} = 0$, quelles que soient les performances mesurées. Le but étant idéalement de concevoir un protocole passable à l'échelle vis à vis de la cardinalité du réseau en fonction de tout paramètre. De façon pratique, si nous observons sur un graphe les performances de l'algorithme en fonction du paramètre λ_i, la courbe ne devrait pas s'approcher asymptotiquement de l'axe des abcisses. [47] a montré qu'aucun protocole de routage actuel ne passe totalement à l'échelle.

2.3.5 Robustesse à la dynamique du réseau

Puisque chaque mobile est libre de se déplacer indépendamment des autres, la topologie est continuellement changeante. En conséquence, un algorithme doit s'adapter aux changements de topologies. Il peut fonctionner en mode dégradé pendant le laps de temps nécessaire à la mise à jour de ses données : plus cet intervalle de temps est faible, plus le protocole réagit rapidement aux changements, et meilleure est sa robustesse. Puisqu'un terminal ne possède, comme décrit précédemment, que des informations localisées, des incohérences dans la vue du réseau peuvent survenir. C'est pourquoi un protocole de routage par exemple peut voir une de ses routes casser du fait de l'absence d'une mise à jour immédiate. Le but du concepteur est donc de réduire ces incohérences et/ou son impact. Il est important de noter qu'un algorithme peut optimiser sa robustesse en proposant des règles permettant de bien réagir aux incohérences : même si la vue est incohérente, la décision prise n'engendrera aucun conflit.

2.3.6 Persistance

La structure d'auto-organisation doit être stable dans le temps. La structure virtuelle, la hiérarchie sera utilisée ultérieurement pour l'adressage, le routage... Il est donc fort probable qu'un nœud nouvellement élu soit contraint d'échanger des informations avec les autres nœuds du réseau pour mettre à jour des informations de routage par exemple. De même, il se peut qu'un changement dans la structure virtuelle crée des changements de routes. Les nœuds constituant la structure virtuelle doivent dans la mesure du possible rester inchangés.

[1] Lors d'une faute byzantine, un nœud malicieux ou non prend une décision contraire à celle qu'il aurait du prendre.

2.3.7 Flexibilité d'adaptation

Un réseau ad-hoc se meut dans un environnement variable. Ainsi, la densité, la mobilité ou le nombre de nœuds peut évoluer avec le type d'application ou même avec le temps. Les algorithmes proposés doivent donc pouvoir être aisément paramétrables afin de les adapter aux conditions environnementales. Idéalement, une telle adaptation devrait être automatique et dynamique. [41] identifie 3 niveaux d'adaptation :

1. 1^{er} niveau : l'algorithme doit s'adapter aux changements de topologie (comme décrit dans un des paragraphes précédents).
2. 2^{ieme} niveau : l'algorithme doit adapter la valeur de ses propres paramètres. Par exemple, lorsque le degré du réseau augmente, un protocole peut par exemple choisir de minimiser les informations échangées avec les voisins directs.
3. 3^{ieme} niveau : l'algorithme doit être remplacé par un autre algorithme plus efficace dans ces conditions. Par exemple lorsque les changements de topologie sont trop importants, un mécanisme d'inondation aveugle[1] peut être mis en place.

2.3.8 Flexibilité d'utilisation

Comme nous le décrivions dans le paragraphe précédent, cette structure d'auto-organisation nous sera utile pour déployer tout service réseau. En conséquence, elle doit être assez générique pour remplir tous les objectifs de routage, d'allocation de bande passante... Nous verrons dans les chapitres suivants que les algorithmes proposés dans ce chapitre peuvent être utilisés pour l'économie d'énergie, pour le routage dans un réseau ad-hoc, pour la localisation d'un nœud dans un réseau cellulaire multisauts.

2.4 Préliminaires

Nous allons dans un premier temps introduire la modélisation que nous utilisons pour la représentation formelle d'un réseau. Ensuite, nous détaillerons notre notation et nous présenterons une structure bien connue dans les graphes.

2.4.1 Modélisation

Un réseau ad-hoc se représente souvent sous la forme d'un graphe : à chaque terminal du réseau correspond un sommet du graphe, et il existe une arête dirigée d'un sommet A vers un sommet B si le terminal correspondant à A peut envoyer des paquets sur son interface radio de telle sorte que le terminal correspondant au sommet B les réceptionne. Si un lien radio est symétrique, i.e. les deux extrémités peuvent envoyer un paquet sur un lien radio de telle sorte que l'autre le reçoive, le réseau ad-hoc peut être modélisé à l'aide d'un graphe non dirigé (exemple sur la figure 2.1).

Si chaque terminal possède une même puissance d'émission et que le système radio est *idéal*[2], la portée radio est circulaire : tous les nœuds à une distance inférieure ou égale à R reçoivent correctement le signal radio. Un tel réseau peut donc être représenté par un Unit Disk Graph (UDG) : 2 sommets possèdent une arête commune si et seulement si les deux cercles de rayon unité centrés sur les deux sommets possèdent une intersection non vide. Les UDG présentent des propriétés bien connues dans les graphes [17] permettant de borner la cardinalité de certaines structures. Cependant, nous devons garder à l'esprit que cette modélisation est simplificatrice :

[1] Une inondation permet d'envoyer une information à tous les nœuds du réseau. Une inondation aveugle est accomplie en obligeant chaque nœud recevant un paquet, à le relayer.

[2] Nous appelons un *système radio idéal* un système possédant des antennes omnidirectionnelles parfaites, en environnement libre, sans obstacle.

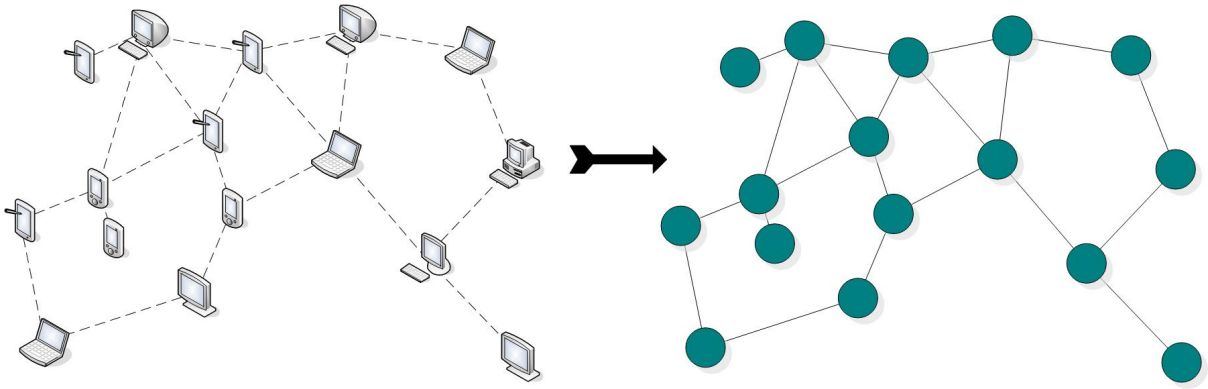


FIG. 2.1 – Modélisation d'un réseau ad-hoc en graphe

un environnement radio ne présente jamais dans la réalité de telles propriétés. La portée radio ne peut être modélisée que par une forme convexe potentiellement complexe. De même, il peut arriver que des liens radio unidirectionnels apparaissent si par exemple 2 terminaux n'ont pas exactement la même antenne ou la même orientation. Les UDG sont donc utiles pour donner une idée de la borne sur la cardinalité de certaines structures. Mais ils ne doivent pas être considérées comme un outil reproduisant parfaitement un réseau ad-hoc quelconque dans un cas non idéal.

2.4.2 Notations

Nous introduisons ici les notations que nous utiliserons tout au long de ce document :

- G(V,E) : le graphe modélisant le réseau ad-hoc, comportant l'ensemble de sommets V et l'ensemble d'arêtes E.
- $|X|$: traduit la cardinalité de l'ensemble X
- n : la cardinalité du réseau($= |V|$)
- $N_k(u)$: le k-voisinage de u, i.e. l'ensemble des nœuds à au plus k sauts de u. Nous pouvons par ailleurs remarquer que $\forall k, u \in N_k(u)$. Par simplification d'écriture, $N_1(u) = N(u)$
- $\Delta_k(u)$: le nombre de k-voisins de u ($\Delta_k(u) = |N_k(u)|$). Par convention, $\Delta_1(u) = \Delta(u)$
- $w(u)$: le poids du nœud u
- $dist(u, v)$: la distance en saut de u à v
- $chemin_{s \to d} =< s, u_1, u_2, \ldots, u_{k-1}, \ldots, d >$: un chemin de s à d. u_i est le $i^{ième}$ nœud intermédiaire dans ce chemin. k est appelée la longueur du chemin.
- P_A : puissance en réception d'un signal radio A

2.4.3 Ensembles indépendants

Nous introduisons ici une structure connue dans les graphes, qui nous sera utile par la suite. Un Ensemble Indépendant (Independent Set, IS) est un ensemble de sommets du graphe tel qu'aucune paire de sommets de l'IS ne se trouvent voisins dans le graphe. Soit IS un ensemble indépendant. Il est défini formellement comme suit :

$$\forall u \in IS, \neg(\exists v \in IS / v \in N(u)) \tag{2.2}$$

Un ensemble indépendant est complet pour l'inclusion si un sommet du graphe ne peut être ajouté à l'ensemble indépendant sans qu'il perde sa propriété d'indépendance. Un ensemble indépendant maximum (Maximum Independent Set, MIS) est un ensemble indépendant présentant une cardinalité maximale. Par ailleurs, un MIS ne possède pas dans le cas général la propriété d'unicité.

2.5 Un panorama des structures d'auto-organisation dans les réseaux ad-hoc

Nous allons maintenant présenter un panorama des différentes propositions de structures de clustering ou de dorsales classées, selon notre vision, dans la catégorie de l'auto-organisation. Nous avons défini deux grandes catégories de structures virtuelles : les dorsales permettent de collecter le trafic de contrôle et forment une dorsale centralisatrice dans le réseau. Les clusters, eux, introduisent une hiérarchie dans le réseau, découpant le réseau en zones homogènes. Puisque les algorithmes doivent pouvoir fonctionner dans les réseaux ad-hoc, les algorithmes de construction mais également de maintenance seront présentés.

2.5.1 Dorsales

Les dorsales représentent un élément clé des réseaux filaires, servant à collecter le trafic et faire transiter les données. Ainsi, dans un réseau local, les clients représentent les feuilles de la dorsale, les équipements réseau étant organisé en une dorsale redondante ou non. Cette dorsale peut elle même être organisée hiérarchiquement. Une telle structure a été très largement éprouvée et a prouvé ses avantages. Cependant, une telle structure, par définition, n'existe pas dans les réseaux ad-hoc. Des auteurs ont donc proposé l'adaptation avec la construction dynamique de dorsales.

2.5.1.1 Qu'est ce qu'une bonne dorsale ?

Une dorsale pour les réseaux ad-hoc doit pouvoir servir de diffuseur d'informations. Lorsqu'une information doit être envoyée dans le réseau, seuls les nœuds de la dorsale sont autorisés à relayer le paquet, minimisant la charge réseau. Tout nœud doit donc être proche de la dorsale pour lui envoyer les informations à diffuser. Par contre, le nombre des nœuds constituant la dorsale doit être minimisé afin de réduire le nombre de nœuds relayant les paquets, en minimisant la charge réseau.

Nous attirons l'attention du lecteur sur le fait qu'une dorsale n'est qu'une vision logique de la topologie radio : bien que certains nœuds se retrouvent à portée radio l'un de l'autre, ils peuvent ne pas être voisins dans la vue logique de la dorsale. Dans le cas qui nous intéresse, la vue logique doit constituer un sous-ensemble de la vue radio réelle.

La dorsale doit être stable dans le temps, i.e. les nœuds la constituant centralisant les informations doivent rester inchangés pendant un laps de temps suffisant. Il peut être opportun de créer une structure redondante dans la dorsale afin d'en optimiser la robustesse.

Enfin, la dorsale doit nécessairement respecter les propriétés décrites dans le paragraphe 2.3 page 11.

2.5.1.2 Arbre Recouvrant

L'arbre recouvrant (spanning tree) est une structure bien connue dans les réseaux [20, 22]. Tout nœud doit faire partie du spanning tree (ST), le ST doit former une structure connexe, et il ne doit exister qu'une seule route entre tout point A et tout point B. Si chaque lien porte un coût, il est possible de construire un minimum spanning tree (MST), minimisant le coût global des arêtes utilisées (exemple figure 2.2). Une telle structure peut être utile pour organiser des commutateurs réseaux reliés entre eux de façon redondante [48] : un MST est construit et maintenu de façon distribuée. Lorsqu'une information à relayer est reçue par un commutateur, il ne peut la relayer que via un lien faisant parti du MST. Le nombre d'envois est optimisé et une information ne peut boucler.

Cependant, un MST devient inutile dans un réseau ad-hoc. Puisqu'il n'existe plus la distinction entre un client et un équipement spécialisé, la diffusion d'information n'est aucunement

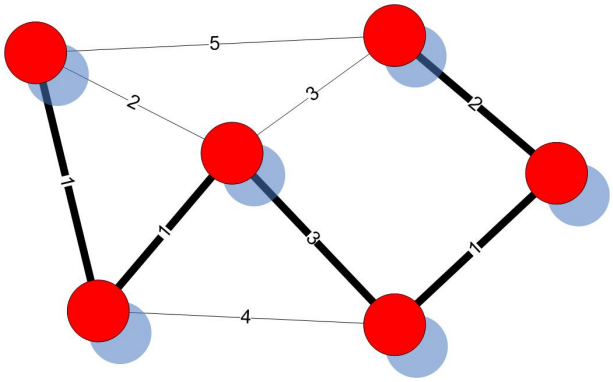


FIG. 2.2 – Exemple d'arbre recouvrant de poids minimum (MST)

optimisée. De même, lorsqu'un nœud envoie un paquet, tous ses voisins le reçoivent : on dit que le médium de communication est diffusant. Les liens ne peuvent donc pas être considérés indépendamment. Or, les MST ne prennent pas en compte une telle propriété. Un autre type de dorsale doit donc être proposé pour les réseaux ad-hoc.

La construction d'un MST demandant une information globale, [32, 42] ont proposé des versions localisées, utilisées pour l'adaptation de portée, dans le contrôle de topologie. Nous verrons dans la section 2.5.3 en quoi de telles propositions sont complémentaires d'une auto-organisation.

2.5.1.3 K-Arbre

La notion de k-arbre fut la première fois introduite par [40] dans le domaine d'application des bases de données distribuées afin d'optimiser le placement de k informations dans une structure répartie. Un k-arbre est un arbre à exactement k feuilles minimisant la distance moyenne entre un nœud quelconque et le nœud du k-arbre lui étant le plus proche (fig. 2.3). Un k-arbre est donc une structure connexe minimisant la distance moyenne séparant un nœud de la structure. Cette structure pourrait donc être un bon candidat pour une dorsale dans les réseaux ad-hoc.

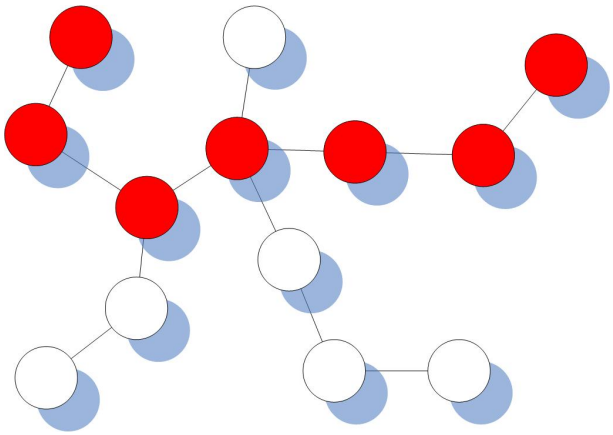


FIG. 2.3 – Exemple de k-tree core (k=2)

[54] présente la construction en partant d'un arbre d'un cœur formant un k-arbre (k-tree core) : de façon itérative, certaines branches sont sélectionnées. Bien que l'algorithme soit prouvé comme optimal, il ne l'est que vis à vis de l'arbre initial. En d'autres termes, le k-arbre n'est pas optimal vis à vis du graphe radio sous-jacent au spanning tree. [49] suit le même algorithme mais détaille également la construction d'une forêt en utilisant des informations de localisation des nœuds. [50] exploite ce k-arbre pour le routage, tous les nœuds ayant choisi le même père

formant une zone de routage, l'arbre formant un deuxième niveau de hiérarchie. Cependant, aucune maintenance n'est détaillée.

Un k-tree core borne statiquement le nombre de feuilles de la structure, alors qu'il nous semble plus pertinent de borner au contraire la distance maximale entre un nœud et la dorsale. De plus, la maintenance conjointe d'un k-tree core et d'un arbre recouvrant nous semble présenter un coût important pour les réseaux ad-hoc. Enfin, la construction d'un k-tree core suppose l'existence préalable d'un arbre recouvrant optimisé pour la construction d'un k-tree core, une telle co-optimisation nous semblant non triviale.

2.5.1.4 CDS

La structure d'ensemble dominant connecté (Connected Dominating Set, CDS) se rapproche beaucoup des dorsales classiques. Un CDS est un ensemble de nœuds tel que tout nœud du réseau est voisin d'au moins un nœud du CDS, et tel que le CDS forme une structure connexe. Si V' est l'ensemble des nœuds du CDS, la définition plus formelle est :

$$\forall u \in V, \quad \exists v \in V' \quad / \quad v \in N(u) \tag{2.3}$$

$$\forall (u,v) \in V'^2, \quad \exists c = chemin_{u \to v} \quad / \quad \forall w \in c, w \in V' \tag{2.4}$$

Un nœud appartenant au CDS est appelé *dominant*, les autres nœuds étant des *dominés*. Un exemple de CDS ou les dominants sont colorés en rouge est donné sur la figure 2.4.

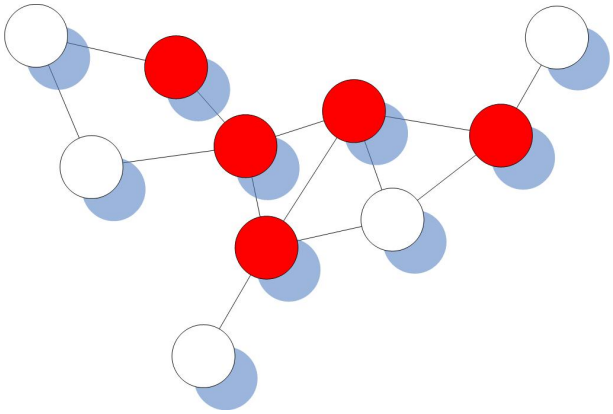


Fig. 2.4 – Exemple d'ensemble dominant connecté (CDS)

On appelle MCDS un CDS de cardinalité minimale. Nous pouvons d'ailleurs remarquer l'équivalence entre la construction d'un MCDS et d'un ST à nombre de feuilles maximal [11]. La construction d'un MCDS de façon centralisée est un problème NP-difficile dans les graphes de disque unité [36]. Sa construction distribuée dans un graphe quelconque l'est donc également. De nombreux articles proposent donc des heuristiques de construction d'une telle structure.

Algorithmes distribués L'approche la plus répandue pour construire un CDS est d'abord d'élire un ensemble dominant, puis de le connecter. Les algorithmes de construction d'ensembles dominants sont similaires. Un nœud possède l'un des quatre états suivants : dominant (membre du CDS), dominé (voisin d'un dominant), actif (en élection), isolé (en attente). Un nœud actif qui possède le plus fort poids parmi ses voisins actifs devient dominant, ses voisins devenant ses dominés. Ensuite, les dominés vont obliger d'autres nœuds à devenir actifs. Le poids pour l'élection peut prendre la valeur du degré [1, 7, 33, 46], l'identifiant du nœud [9], ou un poids non explicité [14].

Il faut ensuite interconnecter l'ensemble dominant en limitant toutefois le nombre de dominés à colorer en dominants. Plusieurs approches ont été proposées :

Interconnexion dans un ensemble dominant à propriétés particulières Si seuls les nœuds *isolés* voisins d'un dominé deviennent actifs, il suffit que deux dominants à moins de 2 sauts l'un de l'autre se connectent pour former un CDS. Les auteurs choisissent souvent de sélectionner plus finement le nombre de ces interconnexions pour réduire la cardinalité du CDS. [9] propose une exploration itérative partant d'un leader dans le réseau. Ce message d'exploration va permettre de créer un arbre d'interconnexion pour les dominants. [7] présente une approche semblable. [38] propose une exploration similaire mais en forçant la séparation de la construction de l'ensemble dominant de son interconnexion. Dans [16], un dominant nouvellement élu choisit simplement de colorier un de ses voisins dominés en dominant (celui-ci est par construction voisin d'un dominant élu antérieurement).

Interconnexion dans un ensemble dominant général Si l'ensemble dominant est quelconque, un dominant peut se trouver à exactement trois sauts de tout autre dominant. [1, 53] suivent une approche de type *best-effort* : des messages d'invitation sont propagés dans le réseau, partant du leader. Un dominant choisit de s'interconnecter via la première invitation reçue. [53] suit une approche similaire. [35] propose la connexion via la construction d'un arbre de Steiner. Cependant, un tel algorithme est quasi centralisé et nécessite un trafic de contrôle important.

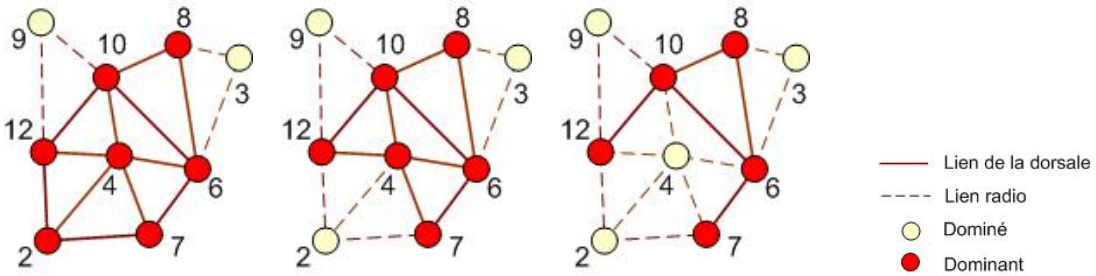
Interconnexion tirant parti d'un protocole [46] se base sur les mécanismes protocolaires sur-jacents : tous les nœuds doivent envoyer périodiquement des messages de présence, inondés par la dorsale. Si un nœud N n'entend pas un voisin dominant relayer les messages de présence issus d'un de ses voisins, alors N devient dominant. Cette simplicité fait sa force et sa faiblesse : un protocole requérant une inondation périodique totale est requis, de même la distance entre la dorsale et un dominé ne peut être fixée comme un paramètre de la dorsale.

Interconnexion maillée Dans [5], chaque dominant tente de s'interconnecter à tous les autres dominants à moins de 3 sauts. En utilisant les informations de ses dominés, un dominant choisit explicitement des chemins pour se connecter aux dominants connus dans le 3-voisinage. [29] propose lui aussi une telle interconnexion, commandée par le dominant si deux dominants sont séparés par au plus 2 sauts, initiée par un dominé si au contraire 2 dominants sont éloignés de 3 sauts. Cependant, la redondance introduite par cette dernière interconnexion peut être importante. [55] étend ces algorithmes pour prendre en compte des chemins d'interconnexion pondérés.

[33] présente la construction de bases de données distribuées se rapprochant des algorithmes de CDS. Un ensemble r-dominant est au préalable construit de façon localisée. Puis, chaque base de données envoie des paquets de contrôle à $2r+1$ sauts pour découvrir les autres dominants et maintenir des *liens virtuels* vers chacun d'eux. Au final, un r-CDS est donc construit et maintenu naturellement, les algorithmes étant localisés. [6] suit une approche similaire mais la distance entre un nœud et la dorsale est fixée à un seul saut.

Algorithmes localisés [58] présente la construction localisée d'un CDS. Chaque nœud envoie en *broadcast* des `hellos` et exécute les règles suivantes : *un nœud est dominant si au moins deux de ses voisins ne sont pas connectés par un lien radio.* Soit la topologie représentée sur la figure 2.5(a). Le nœud 9 possède deux voisins (12 et 10) et ces voisins sont connectés l'un à l'autre. 9 est donc dominé. Par contre, le nœud 8 est dominant car 3 et 10 sont ses voisins et ne sont pas connectés. Nous pouvons remarquer que la cardinalité du CDS est élevée. Les auteurs ont donc proposé les règles 1 & 2 pour réduire cette redondance. Dans la règle 1, un nœud u devient dominé s'il est couvert par un voisins v, i.e. $N(u) \subset N(v)$ et $id(u) < id(v)$. Dans la règle 2, un nœud u devient dominé s'il est couvert par deux de ses voisins v et w, i.e. $N(u) \subset N(v) \cup N(w)$, $id(u) < id(v)$ et $id(u) < id(w)$. Les auteurs prouvent de plus que les plus courtes routes passent obligatoirement par l'ensemble des dominants. Soit la topologie de la figure 2.5(a). Le nœud 2

est dominé car il possède un voisin (4) de plus fort id et qui couvre tout son voisinage. [10] a proposé une version généralisée de cette règle : *un nœud est couvert si un sous-ensemble de ses voisins forme un ensemble connexe et dominant de son voisinage*. Dans la figure 2.5(c), 4 est dominé car l'ensemble $\{6, 10, 12\}$ forme un ensemble dominant connexe de tout son voisinage $\{2, 6, 7, 10, 12\}$.



(a) Élimination de la redondance (b) Règle 2 [58] (c) Règle généralisée [57]

FIG. 2.5 – Exemple de construction d'un CDS avec l'algorithme de Wu & Li

[51] propose la création d'un CDS orienté source afin d'optimiser la diffusion d'information. Chaque nœud N lorsqu'il reçoit un paquet à relayer va armer un temporisateur. Durant ce laps de temps, il surveille les retransmissions de ses voisins. Lorsqu'un voisin V relaie le paquet, N inscrit tous les voisins de V comme couverts. Si au bout du temporisateur, il existe encore des voisins non couverts, N relaie le paquet. Sinon, N peut supprimer le paquet car une transmission serait inutile. La connaissance du voisinage est donc la seule donnée nécessaire. L'ensemble des nœuds-relais forme un CDS. Cependant, si le paquet provient d'un nœud différent, les nœuds membres du CDS ont une forte probabilité de changer, le CDS étant orienté source, ce qui contredit la propriété recherchée de persistance comme décrit dans la section 2.3.

2.5.1.5 WCDS

Un ensemble dominant connecté faiblement (Weakly Connected Dominating Set, WCDS) a également été introduit dans les réseaux ad-hoc afin de modéliser une dorsale. Un WCDS est un ensemble dominant tel que l'ensemble des dominants avec les arêtes dont une des extrémités est un dominant forme un ensemble connexe [18] (fig. 2.6).

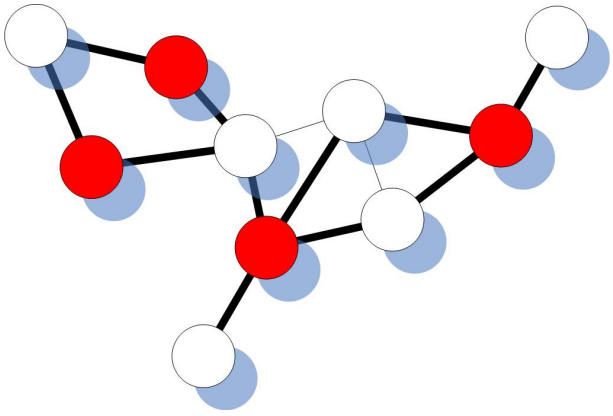


FIG. 2.6 – Exemple de Weakly Connected Dominating Set (WCDS)

[15] construit un WCDS en colorant à chaque étape un nœud à au plus 2 sauts d'un dominant. Les auteurs proposent d'implémenter cet algorithme en distribué en créant plusieurs pièces et

en les fusionnant dans un deuxième temps. Cependant, le trafic de contrôle, le temps de convergence et la robustesse aux incohérences nous semblent dans un tel algorithme problématique. De plus, il reste nécessaire lorsqu'une inondation doit être réalisée dans la dorsale de sélectionner dynamiquement les dominés chargés de relayer les paquets de diffusion. En conclusion, le WCDS nous semble difficilement adaptable pour constituer une bonne dorsale dans les réseaux ad-hoc.

2.5.1.6 Dorsales dans des environnements naturellement hétérogènes

Dans certaines applications notamment militaires, les nœuds peuvent être classés en différentes catégories. La première catégorie est constituée de terminaux mobiles agissant en tant que clients tandis que les autres nœuds sont des équipements mobiles dédiés au réseau, que nous nommerons BCN (Backbone Capable Node). Les BCN pouvant avoir une grande portée radio, la hiérarchie des réseaux filaires classiques peut être calquée sur les réseaux ad-hoc. L'objectif est par conséquent de sélectionner certains BCN afin qu'ils agissent comme une épine dorsale [23, 37, 44, 45].

L'hypothèse d'un tel environnement hétérogène découpé en classes de nœuds nous semble une hypothèse forte pour les applications que nous visons. Nous ne pouvons donc pas adopter une telle démarche.

2.5.1.7 Conclusion

Les dorsales ont été initialement conçues afin de réduire le trafic de contrôle : seuls les nœuds de la dorsale sont autorisés à relayer un trafic de *broadcast*. Ainsi, la cardinalité d'une dorsale représente un critère important de performance. Les approches d'interconnexion maillée de la dorsale présentent notamment une cardinalité élevée, puisque la dorsale créée est très redondante.

Cependant, la cardinalité ne doit pas pour autant pénaliser la robustesse de la structure. Si la dorsale perd temporairement sa connexité et qu'elle est utilisée par exemple de façon concomitante pour une inondation, le paquet ne sera pas délivré à tous les destinataires. Un algorithme de construction de dorsale doit donc proposer un compromis entre cardinalité et robustesse. Cependant, très peu de propositions prennent en compte la maintenance, excepté les algorithmes localisés et les solutions d'interconnexion maillée (cf. section 2.5.1.4 page 18). Les auteurs justifient souvent leur approche en proposant une reconstruction périodique. Par contre, le moment opportun d'une telle reconstruction n'est jamais étudié. De plus, le trafic de contrôle et le temps avant la convergence durant lequel le réseau est inexploitable nous semblent des inconvénients très forts. Les interconnexion maillées de dorsales (comme [33] par exemple) proposent en créant des liens virtuels entre dominants de former et maintenir une dorsale connexe. Cependant, ces liens virtuels sont rapidement sous-optimaux, et le trafic de contrôle généré peut être important, comme nous l'avions mentionné.

Par ailleurs, la dorsale devrait prendre en compte l'hétérogénéité du réseau. Si la dorsale a été construite de telle sorte que seuls les nœuds forts (en énergie, mémoire...) sont membres de la dorsale, les nœuds plus faibles peuvent participer moins. A notre connaissance, un tel aspect n'a jamais été exploré.

Enfin, la dorsale doit rester la plus stable possible : une hiérarchie changeant constamment au niveau logique impactera sans aucun doute sur les protocoles l'exploitant, générant notamment du trafic de contrôle. Cependant, un tel critère n'a, selon l'état de nos connaissances, jamais été pris en compte. Les algorithmes purement localisés n'ont notamment pas été conçus pour remplir un tel objectif.

2.5.2 Clusters

Le découpage en zones d'un réseau étendu permet de l'organiser pour des problématiques d'adressage, de routage, d'agrégation de flux. Si un chef est élu dans chaque zone, une hiérar-

chie est créée. De plus, la topologie de clusters peut former une topologie virtuelle utile pour l'agrégation d'informations et pour cacher les changements dans la topologie radio. Les clusters peuvent être utiles pour le routage [27], la qualité de services [8], la couche MAC [25]...

2.5.2.1 Qu'est ce qu'un bon découpage hiérarchique ?

Le découpage hiérarchique du réseau doit naturellement être distribué et suivre les propriétés décrites dans la section 2.3. Par ailleurs, il est intéressant d'avoir des clusters homogènes en taille : deux zones doivent comporter un nombre comparable de membres. Il peut également être intéressant de borner la distance entre un membre de la zone et le chef de zone afin de limiter le trafic de contrôle immanquablement induit par les fonctions réseau à implémenter (routage, découverte de services...). Enfin, les zones doivent être les plus stables possibles, i.e. les chefs de zones doivent garder leur rôle suffisamment longtemps afin d'être efficaces. De même, si un nœud très mobile change fréquemment de zone, des problèmes au niveau réseau peuvent survenir : si un nœud change trop souvent d'adresses, les routes cassent continuellement...

2.5.2.2 Construction et Maintenance de Clusters

Dans [30], une liste des clusters et de leurs membres est maintenue dans le réseau. Lorsqu'un nouveau nœud joint le réseau, il reçoit la liste et la met à jour en fonction du voisinage qu'il possède. Un mécanisme de suppression des clusters redondants permet de limiter ce nombre. Cependant, aucune procédure d'élimination des incohérences, de reprise sur erreur et d'amélioration de la robustesse n'est proposée. Un tel algorithme est donc difficilement applicable aux réseaux ad-hoc.

[34] présente un algorithme très simple de construction de clusters : chaque nœud possédant un identifiant minimum parmi l'ensemble de ses voisins sans cluster devient chef de cluster, un *clusterhead*.. En outre, tous les voisins sans cluster d'un nouveau clusterhead, joignent ce cluster. Finalement, des clusters sont créés de telle sorte que tout nœud est voisin de son clusterhead. [3] propose de créer des clusters homogènes en mobilité : des nœuds possédant une mobilité relative faible peuvent faire partie d'un même cluster. Inversement, deux nœuds possédant une trajectoire différente auront tendance à former des clusters disjoints. Les auteurs supposent ici l'existence d'un GPS, ce qui nous semble une contrainte forte dans certains environnements embarqués.

[52] utilise la programmation linéaire afin de minimiser la différence de cardinalité entre clusters. L'algorithme prend en entrée l'ensemble des dominants possibles et donne l'ensemble de dominants permettant de minimiser les différences entre clusters. Cependant, aucun détail n'est donné quant à l'adaptation d'une telle optimisation à un environnement distribué.

Dans [31], la construction de clusters est déclenchée par l'inondation d'un message. Le premier nœud qui relaie un message devient clusterhead. Les nœuds voisins deviennent membres. L'overhead induit réside dans l'ajout d'un champ dans les paquets à inonder. Si un nœud entend plusieurs clusterheads, alors il devient passerelle (ou *gateway*). Une passerelle relaie un paquet d'inondation selon le nombre de clusterheads et de passerelles qu'elle entend. Cependant, une telle structure est orientée source et ne présente donc aucune persistance dans le temps, i.e. l'ensemble des clusters change continuellement.

Politiques de Maintenance Nous distinguons deux cas de maintenance de clusters : soit le chef a besoin d'être maintenu, soit il est inutile.

Zones sans Hiérarchie Si le chef est inutile, seule la contrainte de diamètre ne doit pas être violée. Tout nœud doit donc être à au plus 2 sauts de tout autre nœud du même cluster [34]. Lorsque cette contrainte est violée, une décision doit être prise collectivement et de façon distribuée sur la scission du cluster. Si le comportement du réseau n'est pas prévisible,

il est difficile de proposer un algorithme permettant de minimiser les changements à la fois au moment de la scission et dans le futur. De même, deux clusters peuvent fusionner si la contrainte de diamètre reste inviolée après fusion.

Maintenance d'un chef Si un chef est requis durant la phase d'exécution, nous avons dégagé deux objectifs possibles :

- Le chef doit obligatoirement être le nœud possédant le poids le plus élevé dans le cluster. Ainsi, le chef du cluster est le nœud fort du cluster, les nœuds hors de sa portée formant de nouveaux clusters. Par contre, la persistance n'est pas optimisée : un cluster peut changer souvent de chefs. De même, le changement de chef dans le cluster peut obliger à casser le cluster, et donc augmenter le nombre de changements dans la topologie virtuelle.
- Le chef de cluster reste chef le plus longtemps possible, même s'il ne possède pas le poids maximum dans le cluster. La persistance est donc maximisée. Un clusterhead perd son rôle lorsqu'il le décide : il ne possède plus l'énergie suffisante, il s'éteint...

2.5.2.3 Construction de k-Clusters

D'autres articles traitent de la construction de clusters dont la distance maximale entre un nœud et son chef de cluster, i.e. le rayon du cluster, est bornée par le paramètre k.

[14] exécute un algorithme semblable à [34] sur le k-voisinage d'un nœud : tout nœud sans cluster qui possède le plus fort poids parmi tous ses k-voisins sans cluster devient clusterhead. Les auteurs adaptent l'algorithme de maintenance de [34] pour prendre en compte la nouvelle contrainte en terme de rayon.

[2] propose un algorithme en 2 étapes. La première étape durant k rondes permet de propager les plus grands identifiants à k sauts : chaque nœud relaie l'identifiant le plus élevé entendu dans la ronde précédente. La deuxième étape qui dure également k rondes permet de prévenir les nœuds qu'ils ont été élus clusterheads : les nœuds relaient les identifiants les plus petits entendus dans la ronde précédente. Un nœud s'élit clusterhead s'il entend son identité propagée durant la deuxième partie de l'algorithme. L'algorithme permet donc de construire des clusters connexes, i.e. un nœud peut passer par un chemin de moins de k nœuds du même cluster pour joindre son clusterhead. Cependant, aucune maintenance n'est présentée. Par ailleurs, la ré-exécution périodique de l'algorithme afin de tout reconstruire nous semble peu optimisée en terme de trafic de contrôle.

[21] propose de construire dans un premier temps un arbre recouvrant. Puis les branches sont étayées quand elles possèdent une hauteur égale à k sauts. Une branche étayée forme un même cluster. Cependant, les auteurs ne présentent pas d'algorithme de maintenance. De plus, la construction et la maintenance d'un arbre sont requises, pouvant créer un trafic de contrôle important et des problèmes en terme de robustesse et de temps de convergence.

Dans [4], un k-cluster est limité par le nombre de ses membres plutôt que par son rayon. Dans un premier temps, un arbre recouvrant est construit. Puis, en partant des feuilles, l'algorithme étaye une branche quand elle comporte entre k et 2k nœuds. Ainsi, des clusters de cardinalité comprise entre k et 2k sont construits. Cependant, la construction d'un arbre recouvrant semble coûteuse en temps et en trafic de contrôle. Les auteurs proposent également un algorithme de maintenance, permettant de fusionner les clusters trop petits ou au contraire de les scinder. Cependant, les auteurs présupposent l'existence d'un algorithme maintenant l'arbre recouvrant présentant un coût faible.

2.5.3 De la complémentarité du contrôle de topologie

Le contrôle de topologie tente de modifier la topologie radio afin qu'elle présente certaines particularités : le degré peut être borné ou du moins abaissé, le graphe représentant le réseau peut être rendu planaire... La modification de la topologie radio implique d'agir directement au

niveau MAC, et donc d'ajuster la portée radio en diminuant la puissance d'émission, ou alors d'utiliser des antennes intelligentes afin de *choisir* ses voisins réels. La plupart des approches tentent de réduire la portée radio, permettant par la même de réduire la consommation d'énergie. Si la puissance varie en fonction d'une puissance en α de la distance[1], une division par 2 de la porté de communication permet de diviser par 2^{α} la consommation en énergie. De plus, la réduction du degré permet également de réduire les interférences radio et donc d'augmenter la capacité au niveau réseau.

En conclusion, l'auto-organisation permet d'organiser hiérarchiquement le réseau en créant une topologie virtuelle plus exploitable pour les services sur-jacents. Le contrôle de topologie permet lui d'adapter la topologie radio pour obtenir des propriétés différentes. Cependant, de telles techniques ne sont en aucun cas antinomiques, bien au contraire. Ainsi, il pourrait être opportun de combiner un contrôle de topologie et une auto-organisation, avec une optimisation conjointe. La figure 2.7 illustre une architecture possible.

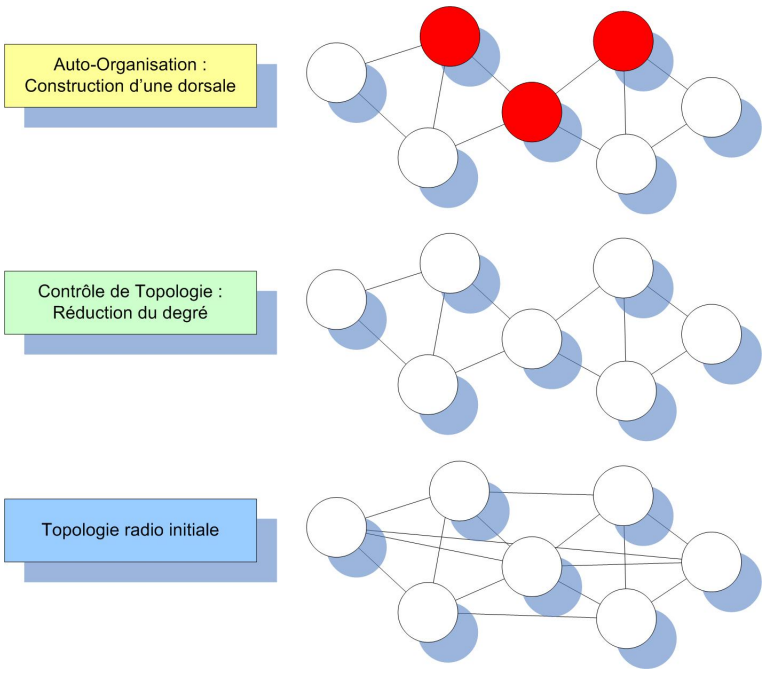


FIG. 2.7 – Exemple d'architecture combinant les techniques d'auto-organisation et de contrôle de topologie

Nous allons maintenant présenter deux types de contrôle de topologie permettant d'améliorer l'efficacité d'un réseau ad-hoc.

2.5.3.1 RNG

Un graphe est un Relative Neighborhood Graph (RNG) [26] si, *quels que soient les points A et B, il n'existe aucun point C dans l'intersection des cercles centrés sur A et B et de rayon AB.* Un RNG est planaire, i.e. aucune arête ne s'intersecte, et présente un faible degré. La propriété de planarité peut être utilisé par un protocole de routage glouton utilisant le GPS [28].

[12] propose la construction distribuée d'un RNG afin d'optimiser la diffusion d'informations dans un réseau ad-hoc. Puisque la règle définissant un RNG est locale, un système de localisation tel quel le GPS permet de construire de façon localisée le RNG associé au graphe radio d'origine. Cependant, un tel équipement étant coûteux, et son installation inconcevable dans

[1] la valeur de α est de 2 en environnement libre sans obstacle, et peut être beaucoup plus élevée en environnement fortement perturbé de type indoor.

certains équipements embarqués, les auteurs proposent de remplacer la distance euclidienne par la différence de voisinage. La distance entre deux points u et v est dénotée par :

$$d_{uv} = \frac{N(u)/N(v) \cup N(v)/N(u)}{N(u) \cup N(v)}$$

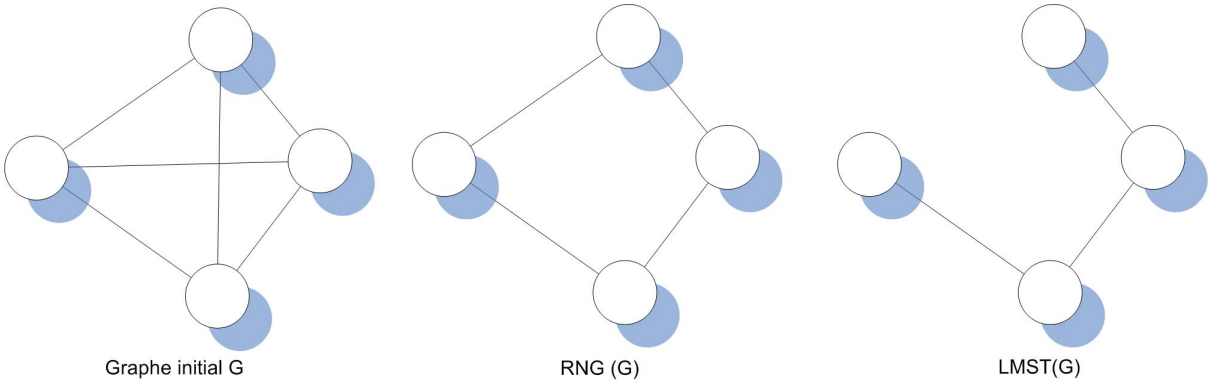


FIG. 2.8 – RNG et LMST d'un graphe G

2.5.3.2 LMST

La construction d'un MST de façon centralisée étant considérée comme non acceptable, des auteurs ont proposé les LMST (localized MST) [32]. Chaque nœud va construire un MST sur le graphe de son voisinage, et ne garder que les voisins radio qui sont également voisins dans le LMST. Comme le montre [13], le LMST d'un graphe G est un sous-graphe du RNG du même graphe G. Le LMST d'un graphe est donc également planaire, et présente un degré moyen (2.04) plus faible que le RNG (2.6). Cependant, la position des voisins étant nécessaire pour la pondération des arêtes, un système de localisation est nécessaire.

2.6 Conclusion

Nous avons présenté ici notre vision de l'auto-organisation. Pour nous, une auto-organisation consiste en la création d'un comportement émergent, structurant le réseau, et permettant par ce biais d'en faciliter l'exploitation. Nous avons exposé différentes solutions de construction de structures virtuelles que nous classons, selon notre vision, dans la catégorie de l'auto-organisation. Cependant, ces solutions ne répondent pas, comme nous l'avons vu, à la totalité de nos exigences en terme de persistance, de robustesse, de passage à l'échelle et de distributivité. Nous allons donc présenter dans le chapitre suivant la structure d'auto-organisation que nous avons proposée.

Bibliographie

[1] 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.

[2] A. Amis, R. Prakash, T. Vuong, and D. Huynh. Max-min d-cluster formation in wireless ad hoc networks. In *INFOCOM*, pages 32–41, Tel-Aviv, Israel, March 1999. IEEE.

[3] B. An and S. Papavassiliou. A mobility-based clustering approach to support mobility management and multicast routing in mobile ad-hoc wireless networks. *International Journal of Network Management*, 11(6) :387–395, November/December 2001.

[4] S. Bannerjee and S. Khuller. A clustering scheme for hierarchical control in wireless networks. In *INFOCOM*, pages 1028–1037, Anchorage, Alaska, April 2001. IEEE.

[5] L. Bao and 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.

[6] S. Basagni, D. Turgut, and S. K. Das. Mobility-adaptive protocols for managing large ad hoc networks. In *International Conference on Communications (ICC)*, pages 1539–1543, Helsinki, Finland, June 2001. IEEE.

[7] 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.

[8] I. Cardei, S. Varadarajan, A. Pavan, L. Graba, M. Cardei, and M. Min. Resource management for ad hoc wireless networks with cluster organization. *Cluster Computing*, 7(1) :91–103, January 2004.

[9] 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.

[10] J. Carle and D. Simplot-Ryl. Energy efficient area monitoring by sensor networks. *IEEE Computer Magazine*, 37(2) :40–46, February 2004.

[11] Y. Caro, D. B. West, and R. Yuster. Connected domination and spanning trees with many leaves. *SIAM Journal on Discrete Mathematics*, 13(2) :202–211, 2000.

[12] J. Cartigny, D. Simplot, and I. Stojmenovic. Localized minimum-energy broadcasting in ad-hoc networks. In *INFOCOM*, pages 2210–2217, San Francisco, USA, March-Avril 2003. IEEE.

[13] J. Cartigny, D. Simplot-Ryl, and I. Stojmenovic. An adaptive localized scheme for energy-efficient broadcasting in ad hoc networks with directional antennas. In *Personal Wireless Communications (PWC)*, Delft, Netherlands, September 2004. IFIP.

[14] G. Chen, F. Garcia, J. Solano, and I. Stojmenovic. Connectivity based k-hop clustering in wireless networks. In *Proceedings of the 35th Hawaii International Conference on System Sciences*, Maui, Hawaii, January 2002. IEEE.

[15] Y. P. Chen and A. 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.

[16] X. Cheng and D.-z. Du. Virtual backbone-based routing in multihop ad hoc wireless networks. Technical Report 02-002, University of Minnesota, Minnesota, USA, January 2002.

[17] B. N. Clark, C. J. Colburn, and D. S. Johnson. Unit disks graphs. *Discrete Mathematics*, 86 :165–177, December 1990.

[18] J. E. Dunbar, J. W. Grossman, J. H. Hattingh, S. T. Hedetniemi, and A. A. McRae. On weakly connected domination in graphs. *Discrete Mathematics*, 167-168 :261–269, April 1997.

[19] Encyclopædia Universalis.

[20] D. Eppstein. Spanning trees and spanners. Technical Report 96-16, University Of California, Irvine, USA, May 1996.

[21] Y. Fernandess and D. Malkhi. K-clustering in wireless ad hoc networks. In *International Workshop on Principles of Mobile Computing (POMC)*, pages 31–37, Toulouse, France, October 2002. ACM Press.

[22] R. G. Gallager, P. A. Humblet, and P. M. Spira. A distributed algorithm for minimum-weight spanning trees. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 5(1) :66–77, January 1983.

[23] M. Gerla and K. Xu. Topology management of hierarchical mobile ad hoc networks. In I. C. M. Cardei and D. Z-Du, editors, *Resource Management in Wireless Networking*. Kluwer Academic Publisher, 2004.

[24] F. Heylighen. Principles of systems and cybernetics : an evolutionary perspective. *Cybernetics and Systems*, 23 :3–10, 1992.

[25] T.-C. Hou and T.-J. Tsai. An access-based clustering protocol for multihop wireless ad hoc networks. *IEEE Journal on Selected Areas in Commnications*, 19(7) :1201–1210, July 2001.

[26] J. Jaromczyck and G. T. Toussaint. Relative neighborhood graphs and their relatives. In *Proceedings of the IEEE*, volume 80 of *9*, pages 1502–1517, September 1992.

[27] M. Jiang, J. Li, and Y. C. Tay. Cluster based routing protocol (CBRP). Internet draft version 01, IETF, July 1999.

[28] B. Karp and H. Kung. GPSR : Greedy perimeter stateless routing for wireless networks. In *International Conference on Mobile Computing and Networking (MOBICOM)*, pages 243–254, Boston, USA, August 2000. ACM.

[29] U. C. Kozat, G. Kondylis, B. Ryu, and M. K. Marina. Virtual dynamic backbone for mobile ad hoc networks. In *International Conference on Communications (ICC)*, Helsinki, Finland, June 2001. IEEE.

[30] 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.

[31] T. J. Kwon and M. Gerla. Efficient flooding with passive clustering (pc) in ad-hoc networks. *ACM SIGCOMM Computer Communication Review*, 32(1) :44–56, January 2002.

[32] N. Li, J. C. Hou, and L. SHa. Design and analysis of an MST-based topology control algorithm. In *INFOCOM*, pages 1702–1712, San Francisco, USA, April 2003. IEEE.

[33] B. Liang and Z. J. Haas. Virtual backbone generation and maintenance in ad hoc network mobility management. In *INFOCOM*, pages 1293–1302, Tel-Aviv, Israel, March 2000. IEEE.

[34] C. R. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. *IEEE Journal of Selected Areas in Communications*, 15(7) :1265–1275, 1997.

[35] J.-H. Lin, C.-R. Dow, and S.-F. Hwang. A distributed virtual backbone development scheme for ad-hoc wireless networks. *Wireless Personal Communications*, 27(3) :215–233, 2003.

[36] M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi, and D. J. Rosenkrantz. Simple heuristics for unit disk graphs. *Networks*, 25 :59–68, December 1995.

[37] R. Meraihi, G. Le Grand, N. Puech, M. Riguidel, and S. Tohmé. Improving ad hoc network performance with backbone topology control. In *Vehicular Technology Conference (VTC Fall)*, Los Angeles, USA, September 2004. IEEE.

[38] M. Min, F. Wang, D.-Z. Du, and P. M. Pardalos. A reliable virtual backbone scheme in mobile ad-hoc networks. In *International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, pages 60–69, Fort Lauderdale, USA, October 2004. IEEE.

[39] M. Mouly and M.-B. Pautet. *The GSM System for Mobile Communications*. Cell & Sys, 1992.

[40] S. Peng, A. Stephens, and Y. Yesha. Algorithms for a core and k-tree core of a tree. *Journal of Algorithms*, 15(1) :143–159, July 1993.

[41] C. Prehofer and C. Bettstetter. Self-organization in communication networks : principles and design paradigm. *IEEE Communications Magazine*, 43(7) :78–85, July 2005.

[42] S. Radhakrishnan, G. Racherla, C. N. Sekharan, N. S. Rao, and l. S. Batsel. Protocol for dynamic ad-hoc networks using distributed spanning trees. *Wireless Networks*, 9(6) :673–686, November 2003.

[43] Y. Rekhter and T. Li. BGP version 4. RFC 1771, IETF, March 1995.

[44] I. Rubin, A. Behzad, H.-J. Ju, R. Zhang, X. Huang, Y. Liu, R. Khalaf, and J. Hsu. *Ad Hoc Wireless Networks with Mobile Backbones*, chapter 9. Kluwer, 2005.

[45] I. Rubin, X. Huang, Y. C. Liu, and H.-j. Ju. A distributed stable backbone maintenance protocol for ad hoc wireless networks. In *Vehicular Technology Conference (VTC Spring)*, Jeju, Korea, April 2003. IEEE.

[46] B. Ryu, J. Erickson, J. Smallcomb, and S. Dao. Virtual wire for managing virtual dynamic backbone in wireless ad hoc networks. In *International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL'M)*, Seattle, USA, August 1999. IEEE.

[47] C. A. Santivanez, B. McDonald, I. Stavrakakis, and R. Ramanathan. On the scalability of ad hoc routing protocols. In *INFOCOM*, New York, USA, June 2002. IEEE.

[48] M. Seaman. Spanning tree. Standard 802.1D, IEEE, 2004.

[49] S. Srivastava and R. Ghosh. Distributed algorithms for finding and maintaining a k-tree core in a dynamic network. *Information Processing Letters*, 88(4) :187–194, November 2003.

[50] S. Srivastava and R. K. Ghosh. Cluster based routing using a k-tree core backbone for mobile ad hoc networks. In *International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL'M)*, pages 14–23. ACM Press, 2002.

[51] 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*, 12(12) :14–25, December 2001.

[52] D. Turgut, B. Turgut, R. Elmasri, and T. V. Le. Optimizing clustering algorithm in mobile ad hoc networks using simulated annealing. In *Wireless Communications and Networking Conference (WCNC)*, New Orleans, USA, March 2003. IEEE.

[53] P.-j. Wan, K. M. Alzoubi, and O. Frieder. Distributed construction of connected dominating set in wireless ad hoc networks. *Mobile Networks and Applications*, 9(2) :141–149, April 2004.

[54] B.-F. Wang. Finding a k-tree core and a k-tree center of a tree network in parallel. *IEEE Transactions on Parallel and Distributed Systems*, 9(2) :186–191, 1998.

[55] Y. Wang, W. Wang, and X.-Y. Li. Low-cost routing in selfish and rational wireless ad hoc networks. In *International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, Urbana-Champaign, USA, May 2005. IEEE.

[56] Wikipedia, l'encyclopédie libre. http://wikipedia.org.

[57] J. Wu. An enhanced approach to determine a small forward node set based on multipoint relays. In *Vehicular Technology Conference (VTC Fall)*, pages 2774–2777, Orlando, USA, October 2003. IEEE.

[58] J. Wu and H. Li. Dominating-set-based routing in ad hoc wireless networks. In *International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL'M)*, pages 7–14, Seattle, USA, August 1999. ACM.

Publications

Chapitre de livre

[1] F. Theoleyre and F. Valois. *Réseaux mobiles ad hoc et réseaux de capteurs (Traité IC2, série Réseaux et télécommunications)*, chapitre Auto-organisation de réseaux ad hoc : concepts et impacts, pages 101–128. Hermes, 2006.

Séminaire

[2] F. Theoleyre and F. Valois. Construction de structures virtuelles dans les réseaux ad hoc. In *Techniques Algorithmiques, Réseaux et d'Optimisation pour les Télécommunications (TAROT)*, Paris, France, January 2004.

Chapitre 3

Une proposition d'auto-organisation pour réseaux ad hoc et hybrides

3.1 Introduction

De nombreuses structures virtuelles d'auto-organisation ont déjà été proposées, comme nous l'avons vu précédemment. Les propositions considèrent soit la construction d'une dorsale permettant d'optimiser la diffusion d'information, soit le découpage du réseau en zones plus petites, en formant une structure hiérarchique. Cependant, comme nous l'avons vu précédemment, ces structures souffrent pour la plupart de carences en termes de robustesse aux changements de topologie, de persistance dans le temps ou de trafic de contrôle.

Ainsi, nous proposons ici une structure virtuelle d'auto-organisation permettant de structurer le réseau en offrant une vue logique plus stable de la topologie radio. Cette structure doit présenter une utilité par exemple pour le routage, comme nous les verrons par la suite. Nous proposons de combiner les avantages des infrastructures virtuelles de dorsales et de clusters. Une hiérarchie, une organisation est maintenue, et tous les processus chargés de construire et maintenir cette hiérarchie partagent des informations communes afin de réduire le trafic de contrôle induit.

Ce chapitre est consacré à la présentation de la structure d'auto-organisation tandis que le chapitre suivant analysera plus en détails les algorithmes proposés ici, en démontrant notamment leur propriété d'auto-stabilisation. Dans ce chapitre, nous allons dans un premier temps exposer les motivations de notre travail et donner un aperçu général de notre solution d'auto-organisation. Ensuite, nous présenterons dans la section 3.3 les algorithmes de construction, puis la section 3.4 détaillera les algorithmes de maintenance. La section 3.5 donnera une méthode pour prendre en charge plusieurs dorsales. Enfin, la section 3.6 étudiera les performances de cette structure au travers de simulations.

3.2 Motivations et Description générale

Nous rappelons rapidement avant la description de notre proposition les fonctions recherchées de notre structure d'auto-organisation :

- Tirer parti de l'hétérogénéité en faisant participer plus activement les nœuds forts.
- Optimiser les inondations afin d'empêcher la formation d'une *tempête de broadcast* [13].
- Créer un découpage logique proche d'un découpage physique permettant de *situer*, *adresser* des nœuds.
- Structurer le réseau en créant des leaders de zones

Naturellement, cette structure doit suivre les propriétés décrites dans le paragraphe 2.3 page 11 (robustesse, persistance, distributivité, passage à l'échelle, faible trafic de contrôle...).

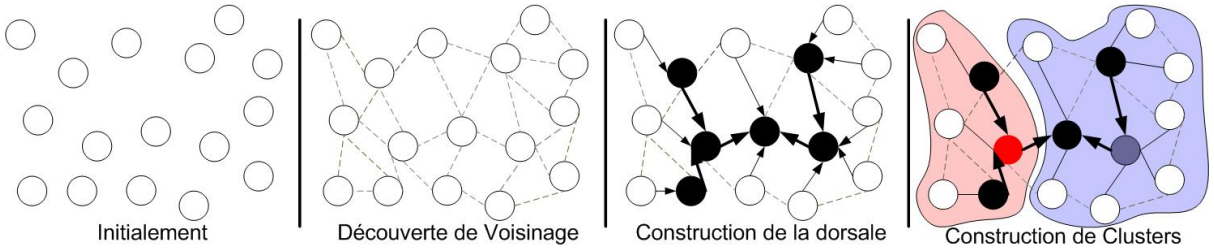


FIG. 3.1 – Schéma de la construction de la topologie virtuelle

La figure 3.1 présente un aperçu général de notre structure virtuelle. Les nœuds initient une découverte de voisinage pour découvrir leurs voisins radio. A partir de ce voisinage radio, la dorsale est construite en élisant de façon distribuée des dominants, et en les interconnectant afin de former une dorsale connexe. Enfin, au dessus de cette dorsale sont construits des clusters qui permettent de servir de zone de localisation ou de zones de services. Bien que la description en soit ici itérative, les différentes phases de l'algorithme s'exécutent en parallèle afin de réduire le temps de convergence et d'améliorer la robustesse. Nous avons proposé des algorithmes de construction mais également de maintenance événementielle. Les deux sections suivantes en donnent les détails.

3.3 Algorithmes distribués de construction d'une structure virtuelle

La construction de la dorsale est réalisée avant la construction des clusters pour les raisons suivantes :

- optimiser le nombre de noeuds participant à l'élection des chefs de zones.
- forcer un chef de zone à être membre de la dorsale.
- tirer profit de la dorsale pour le trafic de contrôle de construction des zones.
- fixer une distance limite entre un noeud et son chef de zone via la dorsale.

Il est toutefois important de noter que la construction de la dorsale et celle des clusters ne se déroulent pas de façon parfaitement séquentielle. Il est par contre nécessaire de coordonner les deux parties de la structure virtuelle, un clusterhead n'étant par exemple élu que parmi les membres de la dorsale. En conséquence, un nœud initie l'élection de chefs de cluster dés que ses voisins ont déterminé leur rôle dans la dorsale. En agissant de façon localisée, la construction présente une convergence plus rapide, et une plus grande robustesse aux fautes. Il est par exemple non acceptable d'attendre que chaque nœud reporte la fin de la construction pour initier le clustering. La construction, tant pour la dorsale que pour les clusters s'appuie sur la connaissance du voisinage, nous allons donc ici détailler ce processus commun.

3.3.1 Découverte de voisinage

Pour découvrir ses voisins, un nœud envoie périodiquement un paquet `hello`. Puisque le médium radio est de type diffusion (*broadcast*), tous ses voisins le recevront. Cependant, il peut se trouver que deux nœuds possèdent une portée radio différente, pouvant mener à la création de liens asymétriques. Par ailleurs, un protocole MAC, bien souvent, utilise des acquittements pour fiabiliser l'envoi de ses trames unicast, et donc limiter l'impact des collisions ou le manque de fiabilité des liens radio. Or, par définition, un lien asymétrique ne permet pas de prendre en charge une telle fonction. Nous avons donc choisi de n'utiliser que les liens bidirectionnels. Un nœud doit donc envoyer dans ses paquets `hellos` la liste des nœuds qu'il entend au niveau radio. En conclusion, les liens unidirectionnels peuvent être éliminés.

Pour construire la dorsale, la connaissance du k_{cds}-voisinage, i.e. l'ensemble des nœuds à k_{cds} sauts ou moins, est requise. Pour remplir un tel but, deux choix sont possibles :

- Approche de type *vecteur de distance* : chaque nœud envoie dans ses paquets `hellos` la liste des nœuds à k_{cds}-1 sauts ou moins. Chaque nœud peut en agrégeant les informations de ses voisins connaître son k_{cds}-voisinage.
- Approche de type *état de liens* : un paquet `hello` est relayé sur k_{cds}-1 sauts. Comme un paquet `hello` contient la liste des voisins de la source, la reconstruction du k_{cds}-voisinage est possible.

La première approche permet de réduire le trafic de contrôle : les paquets sont plus importants en taille mais moins nombreux. Or, le nombre de paquets possède en radio un impact beaucoup plus important que la taille des paquets. Cependant, la deuxième approche converge beaucoup plus rapidement. Nous avons donc employé cette méthode dans notre implémentation.

Afin que les changements de topologie soient détectés par les nœuds, l'envoi de paquets `hellos` est périodique.

3.3.2 Construction de la dorsale

Nous avons choisi de construire une dorsale sous la forme d'un k_{cds}-CDS : la distance entre un noeud et la dorsale est un paramètre de notre solution. Dans un réseau statique, k_{cds} peut être élevé afin de limiter la cardinalité de la dorsale. Dans un réseau fortement mobile, k_{cds} devra être faible pour limiter les déconnexions. Un noeud peut prendre l'un des états suivants :

- isolé : en état d'initialisation, le noeud attend le signal déclencheur pour la détermination de son état
- actif : en processus d'élection pour devenir dominant
- dominant : membre de la dorsale
- dominé : client de la dorsale, possédant un dominant à moins de k_{cds} sauts

Nous avons fait le choix de ne pas nous focaliser uniquement sur la minimisation de la cardinalité de la dorsale. Nous pensons au contraire que la robustesse d'une dorsale représente une propriété beaucoup plus importante. Par exemple, il est inutile de minimiser le nombre de nœuds relais lorsqu'un paquet inondé n'est reçu que par un faible nombre des terminaux auxquels il aurait du être délivré. Pour nous, une dorsale doit avant tout remplir la tâche pour laquelle elle a été conçue, plutôt que de minimiser par exemple le trafic de contrôle, qui est une propriété subsidiaire.

Par ailleurs, la construction se déroule en deux phases : dans un premier temps, un ensemble dominant est construit. Puis il est interconnecté. Nous utilisons dans nos algorithmes la présence d'un leader. Dans un réseau hybride, le point d'accès (AP) peut jouer le rôle de leader naturel. Si au contraire aucun AP n'existe, il est nécessaire d'en élire un en utilisant par exemple l'algorithme décrit dans [12].

3.3.2.1 Création d'un ensemble dominant

La première étape permet de construire un ensemble dominant, i.e. tout nœud dominé est à au plus k_{cds} sauts d'un dominant. Lorsqu'un nœud change d'état, il envoie un `hello` immédiatement, sans attendre la fin de son temporisateur. Sur réception d'un tel paquet, un nœud applique les règles suivantes :

- un noeud isolé ou actif recevant un message d'un dominant à moins de k_{cds} sauts devient dominé et fixe la source comme père
- un noeud isolé recevant un message d'un dominé à moins de k_{cds}+1 sauts devient actif et arme un temporisateur pour l'élection
- un noeud actif pour lequel le temporisateur est écoulé, qui possède le poids le plus élevé parmi tous ses k_{cds}-voisins actifs devient dominant.

Le leader devient le premier dominant du réseau. Il va donc déclencher le changement en dominé de ses k_{cds}-voisins. De même, les dominés entraînent le changement en actif des k_{cds}+1-voisins. Parmi ces nœuds actifs seront élus les dominants qui possèdent le plus fort poids. Finalement, ces dominants vont engendrer de nouveaux changements. Ainsi, l'algorithme de construction procède par vagues de changements d'états. Ces changements sont illustrés sur la figure 3.2 (ils correspondent aux transitions préfixées par DS (Dominating Set)).

La deuxième règle fait intervenir le (k_{cds}+1)-voisinage d'un nœud. Cependant, une telle connaissance n'est requise que pour limiter la redondance dans l'élection des dominants. Ainsi, lorsqu'un nœud change d'état, il envoie un paquet `hello` avec son nouvel état avec un TTL d'exactement $k_{cds} + 1$, et seulement dans ce cas.

Par ailleurs, nous pouvons remarquer que durant la première phase de la construction, seul un dominé possède un père dans la dorsale. Le dominant fixera son père dans la deuxième étape de la construction.

Un exemple de cette construction est donné figure 3.3 page suivante (étapes 1 à 4). Dans la première étape, le leader devient le premier dominant. Puis les voisins du dominant deviennent dominés, et les 2 voisins des dominés deviennent actifs. Durant l'étape 3, 2 noeuds actifs ont été élus dominants car ils possédaient le poids le plus élevé parmi leurs voisins actifs. Finalement, le processus réitérant, un ensemble dominant est bien construit.

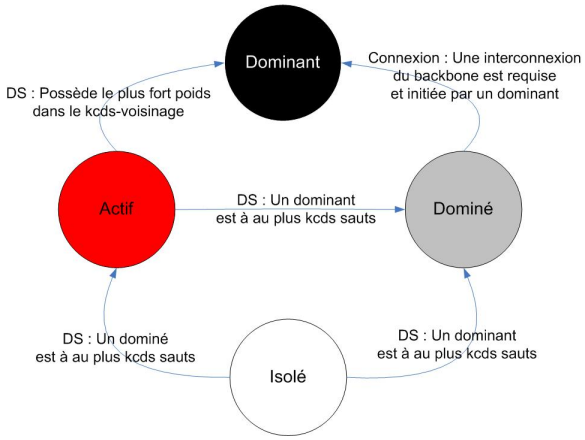


FIG. 3.2 – Diagramme des changements d'états pour la construction de la dorsale

3.3.2.2 Interconnexion de l'ensemble dominant

Il reste maintenant à connecter l'ensemble dominant pour former une dorsale. Nous avons choisi de maximiser la robustesse et la rapidité de convergence. La redondance éventuelle dans la dorsale sera supprimée au cours de la maintenance.

Nous proposons ici un mécanisme d'interconnexion inspiré de [3]. Le leader constitue initialement le seul dominant connecté. Il envoie un paquet d'invitation à la connexion, un `cds-invite`, en *broadcast* avec un TTL égal à $2k_{cds}$+1. Un dominé recevant une invitation avec un TTL t la relaie s'il a relayé au plus $max_{invitations}$ avec un TTL supérieur ou égal à t. Si aucune collision ne se produit, il est conseillé de fixer $max_{invitations}$ à 1. Sinon, $max_{invitations}$ constitue un compromis entre le trafic de contrôle généré et la robustesse aux collisions. Si toutefois les `cds-invite` subissent de nombreuses collisions, la dorsale peut ne pas être connexe à la fin de la construction. Cependant, les algorithmes de maintenance étant auto-stabilisants [1], une dorsale connectée et fonctionnelle sera tout de même obtenue en cours de maintenance.

Lorsqu'un dominant non connecté reçoit un `cds-invite`, il y répond par un `cds-accept`. Ce paquet est envoyé sur la route inverse, vers la source de l'invitation. Les dominés intermé-

[1] Nous reviendrons en détail sur ce point dans le chapitre 4 page 55

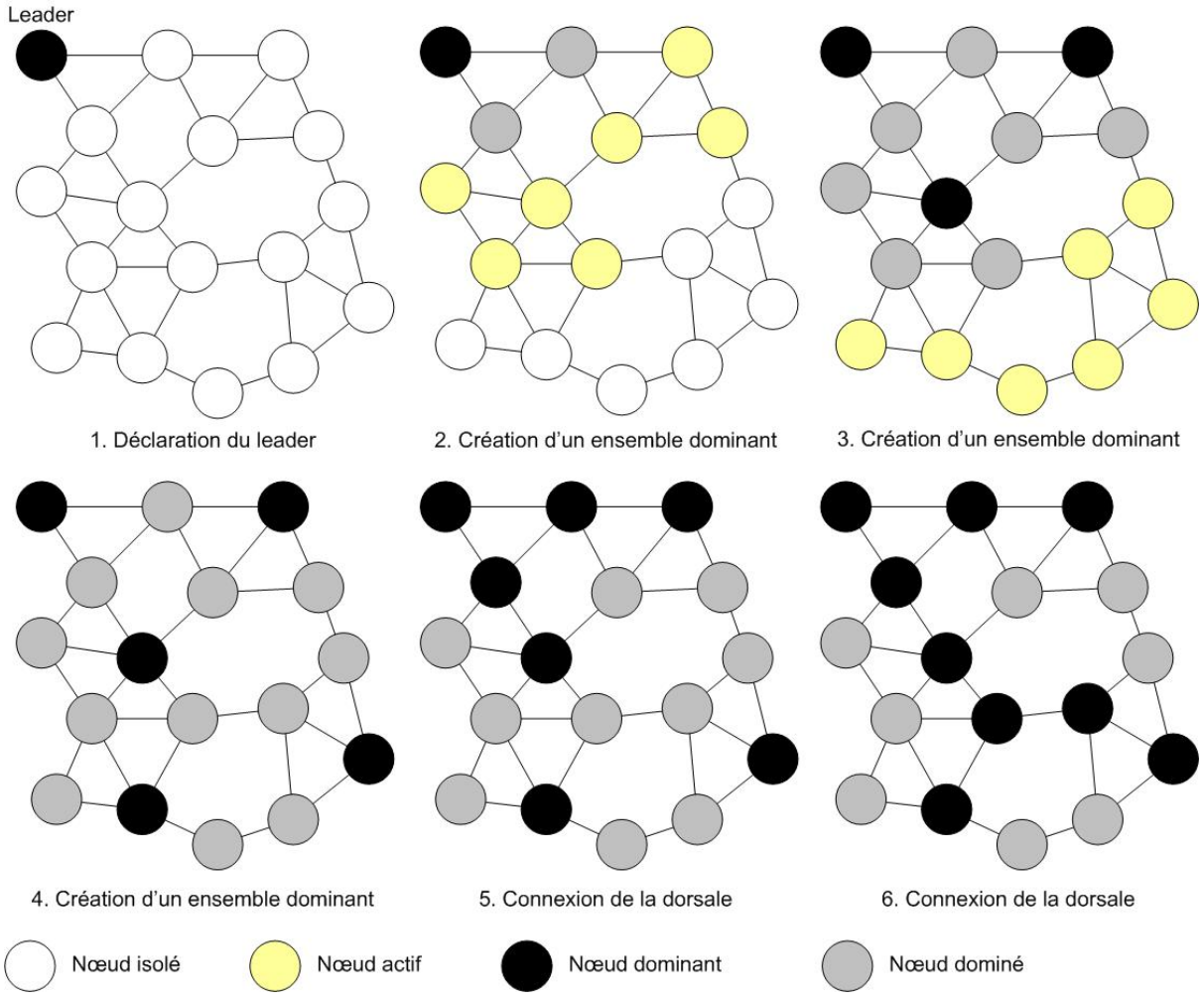


FIG. 3.3 – Construction de la dorsale : exemple avec k_{cds} fixé à 1

diaires relayant un `cds-accept` deviennent dominants et fixent le prochain saut comme nouveau père. Ainsi, le dominant source du `cds-accept` devient connecté et peut lui même envoyer un `cds-invite` pour autoriser les autres dominants à se connecter par son intermédiaire.

Un dominant ne répond pas immédiatement à un `cds-invite`. Il arme au préalable un temporisateur. Si au bout de ce temps, il a reçu plusieurs `cds-invite`, il choisit de répondre à celui maximisant le poids minimum des nœuds intermédiaires du chemin. En effet, si un nœud de faible poids fait partie du chemin suivi par le `cds-invite`, ce dominé sera coloré en dominant, et constituera un point faible dans la dorsale.

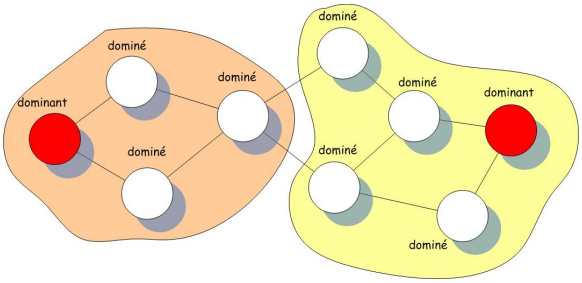


FIG. 3.4 – Espacement de deux dominants dont les zones de dominances sont voisines (k_{cds}=1)

Nous allons expliquer ici intuitivement la raison pour laquelle fixer la valeur du TTL des invitations à $2k_{cds}+1$. Nous appelons zone de dominance un groupe de nœuds dominés possédant le même père, plus le dominant en question. Nous pouvons remarquer que deux dominants appartenant à deux zones de dominance voisines sont espacés d'au plus $2k_{cds}+1$ sauts : les deux dominés à la frontière des deux zones sont à au plus k_{cds} sauts de leur propre dominant respectif (fig. 3.4). Ainsi, les invitations permettront une interconnexion totale de la dorsale, comme démontré de façon rigoureuse dans le chapitre suivant.

Finalement, à la fin de la deuxième phase, la dorsale forme un semble connexe de dominants. de plus, chaque dominant possède un père dans la dorsale, constituant le prochain saut vers le leader. Enfin, chaque dominé possède un père dominant et à au plus k_{cds} sauts. Il existe un chemin qui l'unit à ce père et qui ne contient que des dominés ayant choisi le même père. Ainsi la dorsale forme un arbre dirigé, dont le leader est la racine.

Reprenons l'exemple de la figure 3.3 page précédente. Les étapes 5 et 6 représentent la phase d'interconnexion. Dans l'étape 5, les 2 dominants à 2 sauts du leader ont reçu l'invitation du leader et ont forcé les dominés intermédiaires à devenir dominants. L'étape 6 connecte les 2 derniers dominants. Nous pouvons là encore remarquer que le processus de connexion se déroule en vagues.

3.3.3 Construction des clusters

Afin de minimiser le trafic de contrôle induit par la construction des clusters, seuls les dominants participent à la construction des zones. Un dominé rallie automatiquement le même cluster que son père. En conséquence, un dominant doit inscrire l'adresse de son clusterhead dans ses `hellos` afin que ses dominés sachent dans quel cluster ils se trouvent.

Les clusters sont construits en utilisant la topologie de la dorsale : la construction doit donc être terminée. Cependant, un algorithme de construction séquentiel (dorsale puis clusters) pourrait augmenter le temps de convergence. De plus, il n'est par exemple pas nécessaire que les voisins du leader attendent que le réseau entier ait fini de construire la dorsale. Ainsi, tout dominant connecté et qui ne possède aucun voisin soit dominé soit isolé considère que sa zone locale a terminé la phase de construction de la dorsale. Il initie donc la construction des clusters.

Nous souhaitons limiter la distance via la dorsale entre un nœud et son chef de cluster. Ainsi, seule la topologie de la dorsale doit être empruntée : les dominants initient une découverte de leur

voisinage virtuel. Un voisin virtuel est soit un fils[1] soit un père dans la dorsale. Un voisin virtuel est un sous-ensemble des voisins radio réels. Chaque dominant envoie donc périodiquement un `cluster-hello` avec un TTL de $k_{cluster}$-k_{cds}. Un dominant relaie un `cluster-hello` seulement s'il provient d'un voisin virtuel. Un dominant stoppe l'émission de `cluster-hellos` lorsqu'il ne possède plus aucun dominant sans clusterhead dans sa table de voisinage.

Un dominant qui possède le plus fort poids parmi tous ses $k_{cluster}$-k_{cds}-voisins virtuels sans clusterhead s'élit clusterhead. Conséquemment, il envoie immédiatement un `cluster-hello` afin d'avertir de sa décision. Un dominant D relayant un `cluster-hello` choisit la source du paquet comme clusterhead si :

- D ne possède pas de clusterhead
- La source du paquet est à au plus $k_{cluster}$-k_{cds} sauts
- Le précédent dominant a choisi le même clusterhead

Ainsi, des clusters connexes sont formés : un nœud peut joindre son clusterhead via un chemin de nœuds de son propre cluster.

Un dominant possède un clusterhead à au plus $k_{cluster}$-k_{cds} sauts. De plus, un dominé est à au plus k_{cds} sauts de son père dominant. Ainsi, les clusters formés présentent bien un rayon maximum de $k_{cluster}$.

Les `cluster-hello` étant envoyés seulement par les nœuds de la dorsale, leur impact sur le trafic de contrôle est faible. D'autre part, ces paquets ne sont utilisés que lors de la construction, la maintenance extrayant les informations dont elle a besoin directement des paquets `hellos`.

3.4 Algorithmes distribués de maintenance événementielle d'une structure virtuelle

Il est absolument primordial, dans un réseau ad hoc de continuellement maintenir une structure virtuelle. La construction est exécutée à l'initialisation du réseau tandis que la maintenance doit être continuellement active, tout en présentant un trafic de contrôle réduit. De plus, il doit être possible d'exécuter directement l'algorithme de maintenance, sans la phase de construction.

L'information du k_{cds}-voisinage est nécessaire pour la maintenance de la structure virtuelle. Un nœud doit donc surveiller en permanence les changements de topologie. Chaque noeud envoie donc périodiquement dans un hello son poids, son état, son père, sa distance au père, son chef de zone, ses voisins et leur poids. Ainsi, chaque dominant peut maintenir la liste de ses dominés, i.e. dominés desquels il est père, et de ses fils, i.e. dominants desquels il est père dans la structure virtuelle.

Dans un environnement radio, il est plus coûteux d'envoyer plusieurs paquets qu'un paquet de taille plus importante. En effet, l'envoi du médium par la couche MAC nécessite des paquets de contrôle, des temps d'acquisition du médium radio. Nous avons donc choisi d'ajouter les informations pour la maintenance directement dans les `hellos` :

- Le poids de la source pour les élections de dominants
- L'état (dominant, dominé, actif ou isolé) de la source de telle sorte que ses voisins puissent changer leur propre état
- L'identité du père et sa distance afin que chaque dominant maintienne l'identité de ses fils et dominés
- L'identifiant de l'AP et sa distance sont optionnels, ils peuvent servir à un protocole de gestion de la mobilité comme Mobile IP
- Le nombre et la liste des voisins. Un drapeau *bidirect* permet de détecter les liens unidirectionnels. Comme l'état des voisins à exactement k_{cds} sauts est requis, la liste des états de chaque voisin est également reportée dans le paquet

[1] un fils de N est un dominant ayant choisi N comme père dans la dorsale

3.4.1 Maintenance de la dorsale

La dorsale doit garder sa propriété de dominance, et doit de plus rester connexe. En conséquence, nous proposons la maintenance événementielle suivante.

3.4.1.1 Propriété de dominance

Chaque dominé vérifie la validité de son père. Un père P à une distance d_P est valide si :

- P est un dominant
- P est à au plus k_{cds} sauts ($k_{cds} \geq d_P$)
- Il existe un autre dominé ayant choisi P comme père et qui déclare dans ses paquets `hellos` qu'il est à $d_P - 1$ sauts de lui. Nous forçons ainsi la connexité des zones de dominance

Si l'ancien père est toujours valide, le dominé le garde. Nous optimisons ainsi la persistance de la structure : la structure reste inchangée si aucune collision et aucun changement de topologie ne se produisent.

Si un dominé possède un père invalide, il cherche un nouveau dominant valide dans sa table de voisinage. S'il en trouve un, il envoie un `hello` gratuit, i.e. il envoie immédiatement un `hello` même si son temporisateur n'est pas terminé. Ainsi, le dominant choisi mettra à jour la liste de ses dominés pour éviter de se déclarer comme redondant (cf. section 3.4.1.4 page suivante).

Si au contraire un dominé ne trouve pas de de dominant possible, il devient actif. Une élection parmi les nœuds actifs permettra d'élire un ou plusieurs dominants selon les mêmes règles que la création d'un ensemble dominant lors de la construction. Enfin, les actifs élus dominants exécuteront la procédure de maintenance réservée aux dominants afin de se reconnecter à la dorsale. Ainsi, lorsqu'un dominant part en laissant ses dominés isolés, ces dominés deviendront actifs et une élection permettra de reconstituer un ensemble dominant.

3.4.1.2 Maintien de la connexité

La dorsale doit rester connexe. Nous avons donc choisi que l'AP envoie périodiquement des `ap-hellos` avec un numéro de séquence croissant. Ces `ap-hellos` sont relayés en multicast via la dorsale uniquement par les dominants afin de limiter le nombre de transmissions. Lorsqu'un dominant reçoit un `ap-hello` venant de son père, il le relaie. Sinon, il ajoute la source comme père secondaire si le numéro de séquence est strictement supérieur à celui du dernier `ap-hello` envoyé par son père. Ainsi, un dominant ne peut choisir comme père secondaire un de ses descendants dans la dorsale.

Un dominant D se considère déconnecté dans l'un des cas suivants :

- Le père de D n'est plus dominant, ou n'est plus voisin
- D n'a reçu aucun des $\max_{lost}$ derniers `ap-hellos` de son père

Lorsqu'un dominant est déconnecté, il choisit comme nouveau père principal son père secondaire de plus fort poids. Il l'avertit de sa décision en envoyant un `hello` gratuit s'il est voisin. Cependant, la liste de pères secondaires peut se trouver vide. Ainsi, nous proposons le mécanisme suivant de découverte :

1. D génère un `cds-reconnnect` avec le numéro de séquence du dernier `ap-hello` entendu. Il envoie le paquet en *broadcast* avec un TTL fixé à $2 \cdot k_{cds} + 1$
2. Les dominés de D relaient le paquet en *broadcast*
3. Les autres dominés le relaient en unicast vers leur dominant afin d'optimiser le trafic de contrôle
4. Si un dominant reçoit la demande et a reçu un `ap-hello` de son propre père avec un numéro de séquence supérieur à celui demandé, il répond avec un `cds-invite`, relayé en unicast sur la route inverse

Finalement, chacun des dominants entendant un `cds-invite` peut inscrire la source comme père secondaire. Si un dominant choisi de se reconnecter à un père secondaire venant d'une découverte explicite, il envoie un `cds-accept`, forçant les dominés intermédiaires entre lui et la source du `cds-invite` à devenir dominants (agissant donc comme les invitations lors de la construction).

Le mécanisme de maintenance proactive des pères secondaires permet de créer des liens de secours dans la dorsale. Une reconnexion de la dorsale est donc possible sans latence et sans trafic de contrôle.

3.4.1.3 Cassure de dorsale

Si le médium radio est chargé, de nombreux paquets peuvent être perdus. Les demandes de reconnexion vont s'enchaîner, aggravant la surcharge réseau. Ainsi, un dominant pour lequel $max_{reconnect}$ tentatives de reconnexion ont échoué, ordonne la cassure de sa branche en envoyant un `cds-break` en multicast à ses fils et dominés. Un nœud possédant comme père la source du `cds-break` relaie le message en multicast aux autres nœuds concernés, puis prend l'état *isolé*, et attend une sollicitation extérieure afin d'initier le processus de reconstruction.

Nous pouvons remarquer qu'au moins un nœud isolé se trouve à au plus $k_{cds}+1$ sauts d'un dominant : un dominé est voisin de la zone des nœuds isolés, et ce dominé est à au plus k_{cds} sauts de son propre dominant.

Ainsi, un dominant s'apercevant qu'il possède dans sa table de voisinage un k_{cds}-voisin d'état isolé envoie un `cds-invite` avec un TTL de $k_{cds}+1$. De même un dominé voisin de son dominant et qui possède un voisin isolé à exactement k_{cds} sauts ordonne à son père d'envoyer un `cds-invite`. Un dominé recevant un `cds- invite` devient actif, en stockant la source comme père secondaire afin de se connecter plus tard s'il est élu dominant.

3.4.1.4 Réduction de la redondance

Afin d'éviter une redondance inutile dans la dorsale, et minimiser ainsi sa cardinalité, nous proposons un mécanisme de suppression des dominants inutiles. Un dominant est inutile si et seulement s'il ne possède que des dominés à au plus $k_{cds} - 1$ sauts et aucun fils dominant. Un tel dominant envoie à ses dominés en *broadcast* un `cds-useless`, les forçant à choisir son propre père comme nouveau dominant. Puis il prend l'état dominé en maintenant le même père.

D'autre part, un dominant D_1 peut optimiser la distance qui le sépare de l'AP : quand il reçoit d'un dominant D_2 un `ap-hello` de numéro de séquence supérieur au dernier `ap-hello` reçu de son père, et que D_2 est plus proche de l'AP que le père de D_1 (information tirée du TTL du `ap-hello`), alors D_1 choisit D_2 comme père. Ainsi, la hauteur du CDS est réduite, ce qui permet d'obtenir un CDS possédant plus de branches, et donc potentiellement, plus de dominants peuvent se déclarer inutiles. D'autre part, le diamètre du CDS étant également plus réduit, un paquet diffusé dans la dorsale a moins de risques de subir une collision et d'être perdu.

3.4.2 Maintenance des clusters

Un dominé ne participe pas à la maintenance des clusters. Lorsqu'un `hello` vient du nœud choisi comme relais vers son dominant, le dominé peut mettre à jour l'identifiant de son chef de cluster. Par contre, un dominant doit envoyer certaines informations additionnelles dans ses `hellos` : la distance de son clusterhead via la dorsale et le prochain saut via la dorsale pour l'atteindre.

3.4.2.1 Maintien d'un ensemble dominant

Un dominant D, possédant le relais vers son clusterhead R, considère son chef perdu si l'une des conditions suivantes est remplie :

- R n'est plus un voisin

- R possède un chef différent
- R annonce une distance à son clusterhead strictement supérieure à $(k_{cluster} - k_{cds} - 1)$ sauts

Si son relais R annonce un nouveau chef C à au plus $(k_{cluster} - k_{cds} - 1)$ sauts, alors D prend ce nouveau chef et met à jour la distance qui le sépare de C. D envoie un `hello` gratuit pour forcer les éventuels dominants l'ayant choisi auparavant comme relais vers leur chef à mettre à jour leurs informations et changer leur décision.

Un dominant D dont le chef C_1 n'est plus valide va tenter de se reconnecter. D cherche dans ses voisins virtuels un candidat remplissant l'une de ces conditions :

- un dominant D' est voisin, possède un chef C_2 différent de C_1 et annonce une distance à C_2 via la dorsale d'au plus $(k_{cluster} - k_{cds} - 1)$ sauts,
- un dominant D' est voisin, possède C_1 comme chef, annonce une distance à C_2 via la dorsale d'au plus $(k_{cluster} - k_{cds} - 1)$ sauts, et D n'est pas le relais vers C_1 pour D'.

D prendra donc ce nouveau chef et mettra à jour dans ses prochains `hellos` la distance qui le sépare de lui via la dorsale. Si un dominant ne peut rallier aucun cluster existant, alors il s'élit chef. Il annonce sa décision immédiatement avec un `hello` gratuit. Une telle maintenance est efficace car elle s'appuie sur la structure en arbre de la dorsale, limitant ainsi le problème de l'horizon infini inhérent à tout algorithme de type vecteur de distance.

Prenons l'exemple de la figure 3.5 : le clusterhead du nœud 2 disparaît. Le nœud 2 va donc rechercher un clusterhead candidat pour sa reconnexion. Le nœud 3 possède un clusterhead, mais 2 est le relais pour 3. Donc aucun candidat n'est disponible : 2 devient clusterhead et envoie un `hello` gratuit. 3 va donc automatiquement prendre le clusterhead de son ancien relais, et joindre son cluster.

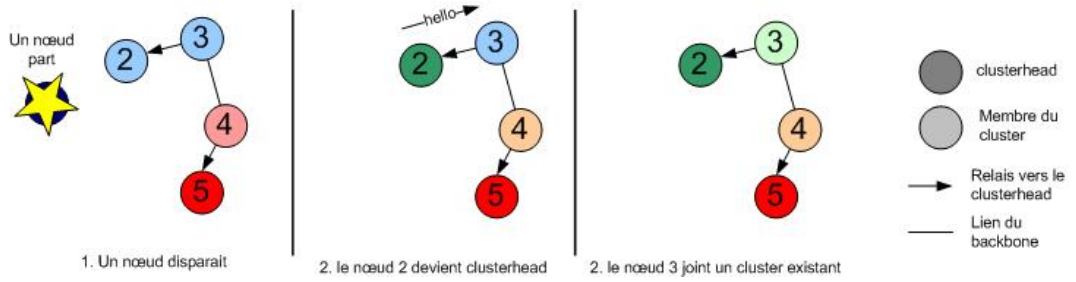


FIG. 3.5 – Maintenance d'un cluster : reconnexion $(k_{cluster} - k_{cds} = 2)$

3.4.2.2 Élimination de la redondance

Un clusterhead est inutile s'il ne possède aucun dominant voisin l'ayant choisi comme chef. Les clusters étant connexes, a fortiori aucun autre dominant dans le réseau ne peut l'avoir choisi. Un tel nœud inutile cherche un cluster existant à rallier. Si un tel cluster existe, il s'y attache en perdant son rôle de clusterhead.

3.4.2.3 Perspectives

Un prochain travail pourrait s'intéresser au remplacement des paquets `ap-hello` par un champ spécifiant la distance au leader dans les paquets `hellos`. Un dominant fixerait sa distance au leader comme la distance minimale entendue des dominants voisins, incrémentée d'une unité. Nous pourrions forcer un dominant à ne se reconnecter qu'à un père secondaire possédant une distance au leader inférieure à la sienne, évitant ainsi les boucles dans la dorsale. Cependant, la convergence pourrait être plus lente, un tel algorithme présentant le problème classique de l'horizon infini des algorithmes à vecteurs de distance. De plus, les reconnexions de dorsales tout en évitant la création de boucles peuvent être problématiques.

Voisinage (pour chaque voisin)	Dorsale	Clusters
Adresse	Père principal	Clusterhead
Poids	Pères secondaires	Relais vers le clusterhead
État (actif, dominant, dominé ou isolé)	Fils et dominés	Distance au clusterhead
Liste des voisins	Prochain saut vers le père (pour les dominés)	

TAB. 3.1 – Informations disponibles durant le fonctionnement du réseau, grâce au protocole de maintenance de la structure virtuelle

3.5 Interconnexion de dorsales dans un réseaux à leaders multiples

Dans un réseau hybride, les terminaux accèdent à Internet via un point d'accès. L'AP est donc un point névralgique, requérant une redondance d'équipements. Nous proposons donc simplement de construire une dorsale par point d'accès. Un nœud identifie l'AP qui le dessert grâce aux `ap-hellos` et `hellos`.

La maintenance de la dorsale est ensuite semblable. La seule différence réside dans le fait qu'un dominant puisse choisir de se reconnecter à un nœud de la même dorsale ayant reçu un `ap-hello` récemment, ou à un dominant d'une autre dorsale. Dans ce dernier cas, le dominant réalise un *handover* pour sa branche, changeant le point d'accès de tous ses descendants.

3.5.1 Interconnexion des dorsales

Les points d'accès sont interconnectés par le réseau filaire, constituant ainsi une racine commune de l'ensemble des dorsales. Cependant, une interconnexion des dorsales peut être opportune lorsque le chemin suivi par le paquet de contrôle est important. Ainsi, lorsqu'une requête de routes est envoyée, l'obligation de passer via son point d'accès lorsque la destination est desservie par un autre point d'accès peut être inadéquate (fig. 3.6 page suivante).

Nous proposons d'élire des dominés afin qu'ils relaient les messages de contrôle entre les dorsales. Cependant, le nombre de dominés élus pour une telle interconnexion doit être réduit, afin de limiter le trafic de contrôle généré. Si le chef de zone élit les dominés-connecteurs, un volume important de trafic de contrôle doit être échangé dans un cluster pour une telle élection. Ainsi, nous avons choisi que les dominants élisent leurs dominés-connecteurs à partir des `hellos` de leurs dominés, incluant les identifiants de dorsales entendus dans le 1-voisinage. Un dominant possède au plus un dominé-connecteur par dorsale voisine. Le trafic de contrôle additionnel consiste donc en l'ajout de quelques champs dans le paquet `hello`.

Lors d'une inondation, un dominant relaie un paquet de contrôle en multicast le long de la dorsale, et en unicast vers les dominés-connecteurs. Un dominant n'utilise qu'une seule transmission radio en *broadcast* pour transmettre à ses voisins de la dorsale et à ses dominés-connecteurs afin de réduire le nombre d'accès au médium. Un dominant peut choisir un dominé-connecteur à plus d'un saut. Il envoie donc le paquet de contrôle à un nœud relais, dont il tire l'identifiant de sa table de voisinage. Ce nœud relaie le paquet vers le dominé-connecteur, dont l'adresse est contenue dans le paquet. Enfin, le dominé-connecteur relaie à la dorsale dont l'identifiant est contenu dans le paquet. Cette dorsale pouvant se trouver à plus d'un saut, le paquet passe de nouveau par d'autres dominés.

3.5.2 Acquittements

Les inondations le long de la dorsale sont envoyées en multicast, correspondant au niveau MAC à un *broadcast*. Ainsi, un mécanisme d'acquittements passifs peut être mis en place le

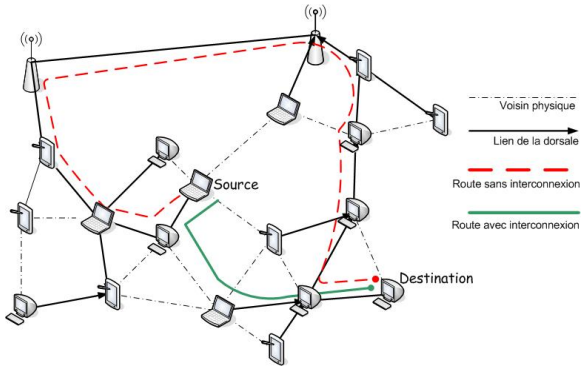


FIG. 3.6 – Schéma d'interconnexion de dorsales

long de la dorsale. Un dominant ayant envoyé une inondation scrute l'envoi de ses voisins de la dorsale. S'ils ont relayé le paquet (en *broadcast*), le dominant peut légitimement considérer la transmission comme réussie. Sinon, il retransmet le paquet d'inondation.

Cependant, un tel acquittement n'est pas possible pour les paquets entre les connecteurs, transmis en unicast. Un nœud peut donc choisir d'envoyer explicitement un paquet d'acquittement, de se mettre en mode *promiscuous* pour écouter la retransmission de ses paquets, ou de tirer des informations de la couche MAC pour la validation de la transmission au niveau radio.

3.6 Évaluation de performances de la structure virtuelle d'auto-organisation

Nous présentons dans cette partie les résultats de campagnes de simulation effectuées sous OPNET Modeler [14]. Les réseaux ad hoc ne préjugent pas de l'utilisation d'une couche MAC particulière : ils devraient être conçus de la façon la plus générique possible, et s'affranchir de telles contraintes. Si un réseau ad hoc devait être déployé de nos jours, il pourrait s'appuyer sur les technologies telles que le Bluetooth [2] ou IEEE 802.11 [1]. De même, rien n'interdit d'intégrer dans l'avenir une couche MAC différente, plus adaptée aux réseaux sans-fil multisauts. Cependant, IEEE 802.11 constitue actuellement le meilleur candidat pour servir de couche MAC : il représente de facto un standard pour les accès sans-fil à Internet, présentant un débit et une portée radio acceptables pour de nombreuses applications. En conclusion, nous avons utilisé l'environnement radio 802.11b fourni par OPNET en mode DCF et sans RTS/CTS [1] , avec une modulation de type Direct Sequence Spread Spectrum (DSSS). OPNET présente une granularité fine de la modélisation radio en utilisant un modèle basé sur le calcul du Signal To Noise Ratio (SNR). Cependant, nous avons utilisé une modélisation idéale de la propagation radio : aucun mécanisme d'évanouissement ou d'ombrage n'est modélisé.

Les nœuds se déplacent sur une surface de simulation carrée de dimension finie (par défaut 1900x1900m) suivant un modèle de mobilité. La plupart de nos résultats sont obtenus en utilisant le *Random Waypoint*, mais nous avons vérifié que d'autres modèles de mobilité n'impactent que légèrement les performances et exhibent les mêmes tendances. Un nœud possède une portée radio de 300m. Les résultats présentés ici sont tous calculés avec un intervalle de confiance de 95%. Nous considérons comme générique une vitesse de 5m.s^{-1}, 40 nœuds et un degré de 10. Un des nœuds de la simulation est statique, constituant le point d'accès, et agit donc en tant que leader naturel pour la dorsale.

[1] `Request-To-Send` et `Clear-To-Send` sont des paquets de IEEE 802.11 permettant de réserver le médium afin d'éviter le problème du nœud caché. Un tel problème survient lorsqu'un nœud N souhaite envoyer un paquet alors qu'un de ses voisins est en train de recevoir un paquet venant d'un nœud non entendu par N. Cependant, les RTS/CTS offrent un débit très dégradé, et ne résolvent que partiellement le problème. Le lecteur pourra se référer à [9] pour une description plus précise de cette problématique.

Les résultats détaillés visent à montrer la pertinence de l'organisation proposée. En particulier, nous nous intéressons à l'influence des paramètres k_{cds} et $k_{cluster}$ sur le nombre de nœuds participant à la vie du réseau et nous montrons la robustesse de la topologie virtuelle à la mobilité, au nombre de nœuds et au degré. Nous évaluons également le surcoût protocolaire en nombre de messages et nous prouvons la robustesse des structures à travers l'interconnexion de plusieurs épines dorsales virtuelles.

Nous avons comparé la structure virtuelle présentée ici (dénommée CDCL pour CDS-Clusters) avec la dorsale proposée par Wu & Li, utilisant la dernière formulation de l'algorithme car la plus efficace, décrite dans [16].

3.6.1 Modèles de mobilité

Il existe de nombreux modèles de mobilité permettant de simuler la mobilité des nœuds dans un réseau ad-hoc (cf. [7] pour en avoir un aperçu). Le Random Waypoint Mobility Model [11] est le modèle le plus utilisé par la communauté dans les simulations. Un nœud choisit aléatoirement une vitesse (direction et norme) et une destination. Il se rend à cette destination avec la vitesse tirée. Une fois à cette destination, il tire un temps de pause aléatoire uniformément réparti entre 0 et $Temps_{max}$. Puis il retire une nouvelle destination et vitesse. Un tel modèle présente des avantages certains en terme de simplicité d'implémentation. Cependant, des travaux [5] ont montré qu'un nœud possède une plus forte probabilité de se trouver proche du centre de la surface de simulation que les bords. Ainsi, de nombreux autres modèles de mobilité ont été proposés afin de simuler des comportements individuels ou de groupes, avec une trajectoire plus lissée, une densité de présence uniforme sur le plan... La figure 3.7 page suivante présente à titre d'illustration le motif des déplacements de différents modèles de mobilité.

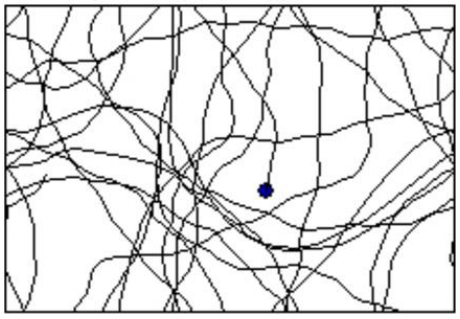
Cependant, les applications des réseaux ad-hoc étant actuellement peu développées, il nous est actuellement impossible de déterminer quel est le modèle de mobilité le plus réaliste. Si des traces de mobilité issues d'utilisations réelles étaient disponibles, nous pourrions évaluer quel modèle de mobilité s'approche le plus de telle application. Nous avons donc choisi d'utiliser dans nos simulations le Random Waypoint. Nous avons choisi de remplacer la pause par une vitesse maximale variable. Nous avons conduit les mêmes simulations avec différents modèles de mobilité, ce qui nous a permis de corroborer la relative indépendance des résultats vis à vis du modèle utilisé. Nous présenterons un peu plus loin dans ce chapitre des résultats obtenus avec le *Boundless Mobility Model*.

3.6.2 Paramétrisation

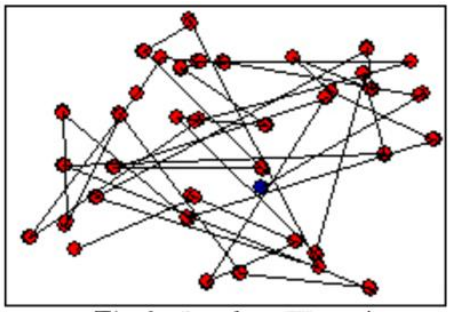
Nous avons dans un premier temps étudié l'évolution de la cardinalité de la dorsale et des clusters en fonction du rayon de dorsale et de cluster. Plus la distance maximale entre un nœud et la dorsale (k_{cds}) est grande, moins la dorsale comporte de dominants (fig. 3.8(a) page suivante). Nous pouvons également remarquer que notre algorithme demande quasiment autant de dominants que Wu et Li lorsque k_{cds}=1, mais demande moins de dominant dés que $k_{cds} \geq 2$. Si la distance maximale entre un dominé et la dorsale (k_{cds}) augmente, moins de dominants sont requis, ce qui est bien une des propriétés recherchées d'adaptation de la structure. De même, plus le rayon de cluster ($k_{cluster}$) est grand, moins le nombre de chefs de clusters est important (fig. 3.8(b) page suivante).

3.6.3 Impact de la mobilité

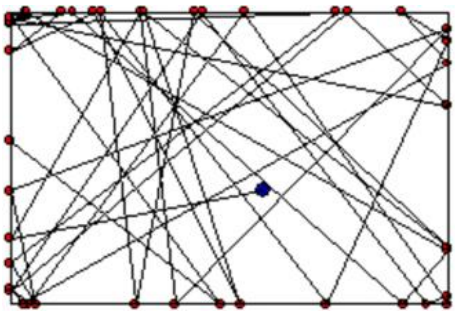
Nous avons ensuite mesuré l'impact de la mobilité sur la cardinalité de la structure et sa stabilité (fig. 3.8 page 45). La connexité représente une métrique importante pour une dorsale : elle reflète l'aptitude d'une structure à optimiser la diffusion avec une fiabilité maximale. Pour mesurer une telle propriété, nous n'avons gardé que les liens radio entre dominants accompagnés



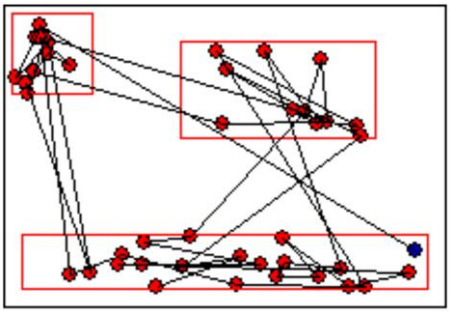
(a) Boundless [10] : un nœud choisit sa direction et sa vitesse selon une distribution exponentielle centrée sur les valeurs précédentes de ces paramètres. Par ailleurs, un nœud dépassant un bord de la simulation est ré-injecté sur le bord symétrique



(b) Random Waypoint [11] : un nœud choisit une destination et une vitesse aléatoirement. Lorsque la destination est atteinte, un nouveau tirage est effectué

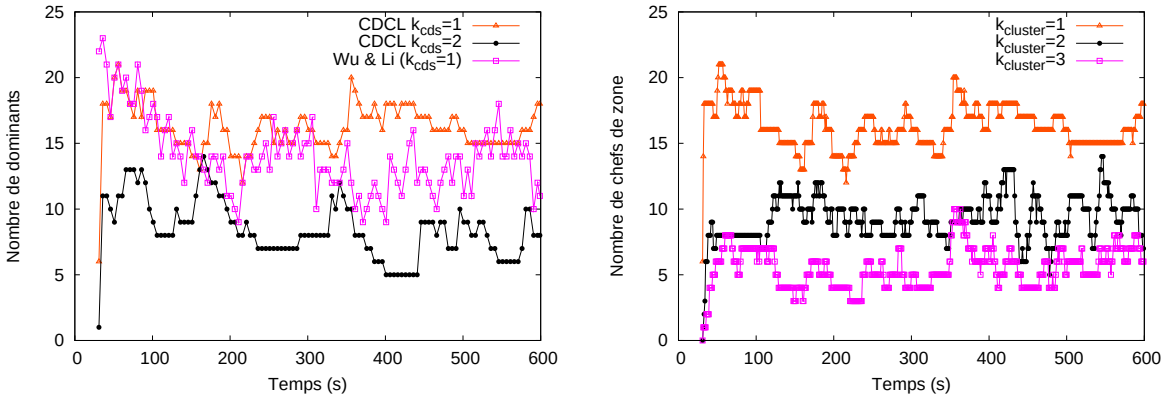


(c) Random Direction [15] : un nœud choisit une direction et une vitesse aléatoires. Lorsque le bord de la simulation est atteint, de nouvelles vitesse et direction sont tirées



(d) Restricted Random Waypoint [6] : un nœud exécute le random Waypoint Mobility Model, tout en restreignant l'espace dans lequel tirer sa destination

FIG. 3.7 – Motifs de déplacement d'un nœud pour différents modèles de mobilité



(a) Évolution de la cardinalité de la dorsale au cours du temps

(b) Évolution du nombre de clusters au cours du temps

des liens radio [dominés → père dans la dorsale] ou [dominés → relais vers le père]. Sur cette sous-topologie nous avons exécuté un algorithme du plus court chemin en partant du leader pour calculer le ratio de nœuds joignable via la dorsale. Cette métrique est mésurée périodiquement en cours d'exécution et moyennée à la fin de la simulation.

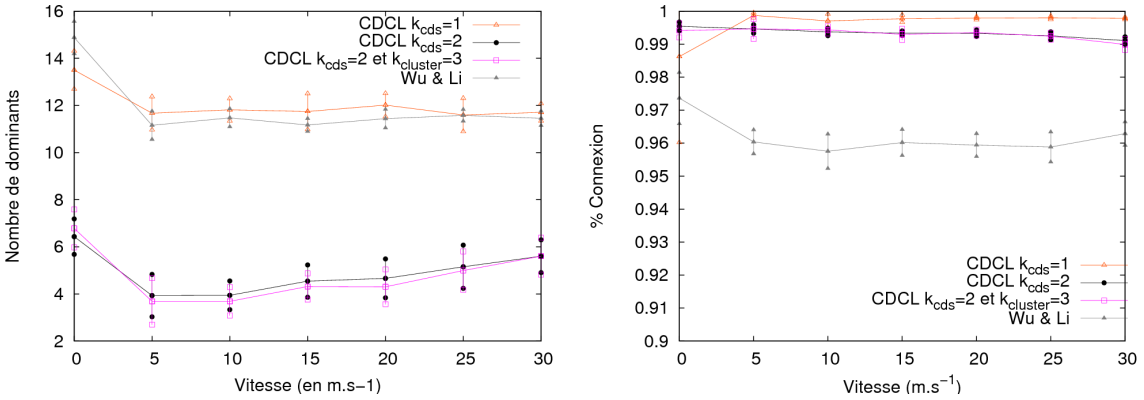


FIG. 3.8 – Impact de la mobilité sur la cardinalité et la connexité

Nous pouvons remarquer que la cardinalité de la dorsale est similaire pour l'algorithme de Wu & Li et notre algorithme avec k_{cds}=1. La cardinalité de notre dorsale diminue lorsque k_{cds} augmente : moins de dominants sont nécessaires. Lorsque la dorsale est configurée avec k_{cds}=1, elle reste connexe plus de 99% du temps, même à des vitesses élevées. **La connexité de la dorsale semble insensible à la mobilité**. Lorsque k_{cds} est plus élevé, moins de dominants sont élus, puisque la distance maximale entre un dominé et la dorsale est autorisée à augmenter. Cependant, la connexité diminue de pair, les reconnexions étant dans ce cas là plus complexes, la dorsale déconnectée devant passer par l'intermédiaire de plus de dominés pour se reconnecter. Ainsi, des valeurs élevées de k_{cds} sont plus adaptées pour un réseau faiblement mobile. D'autre part, nous voyons que l'algorithme de Wu & Li présente une connexité très inférieure : il suffit que des incohérences surgissent dans le voisinage pour potentiellement créer des paritions de la dorsale, diminuant la connexité moyenne. En effet, des décisions incohérentes peuvent être prises du fait du délai entre les envois de paquets `hellos`, mais également de possibles collisions. Dans un tel cas, chaque nœud ne possède pas la même vision du réseau. Or dans un réseau à densité faible, comme c'est ici le cas, une parition non négligeable de la dorsale peut survenir.

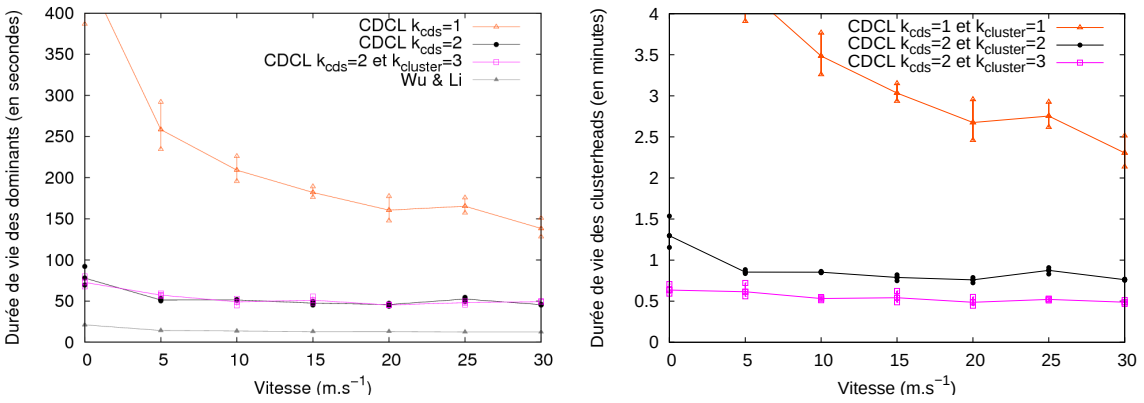


FIG. 3.9 – Persistance des dominants et chefs de cluster selon la mobilité

La persistance de la structure, i.e. le nombre de changements dans ses membres, est également un critère d'évaluation vital dans une structure virtuelle servant à l'auto-organisation (fig. 3.9). Si nous mesurons la durée de vie des dominants, le temps moyen pendant lequel un dominant

garde son rôle, nous pouvons voir qu'il diminue lorsque k_{cds} augmente. En effet, une cassure dans la dorsale crée plus de changements, puisque potentiellement plus de dominés doivent se colorier afin de rendre la dorsale de nouveau connexe. Cependant, même pour des mobilités élevées, **un dominant reste dominant pendant en moyenne plus de 2 minutes** lorsque k_{cds}=1. Lorsque la mobilité est nulle, la dorsale subit tout de même quelques changements : des collisions peuvent se produire, entraînant une vue fausse de la topologie, et obligeant ainsi un dominant à se reconnecter alors que la dorsale reste en réalité toujours valide. Nous pouvons également remarquer que **l'algorithme de Wu & Li présente une persistance très inférieure à nos algorithmes**. En effet, il n'a pas été conçu initialement pour exhiber une telle propriété, puisqu'aucune règle n'oblige un dominant à garder le même rôle. De même, nous pouvons remarquer que la persistance des clusterheads décroît lorsque $k_{cluster}$ augmente : le rayon de cluster étant plus grand, des incohérences ont plus de chance de se produire dans les tables de voisinage, multipliant les élections inutiles de clusterheads, puis leur retour à un état de client.

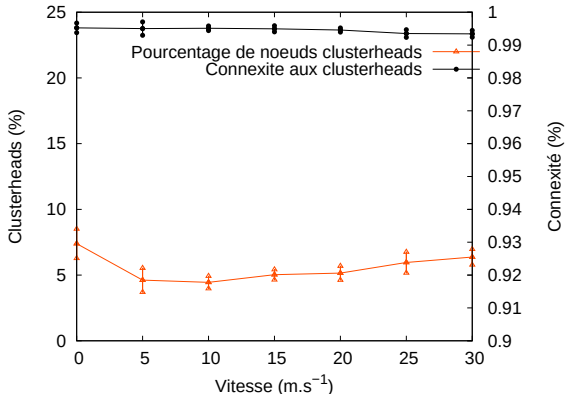


FIG. 3.10 – Impact de la mobilité sur les clusters : cardinalité et connexité (propriété de dominance des clusterheads)

Enfin, nous avons observé le comportement de la structure de clusters (fig. 3.10). Nous pouvons voir que la connexité reste indépendante de la vitesse : le clusterhead étant proche, les clients arrivent à maintenir un clusterhead valide. De même, **la cardinalité et la connexité sont insensibles à la mobilité**.

3.6.4 Impact du modèle de mobilité

Afin d'apprécier l'impact d'un modèle de mobilité différent, nous avons utilisé le boundless mobility model : un nœud choisit ses vitesse et direction suivant une loi exponentielle, centrée sur les précédentes valeurs. Ainsi, le déplacement semble plus régulier. Par ailleurs, un nœud dépassant les bords de la simulation est réinjecté sur la face opposée, simulant des apparitions/disparitions de nœuds.

Nous pouvons observer que la cardinalité suit les mêmes tendances que pour le random waypoint (fig. 3.11 page ci-contre), prouvant ainsi une certaine indépendance du modèle de mobilité. La connexité de notre structure semble un peu plus faible : le boundless mobility model est connu pour créer plus de changements de voisinage, impactant ainsi plus les performances. La connexité et la cardinalité de Wu & Li semblent augmenter : beaucoup de changements donnent une vue erronée du voisinage, surestimant le nombre de voisins. Ainsi, la probabilité d'être élu dominant augmente, créant de la redondance dans la dorsale, et donc une meilleur connexité. Cependant, la connexité de Wu & Li reste inférieure pour des mobilités faibles ou moyennes (inférieures à 20m.s^{-1}).

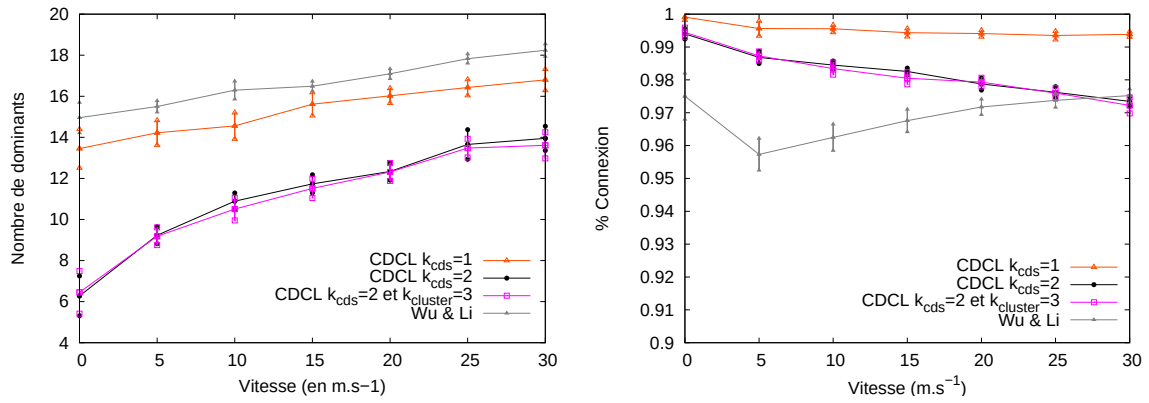


FIG. 3.11 – Impact de la mobilité sur la cardinalité et la connexité (les nœuds exécutant le boundless mobility model)

3.6.5 Densité

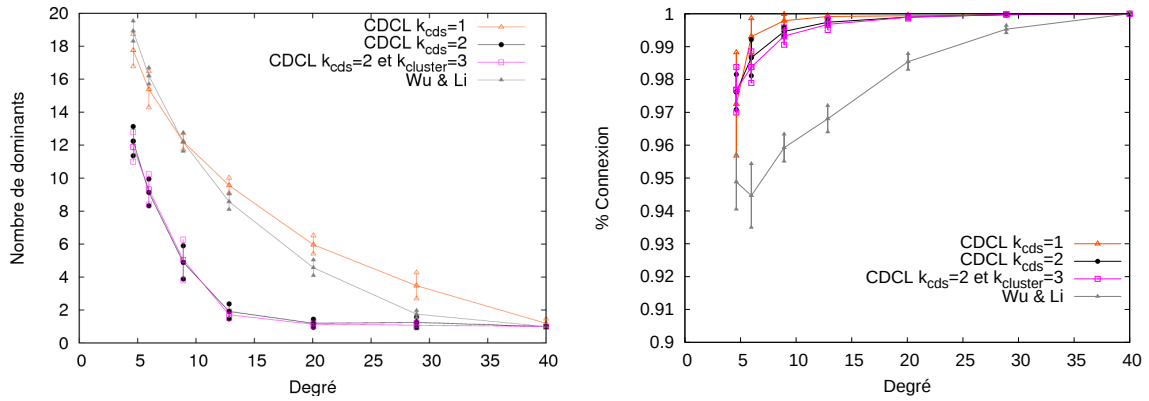


FIG. 3.12 – Impact du degré sur la cardinalité et la connexité de la dorsale

Nous mesurons dans cette série de simulations l'impact de la densité, i.e. nombre moyen de voisins par nœud, sur la topologie que nous proposons. Cependant, nous pensons qu'un protocole ne devrait s'exécuter que sur une topologie à degré limité. En effet, un algorithme de contrôle de topologie (cf. section 2.5.3 page 22) devrait être mis en place afin de limiter le degré en réduisant la portée radio. Un degré limité permet notamment d'économiser de l'énergie et d'augmenter la réutilisation spatiale du spectre radio (la portée étant réduite, les interférences le sont également). Il est donc peu probable qu'un réseau à forte densité (> 20) soit utilisé sans contrôle de topologie.

Nous avons dans un premier temps étudié l'impact du degré sur la dorsale (fig. 3.12). Nous pouvons observer que la cardinalité de la dorsale diminue lorsque le degré augmente, quel que soit l'algorithme utilisé. En effet, une densité plus importante implique qu'un dominant peut couvrir plus de dominés, diminuant ainsi le nombre global de dominants. La cardinalité de Wu & Li et de notre algorithme sont très proches pour k_{cds}=1. Nous pouvons par ailleurs remarquer que la connexité de notre dorsale reste très élevée pour des degrés suffisants. Si le degré est trop faible, il arrive soit que le réseau soit déconnecté, soit que la cassure d'un lien radio critique casse la dorsale et requière une reconnexion très loin du point de cassure, diminuant ainsi la connexité. Cependant, nous pouvons remarquer que Wu & Li présente une connexité plus faible, subissant les incohérences de voisinage. Nous avons également observé le comportement des clusters (fig. 3.13 page suivante). **La connexité reste toujours très élevée, et le nombre de clusters requis diminue lorsque la densité augmente**.

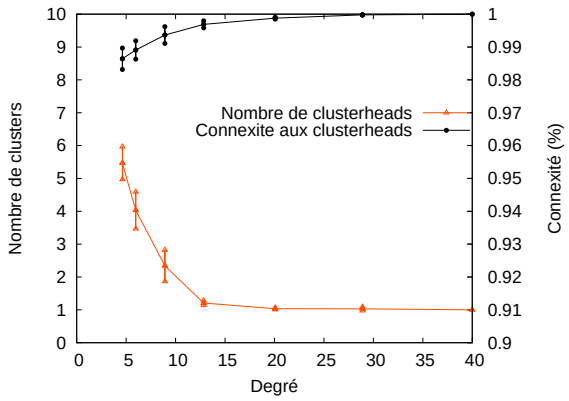


FIG. 3.13 – Impact du degré sur les clusters ($k_{cluster}$=3)

3.6.6 Passage à l'échelle horizontal

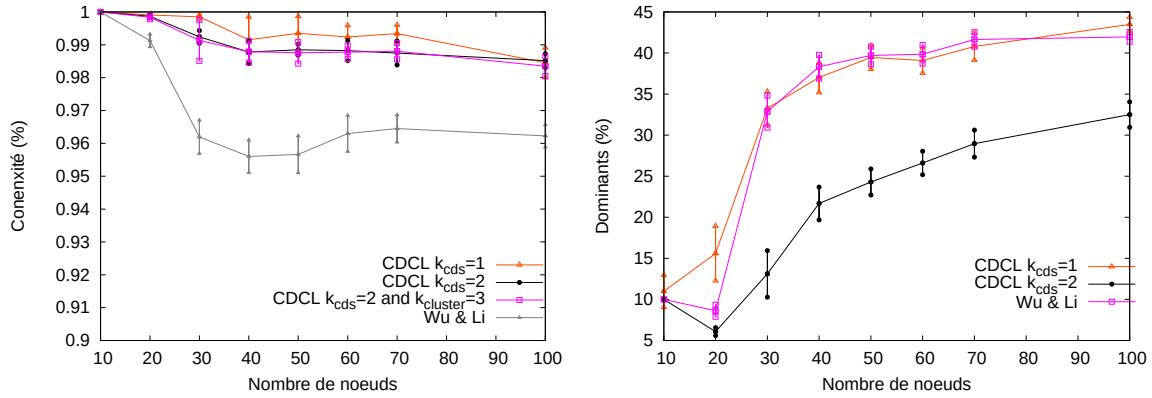


FIG. 3.14 – Impact du nombre de noeuds sur la cardinalité et la connexité

Nous avons ensuite évalué le passage à l'échelle horizontal de notre structure (fig. 3.14). **La connexité de CDCL est très stable : même avec 100 nœuds, la dorsale reste connexe plus de 98% du temps**. Un tel résultat augure donc de ses bonnes propriétés pour une utilisation par un protocole de routage ou de localisation. Par contre, la dorsale de Wu & Li semble moins connexe pour des cardinalités de réseau élevées. Nous pensons que des incohérences dans les tables de voisinage peuvent casser la dorsale en un point, et entraîner potentiellement la déconnexion d'une partie du réseau. Nous pouvons voir que la proportion de dominants augmente quand plus de nœuds participent au réseau. Nous pensons que cette augmentation, qui se stabilise pour les grandes cardinalités est un effet de bord : les bordures du réseau ne sont constituées que de dominés, les dominants étant plus proches du leader. Lorsque le nombre de nœuds augmente, une proportion moins importante de nœuds se trouve sur la bordure. Par ailleurs, nous vérifions bien qu'un k_{cds} plus grand permet d'obtenir une cardinalité de la dorsale plus faible.

3.6.7 Trafic de contrôle

Nous avons également mesuré le trafic de contrôle des différents protocoles et observé leur passage à l'échelle (fig. 3.15 page suivante). Nous pouvons vérifier la propriété de localisation parfaite de l'algorithme de Wu & Li : seul l'envoi périodique par chaque nœud d'un seul paquet `hello` est nécessaire. Ainsi, le trafic de contrôle de Wu & Li passe parfaitement à l'échelle. Notre structure virtuelle présente un trafic de contrôle plus élevé : des `hellos` doivent être

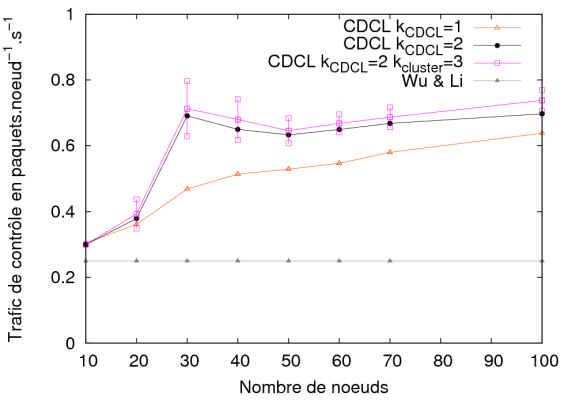


FIG. 3.15 – Trafic de contrôle

envoyés pour construire le k_{cds} voisinage, et des paquets explicites sont requis pour maintenir la connexité de la dorsale et des clusters. Cependant, nous pouvons voir que **CDCL passe à l'échelle horizontalement**, n'augmentant que très peu lorsque le nombre de nœuds augmente. De plus, le trafic de contrôle reste faible, inférieur à un paquet par nœud par seconde. Nous pensons donc que l'amélioration de la connexité, la forme particulière d'arbre de la dorsale, la maintenance conjointe d'une dorsale et de clusters, justifient ce léger surcoût en trafic de contrôle.

Protocole	Hellos	CDS	AP-Hellos	Clusters	Total
CDCL (k_{cds}=1, $k_{cluster}$=2)	0.25	0.1	0.16	0	0.51
CDCL (k_{cds}=2, $k_{cluster}$=3)	0.35	0.2	0.09	0.0016	0.64
Wu & Li	0.25	N/A	N/A	N/A	0.25

TAB. 3.2 – Trafic de contrôle par nœud par seconde dans un réseau de 40 nœuds

Le tableau 3.2 présente la répartition fine des sources du trafic de contrôle pour un réseau de 40 nœuds. Le trafic de contrôle de Wu & Li provient uniquement des `hellos`. Le trafic de CDCL provient en grande partie lui aussi des `hellos` (près de 50%). Le trafic de contrôle est plus élevé quand k_{cds} est plus grand : les paquets `hellos` doivent être propagés plus loin : à k_{cds} sauts durant le déroulement *normal*, et k_{cds}+1 sauts lorsque la source change d'état. La catégorie `CDS` correspond aux tentatives de reconnexion. Plus k_{cds} est important, plus les chemins de reconnexion sont longs et plus les inondations locales de reconnexion sont coûteuses. Ceci explique un trafic de contrôle croissant. Au contraire, les paquets de la catégorie `AP-Hello` ne peuvent être générés que par des dominants pour les détections de déconnexion du backbone. Donc le nombre de dominants étant moins élevé avec un k_{cds} élevé, le trafic de contrôle diminue. Enfin, la construction et la maintenance des clusters est largement négligeable : le deuxième niveau de hiérarchie semble *gratuit*.

Nous voyons que le trafic de contrôle pour la maintenance des clusters est négligeable, et que celui requis pour la maintenance de la dorsale reste acceptable. Ainsi, CDCL présente la construction d'une auto-organisation stable et complète pour un trafic de contrôle qui nous semble acceptable.

3.6.8 Interconnexion de dorsales avec AP multiples

Nous avons mesuré les performances de l'interconnexion de 2 dorsales dans un réseau hybride avec 2 AP. Dans ce but, un paquet est périodiquement inondé dans la dorsale, en empruntant les chemins de connexion entre dorsales. Nous avons comptabilisé séparément les premières transmissions, leur retransmission, et les acquittements.

la fiabilité d'une inondation est très élevée, puisqu'en moyenne plus de 97% des dominants reçoivent un paquet donné[1](tab. 3.3). Lorsque l'acquittement pour les paquets en unicast est activé, le trafic de contrôle augmente de façon logique (tab. 3.4). Cependant, l'inondation est majoritairement prise en charge par les dominants. Comme les dominants constituent des nœuds en moyenne plus forts (en énergie, en mémoire...), une telle inégalité de répartition constitue selon nous une propriété intéressante. De plus, le trafic de contrôle global se trouve réduit : avec k_{cds}=2 et sans acquittement, seuls 16,9 paquets sont en moyenne nécessaires pour inonder la dorsale. En comparaison, une inondation aveugle aurait demandé 40 paquets.

Rayon de dorsale (k_{cds})	acks	taux de livraison	Connexité dorsale
1	oui	97,2	99,3
	non	95,9	99,4
2	oui	97,4	97,5
	non	96,8	97,5

TAB. 3.3 – Inondation de la dorsale - Efficacité

k_{cds}	acks	Trafic de contrôle en nombre de paquets par inondation					
		Dominants			Dominés		
		paquets	retransmissions	acks	paquets	retransmissions	acks
1	oui	15	6,9	2,4	5,2	0,4	9,5
	non	14,2	4,3	0	4,2	0	0
2	oui	8,6	4,2	2,7	6,2	0,3	9,8
	non	8,8	2,3	0	5,6	0	0

TAB. 3.4 – Inondation de la dorsale - Trafic de contrôle

3.7 Conclusion

Dans ce chapitre, nous avons présenté une nouvelle structure virtuelle d'auto-organisation. Une dorsale permet de collecter le trafic de contrôle et de créer un premier niveau de hiérarchie dans le réseau. Au dessus de ce premier niveau, le réseau est découpé en zones, des *clusters*, représentant le deuxième niveau de hiérarchie de la structure virtuelle. La force de cette proposition réside d'une part dans l'intégration fine de ces deux structures virtuelles : le trafic de contrôle est réduit, et les deux structures agissent en symbiose (le clusterhead est élu parmi les membres de la dorsale). D'autre part, nous avons proposé des algorithmes de construction mais également de maintenance : il n'existait pas d'algorithme maintenant une dorsale en forme d'arbre. Enfin, cette dorsale a été conçue pour optimiser sa persistance : un nœud chef garde son rôle de chef pendant un laps de temps élevé. La hiérarchie changeant peu, elle est donc exploitable plus efficacement par un protocole de niveau supérieur, pour le routage ou la localisation par exemple.

Nous avons évalué les performances de cette structure d'auto-organisation via des simulations. La structure est paramétrable, le nombre de dominants et de clusterheads pouvant être ajusté grâce aux paramètres k_{cds} et $k_{cluster}$. De plus, la structure réagit bien aux changements de la topologie : les propriétés de dominance et de cardinalité restent valides même à des fortes mobilités. La dorsale reste par exemple connexe plus de 98% du temps, même à 30m.s^{-1}. Par ailleurs, es algorithmes proposés passent à l'échelle en nombre de nœuds, et l'impact de la densité est faible. Enfin, CDCL reste stable : les dominants et clusterheads gardent leur rôle pendant un

[1] A titre de comparaison, une inondation aveugle à une densité de 20 nœuds atteint un taux de livraison de 95% dans des simulations utilisant ns2 [4]). De plus, il est connu que la multiplication des inondations diminue le taux de livraison, la charge sur le médium radio devenant trop élevée[8].

laps de temps important. La structure de Wu & Li présente une faible stabilité : elle permet donc d'optimiser par exemple les inondations à un moindre coût. Au contraire, CDCL représente une structure candidate pour l'auto-organisation : la vue logique changeant peu, elle peut être exploitée de façon performante par les protocoles des couches réseau supérieures.

Pour toutes ces raisons, il nous semble qu'une telle structure peut être pleinement efficace pour des protocoles de niveau supérieur. Nous avons tout de même souhaité valider ces algorithmes via une étude plus fine : les simulations ne constituent qu'une première étape dans l'évaluation de performances. Nous pensons qu'une étude analytique des propriétés des algorithmes, notamment de leur caractéristiques auto-stabilisantes, permettrait de valider la structure présentée ici. Le chapitre suivant détaille cette partie de notre étude.

Bibliographie

[1] IEEE 802.11, local and metropolitan area networks - specific requirements part 11 : Wireless lan medium access control (mac) and physical layer (phy) specifications, 1999.

[2] IEEE 802.15.1, personal area networks - specific requirements, 2002.

[3] 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.

[4] M. Bani Yassein, M. Ould Khaoua, L. M. Mackenzie, and S. Papanastasiou. Improving the performance of probabilistic flooding in manets. In *International Workshop on Wireless Ad Hoc and Sensor Networks (IWWAN)*, London, UK, May 2005.

[5] C. Bettstetter and P. Resta, Giovanni Santi. The node distribution of the random waypoint mobility model for wireless ad hoc networks. *IEEE Transactions on Mobile Computing*, 2(3) :257–269, July-September 2003.

[6] L. Blazevic, S. Giordano, and J.-Y. Le Boudec. Self organized terminode routing simulation. In *International Workshop on Modeling Analysis and Simulation of Wireless and Mobile systems (MSWiM)*, pages 81–88, Roma, Italy, July 2001. ACM.

[7] 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.

[8] J. Cartigny, I. Francois, and D. Simplot. RNG relay subset flooding protocols in mobile ad-hoc networks. *International Journal on Foundations of Computer Science*, 14(2) :253–266, April 2003.

[9] D. Dhoutaut. *Etude du standard IEEE 802.11 dans le cadre des réseaux ad hoc : de la simulation à l'expérimentation.* PhD thesis, INSA Lyon, December 2003.

[10] Z. J. Haas. A new routing protocol for the reconfigurable wireless networks. In *International Conference on Universal Personal Communications (ICUPC)*, San Diego, USA, October 1997. IEEE.

[11] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, chapter 5, pages 153–181. Kluwer Academic Publishers, 1996.

[12] N. Malpani, J. L. Welch, and N. Vaidya. Leader election algorithms for mobile ad hoc networks. In *International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM)*, pages 96–103, New-York, USA, August 2000. ACM.

[13] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. In *International Conference on Mobile Computing and Networking (MOBICOM)*, pages 151–162, Seattle, USA, August 1999. ACM.

[14] OPNET Modeler. http://www.opnet.com (v8.1a).

[15] E. Royer, P. Melliar-Smith, and L. Moser. An analysis of the optimum node density for ad hoc mobile networks. In *International Conference in Communications (ICC)*, pages 857–861, Helsinki, Finland, June 2001. IEEE.

[16] J. Wu. An enhanced approach to determine a small forward node set based on multipoint relays. In *Vehicular Technology Conference (VTC Fall)*, pages 2774–2777, Orlando, USA, October 2003. IEEE.

Publications

Conférences internationales

[1] F. Theoleyre and F. Valois. A virtual structure for mobility management in hybrid networks. In *Wireless Communications and Networking Conference (WCNC)*, volume 5 of *1*, pages 1035–1040, Atlanta, USA, March 2004. IEEE.

[2] F. Theoleyre and F. Valois. Robustness and reliability for virtual topologies in wireless multihops access networks. In *Mediterranean Ad Hoc Networking Workshop (MedHocNet)*, pages 81–92, Bodrum, Turkey, June 2004. IFIP.

Conférence francophone

[3] F. Theoleyre and F. Valois. Topologie virtuelle pour une organisation des réseaux hybrides multisauts. In *Journees Doctorales Informatique et Reseaux (JDIR)*, Lannion, France, November 2004.

Rapport de recherche

[4] F. Theoleyre and F. Valois. Topologie virtuelle pour réseaux hybrides. Research Report 5035, INRIA, December 2003.

Logiciel

[5] A Virtual-Topology for ad-hoc networks, http://gforge.inria.fr/projects/topo-adhoc/

Chapitre 4

Propriétés de la structure d'auto-organisation

4.1 Introduction

Nous avons présenté dans le chapitre précédent une solution complète d'auto-organisation pour réseaux ad-hoc. Cette structure est constituée d'une dorsale combinée à un découpage en clusters du réseau, suivant tous deux les propriétés décrites dans la section 2.3 page 11. Nous avons pu vérifier au travers de simulations ses qualités en terme de robustesse, de stabilité, de cardinalité. Nous proposons dans ce chapitre d'en étudier plus finement les propriétés.

Une étude en terme de complexité constitue la première étape pour tout algorithme : la complexité en temps permet de refléter le temps de convergence, et la complexité en message le trafic de contrôle. Par ailleurs, nous avons vu que les réseaux ad-hoc présentent une topologie continuellement changeante. Or l'auto-stabilisation permet de prouver qu'un algorithme amène à un état valide, quels que soient les changements de topologie.

Nous proposons donc dans une première partie de détailler l'utilité de l'auto-stabilisation dans le domaine des réseaux ad-hoc. Ensuite, la section 4.3 présentera la complexité des algorithmes d'auto-organisation présentés dans le chapitre précédent. La section 4.4 démontrera les propriétés auto-stabilisantes des algorithmes proposés. La cardinalité de la dorsale sera également étudiée. Une étude des propriétés de convergence, de robustesse, et de stabilité de la structure à travers des simulations sera exposée en section 4.6. Enfin, la section 4.7 présentera une étude de l'impact du poids dans le processus d'élection que nous utilisons.

4.2 Notations

Nous introduisons ici les notations propres à ce chapitre venant compléter les notations décrites dans le chapitre 2.4.2 page 14 :

- $N'_k(u)$: l'ensemble des voisins virtuels à au plus k sauts. Un voisin virtuel de N est soit un père soit un fils de N dans le CDS
- $\Delta'_k(u)$: le nombre de k-voisins virtuels. Ainsi, $\Delta'_k(u) \leq \Delta_k(u)$
- D : l'ensemble des dominants, $|D|$ est la cardinalité de l'ensemble dominant
- C : l'ensemble des clusterheads.
- h_T : la hauteur de l'arbre T, i.e. la distance maximale en sauts d'un nœud de T à la racine
- $dominator(u)$: le père choisi par u, u étant un dominé. $dominator(u) \in N_{k_{cds}}(u)$
- $parent(u)$: le père choisi par u, u étant un dominant. $parent(u) \in N(u)$

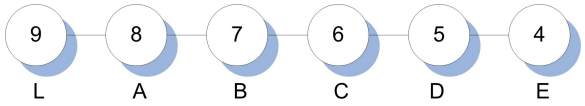


FIG. 4.1 – Réseau Linéaire de poids décroissant

4.3 Complexité

Nous proposons dans un premier temps d'étudier les complexités en temps et en message des différentes étapes des algorithmes que nous avons proposés. La complexité d'un algorithme vis à vis d'un paramètre dénombre et quantifie les facteurs influençant l'évolution asymptotique de ce paramètre. Ainsi, la complexité en temps recense les facteurs qui allongeront le temps de convergence de l'algorithme, et quantifie la part de chaque facteur dans cette évolution. Nous aurons ainsi le comportement asymptotique des algorithmes de construction et de maintenance.

4.3.1 CDS

Nous étudions ici les complexités en temps et messages des différentes phases des algorithmes de construction et de maintenance du CDS. Pour la complexité en temps, nous négligeons le temps de propagation et de traitement d'un paquet devant le temps nécessaire à une élection. Nous obtenons ainsi les complexités suivantes :

		En messages	En temps
Construction	totale	$O(n \cdot \Delta_{k_{cds}-1}) = O(n^2)$	$O(\lceil \frac{n}{k_{cds}+1} \rceil) = O(n)$
	étape 1	$O(n \cdot \Delta_{k_{cds}-1}) = O(n^2)$	$O(\lceil \frac{n}{k_{cds}+1} \rceil) = O(n)$
	étape 2	$O(n \cdot (2 \cdot k_{cds}+1)) = O(n^2)$	$O(\lceil \frac{n}{k_{cds}+1} \rceil) = O(n)$
Maintenance	hellos	$O(\Delta_{k_{cds}-1}) = O(n)$	$O(k_{cds}-1)$
	ap-hellos	$O(\|D\|)$	$O(H)$
	reconnexion d'un dominé	$O(1)$	$O(1)$
	reconnexion d'un dominant	$O((3 \cdot k_{cds}+1) \cdot \Delta_{k_{cds}+1}) = O(n)$	$O(k_{cds})$
	dominant inutile	$O(\Delta_{k_{cds}-1}) = O(n)$	$O(k_{cds}-1)$
	cassure de zone	$O(n)$	$O(h_{branche})$

4.3.1.1 Construction

Le processus de construction se déroule en 2 phases : la première permet d'élire un ensemble k_{cds}-dominant, la deuxième d'interconnecter ces dominants pour former un ensemble connexe. Nous allons donc étudier les complexités en temps et en messages des 2 étapes respectives de l'algorithme de construction.

Création de l'ensemble dominant Pour élire un ensemble k_{cds}-dominant, chaque nœud change au maximum 2 fois d'état :

- isolé → actif → dominant : pour tous les dominants devant passer par une phase d'élection
- isolé → dominé : pour les nœuds devenant directement dominé, un de leur k_{cds}-voisin étant dominant
- isolé → actif → dominé : pour les nœuds entrant en phase d'élection, mais qui ne sont pas élus dominants car ils ne possèdent pas un poids maximal localement

A chaque changement d'état, un nœud doit envoyer un paquet, relayé par les $k_{cds}-1$-voisins. La complexité en messages de cette première phase est donc en $O(n \cdot \Delta_{k_{cds}-1}) = O(n^2)$

La complexité en temps peut être étudiée sur l'exemple d'un réseau linéaire de poids décroissant (fig. 4.1 page précédente), pire cas pour le délai de convergence de cette phase de l'algorithme. Le leader L est placé à une extrémité de la ligne. Durant le premier round, k_{cds} nœuds vont donc devenir *actifs* et entrer en phase d'élection. Dans le pire des cas, le voisin A du leader devient *dominant*. Ses k_{cds}-voisins deviennent *dominés*, et tous les nœuds entre k_{cds} et $2.k_{cds}$ sauts de A deviennent *actifs*. A chaque round, k_{cds} nœuds deviennent *actifs* et un devient *dominant*. Ainsi, la complexité en temps est en $O(\left\lceil \frac{n}{k_{cds}+1} \right\rceil) = O(n)$)

Interconnexion Durant la phase d'interconnexion, chaque *dominant* envoie un `cds-invite` et un `cds-accept`. De même, chaque *dominé* relaie un `cds-invite` de TTL x sous condition qu'il ait relayé nb_{max}-1 `cds-invite` ou moins de TTL supérieur ou égal à x. Le TTL d'un `cds-invite` est au maximum de $2 \cdot k_{cds} + 1$, i.e. la distance maximale séparant deux *dominants*. Ainsi, un *dominé* relaie au maximum $nb_{max} \cdot (2 \cdot k_{cds} + 1)$ `cds-invites` (au plus nb_{max} `cds-invite` arrivant avec le TTL x). La complexité en messages est donc en $O(n \cdot (nb_{max} \cdot (2 \cdot k_{cds} + 1)) = O(n)$

Le pire cas en terme de délai d'interconnexion est là encore le réseau linéaire de nœuds de poids strictement croissant (fig. 4.1 page ci-contre). On cherche le nombre de rondes nécessaires à l'interconnexion, un round étant constitué par le rattachement de tous les *dominants* qui le peuvent à l'arbre déjà formé, étant à moins de $(2 \cdot k_{cds} + 1)$ sauts d'un *dominant* déjà connecté. Sur une ligne, un seul *dominant* peut se rattacher par round. Les dominants étant au minimum espacés de $(k_{cds} + 1)$ saut, la complexité en temps est donc en $O(\left\lceil \frac{n}{k_{cds}+1} \right\rceil) = O(n)$

4.3.1.2 Maintenance

La complexité en temps et messages de la maintenance du CDS dépend de la mobilité, du nombre moyen de voisins, de la topologie. Nous avons donc choisi de calculer indépendamment la complexité en messages des différents cas de maintenance du CDS. La complexité en temps est gouvernée par le temps de propagation et de traitement d'un paquet sur un saut.

hellos Un paquet hello est relayé en broadcast par tous les nœuds à moins de k_{cds}-1 sauts. La complexité en messages est donc en $O(\Delta_{k_{cds}-1})$ et celle en temps en $O(k_{cds} - 1)$.

Ap-hellos Les `ap-hellos` sont relayés seulement par les dominants. La complexité en messages est en $O(|D|)$ et celle en temps en $O(H)$[1].

Reconnexion d'un dominé Un *dominé* déconnecté cherche dans sa table de voisinage un nouveau point de reconnexion dans l'arbre, i.e. un dominant à moins de k_{cds} sauts de lui. La complexité en temps et en messages est donc nulle.

Reconnexion d'un dominant Un dominant déconnecté doit envoyer un paquet de reconnexion, `cds-reconnect`. Ce paquet est envoyé en broadcast par ses dominés. Le nombre de paquets générés est en $O(\Delta_{k_{cds}})$. Les autres dominés relaient ce `cds-reconnect` en unicast vers leur dominant, au maximum à k_{cds} sauts. Ce dernier peut répondre par un `cds-reply`, envoyé vers le demandeur, au maximum à $2 \cdot k_{cds} + 1$ sauts. La complexité en messages est donc en $O((\Delta_{k_{cds}+1}) \cdot (1 + 3 \cdot k_{cds})) = O(n)$.

De même, une reconnexion est relayée sur au maximum $2(k_{cds} + 1)$ sauts. La complexité en temps est donc en $O(k_{cds})$.

[1] Nous rappelons que D est l'ensemble des dominants, et H la hauteur de la dorsale

Étayage du CDS Un dominant peut se déclarer inutile s'il ne possède pas de dominé à exactement k_{cds} sauts, et s'il n'est le père d'aucun dominant. Il envoie un paquet `cds-useless`[1], relayé par ses dominés. La complexité en messages est donc en $O(\Delta_{k_{cds}-1} = O(n))$ et en temps en $O(k_{cds} - 1) = O(1)$.

Cassure de zone Un dominant peut initier la cassure de sa branche s'il n'arrive pas à se reconnecter. Soit $|branche|$ la cardinalité de cette branche, et $h(branche)$ sa hauteur. Un `cds-break` est envoyé par l'initiateur de la cassure, et relayé par l'ensemble de ses descendants. Un `cds-invite` est ensuite envoyé par un dominant connecté afin d'initier la reconstruction de cette branche morte. Dans le pire cas, il peut exister $O(n - |branche|)$ dominants connectés. La complexité en messages est donc en $O\left[|branche| + (n - |branche|) \cdot (2 \cdot k_{cds} + 1)\right] = O(n)$ et celle en temps en $O\left[h(branche)\right] = O(n)$.

4.3.2 Clusters

Nous présentons ici les complexités en temps et messages des algorithmes construisant et maintenant les clusters. Nous détaillons dans cette partie les résultats synthétisés comme suit :

	Messages	Temps
Construction	$O(\Delta'_{k_{cluster}})$	$O(\frac{\|D\|}{k_{cluster}})$
Maintenance	0	0

4.3.2.1 Construction

Chaque dominant envoie initialement un `cluster-hello` pour permettre à ses voisins de construire leur voisinage, et un `cluster-hello` pour notifier son choix de clusterhead. Chacun de ces messages est relayé par tous les dominants à moins de $k_{cluster}$ sauts via le CDS. Ainsi, la complexité en messages est en $O(\Delta'_{k_{cluster}})$.

Le temps de construction réside dans le temps passé aux élections, rendant négligeable le temps de traitement et de propagation d'un paquet. Le pire cas est une chaîne linéaire de dominants, de poids décroissant. Dans un round, un seul dominant à la fois peut devenir clusterhead, obligeant les dominants à moins de $k_{cluster}$ sauts de lui via le CDS à devenir ses clients. Ainsi, la complexité en temps est en $O(\frac{|D|}{k_{cluster}})$.

4.3.2.2 Maintenance

Un dominant déconnecté de son cluster recherche dans sa table de voisinage un cluster candidat. Si aucun n'existe, il s'élit lui même et forme son cluster. La complexité en temps et en messages est donc nulle, intégrées dans les `hellos`.

4.4 Propriété d'auto-stabilisation

Dans cette section, nous démontrons les propriétés auto-stabilisantes des algorithmes proposés dans le chapitre précédent. Nous allons démontrer que les algorithmes convergent bien vers un état stable et valide. La dorsale doit notamment respecter les propriétés suivantes :

- Connexité : un dominant possède un chemin constitué uniquement de dominants vers tout dominant quelconque du réseau
- k_{cds}-dominance : tout nœud doit être à au plus k_{cds} sauts d'un dominant

[1] voir la section 3.4.1.4 page 39

La structure de clusters doit, elle, respecter la propriété de $k_{cluster}$-dominance. Nous allons dans un premier temps définir de façon plus précise l'auto-stabilisation et expliquerons pourquoi une telle propriété est intéressante dans les réseaux ad hoc. Suivra la démonstration de l'auto-stabilisation de l'ensemble de l'algorithme présenté dans le chapitre précédent.

4.4.1 De l'utilité d'un algorithme auto-stabilisant

L'auto-stabilisation a été définie la première fois par Dijkstra [5] :

> un système est auto-stabilisant si quel que soit son état initial, il est garanti d'arriver à un état légitime en un nombre fini d'étapes.

Une des premières applications de ce domaine se trouve dans la tolérance aux fautes [12]. Un tel algorithme permet de constamment s'adapter à un changement et converge vers un état légitime sans réinitialisation. L'auto-stabilisation permet notamment de supprimer les incohérences et peut être obtenue en effectuant par exemple de la réplication (bases de données distribuées), de la correction d'erreur (codes correcteur d'erreur de type FEC). Naturellement, il faut que toutes les briques de l'architecture présentent la même propriété d'auto-stabilisation : l'algorithme, l'implémentation et le système d'exploitation sur lequel le code s'exécute. Si une des briques du système présente des problèmes de convergence, tout le système est mis en péril. Dans de telles conditions, un protocole de réseau auto-stabilisant est garanti de réagir positivement aux fautes, ne les rendant que temporaires.

L'auto-stabilisation prend naturellement une importance fondamentale dans les réseaux ad-hoc : un algorithme doit converger rapidement dans un environnement volatile, ou doit pouvoir converger même en présence de fautes passagères, lorsque par exemple un terminal s'éteint. [8] présente un protocole de routage multicast auto-stabilisant pour les réseaux ad-hoc : l'algorithme des auteurs permet de maintenir un Minimum Spanning Tree dont la racine est le point de rendez-vous du protocole multicast. Les poids minimum et maximum des arêtes sur les chemins vers la racine sont propagés de proche en proche, permettant à chaque nœud de distinguer localement ses voisins dans le MST. L'arbre étant construit, les auteurs étayent les branches sans récepteurs. Bien que le protocole ne soit pas évalué en terme d'efficacité (fiabilité, résistance à la charge...), l'auto-stabilisation présente un avantage certain. De même, [2] présente un algorithme auto-stabilisant de construction de Spanning Tree dans les réseaux sans-fil, en se basant lui sur les marches aléatoires et la répartition d'agents dans le réseau.

[7] traite de la localisation de caches dans les réseaux ad-hoc. Dans un premier temps, les auteurs proposent de construire un ensemble indépendant de nœuds, puis de les interconnecter, s'inspirant de [1] présenté dans la section 2.5 page 15. De même, les auteurs proposent une version auto-stabilisante de l'algorithme localisé de construction de CDS de Wu & Li [13]. Les auteurs ajoutent des règles d'élimination de redondance et de changement d'état sur erreur afin d'améliorer le temps de convergence des algorithmes modifiés. Le réseau logique est ensuite utilisé pour la diffusion de requêtes auprès des caches dispersés dans le réseau.

Enfin, l'auto-stabilisation permet de prouver qu'un algorithme converge en un temps fini, mais ne permet par contre pas d'estimer de façon quantitative la vitesse de convergence. Il peut donc être utile soit de borner analytiquement ce temps si cela est possible, soit de corroborer les résultats précédents par une étude utilisant la simulation.

4.4.2 Hypothèses

Afin de prouver que les algorithmes sont auto-stabilisants, nous avons posé quelques hypothèses dans notre modélisation. Il est avant tout nécessaire de modéliser la mobilité. Aucune hypothèse n'est faite sur le type de changement de topologie. Cependant, nous supposons que la topologie est stable suffisamment longtemps pour que les algorithmes convergent :

Hypothèse 4.1 *Nous supposons que la topologie est stable après une série de changements élémentaires de topologie, i.e. l'addition ou la suppression d'un noeud ou d'un lien radio. Le temps entre deux séries de changements de topologie est suffisamment grand pour que les algorithmes convergent.*

Nous verrons que les résultats de simulations mettant en oeuvre des modèles de mobilité permettent de corroborer cette hypothèse. De plus, nous pouvons également remarquer que si cette hypothèse n'est pas vérifiée, le temps entre deux séries de changements de topologie peut être arbitrairement petit, ce qui, selon nous, rend impossible la convergence de tout algorithme distribué.

Puisque nous modélisons un réseau radio, nous supposons que la perte d'un paquet est possible. Comme nous pourrons le voir, la création d'un ensemble dominant pour la dorsale et les clusters met en oeuvre un envoi périodique de paquets. Il en est également de même pour les algorithmes de maintenance. Ainsi, l'hypothèse suivante est faite :

Hypothèse 4.2 *Si une série d'au plus x paquets est envoyée sur un lien radio (A,B), au moins un paquet est acheminé sans erreur ni collision.*

Cependant, nous pourrons remarquer par ailleurs que la connexion de la dorsale lors de la construction n'est pas un processus périodique. Ainsi, aucune reprise sur erreur n'est prévue. Cependant, s'il existe plusieurs chemins de connexion dans le réseau, les algorithmes de construction convergeront. Si tel n'est pas le cas, les algorithmes de maintenance permettront de prendre en charge l'erreur puisqu'ils sont auto-stabilisants.

Enfin, nous supposons que le réseau de communication est connecté :

Hypothèse 4.3 *le graphe sous-jacent au réseau est connexe.*

4.4.3 Démarche générale

Pour démontrer qu'un algorithme est auto-stabilisant, il est nécessaire de démontrer deux propriétés fondamentales :

- Terminaison : l'algorithme converge, i.e. au bout d'un temps fini, l'algorithme maintient un état inchangeant au cours du temps.
- Validité : l'état à la terminaison est un état légal, i.e. toutes les propriétés requises sont vérifiées (par exemple l'unicité pour une attribution d'adresse, l'existence de tous les chemins pour un algorithme de routage...)

4.4.4 CDS

Nous allons démontrer les propriétés de terminaison et de validité, dans un premier temps pour les algorithmes de construction puis pour les algorithmes de maintenance.

4.4.4.1 Construction

Création d'un ensemble k_{cds}-dominant

Théorème 4.1 *L'algorithme de la première phase se termine et construit un ensemble k_{cds}-dominant*

Terminaison

Lemme 4.1 *Tous les sommets possèdent un état dominant ou dominé à la fin de la première phase de l'algorithme.*

Preuve : Démontrons le par l'absurde :

• Supposons qu'il existe un sommet I isolé. Supposons qu'il existe un sommet non isolé N dans la composante connexe incluant I. Soit $c = \langle N, c_1, c_2, ..., I\rangle$ un chemin de I à N. Tous les k_{cds}-voisins de I sont isolé, sinon I aurait changé d'état et ne serait plus isolé. Ainsi, $\{c_1, ..., c_{k_{cds}}\}$ sont isolé. De même, si $\{c_{i\cdot k_{cds}+1}, ..., c_{(i+1)\cdot k_{cds}}\}$ sont isolé, alors $\{c_{k_{cds}(i+1)+1}, \ldots, c_{k_{cds}(i+2)}\}$ le sont aussi. Ainsi, N doit être également isolé. La composante connexe est entièrement formée de sommets isolés. Or le leader au moins n'est pas isolé. Ceci amène à une contradiction.

• Supposons qu'il existe un sommet N actif. Si un k_{cds}-voisin est dominant, N serait dominé. De même, si tous les k_{cds}-voisins sont dominés, N serait dominant. S'il est actif de plus fort poids dans son k_{cds}-voisinage de sommets actifs, alors N est élu dominant au bout de Δ temps au maximum. Donc, il existe un actif A_1 à moins de k_{cds} sauts, possédant un poids plus élevé.

Soit le graphe G'_k tel que ses sommets sont les sommets actifs de G au round k, et tel qu'il existe un arc d'un sommet A_i vers un sommet A_j si et seulement si $w(A_i) < w(A_j)$. G'_k est acyclique et possède une cardinalité finie, inférieure ou égale à n. La deuxième propriété est triviale. Démontrons l'absence de cycle en raisonnant par l'absurde. Soit $c = \langle c_0, c_1, ..., c_k\rangle$ un cycle dans G'_k. Un arc existe de c_i vers c_{i+1}, i.e. $w(c_i) < w(c_{i+1})$ avec $i \in [1..k-1]$. Donc transitivement, $w(c_0) < w(c_k)$. Or c est un cycle, donc un arc existe de c_k vers c_0, d'où $w(c_k) < w(c_O)$, ce qui amène une contradiction.

Le graphe G'_k contient au moins un puits A_k, i.e. un sommet possédant un degré sortant nul. Au bout d'un temps fini Δ, A_k sera élu dominant, et ses voisins deviendront dominés. Soit I_k l'ensemble des sommets isolé de G au round k. Au round k, un sommet au moins, a_k, de G'_k devient dominant, donc $a_k \notin G'_{k+1} \cup I_{k+1}$. Parallèlement, certains sommets de W_k sont retirés de W_k et ajoutés soit à G'_k, soit à $G - W_k \cup G'_k$, d'où :

$$\begin{aligned} & |W_k| + |G'_k| \geq |W_{k+1}| + |G'_{k+1}| + 1 \\ \Rightarrow \quad & |W_k| + |G'_k| \succ |W_{k+1}| + |G'_{k+1}| \\ \Rightarrow \quad & \lim_{i\to\infty} |G'_i| = 0 \end{aligned}$$

Ainsi, l'algorithme convergera à la fin de la phase 1 vers un graphe ne comportant aucun sommet actif. □

Remarque 4.1 *Si de nombreux changements de topologie surviennent (contredisant l'hypothèse 4.1 page précédente), des incohérences peuvent survenir dans les tables de voisinage. De la même manière, un pirate avec un poids arbitraire pourrait bloquer l'élection. En conséquence, un nœud actif devient automatiquement dominant s'il est actif pendant plus de $\Delta_{election} \cdot max_{election}$ secondes. L'algorithme converge plus rapidement, mais plus de dominants seront élus. Cette redondance sera éliminée ultérieurement par les algorithmes de maintenance.*

Validité Nous allons maintenant démontrer que l'état dans lequel converge l'algorithme est un état valide.

Lemme 4.2 *Tout sommet est à une distance d'au plus k_{cds} d'un autre sommet dominant, ou est lui même dominant, i.e. le graphe des dominants forme un ensemble k_{cds}-dominant.*

Preuve : La preuve découle directement du lemme 4.1 d'après lequel il n'existe à la fin de la première phase que des dominés ou dominants :

- Un dominé a pris cet état par définition si un dominant est à moins de k_{cds} sauts.
- Un sommet élu dominant reste dominant.

□

Formation d'un k_{cds}-CDS

Théorème 4.2 *L'ensemble des dominants forme à la fin de la construction un ensemble connexe de k_{cds}-dominance, i.e. un k_{cds}-CDS.*

Terminaison

Lemme 4.3 *Tous les sommets possèdent un état dominant ou dominé à la fin de la deuxième phase de l'algorithme, et l'émission de* `cds-invite` *stoppe à la terminaison.*

Preuve : Une dominant reste dominant, et un dominé ne peut changer qu'en dominant. La première partie de la proposition est donc triviale. De plus, un dominant n'envoie qu'un seul `cds-invite` lorsqu'il devient connecté. Enfin, un dominé ne relaie qu'un nombre fini de `cds-invite`. La deuxième partie de la proposition est donc également vérifiée. □

Validité

Propriété 4.1 *Soit c un chemin entre deux dominants D_1 et D_k. c suit la propriété 4.1 s'il comprend une suite de i dominants, i quelconque, éloignés consécutivement l'un de l'autre d'au plus $2 \cdot k_{cds}$ dominés : $\exists c = \langle D_1, d_1, ..., d_i, D_2, d_{i+1}, ..., d_j, D_3, d_{j+1}, ..., D_k \rangle$ tel que les d_l sont des dominés, et que $dist(D_i, D_{i+1}) \leq 2 \cdot k_{cds} + 1$.*

Lemme 4.4 *Il existe à la fin de la première phase de l'algorithme un chemin c suivant la propriété 4.1, liant tout dominant au leader $\mathcal{L}$.*

Preuve : Soit D_k l'ensemble des dominants élus au round k ou avant. $D_0 = \{\mathcal{L}\}$. D_0 ne possède donc que des dominants répondant trivialement à la propriété 4.1.

Supposons que D_k réponde à la propriété 4.1. A la fin du round $k-1$, un ensemble S de sommets a été élu dominants, de telle sorte que $S \cup D_{k-1} = D_k$ et $S \cap D_{k-1} = \emptyset$. Un nœud N de S_{k-1} est actif durant le round $k-1$ avant d'être élu dominant. Soit $c_1 =< N, a_1, ...a_i, d >$ le chemin de N au plus proche dominé d, durant le round $k-1$. N étant actif, par construction, $|c_1| \leq k_{cds} + 1$. De plus, les $\{a_l\}$ ne sont pas, par définition, des dominés, et sont à au plus k_{cds} sauts de d, un dominé. En conséquence, les $\{a_l\}$ sont actifs. Comme N est élu dominant, les $\{a_l\}$ deviendront ses dominés à la fin du round. Soit $c_2 =< d, d_1, ...d_i, D >$ du dominé d à son père, D. Par définition, $D \in D_k$, $|c_2| \leq k_{cds}$ et les $\{d_l\}$ sont dominés. Comme $D \in D_k$, soit $c_3 =< D, ..., \mathcal{L} >$ le chemin qui l'unit au leader, suivant la propriété 4.1. Clairement, la concaténation de $c_1 \cdot c_2 \cdot c_3$ suit la propriété 4.1 à la fin de la première phase de l'algorithme. □

Lemme 4.5 *Si la propriété 4.1 est respectée à la fin de la première phase de l'algorithme de construction, l'algorithme construira au final un ensemble connnexe de dominants à la fin de la seconde phase.*

Preuve : Soit $\mathcal{D}_i$ l'ensemble des dominants tel que pour tout dominant D appartenant à $\mathcal{D}_i$, le chemin c unissant D au leader selon la propriété 4.1 comporte au plus i dominants. $\mathcal{D}_0 = \{\mathcal{L}\}$. $\mathcal{D}_0$ forme un ensemble de dominants connexe. Il enverra selon l'algorithme de construction un `cds-invite` avec un TTL de $2 \cdot k_{cds} + 1$.

Supposons que l'ensemble $\mathcal{D}_i$ forme un ensemble connexe de dominants à la fin de la première phase. Soit un sommet dominant $u \in \mathcal{D}_{i+1}$, et c le chemin vers $\mathcal{L}$ qui suit la propriété 4.1. $c = \langle u, v_1, ..., v_k, \mathcal{L} \rangle$. D'après le lemme 4.4, il existe un dominant v_i appartenant à c, à au plus $2k_{cds}+1$ sauts de u, puisque c suit la propriété 4.1. v_i lui même possède le chemin $c' \subset c$ quit suit la propriété 4.1. De plus, $v_i \in \mathcal{D}_i$. Ainsi, v_i enverra un `cds-invite` avec un TTL de $2k_{cds}+1$. u recevra donc le `cds-invite`, et se connectera à $\mathcal{D}_i$. D'où $u \in \mathcal{D}_{i+1}$. $\mathcal{D}_{i+1}$ forme bien un ensemble connexe de dominants. □

Formation d'un arbre

Définition 4.1 *Soit le graphe $\mathcal{G}_{CDS}$ contenant tous les sommets de G, et tel qu'une arête existe d'un sommet u vers un sommet v ssi v est le père de u si u est dominant, ou ssi v est le relais vers son père si u est dominé.*

Théorème 4.3 *Suivant la précédente définition, $\mathcal{G}_{CDS}$ est un arbre*

Preuve : Nous découpons la preuve en 2 parties : l'arbre des dominants, puis les dominés constituant les feuilles :

• Suivant la définition précédente de $\mathcal{D}_i$, $\mathcal{D}_0 = \{\mathcal{L}\}$ est un arbre trivial formé d'un singleton.

Supposons que $\mathcal{D}_i$ forme un arbre. $\mathcal{D}_i$ possède donc $|\mathcal{D}_i - 1|$ arcs. Soit $u \in \mathcal{D}_{i+1}/\mathcal{D}_i$. u va s'interconnecter au CDS via le `cds-invite` envoyé par un dominant appartenant à $\mathcal{D}_i$. Soit v ce dominant. Le chemin c $\langle u, u_1, ..., u_k, v\rangle$ ne comprend que des dominés, sinon, le `cds-invite` ne serait pas relayé par un dominant et conséquemment ne serait pas reçu par u. Ainsi, les dominés vont devenir dominants. Nous ajoutons donc à $\mathcal{D}_i$ une branche à k dominés et un dominant, comportant k liens d'un dominé vers son nouveau père, et un lien de l'ancien dominant à son nouveau père. Ainsi, $\mathcal{D}_i \cup \{u\} \cup \{u_i\}_{i\in[1..k]}$ comporte $|\mathcal{D}_i| - 1 + 1 + k$ arcs, i.e. $\left|\mathcal{D}_i \cup \{u\} \cup \{u_i\}_{i\in[1..k]}\right| - 1$ arcs. Par conséquent, $\mathcal{D}_{i+1}$ reste un arbre.

• Soit d_i l'ensemble des dominés à i sauts ou moins de leur père. Lorsque j'ajoute un sommet de d_0 à $\mathcal{D}$, j'ajoute le sommet et l'arc vers son père. Ainsi, $d_0 \cup \mathcal{D}$ reste un arbre.

Soit $d_i \cup \mathcal{D}$ formant un arbre. Soit un dominé $u \in d_{i+1}$. u choisit un père et un relais r vers ce père. r est d'un saut plus proche du père. Donc $r \in d_i$. Ainsi, j'ajoute un seul arc et un seul sommet. $d_i \cup \mathcal{D}$ est donc un arbre. Le graphe étant de cardinalité finie, $\lim_{i\to\infty} d_i \cup \mathcal{D} = G$. Le CDS forme donc bien finalement un arbre. □

Remarque 4.2 *Un dominé d est à au plus k_{cds} sauts de son père dans l'arbre. Un chemin $p =< d, d_1, \ldots, d_i, dominator(d) >$ existe, de telle sorte que tous les d_i sont dominés et possèdent le même père dans l'arbre, i.e. $\forall i \in [1..k-1], dominator(d_i) = dominator(d)$.*

Preuve : Un dominé d possède un père valide, $dominator(d)$ ssi un voisin N existe tel que :

- $dominator(N)$ est un voisin bidirectionnel dans la table de voisinage de d.
- N est un voisin bidirectionnel de d (redondant avec la précédente condition, mais minimisant l'impact des incohérences de tables de voisinage).
- N est à au plus $k_{cds} - 1$ sauts de $dominator(N)$ (information tirée des paquets `hellos`).
- $dominator(d) = dominator(N)$

La remarque suit trivialement. □

4.4.4.2 Maintenance

Durant la maintenance, un changement dans la topologie de la dorsale n'intervient que si un changement dans la topologie radio est intervenu. La propriété de terminaison de ces algorithmes est triviale, comme le lecteur pourra le vérifier. Il reste cependant à vérifier les propriété de validité (dominance, connexité, arbre).

Dominance

Théorème 4.4 *Un sommet dominé possède toujours un dominant à une distance d'au plus k_{cds}, i.e. le CDS forme un ensemble k_{cds}-dominant*

Preuve : Nous supposons que la topologie est stable après quelques changements, laissant assez de temps à l'algorithme de maintenance pour converger. Les dominés ayant un dominant voisin le choisissent comme dominant. Ce père est valide. Supposons qu'un ensemble de dominés à i sauts de leur dominant ait un père valide. Un dominé à $i+1$ sauts de son dominant l'a choisi car il est à moins de k_{cds} sauts via un autre dominé ayant choisi le même dominant, mais à i sauts, avec $i < k_{cds}$. Donc, comme le père des dominés à i sauts de leur dominant est valide, chaque dominé qui choisit un dominant possède un père valide.

D'autre part, un dominé n'ayant aucun dominant possible candidat dans sa table de voisinage devient actif et entre en élection pour devenir potentiellement dominant. Un sommet actif

redevient dominé ssi il trouve un dominant valide. Les sommets actifs devenant dominants exécutent eux la maintenance réservée aux dominants. Ainsi, tout dominé possède un dominant à moins de k_{cds} sauts. □

Connexité

Théorème 4.5 *L'ensemble des dominants forme un arbre connexe*

Lemme 4.6 *L'ensemble des dominants reste un arbre (connexe) lorsque la topologie radio est stable*

Preuve : Supposons que la topologie est stable. Chaque dominant reçoit un `ap-hello`, en maintenant la source comme père. Soit $\mathcal{D}_i$ l'ensemble des dominants à i sauts du leader (racine du CDS) via un chemin de dominants. $\mathcal{D}_i$ est supposé connexe. Les sommets de $\mathcal{D}_{i+1}/\mathcal{D}_i$ choisissent un père dans $\mathcal{D}_i$ puisqu'ils reçoivent de leur père l'`ap-hello`, et donc qu'ils sont à un saut de plus du leader. $\mathcal{D}_{i+1}$ est donc connexe.

Supposons $\mathcal{D}_i$ sans cycle. Soit E_i l'ensemble des arêtes de $\mathcal{D}_i$, et V_i l'ensemble de ses sommets. Nous pouvons établir que $|E_i| = |V_i - 1|$. Pour chaque sommet de $\mathcal{D}_{i+1}/\mathcal{D}_i$, on ajoute un sommet dans E_i et une arête dans V_i. D'où :

$$|E_{i+1}| = |V_i| - 1 + [|V_{i+1}| - |V_i|] = |V_{i+1}| - 1$$

Donc $\mathcal{D}_{i+1}$ est connexe et sans cycle. □

Définition 4.2 *Nous considérons un dominant u* connecté *s'il existe un chemin de parenté dirigé de u vers le leader $\mathcal{L}$, dont le premier arc est $(u, dominant(u))$, et constitué ensuite du chemin de parenté $\langle dominant(u), ..., \mathcal{L}\rangle$.*

Remarque 4.3 *Nous supposons que les déplacements sont finis et que les modifications de topologie sont propagées en un temps fini au k_{cds}-voisinage, avant qu'un nouveau changement de topologie apparaisse. Il est certain que si un changement de topologie apparait à l'instant t, l'intégralité des nœuds auront une vision exacte de la topologie à $t+\Delta t$, avant un nouveau changement de la topologie.*

Lemme 4.7 *Lorsqu'un dominant d'une branche se reconnecte, tous ses ascendants et descendants dans la branche sont reconnectés.*

Preuve : Si un dominant u se reconnecte, alors il existe un chemin $\langle u, ..., \mathcal{L}\rangle$ vers le leader. D'autre part, un descendant (respectivement descendant) v de u possède par définition un chemin $\langle u, ..., v\rangle$. Ainsi, v possède un chemin $\langle u, ..., v\rangle \cup \langle u, ..., \mathcal{L}\rangle$ vers le leader. Il est donc connecté.

L'ascendant voisin de u recevra l'`ap-hello` suivant. Il pourra donc choisir u comme père et se considérer reconnecté. Récursivement, les ascendants de u recevront un `ap-hello` du fils dans l'arbre sur le chemin vers u. Ce fils constituera le nouveau père. Les descendants de u recevront l'`ap-hello` suivant de leur ancien père, ils se considéreront reconnectés sans changement local dans le CDS pour eux. Enfin, les descendants des ascendants de u recevront également l'`ap-hello` venant de leur père reconnecté. Ils ne changeront pas de père. □

Lemme 4.8 *Lorsque tous les dominants d'une branche se trouvent déconnectés, un dominant au moins se reconnecte.*

Preuve : Tout changement de topologie peut être décomposé en une addition/suppression élémentaire de liens. L'addition d'une arête dans le graphe ne peut engendrer une perte de connexité du CDS. Supposons que l'arête (u, x) a été supprimée. Au bout d'un temps fini Δt, toute la branche, i.e. les descendants de u, se considère déconnectée. Un dominant se considère déconnecté lorsqu'il n'a reçu aucun `ap-hello` durant Δt. Soit v dominant et descendant de u.

u ne relaie pas d'`ap-hello` d'identifiant supérieur à l, numéro d'identifiant du dernier `ap-hello` relayé avant que l'arête (u, x) casse. Ainsi, le fils de u ne peut pas relayer non plus d'`ap-hello` avec un identifiant supérieur à l. Récursivement, v ne peut pas recevoir ni relayer d'`ap-hello`. Les dominants de la branche de racine u se considèrent ainsi tous déconnectés, et tentent de se reconnecter à un dominant avec un identifiant d'`ap-hello` supérieur à l.

Un dominant au moins arrive à se reconnecter et aucun cycle n'est créé dans le CDS, i.e. v ne peut pas choisir de se reconnecter à un descendant de u. Soit $\mathcal{A}$ l'ensemble des dominants descendants de u, et leurs dominés. $\mathcal{A}$ est connexe. Soit $L = G/\mathcal{A}$. L est connexe : soit d un descendant de $\mathcal{L}$, non descendant de u. Donc $u \notin \langle d, ..., \mathcal{L}\rangle \Rightarrow \langle u, x\rangle$ et $\langle d, ..., \mathcal{L}\rangle$ sont disjoints.

Soit $\mathcal{B}$ l'ensemble des nœuds appartenant à L, voisins de $\mathcal{A}$.

$\mathcal{B} \neq \emptyset$: soit $u \in \mathcal{A}$. Le graphe est supposé connexe. Donc il existe un chemin $\langle u, u_1, ..., u_k\rangle$ avec $u \in \mathcal{A}$ et $u_k \in L$. Il existe u_i tel que $u_i \in L$ et $u_{i-1} \in \mathcal{A}$. Par définition de $\mathcal{B}$, $u_i \in \mathcal{B}$. De plus, dominant(u_{i-1}) est un dominant de la branche déconnectée car il appartient à $\mathcal{A}$. $dominant(u_i)$ est connecté, apartenant à L. $d\,[dominant(u_{i-1}), dominant(u_i)] \leq 2 \cdot k_{cds} + 1$. Ainsi, il existe un dominant dans la branche déconnectée à au plus $2k_{cds} + 1$ sauts d'un dominant connecté.

De plus $V/(L \cup \mathcal{A}) = \emptyset$. Soit v un dominant, et son chemin c de parenté vers le leader :

$$u \in \langle v, ..., \mathcal{L}\rangle \Rightarrow u \in \mathcal{A}$$
$$u \notin \langle v, ..., \mathcal{L}\rangle \Rightarrow u \in L$$

Finalement, $dominant(u_{i-1})$ se reconnectera à $dominant(u_i)$ à l'aide d'un `cds-reconnect` avec un TTL de $2k_{cds} + 1$. Chaque dominant pourra choisir un nouveau père valide et se reconnecter avec l'`ap-hello` suivant selon le lemme 4.7 page précédente. □

Nous pouvons donc en conclure :

Théorème 4.6 *Si la suppression d'une arête implique une déconnexion du CDS, le CDS se reconstruira, et un CDS valide sera créé.*

Lemme 4.9 *Si une cassure du CDS est engendrée, la branche cassée se reconstruira.*

Preuve : Un `break` est envoyé par un dominant lorsqu'il n'arrive plus à se reconnecter. Ce paquet est relayé par les descendants, jusqu'aux dominés feuilles du CDS. Toute la branche reprend un état isolé en attendant la reconstruction.

La branche de sommets isolés forme un ensemble à une distance maximale de $k + 1$ d'un dominant connecté. Soit u un sommet non isolé voisin de la zone isolée. Un tel sommet existe car le leader est obligatoirement dominant, et que le graphe est connexe. Soit v appartient à la zone isolée, et v voisin de u. v est à au plus $k_{cds} + 1$ sauts du dominant de u, connecté. Ce dominant au moins est à au plus $k_{cds} + 1$ sauts de la zone isolée. D'autre part, $V - \mathcal{I}$ forme un ensemble connexe. En effet, si un sommet n'appartient pas à $\mathcal{I}$, alors il existe par définition un chemin vers $\mathcal{L}$ ne contenant que des sommets dominants donc non isolés.

Ainsi, un dominant D à $k_{cds} + 1$ sauts au plus d'un sommet isolé envoie un `cds-invite`, qui, reçu par un sommet isolé l, sert de déclencheur pour la reconstruction de la branche isolée. l devient le leader de la zone. Les algorithmes de construction sont donc exécutés et forment un CDS, comme prouvé dans le théorème 4.2 page 61. Le leader de reconstruction l se connecte lui au dominant D lui ayant envoyé le `cds-invite`, qui n'est par définition pas un de ses descendants. D étant lui même connexe au leader, l est transitivement connexe au leader. Toute la branche reconstruite se connecte donc par l'intermédiaire de i au leader $\mathcal{L}$ selon le lemme 4.7 page précédente. □

Remarque 4.4 *Un dominant inutile passant dominé maintient un CDS connexe en forme d'arbre.*

Preuve : Le processus termine, puisque seul le dominant change d'état. Par ailleurs, si un dominant est une feuille du CDS et qu'il n'a aucun dominé à exactement k_{cds} sauts, il devient

dominant et son père devient le nouveau père de ses dominés. Ainsi, le CDS reste toujours connexe car le dominant n'a aucun descendant dans le CDS. De même, le CDS reste trivialement un ensemble k_{cds}-dominant puisque ses dominés sont à une distance au plus de k_{cds} sauts d'un dominant. □

4.4.5 Clusters

4.4.5.1 Construction

Théorème 4.7 *L'ensemble des clusterheads élus construit un $k_{cluster}$-clustering.*

Preuve : Si un dominant est un clusterhead, il est à une distance inférieure à $k_{cluster}$ d'au moins un clusterhead, trivialement lui même. Si un dominant n'est pas clusterhead, alors il a choisi un dominant clusterhead selon le processus de découverte de voisinage sur le CDS. Les `cluster hellos` sont relayés seulement sur les liens du CDS, à une distance d'au plus $k_{cluster} - k_{cds}$. Donc un dominant choisi un clusterhead à une distance d'au plus $k_{cluster} - k_{cds} < k_{cluster}$.

Un dominé possède le même clusterhead que son dominant. De plus, selon le lemme 4.2 page 61, il est à une distance d'au plus k_{cds} de ce dominant, lui même à $k_{cluster} - k_{cds}$ de son clusterhead. Transitivement, un dominé est à au plus $k_{cluster}$ sauts de son clusterhead.

Selon le lemme 4.1 page 61, tout sommet est soit dominant, soit dominé. Ainsi, un sommet quelconque possède un clusterhead à au plus $k_{cluster}$ sauts. L'ensemble des clusterheads forme donc bien un $k_{cluster}$-clustering du graphe. □

4.4.5.2 Maintenance

Théorème 4.8 *L'algorithme de maintenance maintient un ensemble de clusterheads formant un $k_{cluster}$-clustering de $\mathcal{G}_{CDS}$.*

Lemme 4.10 *Les dominants possèdent tous dans $\mathcal{G}_{CDS}$[1] un clusterhead à une distance d'au plus $k_{cluster}$.*

Preuve : Soit G' l'ensemble des sommets ayant choisi le dominant C comme clusterhead. Il existe une arête dans G' de u vers v si v est le relais vers le clusterhead pour u. Si v possède une distance à son clusterhead de H, u est à une distance de $H+1$ de son clusterhead.

G' est un arbre, i.e. il n'existe aucun cycle dans G'. Supposons au contraire l'existence d'un cycle $\langle u_1, ..., u_k \rangle$. u_i avec $i \in [1..k]$ est à une distance de H_i de C. u_1 est le relais vers le clusterhead de u_k. D'où $H_k = H_1 + 1$. De même, u_{j+1} étant le relais de u_j pour $j \in [1..k-1]$, $H_j = H_{j+1} + 1$. Ainsi, $H_k = H_1 + 1$ et $H_k < H_1$, ce qui aboutit à une contradiction. G' est donc bien un arbre, un dominant choisit donc bien un relais plus proche que lui de son clusterhead C.

Soit D_i l'ensemble des dominants qui possèdent un champs *distance au clusterhead* dans leurs `hellos` fixé à i. Tout dominant de D_i choisit par construction un relais dans D_{i-1}. Supposons que les sommets dans D_{i-1} sont à $i-1$ sauts de C. Alors, les sommets de D_i sont bien à i sauts de C. De plus, $D_0 = C$ et C est bien à 0 sauts de lui même. Enfin, un dominant ne peut choisir un relais que si ce relais est à une distance strictement inférieure à $k_{cluster} - k_{cds}$ de son clusterhead. Ainsi, $D_{k_{cluster}-kcds} = \emptyset$. Tout sommet dominant choisit donc bien un clusterhead à au plus $k_{cluster} - k_{cds}$ sauts de lui. □

Lemme 4.11 *Les dominés sont à au plus $k_{cluster}$ sauts de leur clusterhead*

Preuve : Pour les mêmes raisons que le théorème 4.7, tout dominé est à au plus k_{cds} sauts de son dominant, et donc à au plus $k_{cluster}$ sauts de son clusterhead. □

[1] cf. la définition 4.1 page 62 pour un rappel de la définition de $\mathcal{G}_{CDS}$

4.5 Cardinalité

Nous proposons maintenant de calculer une borne supérieure de la cardinalité de notre solution par rapport à la cardinalité d'un MCDS optimal. Nous supposons que le réseau peut être modélisé par un graphe de disque unité (UDG) [4]. Un UDG modélise bien les réseaux radio avec propagation et affaiblissement idéaux[1]. Un tel type de graphe permet de borner la cardinalité de certaines structures bien connues dans les graphes, notamment d'un Maximum Independent Set.

Nous allons suivre une démarche proche de celle décrite dans [1], en adaptant certains détails puisque nous construisons un k-CDS[2] et non un CDS. Nous allons borner la cardinalité de la dorsale que nous construisons dans un premier temps par rapport à celle d'un k-MCDS, puis dans un deuxième temps par rapport à celle d'un MCDS.

4.5.1 Comparaison par rapport à un k-MCDS

Lemme 4.12 *Soit k-MCDS l'ensemble connexe de cardinalité minimale et représentant un ensemble k-dominant du graphe G. L'ensemble construit k-CDS suit la contrainte suivante :*

$$|k\text{-}CDS| \leq (2k+1) \cdot (4\,|k\text{-}MCDS| + 1) \tag{4.1}$$

Preuve : Soit MIS un ensemble indépendant maximum pour l'inclusion, et k-MIS un ensemble k-indépendant, maximal pour l'inclusion. Un nœud comporte au plus 5 voisins appartenant à un MIS [9]. Pour les mêmes raisons géométriques, un nœud comporte au plus 5 voisins appartenant à un k-MIS. Soit k-MCDS l'ensemble k-dominant connexe de cardinalité minimale. k-MCDS $= \{a_i\}_{i\in[1..|MCDS|]}$. D'après précédemment, a_1 comprend au plus 5 voisins dans le k-MIS. De même, tout nœud $a_{i/i\in[2..|MCDS|]}$ comprend au plus 4 voisins dans le k-MIS, non voisins de $a_{j/j<i}$. D'où

$$|\text{k-MIS}| \leq 4\,|\text{k-MCDS}| + 1$$

Or l'ensemble de dominants construit dans la première phase de l'algorithme est un k-MIS. De plus, chaque dominant de la première phase colore au plus $2k$ dominés en dominants durant la deuxième phase. Ainsi :

$$|\text{k-CDS}| \leq (2k+1) \cdot (4\,|\text{k-MCDS}| + 1)$$

Ce qui donne bien la borne de cardinalité recherchée. □

4.5.2 Comparaison par rapport à un MCDS

Lemme 4.13 *Soit MCDS l'ensemble dominant connexe de cardinalité minimale[3]. La proposition suivante est vérifiée :*

$$|k\text{-}MCDS| \leq |MCDS| - 2(k-1) \tag{4.2}$$

Preuve : Nous définissons 0-Ens $\equiv MCDS$. De plus, k-Ens est construit tel que k-Ens ne contienne que les sommets non feuilles de (k-1)-Ens.
k-Ens suit les propriétés suivantes :

1. k-Ens est un arbre : MCDS est un arbre. De plus, si (k-1)-Ens est un arbre, alors k-Ens l'est aussi car seules les feuilles sont enlevées.
2. |k-Ens| $\leq$ |(k-1)-Ens| $-$ 2 : un arbre possède au moins deux feuilles

[1] Cf. section 2.4.1 page 13 pour un rappel des graphes de disque unité et pour une discussion sur leurs atouts et limites dans la modélisation des réseaux ad-hoc

[2] Pour des soucis de lourdeur d'écriture, et puisque nous ne nous focalisons dans cette section que sur la dorsale, nous utilisons indtistinctement les notations k et k_{cds}

[3] k-MCDS reste l'ensemble connexe de cardinalité minimale et représentant un ensemble k-dominant du graphe

3. k-Ens est un k-CDS : 0-Ens est un MCDS donc a fortiori un CDS. Supposons que tout nœud de G soit à au plus $k-1$ sauts d'un nœud de (k-1)-Ens. Soit N un nœud et D le nœud de (k-1)-Ens le plus proche. Par construction de k-Ens, D est à au plus un saut de k-Ens. Par transitivité, N est à au plus k sauts d'un nœud de k-Ens. De plus, k-Ens est un arbre donc a fortiori connexe.

Or par définition, |k-MCDS| $\leq$ |X| avec X tout ensemble formant un k-CDS. Ce qui conduit à l'inégalité recherchée. □

Au final, en utilisant les théorèmes 4.1 page précédente et 4.2 page précédente, nous obtenons transitivement :

$$|\text{k-CDS}| \leq 4 \cdot (2k+1)\,|\text{MCDS}| - (2k+1)(8k-9) \tag{4.3}$$

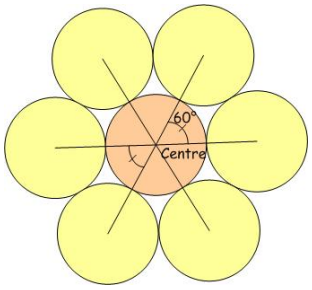


Fig. 4.2 – Borne du nombre de voisins indépendants dans un UDG

Remarque 4.5 *Un nœud comporte au plus 5 voisins appartenant à un MIS pour une raison qui peut s'expliquer intuitivement : le graphe est un UDG, il n'est donc possible de placer qu'au plus 5 cercles ne s'intersectant pas mais intersectant un même 6^{ieme} cercle. Si nous plaçons 6 cercles, ils se trouvent tous en intersection : les distances entre nœuds se trouvent être identiques, les angles sont donc de $\frac{2\pi}{6}$ (fig. 4.2).*

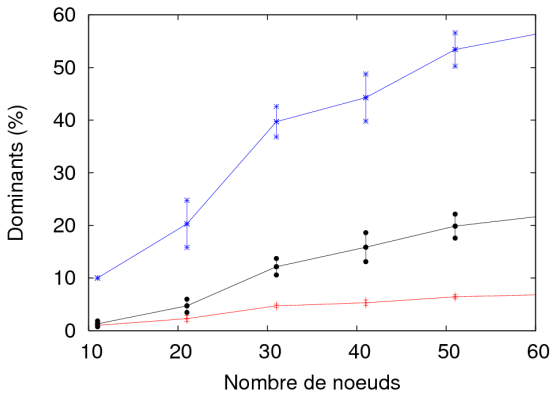


Fig. 4.3 – Comparaison entre la borne supérieure et la cardinalité de la structure obtenue à travers des simulations

Comparaison entre borne supérieure analytique et cardinalité simulée Afin de valider cette borne supérieure, nous avons simulé la structure virtuelle avec les mêmes paramètres que dans la section . Nous avons utilisé une algorithme exhaustif pour calculer la cardinalité du MCDS (le CDS de cardinalité minimale). C'est donc pourquoi aucun résultat n'est présenté

pour un réseau de plus de 60 nœuds, le temps de calcul devenant trop long à partir de cette cardinalité du réseau[1].

Nous pouvons voir sur la figure 4.3 que la cardinalité de la structure simulée est bien comprise entre la cardinalité du MCDS, et la borne supérieure de l'équation 4.3. De plus, la structure simulée prend en compte des paramètres tels que les collisions, l'instabilité des liens radio ou le temps de convergence. Par ailleurs, nous pouvons voir que la structure virtuelle présente une cardinalité proche de celle du MCDS. De plus, le MCDS est calculé de façon centralisée, sur une topologie donnée, en ne tenant compte d'aucune collision ou changement de topologie. En d'autres termes, un MCDS ne prend pas en compte la robustesse, la persistance et les temps de convergence.

4.6 Étude des propriétés de la structure virtuelle au travers de simulations

Pour étudier plus finement le comportement des algorithmes proposés, nous avons conduit une série de simulations. La solution a été simulée en utilisant OPNET Modeler [10]. Le comportement général de la structure a déjà exposé dans le chapitre précédent. L'impact du degré, du nombre de nœuds sur la connexité et la cardinalité de la structure ont notamment été étudiés. Nous avons choisi ici de nous focaliser sur les propriétés en terme de stabilité et de convergence de la structure virtuelle. Cette étude par simulations corrobore ainsi les propriétés d'auto-stabilisation démontrées en section 4.4 page 58.

4.6.1 Paramètres

La présence de collisions au niveau radio et les interférences influent fortement sur les performances d'un protocole dans un réseau ad-hoc. Nous avons donc choisi d'utiliser la couche IEEE 802.11 au niveau MAC, et d'utiliser la modélisation radio standard d'OPNET (affaiblissement de type environnement libre). La portée radio est circulaire, d'un rayon de 300 mètres.

Les paramètres par défaut des simulations sont de 40 nœuds présentant un degré moyen de 10. Afin d'obtenir des résultats représentatifs, nous avons conduit pour chaque graphe 20 simulations avec des graines pseudo-aléatoires différentes et avons tracé les intervalles de confiance de 95%. Les paramètres par défaut du protocole sont présentés sur le tableau 4.1

Paramètre	Valeur
Nombre de nœuds	40
Degré	10
Portée radio	300m
Mobilité	0 m.s-1
$\text{Intervalle}_{hello}$	4 secondes
$\text{Intervalle}_{ap-hello}$	2 secondes

TAB. 4.1 – Paramètres de l'auto-organisation

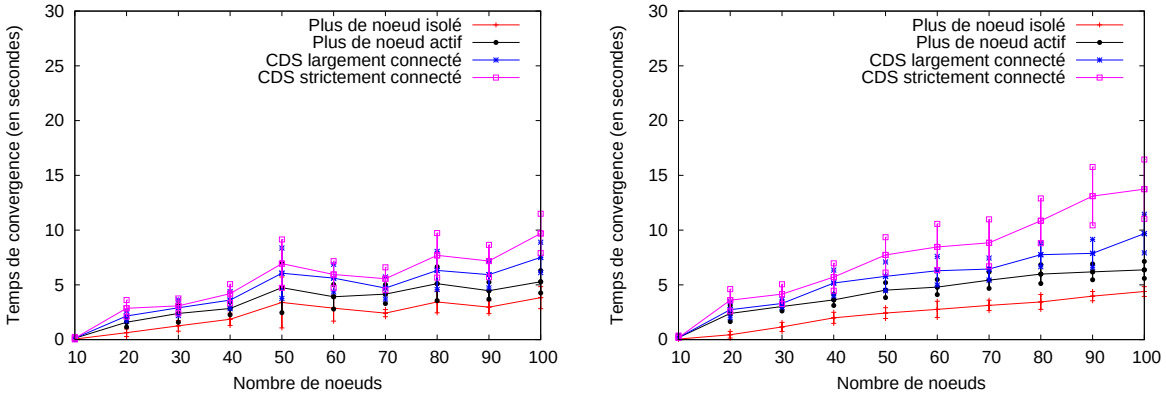
Par ailleurs, nous avons pu remarquer que les clusters sont toujours bien construits avant la fin de la construction de la dorsale. Les clusters sont robustes et ne présentent donc pas la partie la plus sensible de la structure. Les temps de convergence étant très rapides, ils n'ont donc pas été représentés ici. Nous avons par contre choisi de détailler les propriétés concernant la dorsale.

[1] Nous rappelons que le calcul d'un MCDS est un problème NP-difficile.

4.6.2 Temps de convergence

Nous avons dans un premier temps fixé k_{cds} à 1 et $k_{cluster}$ à 2 (fig. 4.4(a)). Nous pouvons remarquer que moins de 5 secondes sont nécessaires pour ne plus avoir de nœud isolé dans le réseau. De même, les élections se terminent rapidement et il n'existe plus de nœud actif après 7 secondes. Nous pensons qu'un tel délai reste très raisonnable pour initialiser un réseau, utilisé pendant un temps qui lui est largement supérieur (de quelques heures à quelques jours). Enfin, moins de 10 secondes sont nécessaires pour avoir une dorsale *largement* ou *strictement* connexe. La dorsale est *strictement connectée* lorsque l'ensemble des nœuds en ne gardant que les arêtes de l'arbre de la forme enfant → parent forme un graphe connexe. La dorsale est *largement connectée* lorsque nous prenons en compte la structure maillée de la dorsale, en relâchant donc la contrainte. Plus précisément, nous avons dans le graphe largement connecté les arêtes suivantes :

- D1 → D2 : si D1 et D2 sont dominants et voisins radio.
- d1 → d2 : si d1 et d2 sont dominés, voisins radio, et ont choisi le même père dans la dorsale.
- d1 → D2 : si d1 est dominé, D2 dominant, d1 et D2 sont voisins radio, et D2 est le père de d1.



(a) $k_{cds} = 1$ et $k_{cluster} = 2$ (b) $k_{cds} = 2$ et $k_{cluster} = 3$

FIG. 4.4 – Temps de convergence de la construction de la dorsale

Les algorithmes de construction, exécutés en parallèle pour la phase de création d'un ensemble dominant puis pour son interconnexion sont efficaces : **ils convergent rapidement et forment une structure valide en moins de 10 secondes**, même avec 100 nœuds. Si nous fixons k_{cds} à 2 et $k_{cluster}$ à 3 (fig. 4.4(b)), le temps requis par les algorithmes de construction est plus élevé : les chemins permettant d'interconnecter l'ensemble dominant sont plus longs et demandent donc un délai additionnel. Cependant, une dorsale valide et opérationnelle est construite en moins de 14 secondes en partant d'un état initial nul même avec 100 nœuds.

4.6.3 Stabilisation

Il est important que l'algorithme converge rapidement vers un état valide. Un nombre important de changements d'états traduirait au contraire des oscillations néfastes présentes dans l'algorithme. Nous avons donc observé le comportement de l'algorithme avant sa convergence (fig. 4.5 page ci-contre). Nous avons relevé le nombre moyen de changements d'états par nœud avant que l'algorithme ne converge. Par exemple, avec 10 nœuds, un nœud prendra en moyenne 0.7 fois l'état actif, 1 fois l'état dominé, et 0.2 fois l'état dominant avant que l'algorithme ne converge. Nous pouvons remarquer qu'un nœud doit en moyenne changer plus d'une fois d'état. Cette propriété est logique : un nœud isolé prendra par exemple d'abord l'état actif avant d'être élu dominant. Nous pouvons de plus remarquer que **le nombre de changements d'états par**

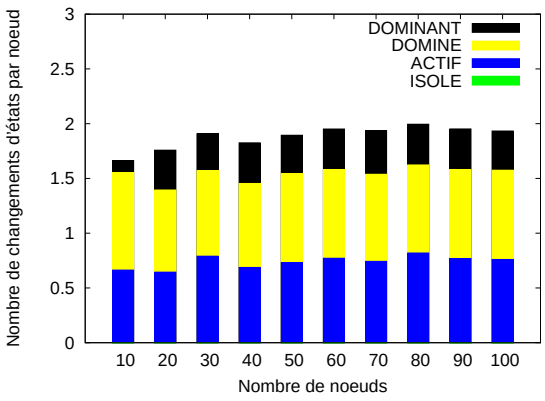
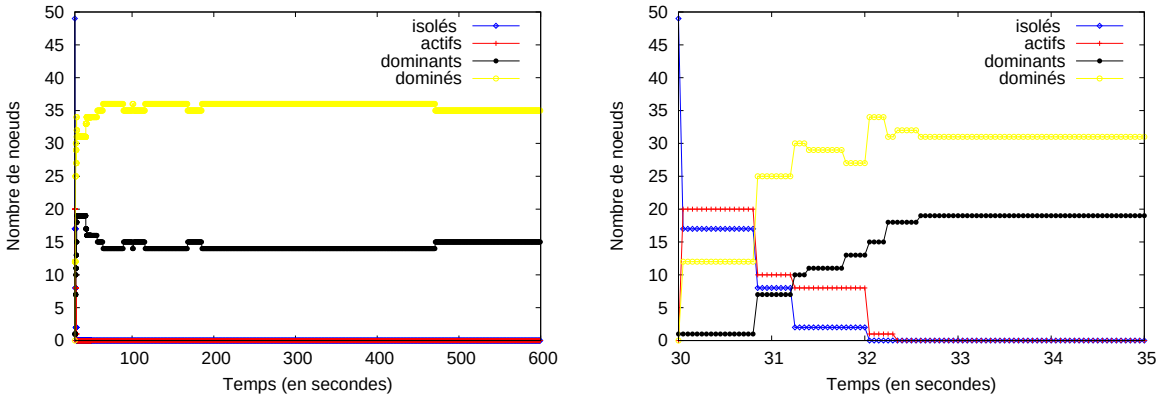


FIG. 4.5 – Nombre de changements d'états par nœud avant convergence de la construction ($k_{cds} = 1$ et $k_{cluster} = 2$)

nœud reste inchangé lorsque le nombre de participants augmente dans le réseau : les algorithmes de construction de la dorsale passent parfaitement à l'échelle. Un nœud ne change en moyenne que 2 fois son état avant qu'une auto-organisation valide ne soit formée.



(a) Durant les 600 premières secondes

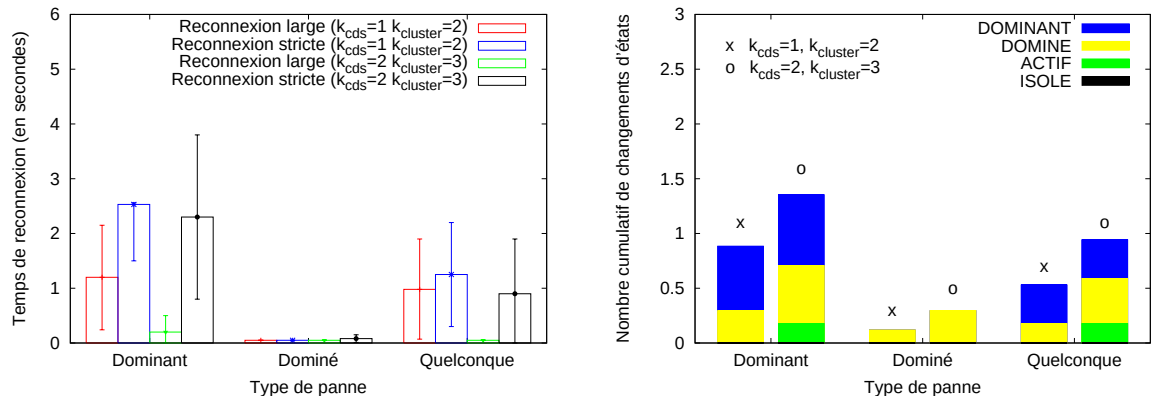
(b) Zoom sur les 5 premières secondes

FIG. 4.6 – Nombre de nœuds isolés/actifs/dominants/dominés ($k_{cds} = 1$ / $k_{cluster} = 2$ / 50 nœuds)

Dans un deuxième temps, nous avons regardé l'évolution des états au cours du temps (fig. 4.6(a)). Dans un réseau de 50 nœuds avec $k_{cds} = 1$ et $k_{cluster} = 2$, les nœuds isolés et actifs ne sont présents que dans les toutes premières secondes d'exécution. Aucune cassure de la dorsale ne se produit, et aucun nœud ne devient isolé à cause d'une telle cassure. De plus, les états restent très stables au cours du temps. Cependant, nous pouvons observer quelques micro-variations : des collisions peuvent se produire à cause du médium radio. Ainsi, des paquets étant perdus, des incohérences peuvent apparaître. Un nœud ayant une mauvaise vision de la topologie, il peut reconstruire localement sa dorsale alors qu'une telle procédure est inutile. Cependant, de tels changements sont peu fréquents.

Nous avons ensuite étudié les toutes premières secondes d'exécution (fig. 4.6(b)). Nous pouvons observer que initialement, trop de dominants sont élus par l'algorithme. Cependant, la redondance est détectée par les algorithmes de maintenance, et la cardinalité de la dorsale diminue peu après. Nous pouvons également observer le fonctionnement de l'algorithme en vagues, ou pseudo-rondes : le nombre de nœuds isolés et actifs diminue par pallier, correspondant aux temporisateurs des élections.

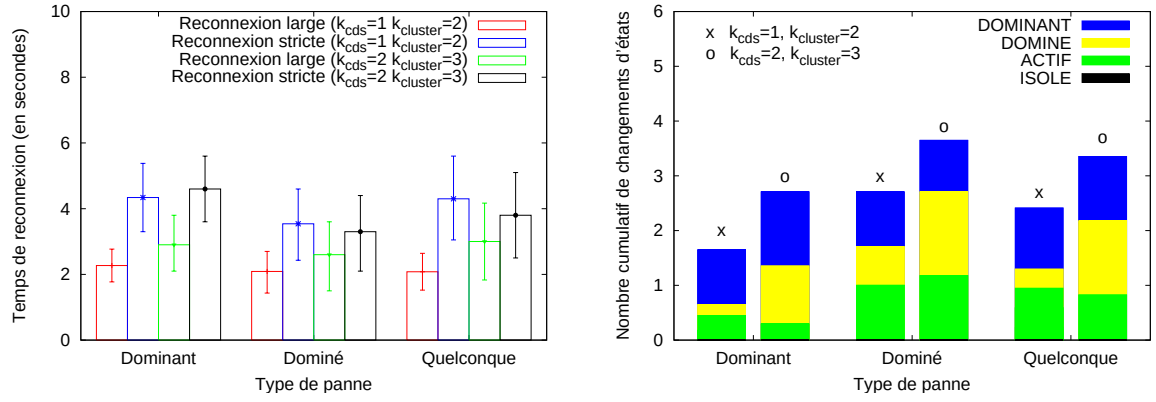
4.6.4 Résistance aux fautes



(a) Temps de reconnexion (b) Nombre de changements d'états

FIG. 4.7 – Simulation d'une faute temporaire

Nous avons simulé dans un premier temps une faute temporaire : un dominé devient arbitrairement dominant, ou au contraire un dominant devient dominé (fig. 4.7(a) et 4.7(b)). Ceci simule la panne d'un nœud (un nœud change d'état suivant une décision incohérente, un nœud part et un nouveau apparaît avec un état différent...). Nous pouvons remarquer que le temps de reconnexion de la dorsale lorsqu'un dominé devient dominant est presque nul. Lorsque $k_{cds} = 1$, un dominé ne possède en effet aucun fils, le changement d'état n'a donc aucun effet. Si $k_{cds} = 2$, le dominé changera rapidement son père de telle sorte que la dorsale soit connexe. Lorsqu'un dominant perd son rôle, il entraîne la déconnexion d'une partie ou de la totalité de ses dominés. Cependant, nous pouvons voir qu'**une seconde seulement est nécessaire à la reconnexion de la dorsale en utilisant sa redondance naturelle**. De même, nous pouvons remarquer que le nombre de changements d'états est très réduit : un changement local n'influe que localement sur la topologie. Les algorithmes localisés de maintenance agissent efficacement. Lorsqu'un dominant devient dominé, nous pouvons remarquer que certains nœuds sont élus dominants pour reconnecter la dorsale.

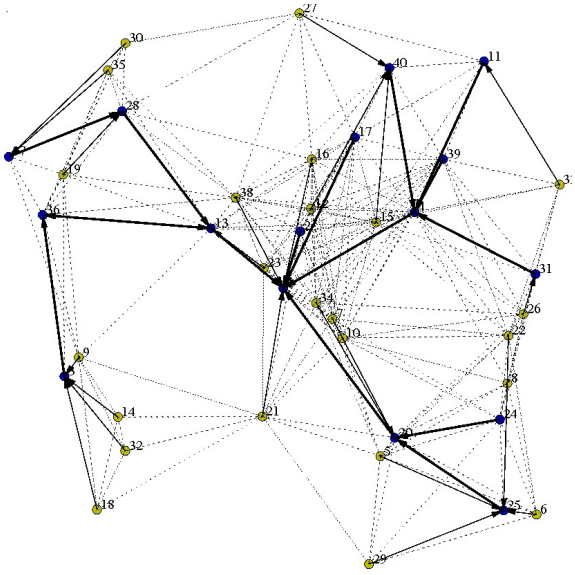


(a) Temps de reconnexion (b) Nombre de changements d'états

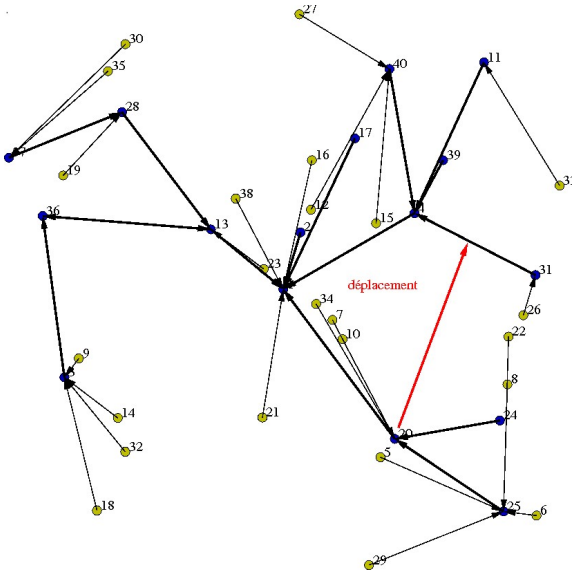
FIG. 4.8 – Simulation du déplacement d'un nœud

Nous avons ensuite simulé le déplacement unitaire d'un nœud : un nœud est aléatoirement choisi, et sa nouvelle position est choisie aléatoirement dans la surface de simulation (fig. 4.8(a)). Ainsi le nœud déplacé doit re-découvrir son voisinage, déterminer son nouvel état, et doit se reconnecter à la dorsale. De la même manière, ses anciens voisins doivent l'invalider, et poten-

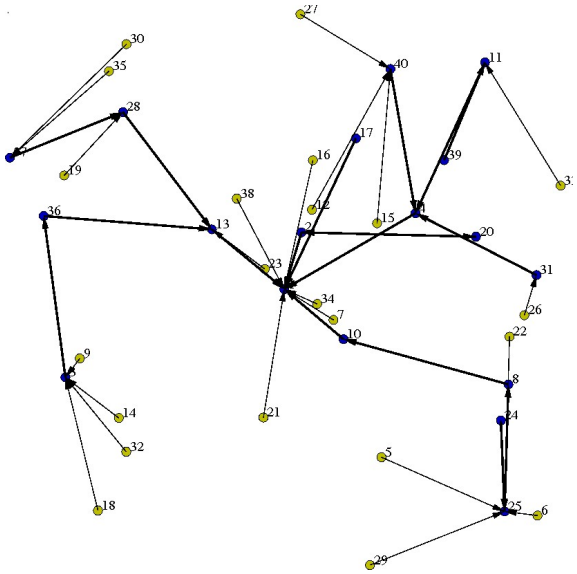
tiellement reconnecter la dorsale. Nous pouvons remarquer que **l'auto-organisation est de nouveau entièrement valide en moins de 3 secondes** : la dorsale est reconnectée, et la dorsale forme un ensemble dominant, quel que soit l'état du nœud déplacé. Le déplacement du nœud n'a qu'un impact local dans la topologie (fig. 4.8(b) page précédente). Nous pouvons observer que des élections de dominants doivent usuellement se produire avant que la dorsale ne se reconnecte. De même, certains nœuds devenus actifs ne sont pas élus et deviennent dominés. La figure 4.9 représente une illustration de la topologie avant et après le changement. Nous pouvons vérifier qu'un changement de la topologie radio n'impacte que localement la topologie de la dorsale.



(a) Avant



(b) Topologie de la dorsale



(c) Après convergence

FIG. 4.9 – Topologie avant et après le changement de localisation d'un nœud (les nœuds bleus sont les dominants, les nœuds jaunes les dominés, les liens de parenté de la dorsale étant représentés sous forme d'arcs dirigés, en gras)

L'impact macroscopique d'un modèle de mobilité a déjà été étudié dans le chapitre précédent. L'impact du Random Waypoint Mobility Model est notamment représenté sur la figure 3.8, et l'impact du Boundless Mobility Model sur la figure 3.11 page 47.

4.6.5 Conclusion

Les simulations conduites nous permettent de valider les propriétés de la structure d'auto-organisation :

- Stabilité : les nœuds ne changent d'état que si un changement de topologie intervient, ou si une collision se produit.
- Passage à l'échelle : les algorithmes réagissent bien à une augmentation du nombre de nœuds.
- Robustesse : la topologie créée réagit très bien à la dynamique du réseau. La dorsale reste connexe même après un changement de topologie.
- Localisation : un changement local de la topologie radio n'influe que localement sur la topologie virtuelle.

4.7 Impact du Poids dans le processus d'élection

De nombreux algorithmes d'organisation (cf. section 2.5 page 15) sont basés sur un processus d'élection. Ainsi, il est nécessaire d'avoir une métrique de décision. La plupart des algorithmes se basent sur l'identifiant ou sur le degré. Cependant, une telle métrique de décision ne nous semble pas refléter fidèlement l'aptitude d'un nœud à agir en tant que coordinateur dans le réseau. Le choix de l'identifiant, par exemple, introduit une sélection aléatoire des dominants sans tirer parti des caractéristiques intrinsèques des nœuds, en particulier de leur hétérogénéité : le réseau n'exploite pas de façon efficace ses nœuds les plus forts. A l'opposé une sélection basée sur le degré le plus important va privilégier les nœuds ayant le plus de voisins radios, autrement dit, cela risque de congestionner le dominant. [11] propose lui de se baser sur la mobilité tirée d'un GPS, équipement que nous écartons à cause de son coût financier. [3] combine plusieurs critères afin de former une métrique (degré, mobilité tirée d'un GPS). Cependant, le degré représentant dans les simulations 70% de la métrique finale, il est difficile d'estimer son impact. Nous proposons donc un poids que nous pensons refléter la stabilité d'un nœud. Afin de l'évaluer, nous décrivons conjointement une solution très simple d'économie d'énergie, ne nécessitant aucune information additionnelle.

4.7.1 Proposition de poids

La construction et la maintenance explicitées auparavant sont basées sur un processus d'élection. Ce processus permet, à partir de la connaissance d'un poids ré-évalué en permanence, de sélectionner les nœuds devenant dominants. Il est évident que le choix de mauvais dominants conduira à des baisses significatives des performances de la structure virtuelle et de sa stabilité en particulier. Ainsi le choix d'un dominant trop fortement mobile introduira une rupture dans la topologie virtuelle tout comme la sélection d'un nœud avec un niveau d'énergie trop faible. Il est donc important d'utiliser une métrique de poids témoignant de l'importance d'un nœud à jouer un rôle prédominant dans la topologie virtuelle créée.

Il nous apparaît donc important d'introduire une nouvelle métrique, appelée métrique de *stabilité* dont l'objectif est de permettre la sélection de nœuds ayant les propriétés de stabilité et de persistance les plus fortes dans le temps et l'espace. Nous proposons donc le poids ainsi défini :

$$P_{stabilite} = \delta(\alpha.D + \beta.M)$$

où :

- D représente la densité d'un nœud,
- M représente l'évaluation de la mobilité locale calculée sur le changement de voisinage au cours d'une période de temps,

- δ est une pénalité calculée sur la base du niveau d'énergie disponible. Cette pénalité varie sous forme de palliers selon l'énergie restante. Un nœud avec peu d'énergie ne pourra donc être dominant,
- α et β sont des facteurs de pondération suivant que nous souhaitons privilégier les nœuds avec un fort voisinage ou une faible mobilité. Dans nos évaluations de performances (section 4.7.3), nous avons choisi $\beta \gg \alpha$

4.7.2 Une solution très simple d'économie d'énergie

Les nœuds du réseau fonctionnant sur batterie, il est important d'apporter des solutions permettant d'augmenter leur durée de vie.

D'après [6], la seule méthode permettant d'économiser de l'énergie est l'endormissement des nœuds. Ne plus émettre des paquets ne suffit pas à économiser les batteries des nœuds mobiles car l'attente de paquets entraîne également une consommation d'énergie. La solution d'auto-organisation que nous proposons permet, ainsi, d'introduire très simplement une meilleure gestion de l'énergie pour l'ensemble des nœuds. En effet, notre proposition sépare les nœuds en deux catégories : les nœuds dominants participant à la vie du réseau et les dominés qui ne sont actifs que lors d'une transmission d'informations. Nous rappelons que parmi les critères de choix d'un dominant se trouve le niveau d'énergie restant. Ainsi, il est possible d'endormir les dominés pendant une période de temps avec un faible impact sur la vie du réseau et sur la topologie virtuelle crée.

Plus précisément, nous proposons qu'un dominant ne possédant pas suffisamment de voisins actifs ne soit pas autorisé à s'endormir ; nous limitons à 6 le nombre minimal de voisins actifs pour qu'un dominant puisse s'endormir. Afin de ne pas pénaliser la maintenance, nous interdisons à un nœud de dormir lorsqu'il est voisin d'une reconnexion de dorsale (un `cds-reconnect` est reçu, ou il existe au moins un voisin isolé dans la table de voisinage). Enfin, nous introduisons une pénalité (P_e) représentant le nombre de 1-voisin de plus faible poids (le nœud considéré inclus). La probabilité d'endormissement P_{sleep} d'un nœud est alors :

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

Les nœuds s'endormant sont donc les dominants les moins importants dans le réseau. Bien entendu, cette métrique peut être modifiée suivant le modèle de consommation des nœuds, l'objectif en terme de durée de vie du réseau, etc.

Une telle solution d'économie d'énergie présente un intérêt de simplicité. De plus, elle découle naturellement de la dorsale construite auparavant : la dorsale différencie déjà les nœuds forts des nœuds faibles en énergie, et aucune information nécessaire additionnelle n'est requise.

4.7.3 Simulations

Afin de valider la métrique de stabilité que nous avons proposée, nous avons conduit une campagne de simulations, sous OPNET Modeler [10]. Les paramètres par défaut sont les mêmes que ceux décrit dans la section 4.6 page 69.

4.7.3.1 Impact de la métrique

Nous avons comparé les performances de la dorsale selon 4 métriques basées sur : l'identifiant, le degré, la mobilité absolue tirée d'un GPS et notre métrique de stabilité. Afin de simuler l'hétérogénéité d'un réseau hybride, un tiers des nœuds possède une mobilité élevée, entre 20 et 30 $m.s^{-1}$, et deux tiers une mobilité entre 0 et 5 $m.s^{-1}$ (tab. 4.2 page suivante). **La connexité est maximale avec la métrique de stabilité**, qui choisit plus de dominants pour connecter la dorsale. Le nombre de reconnexions semble moins important avec une métrique basée sur

le degré, un dominant ayant plus de voisins, la maintenance proactive de la dorsale permet peut être de trouver plus rapidement un chemin de reconnexion. Cependant, de tels dominants peuvent former un goulot d'étranglement en cas de forte charge réseau. Enfin, la connexité des clusters, très élevée, est semblable pour toutes les métriques.

Métrique	CDS				Clusters	
	Connexité	Cardinalité	Nb de reconnexions	Nb de cassures	Connexité	Cardinalité
Id	94,2	9,8	85,6	1,5	99,7	4,8
Degré	93,8	9,7	79,4	1,6	99,7	4,8
Mobilité	93,9	10,1	86,5	3	99,6	5
Stabilité	94,8	10,4	88	1,7	99,7	5,4

TAB. 4.2 – Impact de la métrique sur les performances

4.7.3.2 Performances de la solution d'économie d'énergie

Métrique	Connexité dorsale	Connexité clusters	Temps d'endormissement (en s)	
			moyen	du nœud le plus faible
Degré	93,6	99,1	70,7	65,2
Id	93,6	99	69,5	40
Mobilité	94,3	99,2	71,2	59,5
Stabilité	93	99,2	80,4	128

TAB. 4.3 – Performances de la solution d'économie d'énergie

Nous avons enfin simulé les performances de la solution d'économie d'énergie proposée. Un nœud à énergie très faible a été créé dans le réseau afin de mesurer la prise en compte de l'hétérogénéité des nœuds. La dorsale subit une légère perte de connexité due à l'économie d'énergie de certains dominés (tab. 4.3). Cependant, les clusters gardent la même connexité. La métrique de stabilité présente une connexité de dorsale légèrement moins importante que les autres. Par contre, **le temps d'endormissement moyen d'un nœud est beaucoup plus important avec la métrique de stabilité** (+12%). De même, le réseau favorise les nœuds faibles. Le nœud le plus faible dort en moyenne 100% plus qu'avec les autres métriques. La métrique de stabilité multi-critères est donc particulièrement efficace dans une telle solution.

4.8 Conclusion

Dans ce chapitre, nous avons présenté une évaluation analytique de la structure virtuelle d'auto-organisation proposée dans le chapitre précédent. Tout d'abord, la construction et la maintenance de la structure est asymptotiquement peu coûteuse en temps et en messages. Nous avons également démontré une propriété forte : les algorithmes, tant de construction que de maintenance, sont auto-stabilisants. Ainsi, les algorithmes convergent même si le réseau est dans un état initial erroné, si un changement de topologie survient, ou si un nœud tombe en panne ou prend une mauvaise décision. De ce fait, une auto-organisation efficace et valide est maintenue de façon continue. Nous pensons que l'auto-stabilisation représente une caractéristique forte de notre proposition.

Des simulations ont permis de valider ce comportement recherché : la topologie virtuelle est stable et s'auto-répare naturellement, avec un impact localisé. Un poids de stabilité très simple a été également introduit, montrant l'efficacité de la structure à prendre en charge et tirer parti de l'hétérogénéité du réseau : un nœud *fort* sera naturellement élu, remplissant plus de fonctions au niveau réseau, et stabilisant ainsi la structure. Enfin, nous avons étudié les performances de la

structure d'auto-organisation lorsqu'une solution très simple d'économie d'énergie est proposée. Nous avons donc montré via ces exemple que la structure d'auto-organisation remplit bien son rôle : elle est flexible et peut s'adapter facilement à différentes applications.

Nous avons donc proposé une structure virtuelle d'auto-organisation présentant toutes les propriétés que nous recherchions : passage à l'échelle, persistance, localisation, auto-stabilisation, flexibilité, etc. (cf. section 2.3 page 11). Il est donc maintenant logique d'exploiter cette structure virtuelle. Nous rappelons que l'auto-organisation ne constitue pas une fin en soi mais représente au contraire une couche commune d'organisation du réseau, utile aux couches supérieures. Puisque le routage constitue une problématique clé en réseau, nous proposons dans le chapitre suivant d'étudier comment modifier les protocoles de routage existants afin qu'ils tirent pleinement parti de la structure d'auto-organisation.

Bibliographie

[1] 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.

[2] H. Baala, O. Flauzac, J. Gabe, M. Bui, and T. El-Ghazawi. A self-stabilizing distributed algorithm for spanning tree construction in wireless ad hoc networks. *Journal of Parallel and Distributed Computing*, 3(1) :97–104, January 2003.

[3] M. Chatterjee, S. K. Das, and D. Turgut. Wca : A weighted clustering algorithm for mobile ad hoc networks. *Journal of Cluster Computing (Special Issue on Mobile Ad hoc Networks)*, 5(2) :193–204, April 2002.

[4] B. N. Clark, C. J. Colburn, and D. S. Johnson. Unit disks graphs. *Discrete Mathematics*, 86 :165–177, December 1990.

[5] E. Dijkstra. Self-stabilizing systems in spite of distributed control. *Communications of the ACM*, 17(11) :643–644, November 1974.

[6] L. Feeney and M. Nilson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *INFOCOM*, pages 1548–1557, Anchorage, USA, April 2001. IEEE.

[7] R. Friedman, M. Gradinariu, and G. Simon. Locating cache proxies in manets. In *International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pages 175–186, Tokyo, Japan, May 2004. ACM.

[8] S. K. S. Gupta and P. K. Srimani. Self-stabilizing multicast protocols for ad hoc networks. *Journal of Parallel and Distributed Computing*, 63(1) :87–96, 2003.

[9] M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi, and D. J. Rosenkrantz. Simple heuristics for unit disk graphs. *Networks*, 25 :59–68, December 1995.

[10] OPNET Modeler. http://www.opnet.com (v8.1a).

[11] B. Prithwish, K. Naved, and T. D. C. Little. A mobility based metric for clustering in mobile ad hoc networks. In ACM, editor, *International Conference on Mobile Computing and Networking (MOBICOM)*, pages 129–140, Roma, Italy, July 2001. ACM.

[12] M. Schneider. Sef-stabilization. *ACM Computing Surveys*, 25(1) :45–67, March 1993.

[13] J. Wu and H. Li. Dominating-set-based routing in ad hoc wireless networks. In *International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL'M)*, pages 7–14, Seattle, USA, August 1999. ACM.

Publications

Conférence internationale

[1] F. Theoleyre and F. Valois. About the self-stabilization of a virtual topology for self-organization in ad hoc networks. In LNCS, editor, *Self-Stabilization Symposium (SSS)*, volume 3764, pages 214–228, Barcelona, Spain, October 2005. IEEE.

Rapport de recherche

[2] F. Theoleyre and F. Valois. About the self-stabilization of a virtual topology for self-organization in ad hoc networks. Research Report 5650, INRIA, August 2005. This research report is a long version of the article published in SSS 2005.

Chapitre 5

Bénéfice possible de l'auto-organisation dans le routage

5.1 Introduction

Le but principal d'un réseau est d'acheminer des communications. Or dans un réseau ad hoc, les terminaux ne sont pas tous à portée radio les uns des autres. De plus, la topologie étant dynamique, un protocole de routage est nécessaire pour trouver et mettre à jour un chemin de la source à la destination. La problématique du routage dynamique dans les cœurs de réseaux filaires a déjà été bien étudiée. Cependant, nous verrons pourquoi de telles approches ne sont pas directement applicables aux réseaux ad-hoc.

Nous verrons dans un premier temps pourquoi les approches filaires ne peuvent pas convenir pour le routage dans les réseaux ad-hoc. Ensuite, sera présenté dans la section 5.3 un aperçu des nombreux travaux réalisés dans ce domaine. La section 5.4 s'attachera aux limites sur lesquelles se heurtent actuellement ces solutions, et donc pourquoi le routage reste un domaine de recherche à développer. La section 5.5 détaillera notre proposition de framework de protocoles de routage tirant parti de la structure virtuelle d'auto-organisation décrite précédemment. Enfin, nous présenterons une étude de performance du protocole à travers des simulations en section 5.6.

5.2 De l'inadaptation du filaire

Les réseaux classiques sont par nature très hiérarchisés : les terminaux se trouvent en périphérie et un ensemble d'équipements spécialisés et structurés permet de router les paquets entre les terminaux. Internet est notamment segmenté en systèmes autonomes (AS), avec un protocole de routage entre systèmes autonomes, BGP [28]. A l'intérieur d'un système autonome, l'administrateur est libre de déployer un protocole de routage quelconque. Ainsi, la hiérarchie, statique, permet de déployer efficacement et rapidement une série de protocoles indépendants les uns des autres, mais pouvant coopérer. Nous verrons que le concept de clusters permet dans le protocole de routage présenté dans ce chapitre de jouer le rôle des systèmes autonomes, cachant la topologie du cluster même. Par ailleurs, les concepts de BGP ne sont pas applicables directement aux réseaux ad-hoc : aucune hiérarchie n'est présente et les changements de topologie sont trop fréquents.

Les protocoles de routage classiques se classent en deux grandes catégories. Dans l'approche *à états de liens*, chaque routeur diffusera sa présence dans le réseau. Chaque routeur ayant ainsi une vue globale de la topologie, il peut exécuter un algorithme du plus court chemin pour construire sa table de routage (ex : OSPF [19]). Cependant, les mises à jour périodiques entraînent rapidement une tempête d'inondation. Dans l'approche à *vecteur de distance*, chaque

routeur envoie à ses voisins la totalité de ses routes. Ces routes se propagent de proche en proche dans le réseau, amenant à une connaissance globale du réseau. Cependant, une telle approche présente une convergence lente, donc inapplicable aux réseaux ad hoc. Par ailleurs, nous retrouverons cette classification dans les protocoles de routage proactifs proposés pour les réseaux ad hoc.

5.3 Protocoles de routage pour les MANET

Un protocole de routage pour les MANET devrait présenter les propriétés suivantes :

- Trafic de contrôle minimum
- Délai de bout en bout minimum
- Peu de pertes de paquets de données
- Réactivité aux changements de topologie (cette propriété impacte directement sur les pertes de paquets).

Nous proposons de distinguer cinq grandes familles de protocoles (en jaune sur la figure 5.1) :

1. A plat : le réseau est considéré dans sa forme originale, sans hiérarchie
2. Géographique : le protocole s'appuie sur un système de positionnement tel que le GPS
3. Hiérarchique : une hiérarchie basée sur des clusters est dynamiquement maintenue, le protocole de routage l'exploitant
4. Sur dorsale : une dorsale est construite et maintenue, le protocoles de routage en tirant parti
5. A large échelle : le protocole est conçu spécifiquement pour les réseaux à cardinalité très élevée (plusieurs milliers de nœuds). Étant donnée la spécificité de ces contraintes, nous avons choisi de classer ces protocoles à part

Les protocoles tirant parti de clusters ou d'une dorsale peuvent être considérés comme des solutions tirant bénéfice d'une structure virtuelle d'auto-organisation.

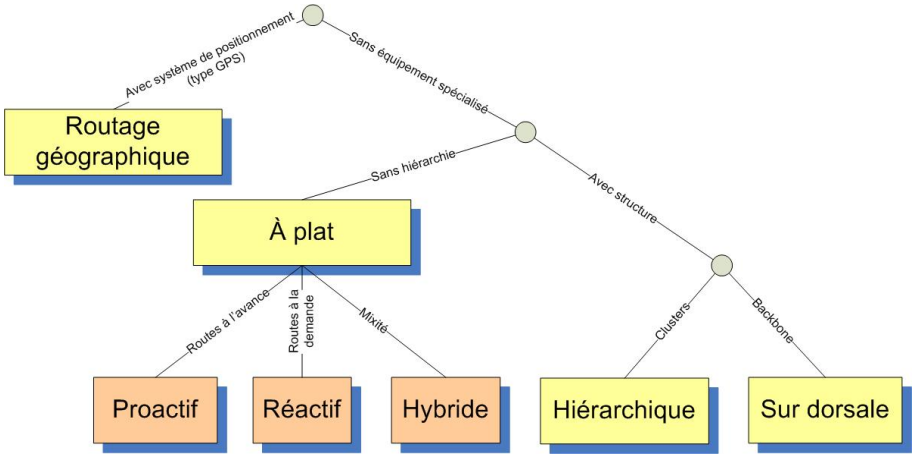


FIG. 5.1 – Proposition de taxonomie des protocoles de routage dans les réseaux ad-hoc

5.3.1 Routage à plat

Dans les réseaux ad-hoc, les terminaux sont à la fois clients et routeurs, sans organisation hiérarchique préalable. Ainsi, les travaux traitant du routage dans de tels réseaux se sont focalisés principalement sur cette nouveauté majeure : traiter de façon égalitaire tous les nœuds, considérés ainsi *à plat*. Le routage doit être réalisable sans intervention extérieure et sans hiérarchie fixée au préalable. Deux approches ont donc été proposées : celle proactive, et celle réactive. Nous présenterons ensuite des propositions essayant de combiner les avantages de ces deux approches.

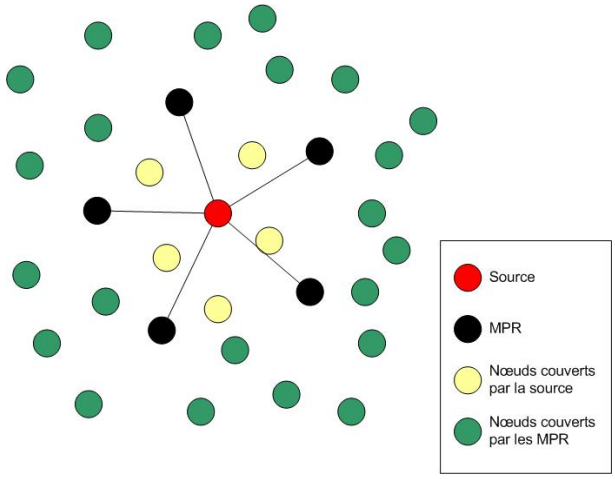


FIG. 5.2 – Sélection des multi-points relais permettant de couvrir l'intégralité du 2-voisinage

5.3.1.1 Proactif

Dans un protocole de routage proactif, chaque nœud maintient une route vers chacune des destinations possibles du réseau. Si chaque nœud envoie un paquet de présence inondé dans le réseau (comme dans OSPF), une tempête d'inondation survient, entraînant de multiples collisions et rendant le réseau inexploitable (cf. paragraphe 1.6 page 4). Les propositions se focalisent donc sur la réduction du trafic de contrôle.

DSDV [26] adopte un algorithme de type vecteur de distance. La destination maintient un numéro de séquence croissant, permettant d'éviter les boucles de routage. Cependant, de telles mises à jour sont déclenchées à chaque changement de route. Ainsi, DSDV est plus particulièrement adapté aux topologies peu changeantes. DSDV est donc plus adapté aux réseaux maillés (mesh networks).

OLSR est une approche de type états de liens optimisant la diffusion d'informations. Des paquets `hellos` contenant la liste des voisins sont émis périodiquement, permettant à chacun de construire la liste de ses 2-voisins. Grâce à cette connaissance, un nœud sélectionne des Multi-Point Relais (MPR) (fig. 5.2) : l'ensemble des MPR de S est tel que si les MPR relaient un paquet, l'intégralité du 2-voisinage de S le reçoit. Ainsi, un paquet de topologie inondé dans le réseau n'est relayé par un nœud N que s'il provient d'un nœud ayant choisi N comme MPR. Cependant, l'envoi périodique de paquets de topologie crée un trafic de contrôle important. De plus, si l'inondation est peu fiable, certaines routes ne seront pas présentes, engendrant des pertes de paquets. [1] propose une extension permettant de prendre en charge des nœuds fortement mobiles en diminuant la période d'émission des `hellos` pour de tels nœuds.

5.3.1.2 Réactif

Les protocoles réactifs représentent l'approche duale : un nœud crée une route seulement à la demande, lorsqu'il en a une utilité directe. Ainsi, un nombre réduit de routes est construit, permettant de limiter le trafic de contrôle.

Dans DSR [14], un nœud initie une découverte de route lorsqu'il possède un paquet à délivrer vers une destination pour laquelle aucune entrée ne correspond dans la table de routage. Cette `Route Request` (RREQ) est envoyée en broadcast aux voisins. Ces nœuds retransmettent le paquet, s'ils ne connaissent aucune route vers cette destination. La route est accumulée dans l'en-tête du paquet. Lorsqu'un nœud connaît la destination, il répond avec une `Route Reply` (RREP) qui copie la route contenue dans la `RREQ`. Chaque nœud relayant la `RREP` peut ainsi mettre en cache la route vers la destination. Cependant, aucun message de contrôle ne permet de maintenir proactivement la validité de la route : il faut attendre que la route casse pour

réinitier une découverte de routes, pouvant occasionner de nombreuses pertes de paquets. De plus, la (re)découverte de routes requiert un délai important avant le premier envoi de paquet, pouvant occasionner des troubles dans certaines applications.

AODV [27] propose une approche similaire, en supprimant toutefois le routage par la source. Chaque nœud maintient une table de routage distribuée, et met à jour le prochain saut à chaque réception de `RREQ` et `RREP`. Ainsi, la route n'est plus contenue dans l'en-tête, diminuant la charge sur le médium radio. AODV propose en outre la transmission périodique de `hellos` afin de maintenir la liste des voisins valides. Lorsqu'un voisin n'est plus présent, les routes passant par lui sont invalidées, notifiant les nœuds empruntant ces routes qu'ils doivent découvrir une nouvelle route. Une cassure de route est donc détectée plus rapidement, conduisant à moins de pertes de paquets. La destination maintient une numéro de séquence permettant d'éviter d'utiliser des informations obsolètes dans les caches de routage. [5] propose d'optimiser la reconstruction d'une route en limitant la recherche d'une nouvelle route à la zone proche de l'ancienne route.

5.3.1.3 Hybride

Certains auteurs ont proposé de combiner les avantages du proactif et du réactif. Dans ZRP [11], chaque nœud maintient proactivement une route vers chaque autre nœud à au plus p sauts, constituant sa *zone*. ZRP, pour découvrir vers une destination hors de sa zone, envoie une `RREQ` seulement à ses *nœuds bordure*, i.e. à exactement p sauts. Un nœud bordure relaie lui même la `RREQ` à ses propres nœuds bordure. Si un nœud est à au plus p sauts de la destination, il peut générer une `RREP` contenant comme route la liste des nœuds bordure à suivre. Cependant, les zones sont recouvrantes, puisqu'une zone n'a de signification que pour un seul nœud, contrairement aux clusters qui regroupent une liste de nœuds, liste commune pour tous les membres du cluster. Par conséquent, l'inondation peut être pire qu'une inondation aveugle quand un nœud doit relayer un `RREQ` plusieurs fois pour des nœuds bordure différents déjà couverts. De plus, le nombre de nœuds bordures augmente de façon drastique quand la densité augmente. [25] propose donc la sélection d'un sous-ensemble de nœuds bordure mais avec la connaissance du $2p + 1$-voisinage, multipliant le trafic de contrôle. [34] propose d'adapter ZRP en créant une connaissance floue pour optimiser l'inondation. Par ailleurs, les zones n'ayant de signification que pour le centre, il pourrait être intéressant d'utiliser une structure plus persistante.

Dans LANMAR [10], les auteurs proposent d'exploiter des réseaux dans lesquels des groupes ont été au préalable configurés. Un nœud landmark est élu par groupe. Le protocole maintient une connaissance proactive de sa zone locale. Parallèlement, un algorithme à vecteur de distance permet de maintenir une route vers chaque landmark. De même, si un nœud d'un groupe est en dehors de la zone de son landmark, une route spécifique est maintenue vers ce nœud grâce au même algorithme à vecteur de distance. Ce protocole semble présenter de bonnes performances. Cependant, des groupes statiques fixés au préalables sont nécessaires. Comme nous ne nous focalisons pas sur les applications militaires, un tel protocole nous semble difficilement adaptable.

5.3.2 Routage Géographique

Les protocoles de routage géographiques s'appuient sur un système de géolocalisation, tel que le GPS. LAR [17] utilise ces informations de position pour optimiser la reconstruction d'une route. L'ancienne position de la destination étant connue, chaque nœud intermédiaire choisit de relayer une `RREQ` selon la position de la destination, de la source, et selon sa propre position. Un tel choix permet de *diriger* l'inondation vers l'ancienne position de la destination.

GPSR [15] est un protocole de routage glouton tel qu'un nœud relaie un paquet de données vers le prochain saut le plus proche de la destination. Si un tel nœud n'existe pas, le paquet est acheminé le long du périmètre de la zone sans nœud. Pour que le périmètre soit bien emprunté sans boucle, il est nécessaire de rendre au préalable le graphe planaire.

Cependant, nous pensons que l'obligation d'embarquer systématiquement un GPS constitue un point bloquant pour beaucoup d'équipements embarqués bon-marché. Aussi, nous ne retenons pas une telle hypothèse.

5.3.3 Routage à large échelle

Certaines propositions se focalisent principalement sur le routage dans les réseaux à très grand nombre de nœuds, censés couvrir des villes ou des régions entières. Le projet *Terminodes* tente d'apporter une réponse à une telle problématique [2, 3].

Dans [3], les auteurs proposent que chaque nœud maintienne une liste d'amis de façon à former un graphe petit monde [16]. Les routes entre amis doivent être maintenues par un protocole de routage proactif. Lorsqu'un paquet doit être envoyé à une destination inconnue, il est envoyé au nœud ami dans le graphe petit monde le plus proche de la destination. Ce nœud s'il connaît la destination peut lui envoyer directement le paquet, sinon il réitère le même processus. Les auteurs proposent un mécanisme permettant de découvrir de nouveaux amis. [2] décrit le mécanisme permettant de découvrir la localisation associée à une destination. Un nœud inscrit sa position dans une région donnée. Si un nœud souhaite atteindre cette destination, une fonction de hachage permet de lui donner la région à interroger. Pour résoudre le problème des *trous* de nœuds dans le réseau[1], les auteurs proposent d'utiliser une route d'ancres : ces ancres constituent un point de passage obligé, accessibles grâce à un protocole de routage géographique.

Naturellement, un tel protocole découvre des routes sous-optimales en nombre de sauts. Ainsi, un tel protocole ne serait exploitable que dans les réseaux à diamètre très élevé. De même, une fonction de hachage adéquate, permettant de retourner une région couverte par de nombreux nœuds est également nécessaire, et les nœuds doivent être uniformément répartis, et de façon dense. Ce type de réseau ne constitue pas notre cas d'étude.

5.3.4 Routage Hiérarchique

CBRP [13] propose une solution de routage sur clusters. Des clusterheads sont élus et sont chargés de coordonner le routage dans leur cluster. Chaque nœud envoie périodiquement des paquets `hellos` contenant la liste de ses 1-voisins, et celle des clusters adjacents. Un clusterhead peut donc choisir un nœud passerelle vers chacun des autres clusterheads à moins de 3 sauts de lui : l'ensemble des nœuds passerelle et des clusterheads forme un sous-ensemble connexe du réseau. Cette topologie est utilisée pour les inondations des `RREQ`. Par ailleurs, lorsque la destination renvoie une `RREP`, chaque clusterhead relayant la réponse vérifie qu'il est nécessaire dans la route : les précédent et prochain sauts ne sont pas voisins l'un de l'autre. Si le nœud est inutile, il met à jour à la volée la route présente dans le paquet. Cependant, CBRP présente les désavantages suivants :

- L'ensemble des passerelles et clusterheads forme un ensemble redondant et donc crée un trafic de contrôle important comme nous le verrons dans les simulations
- La route est constituée d'une liste de nœuds, et non de clusters : il suffit qu'un seul nœud se déplace pour que la route se casse. La hiérarchie n'est pas pleinement exploitée
- Lorsqu'un nœud relayant un paquet n'obtient aucun acquittement du prochain saut, il enclenche une reconstruction locale de route en essayant d'atteindre le prochain saut via un intermédiaire. La route est donc obligatoirement allongée à chaque reconstruction.

[33] propose d'adapter DSR en utilisant une topologie de clusters. Les `RREQ` ne sont ainsi relayées que par les clusterheads et passerelles, comme dans CBRP. Par contre, les auteurs proposent d'insérer des champs spéciaux dans les paquets de données afin de réduire la charge sur le médium radio. Une telle optimisation peut être applicable à tout protocole. Cependant, là encore, toute la hiérarchie n'est, selon nous, pas pleinement exploitée.

[1] Avec un protocole de routage géographique, les *trous* rendent les algorithmes gloutons inefficaces : il n'existe aucun nœud plus proche de la destination, puisqu'il faut s'en éloigner pour contourner le *trou*

5.3.5 Routage sur dorsales virtuelles

Dans DDR [23, 22, 21], chaque nœud choisit comme père un voisin de degré supérieur, conduisant à la création d'une forêt de spanning trees. Chaque arbre forme une *zone*, implémentant un protocole de routage proactif, utilisant l'arbre pour propager les informations de topologie. Le routage inter-zones est réactif, les requêtes étant relayées aux zones voisines par les passerelles. Cependant, une telle inondation ne permet pas d'optimiser le trafic de contrôle puisque chaque nœud relaie la requête de route. De plus, une route passe obligatoirement par la topologie en forêt, augmentant donc ainsi potentiellement la longueur de la route.

[6] propose de créer une dorsale virtuelle maillée (cf. section 2.5.1.4 page 18). Dans un premier temps, un ensemble dominant est élu de façon distribuée, puis les dominants envoient des `hellos` à 3 sauts afin de découvrir les dominants voisins. La dorsale ainsi constituée peut présenter un trafic de contrôle important à cause des différents `hellos`. De plus, l'environnement étant dynamique, les liens virtuels peuvent devenir rapidement sous-optimaux. Cette dorsale est ensuite utilisée par les auteurs afin d'optimiser les inondations de paquets de topologie : seuls les nœuds de la dorsale sont habilités à retransmettre les paquets. Comme tout nœud possède au moins un voisin dans la dorsale, tous les nœuds reçoivent les paquets de topologie. Dans CEDAR [31], les auteurs proposent d'utiliser la dorsale pour inonder les changements dans la bande passante de leurs liens radio. Un mécanisme est proposé pour que seuls les liens radio stables soient propagés loin dans le réseau. La dorsale n'est pas utilisée pour router les paquets de données, afin d'éviter qu'elle constitue un goulot d'étranglement. [32] utilise cette dorsale afin d'optimiser les inondations dans AODV et DSR.

Dans [29], les auteurs proposent d'exécuter l'algorithme de Wu & Li sur la d-fermeture du graphe (un nœud à moins de d sauts est considéré comme voisin direct dans la d-fermeture). Un tel mécanisme nécessite donc la connaissance du d-voisinage. Ainsi, le protocole de routage au sein de la zone est proactif. Une implémentation distribuée utilisant une approche à vecteur de distance est proposée. Par contre, les auteurs ne précisent pas d'algorithme précis pour le routage inter-zone, présentant seulement la possibilité d'adapter un protocole à état de liens sur la topologie des dominants. De plus, aucune évaluation de performances n'est donnée.

5.4 Pourquoi proposer un protocole de routage ?

Les protocoles de routage présentés dans la section précédente utilisent massivement les inondations : les protocoles proactifs pour envoyer des paquets de topologie, les protocoles réactifs pour envoyer des demandes de routes. Or, les inondations créent dans les réseaux ad-hoc un phénomène de tempête : des collisions se produisent, diminuant la fiabilité des transmissions tant pour le trafic de contrôle que pour les paquets de données. Un protocole de routage bien conçu doit donc limiter les inondations.

Des protocoles proposent d'exploiter une structure virtuelle. Cependant, cette structure n'est jamais exploitée dans sa hiérarchie. Par exemple, CBRP n'utilise pas les clusters pour former une route de clusters, et considère donc la topologie du réseau à plat. Les clusters ne sont utilisés que pour l'optimisation des inondations. De même, les dorsales sont souvent utilisées seulement pour les inondations, alors qu'elles forment pourtant un premier niveau de hiérarchie.

Pour ces raisons, nous proposons un protocole de routage tirant bénéfice d'une structure personnelle et combinant les avantages des protocoles de routage précédemment proposés. Nous verrons dans la section suivante comment répondre aux contraintes sus-citées.

5.5 Proposition d'un protocole de routage pour les réseaux ad-hoc tirant parti d'une auto-organisation

Notre proposition, Virtual Structure Routing (VSR), se focalise ici sur le problème du routage dans un réseau ad-hoc mobile : un paquet de données doit arriver à une destination en minimisant les pertes, le délai et le trafic de contrôle. Afin d'optimiser son passage à l'échelle et ses performances, nous nous appuyons sur la structure virtuelle décrite dans le chapitre 3 page 31. VSR présente les avantages suivants :

- La dorsale permet d'introduire un premier niveau de hiérarchie : un dominé laisse agir en proxy son dominant pour toutes les découvertes de routes distantes. La topologie de la dorsale étant stable, une telle propriété est intéressante
- La dorsale permet de limiter les inondations : seuls les dominants sont habilités à relayer des inondations
- La structure en clusters permet d'introduire une hiérarchie de protocoles de routage : les protocoles au sein d'un cluster sont quelconques, et le protocole entre clusters doit être commun. Nous verrons quels protocoles nous proposons d'utiliser dans un premier temps
- Une route distante est constituée d'une suite de clusters à suivre. Comme les clusters sont construits pour leur persistance[1], la route est donc plus stable : un nœud peut se déplacer à l'intérieur d'un cluster sans que la route ne *casse*

Nous allons dans un premier temps décrire le fonctionnement général du protocole de routage proposé. Puis, nous décrirons le routage à l'intérieur d'un cluster. Enfin, le routage inter-clusters, utilisant le deuxième niveau de hiérarchie, sera détaillé.

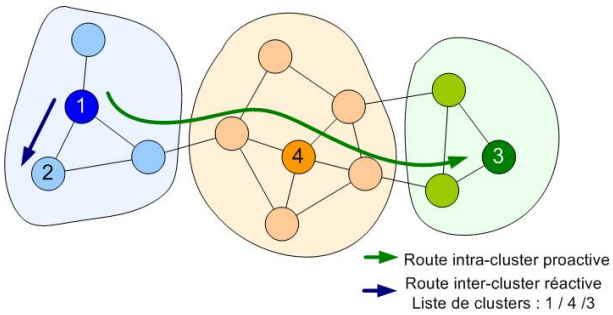


FIG. 5.3 – Description générale du protocole de routage

5.5.1 Description générale

La hiérarchie dans le réseau (dans et entre clusters) nous permet de proposer deux modes de routage différents :

- A l'intérieur d'un cluster, un protocole de routage proactif est implémenté. Les protocoles de routage et de maintien de la structure virtuelle partageant des informations communes agissent en symbiose. Plus précisément, seule une connaissance partielle du cluster est requise, permettant de restreindre l'overhead.
- Entre les clusters, un protocole réactif est implémenté. Une route distante est donnée comme une liste de clusters à suivre, la route entre les clusters étant extraite des informations proactives. La découverte de route est optimisée grâce à la dorsale afin de limiter l'overhead induit.

Ainsi, dans la figure 5.3, le nœud 1 connaît directement une route vers le nœud 2 puisque les deux nœuds se trouvent dans le même cluster. Par contre, la route entre 1 et 3 passe par la route

[1] Un cluster est dit persistant si son clusterhead et la liste de ses membres change peu au cours du temps (cf. section 2.3.6 page 12)

de clusters $(1, 4, 3)$. Une fois que 1 sait qu'il doit joindre le cluster 4, il extrait des informations tirées de son protocole de routage proactif pour le joindre. Les différentes parties du routage sont donc clairement dissociées.

Un nœud possède une table de routage intra-cluster donnant le prochain saut pour les destinations de son cluster, construite grâce au protocole de routage proactif intra-cluster. Parallèlement, un nœud possède également une table de routage inter-cluster contenant la route de clusters à suivre pour atteindre certains nœuds dans le réseau. Ce cache de routage inter-cluster est alimenté par un mécanisme de découverte de routes.

Outre les informations venant de la maintenance de la structure virtuelle (cf. section 3.1 page 41), nous récapitulons ici les informations nécessaires au routage (dont nous détaillerons le rôle et comment les obtenir un peu plus loin) :

- Routage intra-cluster
  1. le $k_{cluster}$-voisinage constituant la table de routage intra-cluster
- Routage inter-cluster
  1. la liste des passerelles à moins de $k_{cluster}$ sauts (et donc la liste des clusters adjacents)
  2. la table de routage inter-cluster constituée de routes de clusters pour atteindre une destination particulière. Cette table est alimentée à la demande à l'aide une découverte réactive de route

5.5.2 Routage intra-cluster

Nous proposons d'utiliser un protocole de routage proactif à l'intérieur d'un cluster. Si le trafic est majoritairement local, i.e. un nœud échange fréquemment des paquets avec des nœuds proches géographiquement et plus rarement avec des nœuds distants, le protocole proactif permet d'optimiser les routes les plus souvent utilisées. De plus, un tel motif de trafic a été démontré comme le seul passant à l'échelle [18]. En outre, même si une telle hypothèse n'est pas vérifiée, le protocole de routage proactif permet d'optimiser localement de longues routes : une route longue est constituée d'une liste de clusters à suivre, et la route à l'intérieur même d'un cluster est optimisée en fonction d'informations récentes, grâce aux paquets de topologies périodiques du protocole de routage proactif.

En conséquence, chaque nœud doit connaître la topologie de son cluster. Plus précisément, nous verrons qu'il n'a besoin que de la connaissance de son $k_{cluster}$-voisinage, $k_{cluster}$ étant le rayon du cluster. Nous avons choisi de fixer k_{cds} à 2, cette valeur représentant un bon compromis : avec une degré moyen de 10 dans le réseau, seuls 25% des nœuds sont des membres de la dorsale (cf. figure 3.14 page 48). Fixer cette valeur nous permet ainsi d'intégrer le protocole de routage proactif à celui de maintenance de la dorsale :

- Comme rappelé précédemment, un nœud connaît son k_{cds}-voisinage. De plus, il doit différencier les liens unidirectionnels de ceux bidirectionnels en incluant la liste des voisins qu'il entend. Si $k_{cds} = 2$, l'envoi en broadcast d'un `hello` suffit pour cette connaissance.
- Nous devons ajouter la connaissance des nœuds à $k_{cluster}$ sauts ou moins appartenant au même cluster. Ainsi, nous proposons qu'un nœud relaie un paquet `hello` s'il vient d'un voisin bidirectionnel du même cluster. Si le TTL est fixé lors de l'envoi à $k_{cluster}$, la connaissance du $k_{cluster}$-voisinage est bien maintenue.

Un nœud connaissant la topologie exacte de son $k_{cluster}$-voisinage peut exécuter un algorithme du plus court chemin, par exemple Dijkstra [8], afin de calculer les routes les plus courtes. Le protocole peut être étendu en ajoutant des informations dans les `hellos` afin de calculer des routes optimales selon des critères de bande-passante, délai, gigue... Cependant, un nœud ne connaît pas la topologie complète de son cluster. Aussi, il crée une route par défaut pointant sur son clusterhead. Lorsqu'un nœud doit relayer ou envoyer un paquet, il applique les règles suivantes :

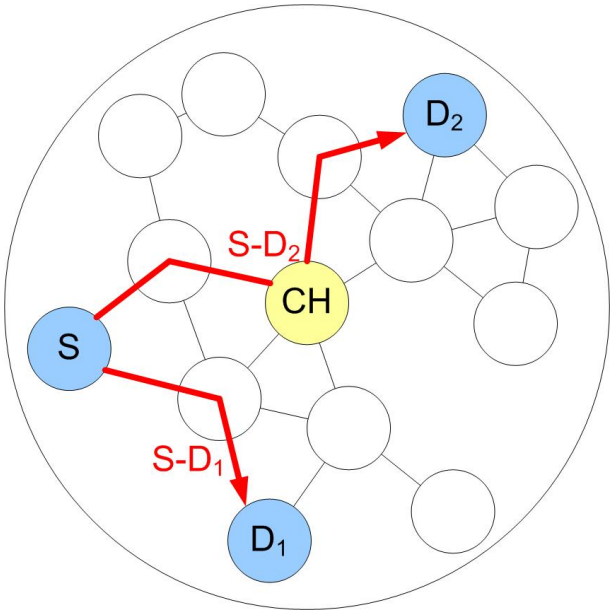


FIG. 5.4 – Routes intra-cluster dans un cluster de rayon 2 avec la connaissance du 2-voisinage

1. Une route vers la destination est connue, le paquet est donc relayé au prochain saut sur cette route
2. Aucune route n'est connue, le paquet est envoyé via la route par défaut. Comme chaque nœud est à au plus $k_{cluster}$ sauts de son clusterhead, le clusterhead connaît toutes les routes de son cluster. Ainsi, au pire, le paquet sera relayé jusqu'au clusterhead.

Nous pourrions penser qu'une telle méthode surchargerait le clusterhead. Cependant, le passage par le clusterhead est obligatoire seulement si la source et la destination sont exactement opposées dans le cluster. Or dans un tel cas, la route la plus courte en nombre de sauts passerait avec une forte probabilité via le clusterhead (fig. 5.4). En conséquence, nous pensons qu'un tel algorithme ne constitue pas un inconvénient sévère, tout en permettant de réduire le trafic de contrôle du protocole proactif à l'intérieur du cluster.

5.5.3 Routage inter-cluster

Les routes entre clusters sont découvertes par un protocole de routage réactif distinct afin d'exploiter pleinement la hiérarchie du réseau. Une route distante est donc constituée d'une liste de clusters à suivre. La topologie en clusters formant une vue macroscopique plus stable, la route cassera donc moins fréquemment. De plus, le protocole proactif nous permettra lui d'optimiser la route au sein d'un cluster.

5.5.3.1 Découverte de la topologie de clusters

Une route longue étant constituée d'une liste de clusters à suivre, un nœud relais doit trouver un nœud de son cluster, une *passerelle*, capable de relayer le paquet au cluster suivant dans la route. Nous devons donc proposer un mécanisme de découverte des clusters adjacents.

Nous avons choisi d'identifier les clusters par l'adresse de leur clusterhead. Un tel schéma d'adressage présente un intérêt évident de simplicité, et n'engendre aucun trafic de contrôle supplémentaire pour propager les identifiants de clusters. Parallèlement, nous proposons qu'un nœud envoie dans ses paquets `hellos` les identifiants de cluster différents du sien annoncés par ses voisins radio. Un tel nœud peut clairement jouer le rôle de passerelle pour les clusters annoncés. Comme un `hello` est propagé à $k_{cluster}$ sauts, tous les $k_{cluster}$-voisins de la passerelle

peuvent mettre à jour leur liste des clusters adjacents. Cependant, un cluster étant de rayon $k_{cluster}$ sauts, certains nœuds ne reçoivent pas les annonces de toutes les passerelles. Ainsi, nous utilisons là encore la route par défaut pointant vers le clusterhead. Dans le pire cas, le clusterhead recevra le paquet, et connaissant l'intégralité des passerelles puisqu'il est à au plus $k_{cluster}$ sauts de tous les nœuds de son cluster, il relaiera le paquet au cluster suivant.

5.5.3.2 Découverte de routes

Lorsqu'un nœud S souhaite envoyer un paquet vers la destination D, les cas suivants peuvent survenir :

- D est à au plus k_{cds} sauts ou fait partie du même cluster et se trouve à au plus $k_{cluster}$ sauts. Ainsi, S exécute directement son algorithme de routage intra-cluster proactif qui lui permet de donner directement le prochain saut.
- une route de clusters vers D est présente dans la table de routage. S exécutera donc directement son algorithme de routage inter-cluster pour trouver le prochain saut
- D est inconnu, S doit donc initier une découverte de route. Nous pouvons noter que si S et D sont dans le même cluster mais à strictement plus de $k_{cluster}$ sauts, S initiera une découverte de route *inutile* : S pourrait utiliser le protocole de routage proactif afin de joindre directement D. Cependant, avec un tel mécanisme, nous évitons que le clusterhead reçoive par défaut tous les paquets de données destinées à des destinations en dehors du cluster et génère lui même les découvertes de routes. Dans un tel cas, le clusterhead pourrait rapidement constituer un goulot d'étranglement dans le réseau. De plus, lorsque la source initie elle-même une découverte de route, elle peut ajouter directement la route découverte dans sa table de routage : au prochain paquet de données, le paquet sera envoyé sans délai à la destination. Naturellement, si le clusterhead centralisait toutes les découvertes de routes, il pourrait utiliser efficacement son cache de routage, mutualisé pour tout le cluster. Cependant, nous considérons le risque d'engorgement trop important.

Nous devons prendre soin de minimiser l'impact d'une découverte de route sur les performances du réseau : la charge au niveau du médium radio doit être minimisée. La dorsale permet de répondre à un tel objectif. La source S génère une `Route Request` (`RREQ`). Si S est un dominé, il l'envoie directement vers son dominant. Le premier dominant ajoute l'adresse de son clusterhead dans la route contenue dans l'en-tête du paquet. Puis il envoie la `RREQ` en multicast à ses voisins membres de la dorsale, initiant une inondation de la dorsale. Un dominant *Dom* recevant une demande de route regarde si D est présent dans sa table de routage intra-cluster :

- Si D est inconnu, *Dom* ajoute l'adresse de son clusterhead si elle n'est pas déjà présente dans la route de clusters de l'en-tête. Puis il envoie le paquet en multicast pour continuer l'inondation de la dorsale
- Si D est présent, *Dom* ajoute son clusterhead et le clusterhead de D dans la route de clusters s'ils n'y sont pas déjà présents. Ensuite, il génère une `Route Reply` (`RREP`) et y copie la route de clusters du `RREQ` après l'avoir inversée. Finalement, *Dom* envoie la `RREP` en unicast à S en exécutant l'algorithme de routage inter-cluster, décrit un peu plus loin. *Dom* agit donc en tant que proxy pour répondre à une demande de route. Puisqu'un nœud est connu à au moins k_{cds} sauts, un tel mécanisme permet d'économiser $2k_{cds}$ transmissions radio (de et vers la destination, k_{cds} sauts dans chaque direction).

Exemple Prenons l'exemple de la topologie représentée sur la figure 5.5 page suivante (Pour des raisons de clarté d'explication, nous avons choisi dans cet exemple le cas $k_{cds} = k_{cluster} = 1$). La source 1.1 souhaite envoyer un paquet de données à la destination 3.1. De plus, 1.1 ne connaît aucune route vers 3.3. Il place le paquet de données dans sa file d'attente et envoie un `Route Request` en unicast vers son dominant, 1. Le paquet contient une route de clusters vide. 1 reçoit le paquet et ajoute son clusterhead dans la route contenue dans l'en-tête. Puis, il relaie la demande de route en multicast aux autres membres de la dorsale. 4 relaie la `RREQ` après avoir

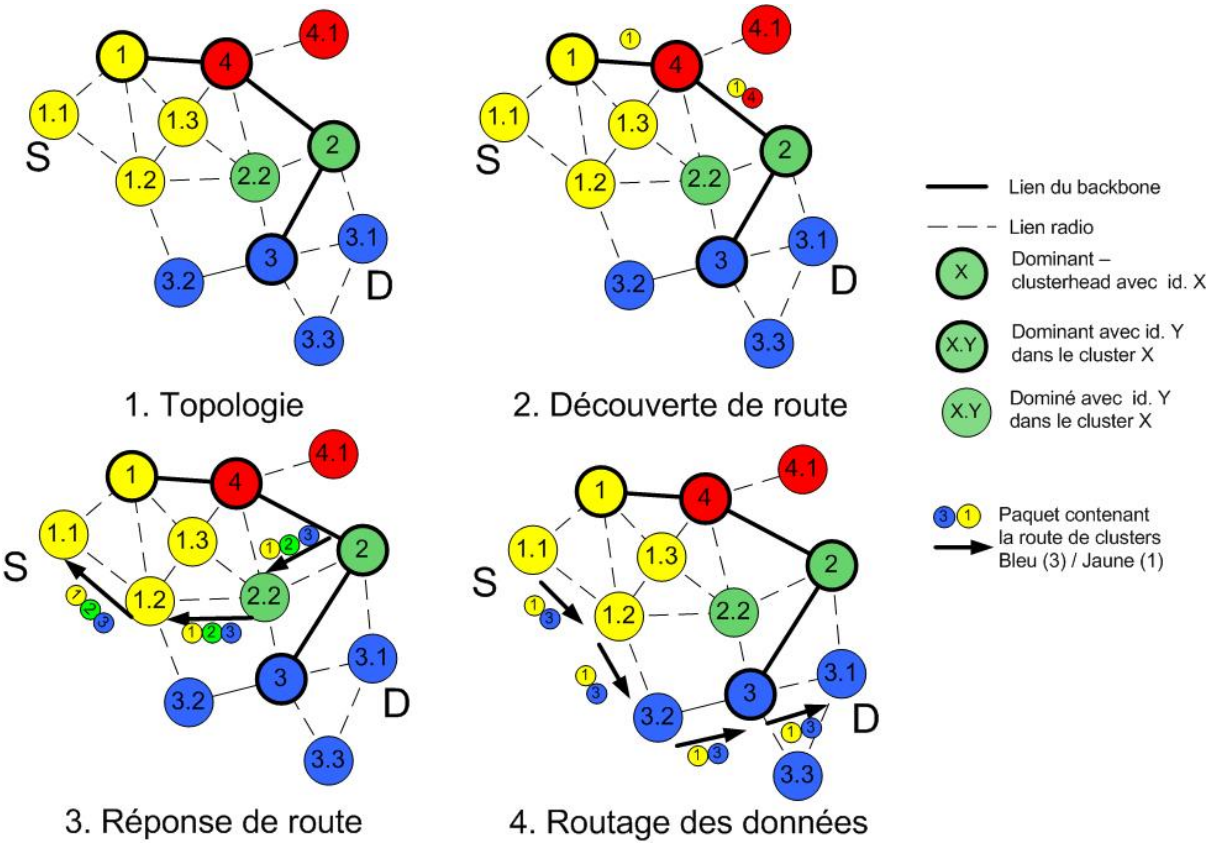


FIG. 5.5 – Description des différentes étapes lors du routage inter-cluster à travers un exemple ($k_{cds} = k_{cluster} = 1$)

ajouté son adresse dans la route de clusters. Finalement, 2 reçoit la requête et se rend compte que 3.1 est présent dans sa table de routage : 3.1 est un voisin radio. 2 ajoute son clusterhead et le clusterhead de 3.1 dans la route. Puis il génère une `RREP` avec la route 3/2/4/1. La `RREP` est ensuite envoyée grâce à la procédure de routage.

Particularités Dans VSR, et contrairement à AODV ou DSR, les paquets de demandes de routes sont uniquement relayés par les nœuds de la dorsale, permettant d'éviter de nombreuses transmissions radio inutiles, chargeant le médium. De plus, la dorsale que nous utilisons forme un arbre. Ainsi, un dominant répondant à un `RREQ` stoppe la propagation de la requête de route dans sa branche. Dans une approche plus classique, une requête de route peut passer par un grand nombre de chemins, et ne peut donc être stoppée dans sa progression : même si une réponse de route est générée par un voisin proche, la requête parcourra le réseau dans son entier, ce qui n'est pas le cas avec notre approche.

Nous avons choisi une approche de routage par la source pour le routage inter-cluster. En effet, si la route de clusters était mise en cache par les nœuds intermédiaires, il n'y aurait aucune possibilité d'adapter à la volée la route au sein d'un cluster : les nouveaux nœuds relais ne posséderaient pas la route de clusters dans leur cache. De plus, une route de clusters contient un nombre réduit de clusters. En conséquence, nous pensons que l'overhead généré est négligeable.

5.5.3.3 Routage

Les paquets de données et les `Route Replies` sont envoyés en unicast, grâce à la route de clusters contenue dans l'en-tête de chaque paquet. Avant de relayer un `RREP`, un nœud devrait ajouter la route de clusters du paquet dans sa table de routage, afin de réduire le nombre de

demandes de routes futures. Lorsque le nœud N_1 dans le cluster C_1 reçoit un paquet à relayer, il exécute l'algorithme de routage intra-cluster : si un prochain saut est trouvé, le paquet est directement envoyé. Sinon, l'algorithme de routage inter-cluster est exécuté. N_1 cherche le cluster connu C_2 le plus proche de la destination. Il applique les règles dans l'ordre suivant :

1. Un voisin N_2 est dans le cluster C_2 : N_2 constitue le prochain saut
2. Un voisin $N2$ s'annonce comme passerelle pour C_2 : N_2 constitue le prochain saut
3. Un nœud dans le cluster C_1 présent dans la table de voisinage de N_1 s'annonce passerelle pour C_2. L'algorithme de routage intra-cluster est ainsi exécuté afin de trouver le prochain saut N_2 vers cette passerelle. Finalement, N_2 constitue le prochain saut

Si aucun cluster connu n'est trouvé, le paquet est relayé vers le prochain saut en direction du clusterhead. Si N_1 est lui même un clusterhead, alors le paquet est supprimé silencieusement. Afin d'éviter les boucles de routage, un nœud ne peut relayer un paquet en direction du cluster C_{next} seulement si C_{next} est dans la route plus proche de la destination que C_1. En conséquence, le paquet est toujours relayé un saut et/ou un cluster plus proche de la destination.

Cependant, des incohérences dans les tables de voisinages et donc dans la table de routage intra-cluster peuvent survenir [35]. Les paquets `hellos` n'étant envoyés que périodiquement, un changement de topologie n'est pas instantanément propagé à tous les autres nœuds. De même, un paquet `hello` peut souffrir de délai et de collisions. En conséquence, un paquet est supprimé silencieusement s'il a déjà été relayé auparavant afin d'éviter les boucles de routage. Les duplicata de paquets sont détectés grâce aux champs source et id.

Un nœud choisit de relayer le paquet au cluster adjacent le plus proche de la destination dans la route de clusters. En conséquence, l'ordre des clusters de la route n'est pas obligatoirement suivi, permettant ainsi de raccourcir la route de clusters. Lorsqu'un nœud relaie un paquet et calcule une plus courte route de clusters, il met à jour la route dans l'en-tête du paquet. Si le paquet est une `RREP`, la source pourra mettre à jour sa route. Si au contraire, le paquet contient des données, seule la destination pourra mettre à jour sa route. Cependant, lorsqu'elle répondra à la source, la nouvelle route sera utilisée, et notifiée à la source.

Finalement, la route n'est pas une plus courte route. Cependant, nous essayons en choisissant de relayer au cluster le plus proche de la destination d'optimiser au mieux sa longueur. En conséquence, la route calculée par nos algorithmes est peu éloignée d'une plus courte route, comme le corroborent les résultats des simulations dans la section suivante.

La route de clusters, fixée par la source, est contenue dans l'en-tête du paquet. Par contre, la route saut par saut est calculée à la volée, utilisant les informations les plus récentes. Le délai de convergence afin d'avoir des informations cohérentes et valides diminue lorsque la destination est plus proche. Ainsi, la route sera plus robuste. Si de nombreux nœuds se déplacent mais que la route de clusters reste tout de même valide, la route sera mise à jour à la volée et finalement délivrée à la destination. La topologie de clusters étant plus stable, les routes sont donc plus robustes à la mobilité.

Exemple Reprenons l'exemple de la figure 5.5 où nous l'avions laissé dans l'exemple précédent : le nœud 2 doit envoyer la `RREP` à 1.1. 2 essaie d'atteindre le cluster 1. Un voisin, 2.1, s'affiche comme passerelle pour ce cluster : il le choisit donc après avoir mise à jour la route dans le paquet. La route est donc devenue 3/2/1. 1.2 reçoit la réponse, qu'il relaie directement à 1.1, qu'il connaît grâce à son algorithme de routage proactif.

Finalement, 1.1 reçoit la `RREP` avec la route 1/2/3 pour joindre 3.1. Il met en cache cette route et envoie le paquet de données après avoir copié la route dans son en-tête. 1 essaie d'atteindre directement le cluster 3 : il trouve la passerelle 1.2. Ainsi, 1.1 met à jour la route dans l'en-tête du paquet, et dans son cache de routage puisqu'il est la source. La route de clusters est donc devenue 1/3. 1 envoie le paquet de données à 1.2 qui le relaie à 3.2. 3.2 ne connaît pas la destination 3.1 qui se trouve à strictement plus de $k_{cluster}$ sauts. De plus, il n'existe pas d'autres

clusters dans la route. Ainsi, il envoie le paquet vers son clusterhead, 3. Finalement, 3 envoie le paquet de données à son voisin 3.1. 3.1 reçoit le paquet et met dans son cache de routage la route 1/3 pour joindre 1.1.

5.5.4 Maintenance de la route

Nous pensons que le taux de livraison est un critère majeur de performances pour un protocole de routage. En conséquence, nous proposons un mécanisme très simple de réparation de route. Plusieurs méthodes d'acquittements sont possibles :

- acquittement MAC : la couche MAC, lorsqu'un envoi échoue car elle n'a reçu aucun acquittement, envoie une notification aux couches supérieures. Une telle méthode n'est actuellement pas implémentable, le firmware des cartes réseaux étant pour la plupart non modifiable.
- acquittement passif : chaque nœud N est en mode *promiscuous* et vérifie que le prochain saut relaie bien le paquet. Si au bout d'un timeout, N n'a détecté aucun paquet relayé, il renvoie le paquet. Si pour une quelconque raison, le saut suivant a relayé le paquet mais que N ne l'a pas entendu, il envoie à N un paquet d'acquittement. De même, le dernier saut dans la route doit envoyer un `acquittement`.
- acquittement actif : lorsqu'un nœud reçoit un paquet, il envoie automatiquement un acquittement au précédent saut. Une telle méthode crée une charge importante sur le médium.

Nous supposons qu'un tel mécanisme d'acquittement des paquets est disponible. Lorsqu'un nœud ne reçoit aucun acquittement, il initie une réparation de route. Il re-exécute juste l'algorithme de routage en interdisant comme prochain saut le nœud défaillant, considéré comme mort. Un tel mécanisme de réparation de route permet de limiter l'impact du délai de convergence des tables de voisinage lorsqu'un changement de topologie survient. Il permet d'augmenter de façon significative le taux de livraison, comme montré au cours des simulations dans la section suivante.

Lorsqu'un nœud échoue à réparer la route, ou que max_{echec} réparations infructueuses ont été tentées, la route est considérée comme cassée. Le nœud envoie donc une `Route Error` (`RERR`) en unicast en direction de la source en exécutant l'algorithme de routage inter-cluster. Tous les nœuds intermédiaires et la source doivent mettre à jour leur table de routage. Finalement, la source reçoit la `RERR`, met en file d'attente les paquets de données suivants dans le flux, et initie une nouvelle découverte de route.

5.5.5 Remarque

Nous insistons sur le fait que le protocole proposé ici devrait être considéré comme un *framework* de routage. Au lieu de proposer un nouveau protocole de routage, nous avons adapté des protocoles réactifs et proactifs existants afin qu'ils tirent pleinement parti d'une structure d'auto-organisation. Par exemple, le protocole de routage intra-cluster est un protocole proactif quelconque. De plus, VSR pourrait parfaitement utiliser des protocoles proactifs différents dans deux clusters distincts. Il est donc envisageable d'utiliser un protocole spécifique à chaque cluster, adapté aux propriétés locales de la zone[1].

De même, le protocole inter-cluster peut également être modifié : une approche proactive pourrait peut être s'avérer adaptée à certaines situations (notamment dans des environnements très faiblement mobiles). Par contre, dans un tel cas, tout nœud doit connaître la topologie des clusters et une liste des membres de chaque cluster. En d'autres termes, de telles informations proactives ne peuvent être détenues seulement par les nœuds de la dorsale. Par ailleurs, pour

[1] DSDV pourrait être utilisé par exemple pour les clusters comportant des nœuds très faiblement mobiles, OLSR pour les clusters à forte densité afin d'optimiser les inondations des paquets `hellos` à $k_{cluster}$ sauts, un flooding aveugle pour les zones à faible densité et très forte mobilité.

établir des routes inter-cluster réactives, un routage par la source est obligatoire puisque les requêtes de routes ne suivent pas le même chemin que les paquets de données.

5.6 Évaluation de performances

Nous présentons ici des résultats de simulation utilisant OPNET Modeler [24]. Nous avons utilisé le modèle IEEE 802.11 avec la portée radio standard de 300m, le mode DCF, sans RTS/CTS. Chaque nœud se déplace suivant le random waypoint mobility model, sans pause. Tous les résultats mesurés sont reportés avec leur intervalle de confiance de 95%. Nous considérons comme génériques une vitesse de 5m.s^{-1}, 40 nœuds, un degré de 10, et 4 flux simultanés.

La génération de trafic est modélisée comme des flux de 20 paquets, espacés de 0.25 secondes. Pour chaque flux, une source et une destination sont choisies aléatoirement. Le temps inter-flux suit une distribution exponentielle centrée sur cinq secondes afin d'obtenir en moyenne le nombre de flux simultanés choisis. La taille des paquets suit une distribution exponentielle centrée sur 128 octets.

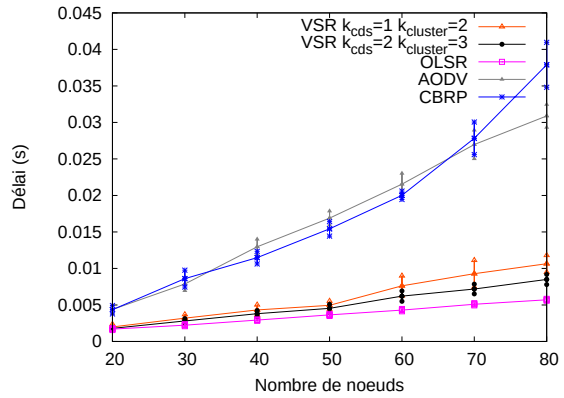
Les résultats détaillent la pertinence du *framework* de routage proposé. En particulier, le passage à l'échelle horizontal et vertical, l'impact de la mobilité et du degré, et l'overhead sont étudiés. Nous comparons les performances de VSR avec celles d'OLSR, CBRP et AODV sous les mêmes conditions. OLSR et AODV sont les deux protocoles de routage standardisés par l'IETF, le premier étant proactif et le deuxième réactif. CBRP est lui un protocole hiérarchique permettant d'exploiter une topologie de clusters. Pour évaluer l'impact de la structure virtuelle utilisée par VSR, nous avons choisi de simuler VSR avec k_{cds}=1 / $k_{cluster}$=2 et k_{cds}=2 / $k_{cluster}$=3. Afin d'avoir une comparaison équitable des protocoles, les retransmissions et réparations de routes ont été désactivées dans VSR et CBRP. Nous considérons que les points suivants représentent des métriques essentielles d'évaluation pour le routage :

1. Le taux de livraison : le pourcentage de paquets arrivant effectuant à la destination
2. Le délai de bout en bout : le temps séparant la génération du paquet de sa réception par la destination. La latence de découverte d'une route peut également être intéressante pour les protocoles réactifs puisqu'elle impacte directement le délai de bout en bout
3. Le trafic de contrôle, diminuant d'autant la bande passante disponible pour les paquets de données

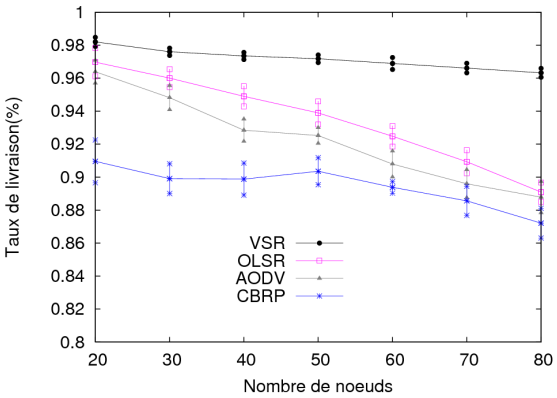
5.6.1 Passage à l'échelle horizontal

Dans un premier temps, nous étudions le passage à l'échelle horizontal, i.e. l'impact du nombre de nœuds sur les performances des différents protocoles de routage. Le délai de bout-en-bout, i.e. le délai entre la réception par la couche routage d'un paquet à envoyer et la réception par la destination de ce paquet, est tout d'abord mesuré (fig. 5.6(a) page ci-contre). Le délai d'AODV et CBRP est plus important et augmente lorsque le nombre de nœuds augmente : les protocoles étant réactifs, plus de nœuds doivent relayer les `RREQ`. Ainsi, le temps de découverte d'une route augmente, impactant le délai. Au contraire, le délai d'OLSR est minimal : le protocole étant proactif, une route est disponible immédiatement. La longueur moyenne de la route tend à augmenter avec le nombre de nœuds puisque le degré est constant, cependant, l'impact sur le délai de bout-en-bout est assez faible. **VSR, quel que soit le rayon de la dorsale, présente un délai stable** : la dorsale permet d'optimiser efficacement les inondations de découvertes de routes, moins de nœuds relayant les requêtes. La structure en clusters permet de maintenir une hiérarchie dans le réseau sans impacter grandement le délai.

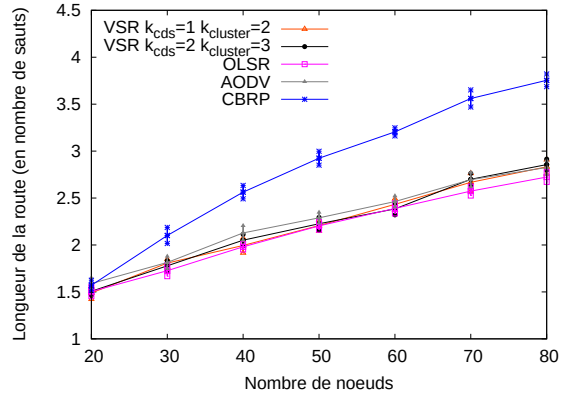
Lorsque nous étudions la longueur moyenne des routes créées par les protocoles (fig. 5.6(c) page suivante), CBRP présente la longueur la plus importante. En effet, la découverte de routes suivant la topologie des clusterheads et passerelles entre clusters, la route est sous-optimale. De



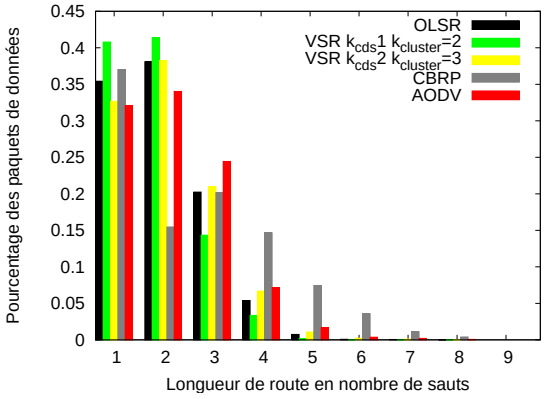
(a) Impact sur le délai



(b) Impact sur le taux de livraison



(c) Impact sur la longueur de la route



(d) La loi de distribution de la longueur des routes utilisées

FIG. 5.6 – Passage à l'échelle horizontal

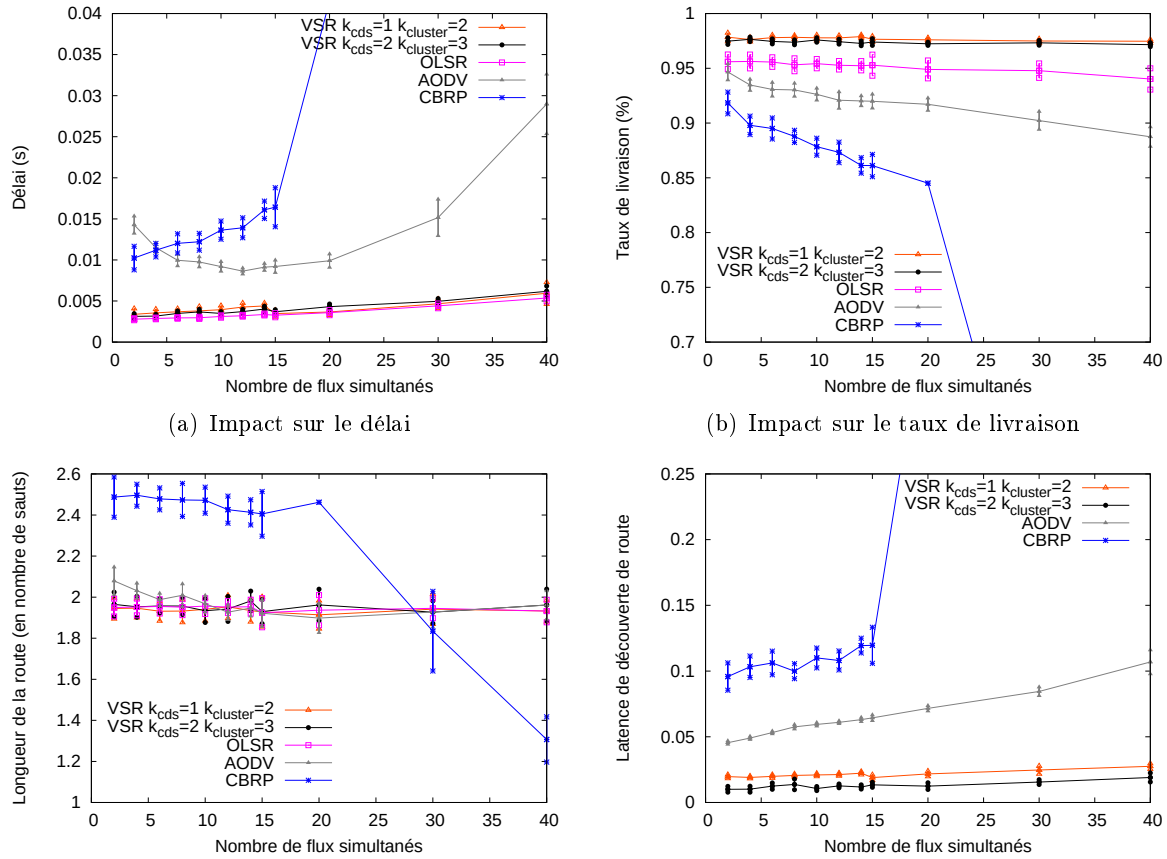
plus, le mécanisme de suppression de redondance lorsqu'un clusterhead relaie une réponse de route ne semble pas efficace. AODV est un protocole réactif, et quelquefois une route autre que la plus courte est créée. Cependant, la longueur moyenne est très inférieure à celle de CBRP. OSLR découvre toujours la plus courte route puisqu'il est un protocole proactif à états de liens. La longueur des routes découvertes par VSR semble indépendante des paramètres k_{cds} et $k_{cluster}$. De plus, la longueur semble très proche de la longueur calculée par OLSR : le mécanisme de relais au cluster le plus proche de la destination semble efficace pour découvrir des routes courtes. **La hiérarchie de clusters peut donc être pleinement exploitée sans pour autant augmenter la longueur moyenne de la route**. Sachant qu'une route plus longue crée en moyenne plus d'interférences et augmente le délai et la probabilité de perte de paquets, ceci constitue une propriété forte.

Ces remarques sont corroborées par la figure 5.6(d) page précédente : la proportion de routes qui sont longues d'exactement x sauts est reportée sur le graphe pour chaque protocole (x variant de 1 à 9). OLSR et VSR (quel que soit le rayon de la dorsale) présentent une distribution très similaire : les deux protocoles arrivent à découvrir des routes courtes. AODV tend à découvrir des routes légèrement plus longues, mais la distribution des longueurs est très proche de celle d'OLSR. Par contre, CBRP découvre des routes beaucoup plus longues : tandis qu'OLSR n'a quasiment aucune route de plus de 5 sauts, 12,6% des routes découvertes par CBRP sont plus longues que 5 sauts. Ainsi, pour les mêmes raisons que citées précédemment, CBRP subira sans doute plus de pertes de paquets.

Enfin, le taux de livraison des paquets, i.e. le pourcentage de paquets arrivant bien à destination, est mesuré (fig. 5.6(b) page précédente). CBRP semble comme prévu souffrir d'une augmentation du nombre de nœuds à cause de sa découverte sous-optimale de routes : le trafic de contrôle est trop important, et les routes découvertes trop longues. Le taux de livraison d'AODV est plus important, mais la découverte de route crée un overhead important lorsque le réseau comporte trop de nœuds. Avec 80 nœuds, seuls 90% des paquets arrivent à destination. OLSR présente un taux de livraison plus élevé mais souffre lui aussi d'une forte cardinalité : l'inondation des paquets de topologie est requise, créant un important trafic de contrôle. A cause du manque de fiabilité, certains paquets de topologie n'arrivent pas, ne créant ainsi aucune entrée dans la table de routage. Ainsi, certains paquets de données sont supprimés. **VSR présente le taux de livraison le plus important : la hiérarchie permet d'exécuter différents protocoles de routage, augmentant le passage à l'échelle**. De même, la dorsale permet de réduire le trafic de contrôle. Enfin, la topologie de clusters est stable : moins de cassures de routes surviennent, diminuant les pertes de paquets. La structure virtuelle permet d'optimiser les performances d'un protocole de routage classique : même avec 80 nœuds, plus de 97% des paquets arrivent à destination.

5.6.2 Passage à l'échelle vertical

Nous mesurons ensuite l'impact de la charge réseau sur les performances. OLSR et VSR présentent le plus faible délai (fig. 5.7(a) page ci-contre), invariant vis à vis du nombre de flux simultanés. Le délai de VSR est légèrement plus faible lorsque k_{cds}=2/$k_{cluster}$=3 que quand k_{cds}=1/$k_{cluster}$=2 : un cluster comprend plus de nœuds, la part de routage proactif est donc plus important. Le nombre de découvertes de routes est donc moins important, diminuant le délai. Le délai d'AODV au début décroît quand le nombre de flux augmente : plus de `RREQ` sont envoyées dans le réseau, plus d'entrées sont créées dans les tables de routage. Ainsi, la probabilité qu'une entrée existe déjà augmente, supprimant le délai occasionné par la découverte de routes. Par contre, trop de connexions engendrent une forte charge sur le médium radio et augmentent le délai. CBRP semble souffrir du routage par la source : le mécanisme d'inondation crée un overhead important, occasionnant des pertes de `RREQ`, augmentant le délai de découverte de routes. Nous pouvons observer qu'une tempête de broadcast apparaît dés que le réseau supporte plus de 20 connexions. Cette remarque est corroborée par la figure 5.7(d) page suivante. La latence



(a) Impact sur le délai

(b) Impact sur le taux de livraison

(c) Impact sur la longueur de la route

(d) Latence de la découverte de route

FIG. 5.7 – Passage à l'échelle vertical

de découverte de routes est maximale avec CBRP. La latence d'AODV augmente légèrement quand plus de découvertes de routes sont réalisées (à cause de la charge du médium et donc des collisions). Cependant, le délai de bout-en-bout d'AODV décroît puisque le nombre de découverte de routes diminue. **VSR présente la latence de découverte de route la plus stable et la plus faible** : la dorsale est bien exploitée pour l'inondation.

Finalement, nous avons étudié le taux de livraison des différents protocoles de routage (fig. 5.7(b) page précédente). CBRP présente le plus faible taux de livraison car beaucoup de découvertes de routes sont infructueuses à cause des collisions. Nous pouvons vérifier que le taux de livraison chute brutalement avec plus de 20 connexions, une tempête de broadcast survenant dans de telles conditions. AODV présente un taux de livraison plus élevé mais ses performances décroissent avec le nombre de connexions : l'overhead des découvertes de routes semble charger le médium radio. OLSR présente un taux de perte de paquet stable puisqu'il est proactif. Enfin, **VSR présente le plus haut taux de livraison grâce à l'utilisation de routes stables**, avec un overhead faible grâce à la structure virtuelle d'auto-organisation.

Enfin, nous pouvons vérifier que la longueur de la route est peu impactée par la charge réseau (fig. 5.7(c) page précédente). La longueur de route de CBRP chute brutalement lorsque la tempête de broadcast se produit : les découvertes de routes sont infructueuses, et seules les communications à 1 saut peuvent être acheminées, diminuant artificiellement la longueur moyenne de routes.

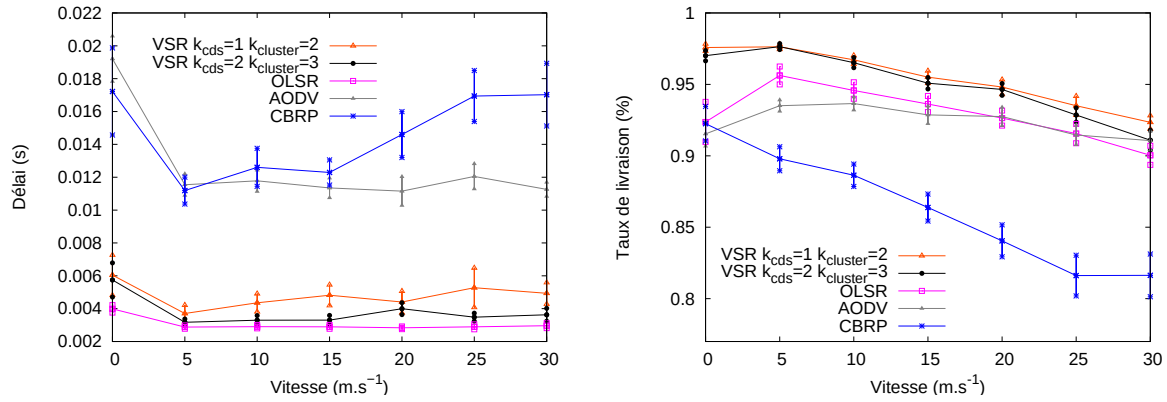
5.6.3 Impact de la mobilité

Comme dans un réseau ad-hoc tous les nœuds sont mobiles, nous avons également étudié l'impact de la mobilité en faisant varier la vitesse maximum du random waypoint mobility model de 0 à 30m.s^{-1}. Le délai d'OLSR est stable avec la mobilité (fig. 5.8(a) page ci-contre) : les protocoles proactifs permettent de mettre à jour périodiquement les informations de topologie. Ainsi, aucun délai n'est requis pour l'envoi. Le délai de VSR est stable : la découverte de routes à travers la dorsale est efficace. Le délai d'AODV est plus important, mais ne souffre pas de la mobilité. Enfin, le délai de CBRP semble augmenter lorsque la vitesse est élevée : les routes sont peu stables, générant de nombreuses redécouvertes de routes.

Les mêmes remarques peuvent être faites sur le taux de livraison (fig. 5.8(b) page suivante) : CBRP souffre de pertes de paquets, présentant le plus faible taux de livraison, et son efficacité décroît rapidement lorsque le nombre de changements de topologie augmente. Le taux de livraison de tous les protocoles de routage décroît quand la mobilité est plus importante : les changements de topologie créent des cassures de route. Cependant, **VSR garde le taux de livraison le plus élevé grâce à des routes de clusters stables**. AODV et OLSR tendent à présenter le même taux de livraison pour des mobilités élevées.

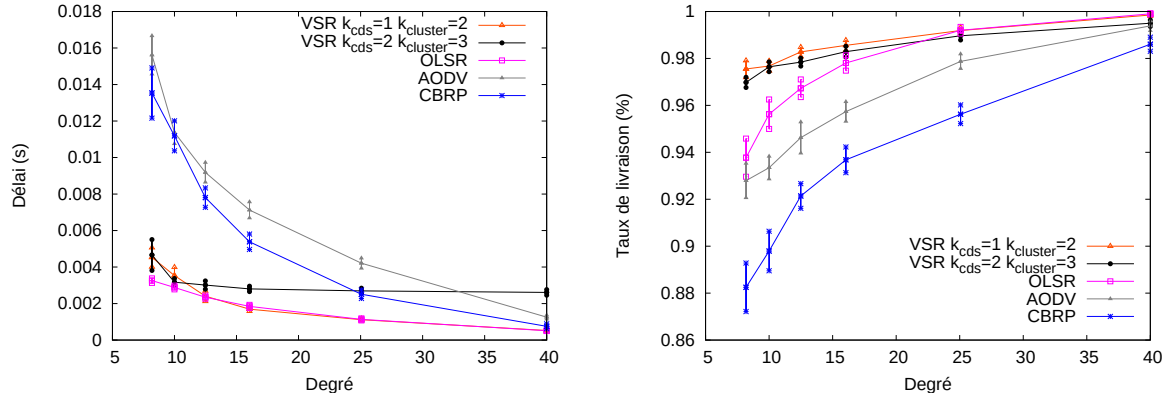
5.6.4 Impact de la densité

Nous pouvons noter que le délai et les pertes de paquets diminuent lorsque la densité augmente (fig. 5.9(a) page ci-contre et 5.9(b) page suivante). En effet, le nombre de nœuds étant constant, le diamètre diminue, réduisant la longueur des routes. Lorsque le réseau est de rayon un (toutes les routes sont en simple saut), tous les protocoles semblent bien réagir. Par contre, pour des faibles densités, les protocoles réactifs présentent un délai plus important, les découvertes de routes étant plus longues. De la même façon, le taux de livraison est plus faible. La cause d'un tel phénomène est peut être le manque de fiabilité des inondations : des `RREQ` sont perdues car le réseau est très éparse, et la redondance de la découverte de routes est trop faible. Le délai de VSR augmente lorsque le degré diminue, la route découverte étant plus longue, mais reste assez similaire à celui d'OLSR. VSR continue à présenter le plus faible taux de pertes de paquets, quel que soit le degré. Finalement, lorsque $k_{cluster}>2$, VSR présente un délai élevé dans les réseaux très denses : les paquets `hello` sont relayés par tous les nœuds du cluster. Ainsi, lorsque le



(a) Délai (b) Taux de livraison

FIG. 5.8 – Impact de la mobilité

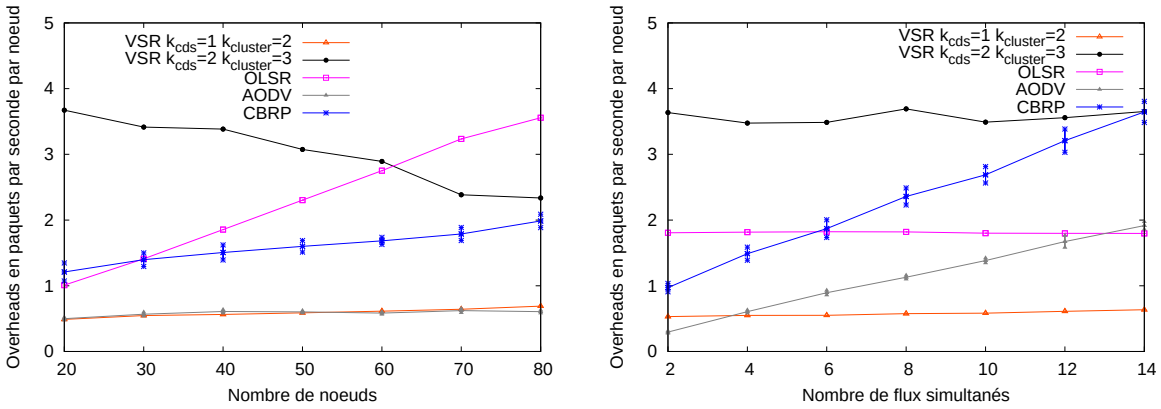


(a) Délai (b) taux de livraison

FIG. 5.9 – Impact de la densité

	Part proactive			Part réactive		Total
	Hellos	Paquets de topologie	Structure virtuelle	RREQ	RREP	
VSR k_{cds}=1 $k_{cluster}$=2	0.25	N/A	0.26	0.06	0.028	0.59
VSR k_{cds}=2 $k_{cluster}$=3	3.1	N/A	0.27	0.009	0.005	3.4
OLSR	0.4	1.4	N/A	N/A	N/A	1.8
AODV	N/A	N/A	N/A	0.39	0.22	0.61
CBRP	0.49	N/A	N/A	0.9	0.09	0.99

TAB. 5.1 – Trafic de contrôle en paquet par nœud par seconde



(a) Passage à l'échelle horizontal

(b) Passage à l'échelle vertical

FIG. 5.10 – Overheads des différents protocoles de routage

réseau est en simple saut, tous les nœuds relaient une fois les `hellos` de tous les autres nœuds, conduisant à une tempête de broadcast. Cependant, un contrôle de topologie peut être maintenu afin de garder un degré acceptable (cf. section 2.5.3 page 22). De même, OLSR pourrait être déployé dans un cluster afin d'optimiser l'overhead dû à la part proactive du routage.

5.6.5 Overhead

L'overhead généré par les différents protocoles de routage est ensuite étudié. Nous avons mesuré l'overhead en nombre de paquets par seconde par nœud. Dans un premier temps, nous avons séparé tous les types de paquets de contrôle afin de comprendre finement la provenance du trafic de contrôle (tab. 5.1). VSR présente une part proactive importante, mais permettant de réduire grandement l'overhead généré par la part réactive. **VSR présente le plus faible overhead lorsque k_{cds}=1/$k_{cluster}$=2** : seule la connaissance du 2-voisinage est requise, ainsi les `hellos` ne sont pas relayés, réduisant de beaucoup l'overhead généré. La connaissance de la topologie du cluster étant partielle, l'overhead global est acceptable. Lorsque k_{cds}=2/$k_{cluster}$=3, l'overhead est trop important : OLSR doit être implémenté afin de réduire la quantité de trafic de contrôle. Inversement, la part réactive diminue lorsque $k_{cluster}$ augmente : moins de découvertes de routes sont nécessaires. OLSR est proactif et requiert l'inondation périodique de paquets de topologie. Ainsi, l'overhead est important : 1.4 paquets par seconde par nœud sont requis pour une connaissance totale de la topologie du réseau. AODV et CBRP présentent un overhead raisonnable lorsque peu de trafic transite sur le réseau. Cependant, nous pouvons voir que la découverte de routes de CBRP est moins efficace car une `RREQ` doit être quelquefois relayée plusieurs fois par un même nœud vers des passerelles différentes.

Enfin, nous étudions l'impact du nombre de nœuds sur l'overhead (fig. 5.10(a)). AODV passe bien à l'échelle : lorsque le réseau véhicule peu de trafic, l'inondation des `RREQ` n'augmente pas de façon drastique l'overhead. CBRP, à cause de ses inondations via les clusterheads et passerelles,

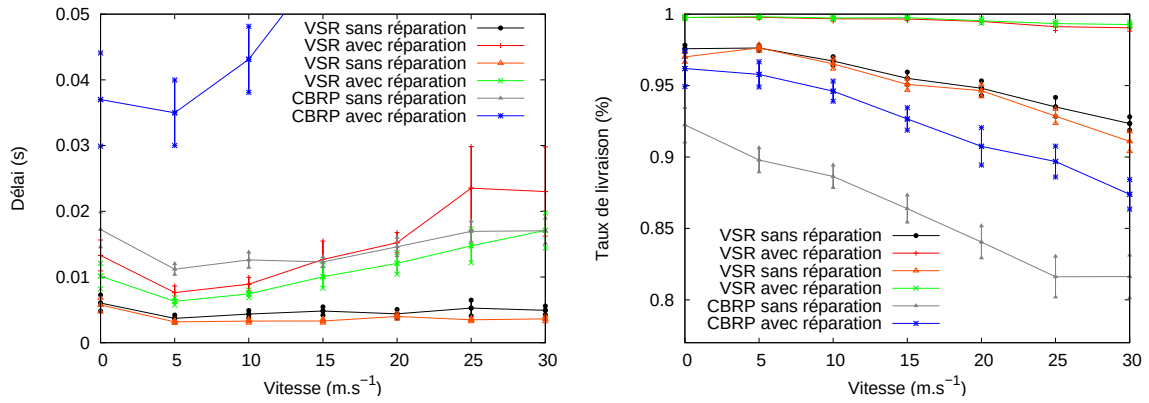


FIG. 5.11 – Impact du mécanisme de réparation de routes

voit son overhead augmenter lorsque le nombre de nœuds augmente. L'overhead d'OLSR augmente : plus de nœuds doivent envoyer des paquets de topologies, envoyés dans le réseau entier. Finalement, l'overhead passe à l'échelle, proposant un compromis efficace proactif / réactif grâce à la hiérarchie introduite par la structure virtuelle. VSR présente le plus faible overhead pour k_{cds}=1.

Nous avons ensuite étudié l'impact de la charge réseau (fig. 5.10(b) page précédente). L'overhead d'OLSR est stable car il est proactif. L'overhead d'AODV et CBRP augmente : plus le nombre de connexions est important, plus le nombre de découvertes de routes augmente. L'overhead d'AODV dépasse celui d'OLSR dés que nous avons plus de 13 connexions dans le réseau. L'overhead de CBRP est toujours plus élevé que celui d'AODV. Enfin, l'overhead de **VSR présente un bon passage à l'échelle** : la diffusion des `RREQ` semble efficace.

5.6.6 Efficacité du mécanisme de réparation de routes

Finalement, nous avons étudié l'impact du mécanisme de réparation de routes (fig. 5.11). Pour avoir l'approche la plus générique possible, nous avons supposé qu'aucune mécanisme d'acquittement particulier n'était possible. Aussi, chaque nœud doit envoyer explicitement un paquet d'`acquittement` à chaque réception d'un paquet reçu en unicast. Si aucun acquittement n'est reçu au bout de trois transmissions, le prochain saut est considéré comme mort, et une tentative de réparation de route est réalisée.

Nous avons comparé l'efficacité de la réparation de routes pour VSR et CBRP. Puisque la réparation de route nécessite des retransmissions et des timeouts, le délai de bout en bout augmente. Cependant, CBRP continue à présenter un délai très largement supérieur à VSR. Le délai de VSR avec réparation de route est même similaire au délai de CBRP sans réparation de route. Par contre, les pertes de paquets sont réduites : VSR semble avec ce mécanisme insensible à la mobilité puisque même à 30m.s^{-1}, VSR permet d'acheminer 99% des paquets de données à destination.

5.7 Conclusion

Le problème du routage dans les réseaux ad-hoc a souvent été considéré à plat : tous les nœuds contribuent également au routage. Cependant, de tels protocoles présentent des problèmes de passage à l'échelle principalement à cause du manque de hiérarchie : un protocole proactif doit obligatoirement annoncer sa présence dans tout le réseau, et de même, un protocole réactif doit chercher la destination dans tout le réseau si elle n'est pas directement voisine.

Dans ce chapitre, nous nous sommes efforcés de proposer une solution efficace au problème du routage, que nous avons nommée VSR. La force de notre proposition est, selon nous, d'adap-

ter des protocoles existants afin qu'ils tirent parti de la structure d'auto-organisation. Au lieu d'inventer un nouveau protocole, nous avons choisi de reprendre des protocoles existants afin qu'ils utilisent une hiérarchie, et ainsi augmenter leur passage à l'échelle. L'auto-organisation permet notamment de différencier le routage intra et inter-clusters. Nous avons utilisé un algorithme d'inondation des états de liens à l'intérieur d'un cluster, mais un protocole proactif quelconque pourrait être utilisé, tel qu'OLSR ou DSDV. De même, nous nous sommes très largement inspirés de DSR pour le routage inter-clusters : tout protocole réactif par la source ou proactif conviendrait également.

Ce protocole de routage hiérarchique est rendu possible par les propriétés de persistance, de rapidité de convergence de la structure d'auto-organisation. Si par exemple les clusterheads changeaient trop souvent, les routes inter-clusters casseraient fréquemment et nuiraient aux performances du protocole. La hiérarchie introduite n'allonge que modérément la longueur de la route, mais améliore par contre la robustesse et le passage à l'échelle.

Dans le futur, il serait également intéressant de proposer un protocole de routage multicast, s'appuyant sur la forme d'arbre de la dorsale. Nous pensons que la stabilité de la structure permettrait au protocole de présenter de bonnes performances. De même, des mécanismes d'auto-configuration des nœuds devraient été étudiés, afin de proposer une connexion ad-hoc transparente à l'utilisateur.

Nous avons présenté ici un protocole de routage pour les MANET. Cependant, nous pensons que des optimisations sont possibles pour router des données de et vers Internet dans le cas d'un réseau hybride. En effet, dans de tels réseaux, le point d'accès (connecté à la fois au filaire et à la bulle ad-hoc) constitue la destination privilégiée. De plus, nous pensons que la structure d'auto-organisation pourrait prouver pleinement son utilité dans ces réseaux hybrides. Nous proposons donc d'étudier dans le chapitre qui suit le cas particulier de ces réseaux ad-hoc connectés à Internet.

Bibliographie

[1] M. Benzaid, P. Minet, and K. Al Agha. Integrating fast mobility in the olsr routing protocol. Research Report 4510, INRIA, June 2002.

[2] L. Blazevic, L. Buttyan, S. Capkun, S. Giordano, and J.-Y. Le Boudec. Self-organization in mobile ad-hoc networks : the approach of terminodes. *IEEE Communications Magazine*, 39(6) :166–174, June 2001.

[3] L. Blazevic, S. Giordano, and J.-Y. Le Boudec. Anchored path discovery in terminode routing. In *Networking*, Pisa, Italy, May 2002. IFIP.

[4] A. Boukerche. Performance evaluation of routing protocols for ad hoc wireless networks. *Mobile Networks and Applications*, 9(4) :333–342, August 2004.

[5] L. H. M. K. Costa, M. D. De Amorim, and S. Fdida. Reducing latency and overhead of route repair with controlled flooding. *Wireless Networks*, 10(4) :347–358, September 2004.

[6] B. Das, R. Sivakumar, and V. Bharghavan. Routing in ad-hoc networks using a spine. In *International Conference on Computer Communications and Networks (ICCCN)*, page 64, Las Vegas, USA, September 1997. IEEE.

[7] D. S. J. De Couto, D. Aguayo, B. A. Chambers, and R. Morris. Performance of multihop wireless networks : shortest path is not enough. *ACM SIGCOMM Computer Communication Review*, 33(1) :83–88, january 2003.

[8] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1 :269–271, 1959.

[9] R. Draves, J. Padhye, and B. Zill. Comparison of routing metrics for static multi-hop wireless networks. In *SIGCOMM*, Portland, USA, August 2004. ACM.

[10] M. Gerla, X. Hong, and G. Pei. Landmark routing for large ad hoc wireless networks. In *Global Telecommunications Conference (GLOBECOM)*, San Francisco, USA, November 2000. IEEE.

[11] Z. J. Haas and M. R. Pearlman. The performance of query control schemes for the zone routing protocol. In *SIGCOMM*, Vancouver, Canada, September 1998. ACM.

[12] C. Hedrick. RIP version 2. RFC 2453, IETF, June 1988.

[13] M. Jiang, J. Li, and Y. C. Tay. Cluster based routing protocol (CBRP). Internet draft version 01, IETF, July 1999.

[14] 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.

[15] B. Karp and H. Kung. GPSR : Greedy perimeter stateless routing for wireless networks. In *International Conference on Mobile Computing and Networking (MOBICOM)*, pages 243–254, Boston, USA, August 2000. ACM.

[16] J. Kleinberg. The small-world graph phenomenom : an algorithmic aspect perspective. Research Report 99-1776, Cornell University, Computer Science Department, 1999.

[17] Y. Ko and N. H. Vaidya. Location-aided routing (lar) in mobile ad hoc networks. In *Conference on Mobile Computing and Networking (MOBICOM)*, pages 66–75, Dallas, Texas, October 1998. ACM.

[18] J. Li, C. Blake, D. S. J. de Decouto, H. I. Lee, and R. Morris. Capacity of ad hoc wireless networks. In *International Conference on Mobile Computing and Networking (MOBICOM)*, Roma, Italy, July 2001. ACM.

[19] J. Moy. OSPF version 2. RFC 2328, IETF, April 1998.

[20] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. In *International Conference on Mobile Computing and Networking (MOBICOM)*, pages 151–162, Seattle, USA, August 1999. ACM.

[21] N. Nikaein and C. Bonnet. Topology management for improving routing and network performances in mobile ad hoc networks. *ACM Monet*, 10(2), April 2005.

[22] N. Nikaein, C. Bonnet, and N. Nikaein. HARP - hybrid ad hoc routing protocol. In *International Symposium on Telecommunications (IST)*, Dusseldorf, Germany, December 2001.

[23] N. Nikaein, H. Labiod, and C. Bonnet. DDR - distributed dynamic routing algorithm for mobile ad hoc networks. In *Interational Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, Boston, USA, August 2000. IEEE.

[24] OPNET Modeler. http://www.opnet.com (v8.1).

[25] M. R. Pearlman and Z. J. Haas. Determining the optimal configuration of the zone routing protocol. *IEEE Journal on Selected Areas in Communications*, 17(8) :1395–1414, June 1999.

[26] 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.

[27] C. E. Perkins, E. M. Belding Royer, and S. R. Das. Ad hoc on-demand distance vector (AODV) routing. RFC 3561, IETF, July 2003.

[28] Y. Rekhter and T. Li. BGP version 4. RFC 1771, IETF, March 1995.

[29] M. Q. Rieck and S. Pai, Sukesh ad Dhar. Distributed routing algorithms for wireless ad hoc networks using d-hop connected d-hop dominating sets. *Computer Networks*, 47(6) :785–799, April 2005.

[30] C. A. Santivanez, B. McDonald, I. Stavrakakis, and R. Ramanathan. On the scalability of ad hoc routing protocols. In *INFOCOM*, New York, USA, June 2002. IEEE.

[31] P. Sinha, R. Sivakumar, and V. Bharghavan. CEDAR : a core-extraction distributed ad hoc routing algorithm. In *INFOCOM*, pages 202–209, New York, USA, March 1999. IEEE.

[32] P. Sinha, R. Sivakumar, and B. Vaduvur. Enhancing ad hoc routing with dynamic virtual infrastructures. In *INFOCOM*, Anchorage, Alaska, USA, April 2001. IEEE.

[33] H. Tan, W. Zeng, and L. Bao. Patm : Priority-based adaptive topology management for efficient routing in ad hoc networks. In *International Conference in Computational Science*, Atlanta, USA, May 2005.

[34] L. Wang and S. Olariu. A two-zone hybrid routing protocol for mobile ad hoc networks. *IEEE Transactions on Parallel and Distributed Systems*, 15(12) :1105–1116, December 2004.

[35] 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.

Publications

Conférences internationales

[1] Fabrice Theoleyre and Fabrice Valois. On the Performances of the Routing Protocols in MANET : Classical versus Self-Organized Approaches. In *Networking*, Coimbra, Portugal, May 2006. IFIP.

[2] Fabrice Theoleyre and Fabrice Valois. Virtual structure routing in ad hoc networks. In *International Conference in Communications (ICC)*, volume 2, pages 3078–3082, Seoul, Korea, May 2005. IEEE.

Conférence francophone

[3] Fabrice Theoleyre and Fabrice Valois. Routage hybride sur structure virtuelle dans les réseaux mobiles ad-hoc. In *Colloque Francophone sur l'Ingénierie des Protocoles (CFIP)*, Bordeaux, France, March 2005.

Chapitre 6

Apport de l'auto-organisation dans un protocole de localisation

6.1 Introduction

Nous pensons que les réseaux ad-hoc vont connaître un fort développement dans leur version hybride, i.e. quand ils sont connectés à Internet. De tels réseaux constituent des réseaux d'accès sans-fil multi-sauts pour les utilisateurs nomades, en créant ainsi un Internet ubiquitaire. Cependant, les solutions classiques en réseaux cellulaires ne fonctionnent plus dans un réseau décentralisé : il faut distribuer la gestion du réseau.

Le domaine de recherche permettant d'interconnecter un réseau ad-hoc à Internet est actuellement en plein essor. Cependant, beaucoup de solutions se focalisent sur l'utilisation d'un protocole conçu pour les réseaux ad-hoc, en ajoutant une interconnexion. Nous pensons au contraire que des protocoles spécifiques doivent être proposés. En effet, s'il existe peu de communications intérieures à la bulle ad hoc, il est inutile de maintenir toutes les routes correspondantes. En particulier, VSR ne nous semble pas adapté à une telle application car il ne tient en aucun cas compte du fait que le point d'accès constitue la seule destination pour les nœuds intérieur à la bulle ad hoc. En conclusion, nous choisissons dans une telle application d'optimiser plutôt les communications de et vers Internet, en permettant si cela est possible les communications de pair à pair.

Dans un premier temps, nous présenterons dans la section 6.2 notre définition de la localisation dans les réseaux hybrides. Ensuite, la section 6.3 présentera un panorama des solutions permettant l'interconnexion d'un réseau ad-hoc à un réseau filaire, en traitant de la problématique de la gestion de la mobilité. La section 6.4 détaillera notre proposition de protocole de localisation. Des résultats de simulations afin d'évaluer les performances de ce protocole seront donnés en section 6.5. Enfin, la partie 6.6 conclura ce chapitre.

6.2 Une définition de la localisation

La localisation est pour nous le mécanisme permettant à un point d'accès de localiser logiquement un terminal dans son réseau. Si un point d'accès reçoit par exemple un paquet de données à relayer pour un des nœuds de sa bulle ad-hoc, il doit trouver une route vers ce terminal pour lui délivrer le paquet. Une telle définition est inspirée des réseaux cellulaires voix où lorsqu'un appel entrant arrive, il est nécessaire de trouver où se trouve le téléphone portable destinataire, i.e. quelle station de base le dessert.

L'objectif principal n'est donc pas d'autoriser les communications directes entre les terminaux clients. Cependant, il peut être intéressant de développer une telle fonctionnalité pour éviter que les points d'accès ne forment un goulot d'étranglement si les terminaux de la bulle ad-hoc

communiquent majoritairement les uns avec les autres.

Dans ce chapitre nous appellerons *bulle ad hoc* le groupe de nœud mobiles rattachés au point d'accès, en opposition à la partie filaire du réseau hybride.

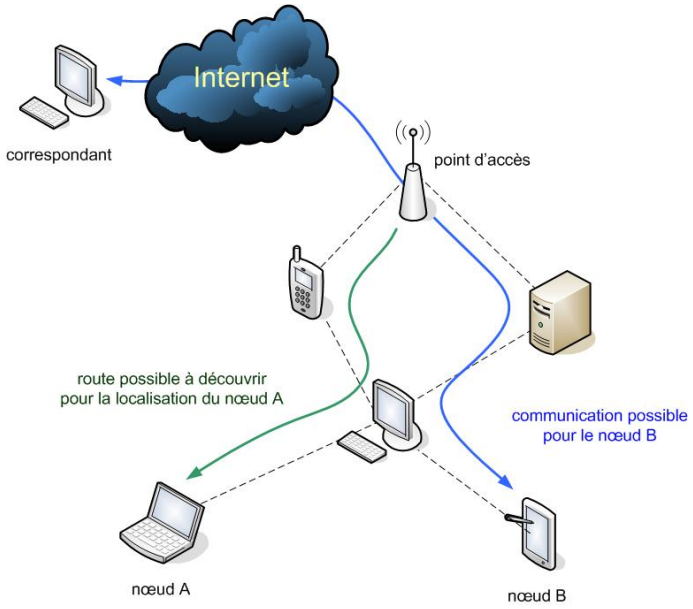


FIG. 6.1 – Définition de la localisation et illustration du type de communication dans un réseau hybride

6.3 Panorama des solutions d'interconnexion ad-hoc / filaire

Les réseaux sans-fil ont été avant tout conçus pour les réseaux cellulaires : tous les clients sont à portée radio d'une station jouant la passerelle entre le monde sans-fil et le monde filaire. Nous allons donc dans un premier temps présenter les solutions de localisation des clients existant dans les réseaux cellulaires, puis nous exposerons les adaptations afin d'interconnecter un réseau sans-fil multi-sauts dans sa globalité.

6.3.1 De l'inadaptation du cellulaire

Un réseau cellulaire est fortement hiérarchique : le client constitue les feuilles terminales du réseau, un équipement spécialisé sert de passerelle entre le sans-fil et le coeur de réseau filaire[1]. Le coeur de réseau est lui aussi hiérarchisé, se tournant de plus en plus vers la hiérarchie classique d'IP.

Nous pouvons distinguer deux grandes solutions de gestion de la mobilité : la macro-mobilité et la micro-mobilité. La première permet à la source de localiser la région où se trouve une destination et d'envoyer ses paquets à la région en question. La micro-mobilité est le protocole permettant à la région de délivrer ensuite les paquets à la destination. Si la destination change de point d'accès mais pas de région, des mises à jours relatives à la micro-mobilité doivent être envoyées, mais les changements ne sont pas propagés dans l'Internet, permettant un meilleur passage à l'échelle.

[1] Il est appelé communément point d'accès (AP) dans les réseaux de données IP, et Base Station Transceiver (BTS) dans les réseaux de type voix

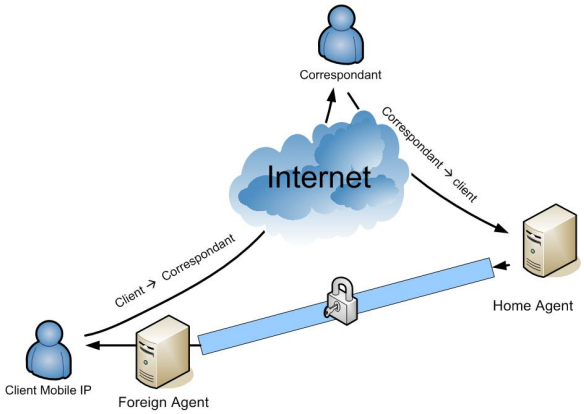


FIG. 6.2 – Schéma de fonctionnement de Mobile IP

6.3.1.1 Solutions de Gestion de la Macro-Mobilité

Mobile IP Mobile IP (MIP) [20, 13, 15, 23] est le standard le plus répandu pour la gestion de la macro-mobilité. Un utilisateur nomade se déplaçant et se connectant à un réseau hôte doit continuer à être joignable via son ancienne adresse par un hôte sur Internet. Lorsqu'un mobile se trouve au sein d'un réseau hôte, il va contacter un Foreign Agent(FA). Ce FA va collecter des informations d'authentification qu'il va faire suivre au Home Agent(HA) du mobile, se trouvant dans le réseau d'origine du mobile et interceptant les paquets. Il vérifie les données d'authentification, enregistre l'adresse du FA, et l'autorise à enregistrer le client. Pour la suite des communications, le mobile peut faire suivre au FA ses paquets à envoyer ou acquérir une adresse temporaire grâce à DHCP [11] par exemple. Dans l'autre sens, les paquets sont envoyés dans le réseau d'origine, interceptés par le HA qui les encapsule pour les rediriger vers le FA. Enfin, ce dernier décapsule les paquets et les délivre au mobile. Cependant,lorsque le mobile se déplace au sein du réseau hôte et qu'il change par exemple de FA (il change de sous-réseau IP), il doit réenclencher tout le processus d'enregistrement qui traverse Internet. Un protocole de micro-mobilité permet de n'exécuter ce processus d'enregistrement que lorsque le mobile sort de la région, ce qui se passe rarement. [25] propose une fonctionnalité additionnelle de paging intégrée à Mobile IP. Un mobile peut s'endormir, la période de rafraîchissement de sa localisation au sein du FA est plus faible, et les zones de localisation pour le paging sont plus étendues. Le FA maintient un cache des adresses temporaires des mobiles qu'il dessert. Si le mobile ne se réenregistre pas au bout d'un certain temps, il place l'entrée en mode paging. Si un paquet arrive, il le diffuse au groupe des FA adjacents constituant la zone de paging jusqu'à la réponse du mobile.

Ainsi, il serait intéressant d'utiliser Mobile IP pour gérer la mobilité d'un client dans un réseau hybride. Lorsqu'un terminal arrive dans une zone ad-hoc, il contacte une des passerelles de la bulle ad-hoc pour s'enregistrer auprès de son Home Agent. S'il change de bulle ad-hoc ou de passerelle, il devra bien entendu s'enregistrer. Par contre, la mobilité à l'intérieur de la bulle ad-hoc ne devrait pas être gérée avec Mobile IP : le trafic de contrôle généré de bout en bout dans Internet à chaque changement de localisation pourrait être problématique. Nous allons voir dans le paragraphe suivant que les protocoles de gestion de la micro mobilité dans le cellulaire, bien que présentant des concepts intéressants, ne sont pas directement applicables aux réseaux ad-hoc.

6.3.1.2 Solutions de Gestion de la Micro-Mobilité

Hierarchical Mobile IP Afin, de diminuer le trafic de contrôle et maintenir un impact local à des changements locaux, Hierarchical Mobile IP (HMIP) [4, 8] propose de créer une hiérarchie de Foreign Agent. Le mobile s'enregistre dans le niveau i avec l'adresse donnée par le niveau

$i-1$. Cependant, une telle structure statique n'est pas présente dans un réseau ad-hoc. HMIP n'est donc pas applicable directement.

Cellular IP Cellular IP (CIP) [5] permet sur une topologie en arbre de maintenir un cache de localisation distribué dans chaque nœud de l'arbre. Lorsqu'un routeur reçoit un paquet venant d'un mobile, il enregistre le routeur fils comme prochain saut vers le mobile en question. Un mobile n'enregistre dans son Home Agent que l'adresse de la racine de l'arbre, CIP se chargeant ensuite de distribuer le paquet dans son arbre grâce au cache de routage distribué. Là encore, CIP s'appuie sur un arbre configuré de façon statique. Par contre, certains pourraient judicieusement remarquer que la dorsale proposée dans le chapitre 3 page 31 constitue un arbre, et se répare automatiquement lorsqu'un changement de topologie survient. Ainsi, il serait intéressant de s'inspirer de CIP pour concevoir un protocole de localisation en utilisant la dorsale créée auparavant.

Des extensions [6, 7, 25] permettent de déterminer réactivement la localisation d'un client en proposant une fonction de *paging*. Ce concept pourrait être adapté aux réseaux ad-hoc : de nombreux clients peuvent être inactifs, et n'enregistrent donc que rarement et de façon macroscopique leur localisation. Le trafic de contrôle est ainsi plus réduit.

IDMP Intra-Domain Mobility Management Protocol (IDMP) [10] est un protocole proposé dans les réseaux voix de $3^{ième}$ génération. Les auteurs ont une approche intéressante : bien que le réseau soit statiquement configuré en arbre, un mobile peut choisir la hauteur d'enregistrement dans l'arbre. Ainsi, le protocole est adaptatif : un client fortement mobile s'inscrira plus haut dans l'arbre afin d'optimiser les redirections de paquets et donc limiter les cassures dans les routes. Un tel concept pourrait avec profit être adapté aux réseaux hybrides.

6.3.1.3 Synthèse

Ainsi, Mobile IP peut être directement utilisé pour la gestion de la mobilité entre les zones ad-hoc, puisque le point d'accès constitue une racine présente dans les réseaux hybrides. Par contre, les protocoles de gestion de la micro-mobilité s'appuient sur une hiérarchie statique configurée au préalable, ils ne sont donc pas directement exploitables dans un réseau ad-hoc. Nous verrons que notre approche s'appuie sur la hiérarchie proposée par la structure virtuelle présentée dans le chapitre 3 page 31 : elle permet d'adapter facilement les protocoles venant du cellulaire, en permettant de déployer toutes les fonctionnalités disponibles dans les réseaux sans-fil classiques.

6.3.2 Du ad-hoc vers le hybride

De nombreuses propositions ont été récemment faites afin d'intégrer les réseaux ad-hoc à l'Internet. Elles s'appuient principalement sur Mobile IP, et sur un protocole de routage ad-hoc pour l'interconnexion. [9] dresse une liste des verrous existants pour l'interconnexion des réseaux cellulaires et MANET :

- Intégration de plusieurs technologies de transmission (bluetooth, IEEE 802.11...)
- Dimensionnement du réseau et placement des points d'accès
- Qualité de service dans la couche MAC
- Passage à l'échelle des protocoles de routage
- Réduction de l'impact du routage sur les couches inférieures (interférences, distribution de la charge,...)
- Découverte efficace des passerelles vers Internet
- Proposition de nouveaux protocoles de gestion de la mobilité réduisant les délais lors d'un handoff

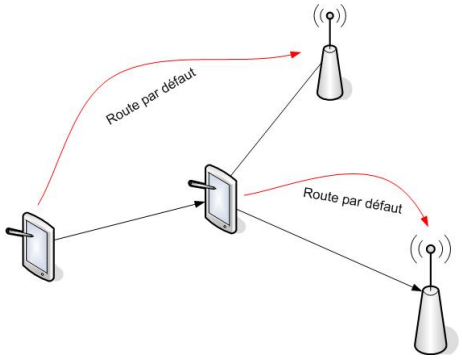


FIG. 6.3 – Incohérence dans le choix d'une route par défaut dans le cas de passerelles multiples

6.3.2.1 MIP MANET

MIPMANET [16] intègre Mobile IP et AODV : Mobile IP permet de gérer la macro-mobilité, puis le protocole de routage AODV permet de trouver une route du point d'accès vers le mobile. Afin qu'un mobile connaisse l'identité du Foreign Agen dans son réseau ad-hoc, le point d'accès inonde périodiquement le réseau avec des `advertisements` donnant tous les paramètres de Mobile IP. Tout mobile relayant ce paquet, chaque nœud connaît l'identité du FA et établit donc une route à la demande avec AODV vers la passerelle lorsqu'il désire joindre un nœud sur Internet. Un nœud considère qu'une destination est extérieure à sa bulle ad-hoc losqu'il ne reçoit aucune réponse à ses découvertes de routes. Nous voyons donc que le trafic de contrôle dû à l'inondation périodique des `advertisements` peut être importante, occasionnant de nombreuses collisions et donc des pertes de paquets. Dans [1], un mobile ad-hoc a l'obligation de s'enregistrer auprès du FA. Ainsi, lorsque le FA reçoit une demande de route, il peut savoir si la destination est présente ou non dans la bulle ad-hoc, et donc envoyer la réponse de route correspondante. Des champs spéciaux sont donc ajoutés dans les paquets `RREQ` et `RREP`. Cependant, l'overhead créé par ces enregistrements peut être élevé. [17] propose de limiter l'impact des inondations périodiques en limitant les `advertisements` à k sauts : certains mobiles peuvent donc ne posséder aucune route vers Internet. [2] suit une approche similaire, en remplaçant AODV par OLSR.

[21] propose une approche combinant le réactif et le proactif : les passerelles envoient périodiquement les `advertisements` dans une zone de rayon borné. Les clients à l'extérieur de cette zone demandent réactivement l'identité de la passerelle, et les paramètre Mobile IP : tout nœud appartenant à la zone proactive pourra répondre à une telle sollicitation. [22] propose une étude comparative des approches d'envois proactif et réactif des `advertisements`. Pour mesurer également l'impact du protocole de routage, OLSR et AODV sont comparés. OLSR présente le trafic de contrôle le plus important. De plus, lorsque la mobilité est importante, les auteurs remarquent que des incohérences peuvent survenir rendant une approche proactive moins performante. [14] présente une étude analytique des mêmes facteurs, montrant l'intérêt d'une approche hybride.

[18] discute des avantages et inconvénients des solutions d'encapsulation ou d'établissement de routes par défaut dans les réseaux hybrides. L'encapsulation augmente le trafic de contrôle mais permet de cacher la destination réelle aux nœuds intermédiaires, rendant la solution transparente. De même l'encapsulation permet de multiplier le nombre de passerelles possibles, en laissant au client toute indépendance dans le choix de sa passerelle. Une route par défaut présente une grande simplicité, mais des incohérences peuvent survenir dans le routage si plusieurs nœuds ne choisissent pas leur route par défaut selon un même objectif (fig. 6.3). Les auteurs proposent qu'un nœud se souvienne de sa passerelle par défaut plutôt que d'enregistrer le prochain saut par défaut. Il pourra ainsi éviter de relayer certains paquets venant d'autres passerelles afin qu'aucun autre nœud ne puisse connaître par son intermédiaire une passerelle différente de la sienne.

6.3.2.2 MEWLANA

MEWLANA-TD [12] propose une approche similaire à MIPMANET en intégrant Mobile IP et DSDV. Le protocole de routage dans la bulle ad-hoc étant proactif, il suffit de fixer une route par défaut pointant vers le Foreign Agent. Cependant, la lenteur de convergence de DSDV, ainsi que son trafic de contrôle auquel s'ajoutent les `advertisements` des FA représentent des contraintes fortes.

Les auteurs proposent dans le même article un protocole optimisé pour les communications de et vers Internet, MEWLANA-RD. Le FA inonde le réseau d'`advertisements`. Lorsqu'un nœud relaie un `advertisement`, il enregistre le nœud précédent comme route par défaut. Parallèlement, il envoie un paquet d'enregistrement vers le FA. Chaque nœud relayant le `registration` enregistre la route vers la source. Ainsi, une table de routage distribuée est créée dans le réseau. Les auteurs montrent que les performances de MEWLANA-RD sont supérieures à MIPMANET et MEWLANA-TD lorsque le nombre de flux intérieurs à la bulle ad-hoc est négligeable. [24] présente une approche similaire. Cependant, dans les deux approches, les `advertisements` et `registrations` périodiques créent un trafic de contrôle important. Un mécanisme plus intelligent devrait donc être plus efficace.

6.3.2.3 Intégration de technologies multiples

[3] propose d'intégrer une architecture multi-sauts dans un réseau cellulaire. Un nœud central permet de collecter auprès de tous les clients leur position et leurs voisins dans la topologie ad-hoc. Ainsi, cet agent peut calculer les plus courtes routes dans un réseau multi-sauts. La requête de route se fait via un réseau cellulaire, donc disjoint. Le mobile va demander à l'agent central une route via un canal radio différent de celui utilisé pour router les données dans le ad-hoc. L'agent central lui répondra via le réseau cellulaire. Le mobile peut ensuite emprunter la route ad-hoc, jusqu'à sa cassure, auquel cas il recontactera l'agent pour obtenir une nouvelle route. Une telle solution requiert donc deux réseaux disjoints, chaque nœud possédant deux radios distinctes. De plus, nous perdons ainsi la flexibilité d'un réseau ad-hoc en terme de déploiement et d'adaptabilité. Nous pensons donc qu'une telle solution est peu adéquate aux réseaux ad-hoc.

6.3.2.4 Limites de ces approches

Ces approches pour la plupart proposent d'intégrer directement Mobile IP et un protocole de routage conçu pour les réseaux ad-hoc. Ainsi, les communications ad-hoc restent possibles tout en offrant une connectivité Internet. Cependant, le ratio de communications de type ad-hoc doit être suffisant afin de justifier l'utilisation d'un tel protocole. Un protocole de routage proactif présente un trafic de contrôlé élevé. De même, l'utilisation d'un protocole réactif requiert l'inondation de plusieurs découvertes de routes infructueuses avant de considérer que la destination est extérieure à la bulle ad-hoc. Ainsi, le trafic de contrôle de toutes ces solutions reste élevé.

Seul MEWALANA-RD est un protocole conçu spécifiquement pour un réseau cellulaire multisauts. L'utilisation d'un arbre est un concept intéressant. Cependant, une telle structure est détruite et reconstruite périodiquement. De même, l'enregistrement des mobiles auprès d'une passerelle est obligatoire. Ainsi, le trafic de contrôle généré est sous-optimal. Nous pensons que l'utilisation d'une structure virtuelle permettrait de proposer un protocole de gestion de la mobilité plus optimisé.

6.4 Proposition d'une solution de localisation adaptée aux réseaux hybrides

Notre proposition se focalise ici sur la localisation des nœuds dans un réseau hybride. Un nœud doit être accessible par son point d'accès : lorsqu'un paquet arrive dans la passerelle, il doit

être envoyé en multisauts jusqu'à la destination. Inversement, un terminal doit pouvoir envoyer ses paquets à des destinataires présents dans Internet. Mobile IP peut directement être utilisé pour gérer la macro-mobilité entre bulles ad hoc. Par contre, nous présentons ici une solution de gestion de la micro-mobilité inspirée de Cellular IP. Comme nous l'avions déjà souligné, la dorsale proposée dans le chapitre 3 page 31 convient particulièrement pour une telle adaptation. En effet, nous avions proposé la création d'une dorsale en forme d'arbre, se réparant de façon autonome lorsque des changements de topologie surviennent. Ainsi, nous pouvons adapter Cellular IP sur cette dorsale, gérant de façon transparente la mobilité des nœuds. Nous détaillons donc ici le fonctionnement de ce protocole, que nous avons appelé Self-organized Mobility Management (SOMOM) .

6.4.1 Description générale

Cellular IP s'appuyant sur une topologie en arbre des routeurs, nous proposons tout simplement d'utiliser la dorsale en arbre proposée précédemment. Les terminaux, bien que mobiles, maintiennent en continu la connectivité et la propriété de dominance de la dorsale. Ainsi, un protocole de localisation peut de façon transparente utiliser cette dorsale, levant le problème des changements de topologie. Naturellement, l'intégration de Cellular IP et de la dorsale ne peut être réalisée de façon totalement transparente puisque la dorsale subit des changements de topologie, et donc que les caches de localisation doivent être mis à jour en conséquence. Cependant, comme nous l'avions remarqué, la dorsale présente une persistance élevée : de telles mises à jour seront donc limitées. Une telle propriété est importante pour limiter la génération intempestive de trafic de contrôle.

Plus précisément, dans le sens download, le point d'accès qui ne connaît aucune route vers la destination inondera la dorsale. Une réponse de route mettra à jour les caches de localisation des terminaux faisant partie de la dorsale afin que les paquets suivants soient routés normalement. Dans le sens upload, nous verrons qu'une route est a priori connue vers le point d'accès, du fait de la forme particulière de la dorsale, le point d'accès constituant sa racine. Nous utiliserons donc cette information pour créer une route par défaut.

6.4.2 Accès à Internet

Le point d'accès peut représenter une route par défaut adéquate pour joindre Internet. Lorsqu'un terminal souhaite envoyer un paquet, il l'envoie à son point d'accès. Ensuite, le point d'accès agira comme proxy afin de trouver une route dans Internet, faire de la translation d'adresses, du filtrage... La dorsale présente des propriétés pouvant convenir à une telle fonctionnalité : elle est en forme d'arbre, le point d'accès étant placé à sa racine. De plus, chaque nœud maintient l'identité de son père dans l'arbre. Ce parent constituera donc la route par défaut.

Plus précisément, pour un dominant, le père dans l'arbre est un voisin. Donc la route par défaut peut directement pointer vers ce père. Pour un dominé, le père peut se trouver à au plus k_{cds} sauts. Cependant, comme la connaissance du k_{cds}-voisinage est requise pour la maintenance de la dorsale, un dominé peut extraire un prochain saut vers son père, constituant sa route par défaut.

La connaissance proactive d'une route par défaut vers Internet représente un grand atout : aucun trafic de contrôle supplémentaire n'est requis, et la latence de découverte de route est nul. Nous pensons donc qu'une telle connaissance est utile dans les réseaux hybrides.

6.4.3 Localisation du mobile

De la même manière, un point d'accès qui reçoit des paquets de données venant d'Internet et à destination d'un de ses mobiles doit pouvoir connaître une route vers ce mobile. Nous proposons dans un premier temps une maintenance gratuite de route, telle que dans Cellular IP.

Lorsqu'un nœud reçoit un paquet de données relayé par un nœud R, il peut mettre jour dans sa table de routage l'entrée pointant vers la source du paquet de données, fixant son prochain saut à R. Ainsi, lorsqu'un nœud initie une communication vers Internet, il envoie directement un paquet de données sur sa route par défaut, créant automatiquement dans chaque nœud intermédiaire une entrée correspondante dans sa table de routage. De même, si un changement de topologie dans la dorsale se produit, il suffit que le nœud envoie un paquet de données pour mettre à jour les tables de routage obsolètes des nœuds relais. De plus, nous pensons que les communications seront, dans un réseau hybride, majoritairement initiées par le terminal : pour l'envoi d'enregistrements Mobile IP, pour initier une requête `http`... Ainsi, le rafraîchissement de la route inverse, lors de chaque envoi de paquet de données permet de limiter les délais de création de route, et de limiter l'overhead. Une telle propriété représente selon nous un atout important pour un protocole de routage dans les réseaux hybrides.

Cependant, il peut arriver qu'un point d'accès reçoive un paquet de données à relayer pour un de ses mobiles et qu'il ne connaisse encore aucune route vers celui-ci. Pensant qu'un tel cas est rare, nous proposons ici une approche réactive, permettant de limiter l'overhead. La recherche réactive d'une route pourrait être comparée au mécanisme de paging de Cellular IP : un routeur Cellular IP doit trouver réactivement la localisation exacte d'un mobile, en inondant ses descendants dans l'arbre des routeurs Cellular IP.

Lorsque le point d'accès reçoit un paquet de données destiné à D et qu'aucune route vers D n'est connue, il engage la procédure suivante :

- Le point d'accès met en file d'attente le paquet de données. Puis il génère un paquet de `Route Request` et l'envoie à ses fils dans l'arbre en multicast (fig. 6.4 page suivante), l'adresse multicast correspondant à l'ensemble des nœuds de la dorsale. L'AP initie donc une inondation de la dorsale.
- Un dominant recevant une `Route Request` regarde si la destination recherchée est présente dans sa table de voisinage :
  - Si la destination n'est pas un de ses dominés, le dominant relaie la requête en multicast à ses fils.
  - Si au contraire, le dominant est lui-même la destination ou la connaît, il génère une `Route Reply` et l'envoie à son père dans l'arbre. Un dominant joue donc le rôle de *mandataire* pour ses dominés.

Dans un tel mécanisme, seul un nœud peut générer une `Route Reply` (le nœud lui même s'il est dominant, ou sinon son père). Cependant, lorsqu'un dominé se retrouve déconnecté de la dorsale lors de la maintenance, un mécanisme de réparation se met en place, demandant une certaine latence avant que les informations convergent. Afin de diminuer l'impact d'une déconnexion de la dorsale, tout dominant recevant une `Route Request` peut renvoyer une `Route Reply` si la destination est présente dans sa table de voisinage. Ainsi, plusieurs `Route Reply` peuvent être générées pour une même destination, augmentant l'overhead. Cependant, les `Route Request` devraient également se propager moins loin dans l'arbre, limitant ainsi le nombre de paquets de contrôle. Nous pensons donc qu'un tel mécanisme présente un impact limité sur le trafic de contrôle. De plus, nous pouvons remarquer que seuls les dominants relaient les `Route Request`, quelle que soit la méthode utilisée. C'est pourquoi l'overhead est beaucoup plus réduit que lors d'une inondation globale du réseau.

Lorsqu'un nœud reçoit une `Route Reply`, il met à jour sa table de routage, en ajoutant ou rafraîchissant l'entrée correspondant à la source du paquet. Puis, la `Route Reply` est envoyée via la route par défaut. Finalement, le paquet arrivera à la racine de la dorsale, le point d'accès. L'AP mettre donc à jour sa table de routage et pourra envoyer les paquets de données bufferisés et destinés à ce nœud. Si d'autres paquets arrivent d'Internet, le point d'accès connaîtra dorénavant une route vers le mobile, et n'aura pas besoin de réinitier une découverte de route.

Si certains nœuds sont fortement actifs et souhaitent maintenir continuellement une route pointant vers eux, ils peuvent supprimer la part réactive en envoyant périodiquement des `Route`

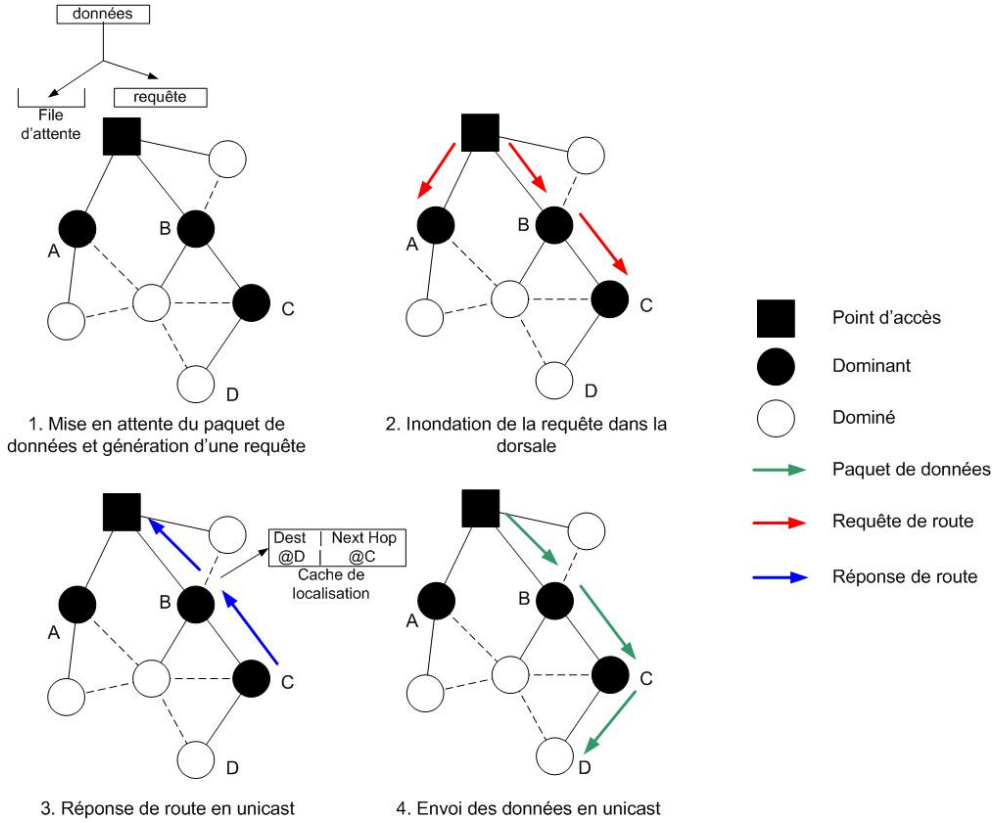


FIG. 6.4 – Exemple de fonctionnement de SOMOM lorsqu'un paquet de données arrive de l'Internet ($k_{cds} = 1$)

`Update` agissant comme des `Route Replies`. Ainsi, aucune latence de découverte de route ne sera introduite. Un tel mécanisme pourrait être intéressant pour un serveur de données par exemple.

Durant la maintenance, des réparations de la dorsale peuvent intervenir, introduisant des changements dans la topologie virtuelle. Un dominant changeant de père va créer un ensemble de routes obsolètes dans les caches de routage de la dorsale. Un dominant voyant qu'un de ses fils dans la dorsale est parti, supprime toutes les entrées de sa table de routage dont il constituait le prochain saut. Puis, il génère un `Route Delete` contenant la liste des entrées supprimées de sa table de routage. Ce paquet est ensuite envoyé à son père. Saut par saut, le `Route Delete` va supprimer les entrées obsolètes dans les tables de routage. Lorsque le point d'accès recevra un paquet destiné à un nœud de la branche reconnectée, l'entrée obsolète aura été supprimée, et une nouvelle découverte de route sera ré-initiée.

Dans le futur, les `AP advertisements` intégreront les paramètres Mobile IP, le préfixe réseau utilisé,... Ainsi, chaque nœud pourra découvrir les paramètres d'accès du réseau. Si le réseau hybride comprend plusieurs points d'accès, une dorsale par AP sera construite, comme décrit dans la section 3.5 page 41. Un dominant qui changera de dorsale fera un handover pour sa branche, récupérera les nouveaux paramètres d'accès tirés des `AP advertisements` et agira en tant que proxy, ou diffusera ces paramètres.

6.4.4 Implémentation d'un mode ad-hoc

Nous avons proposé une solution optimisée pour les communications de et vers Internet. Cependant, il peut être quelquefois intéressant d'autoriser les communications de pair à pair. Nous proposons donc ici une extension permettant les communications internes. Le délai et la longueur de la route ne sont pas optimaux, mais nous pensons que ceci ne constitue pas un inconvénient sévère, ce type de communications n'étant pas majoritaires.

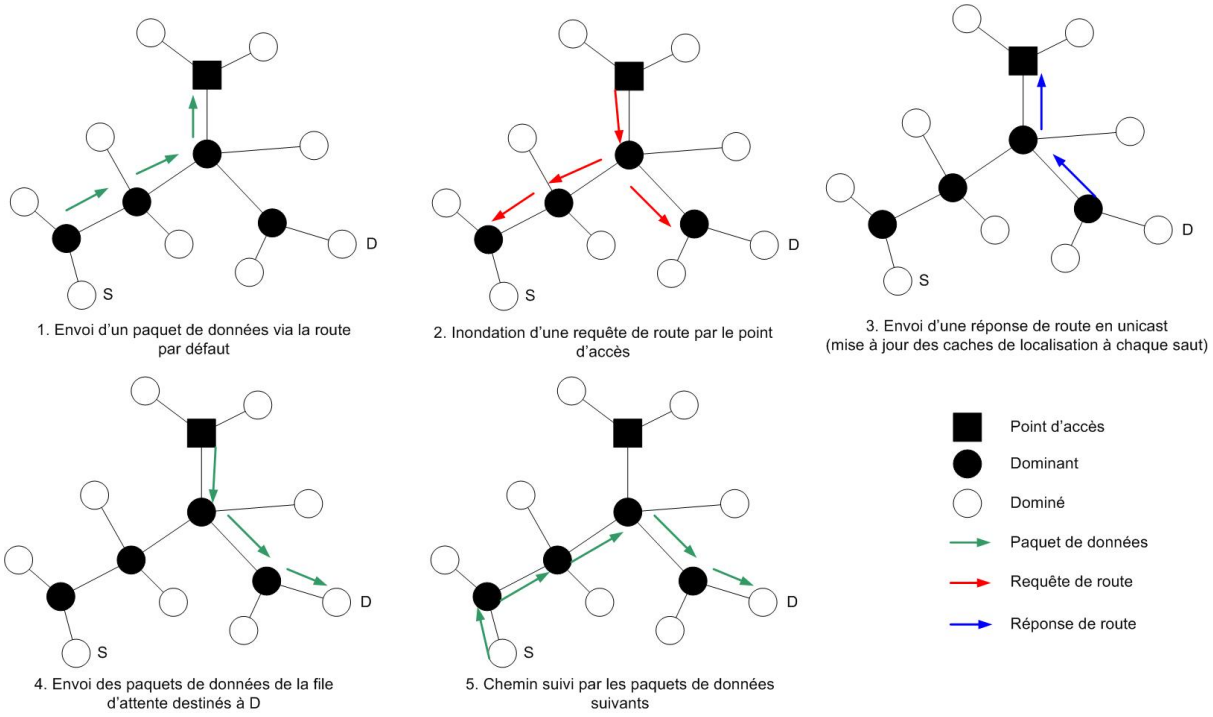


FIG. 6.5 – Fonctionnement du mode ad hoc de SOMOM

Lorsqu'un nœud S souhaite envoyer un paquet de données vers D, S ne sait pas si la destination est à l'intérieur de la bulle ad-hoc ou sur Internet. Ainsi, S va chercher s'il connaît une route vers D dans son cache de routage, ou si D est présent dans sa table de voisinage. Si c'est le cas, il lui envoie directement le paquet. Sinon, il l'envoie via sa route par défaut à son père, qui exécute la même procédure (étape 1 de la figure 6.5). Si finalement, aucun ascendant dans la dorsale ne connaissait la destination, le paquet arrive au point d'accès. Nous supposons que le point d'accès sait si la destination correspond à un de ses clients ou pas (selon le préfixe réseau, son cache de membres...). Si la destination est dans la bulle ad-hoc, il bufferise le paquet, génère une `Route Request` (étape 2 fig. 6.5), et attend une `Route Reply`, comme normalement. Les paquets seront finalement envoyés sur la route découverte (étape 4 fig. 6.5).

Si de nouveaux paquets arrivent pour la même destination, une route est déjà connue. Ainsi, le paquet n'arrivera certainement pas jusqu'à la racine de la dorsale. Il sera envoyé directement sur la bonne route par l'ancêtre commun de la source et de la destination dans la dorsale (étape 5 fig. 6.5). Ainsi, la longueur de la route ne sera pas optimale, mais nous pensons qu'un tel scénario se présentera dans un nombre de cas minoritaire.

6.4.5 Paging

Le paging est utilisé dans les réseaux cellulaire afin de limiter l'overhead des enregistrements. Un nœud ne s'enregistre que rarement, dans sa zone de paging, comprenant plusieurs points d'accès. Ainsi, un mobile peut se déplacer sans changer de zone de paging, et donc sans renouveler son enregistrement. Un *Paging Master* est supposé connecté (directement ou indirectement) à tous les points d'accès de la zone de paging. Le *Paging Master* ajoute le mobile dans son cache de membres avec un timeout long. Lorsqu'un paquet arrive pour un tel mobile, il vérifie que la destination est présente dans son cache de membres. Ensuite, il regarde dans son cache de routage si un point d'accès lui est déjà associé. Si c'est le cas, le paquet de données est envoyé à l'AP correspondant. Sinon, le *Paging Master* initie une découverte de route, déclenchant l'envoi d'un `Route Request` par l'ensemble des points d'accès de sa zone de paging. Un mécanisme de paging peut être utile lorsqu'un terminal est mobile, et que le changement de point d'accès

engagerait un changement d'adresse IP, de reconfiguration, et donc engendrerait un overhead périodique important.

6.5 Performances

Nous présentons ici des résultats de simulation utilisant OPNET Modeler [19]. Nous avons utilisé le modèle IEEE 802.11 avec la portée radio standard de 300m, le mode DCF, sans RTS/CTS. Chaque nœud se déplace suivant le random waypoint mobility model, sans pause. Tous les résultats mesurés sont reportés avec leur intervalle de confiance de 95%. Nous considérons comme générique une vitesse de 5m.s^{-1}, 40 nœuds, un degré de 10, et 4 flux simultanés.

La génération de trafic est modélisée comme des flux de 8 paquets, espacés de 0.25 secondes. Pour chaque flux, une source et une destination sont choisies aléatoirement. Le temps inter-flux suit une distribution exponentielle centrée sur 2 secondes afin d'obtenir en moyenne le nombre de flux simultanés choisis. La taille des paquets suit une distribution exponentielle centrée sur 128 octets.

Nous avons comparé les performances de SOMOM avec celles de MEWLANA-RD [12]. En effet, [12] montre que MEWLANA-RD est plus performant que MIPMANET dans le scénario d'un réseau hybride. Nous considérons que les principales métriques d'efficacité d'un protocole de routage sont

1. Le taux de livraison, i.e. la proportion des paquets de données qui sont effectivement reçus par la destination
2. Le délai de bout en bout, i.e. le temps entre la génération d'un paquet et sa réception par la destination. Ainsi, le délai de bout en bout comprend le temps de découverte d'une route
3. Le trafic de contrôle généré (qui réduit la bande passante restante disponible pour les paquets de données)

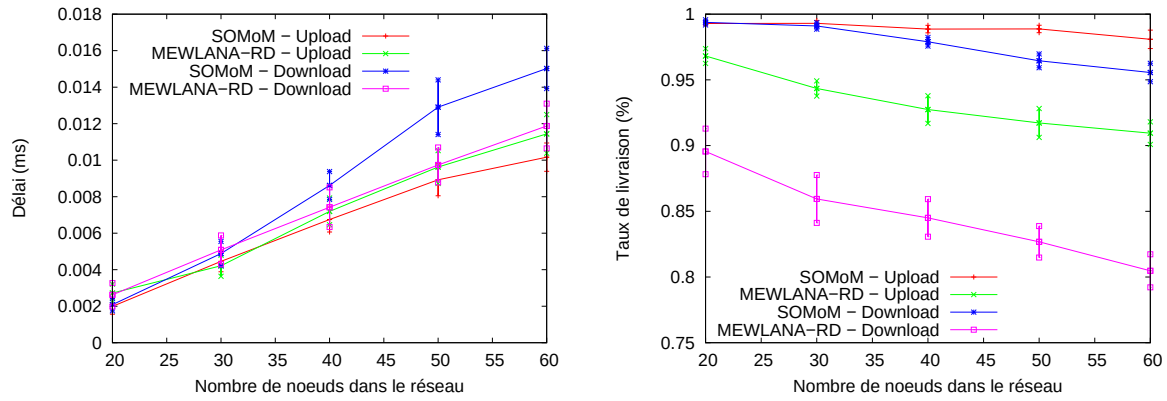
Nous avons implémenté MEWLANA-RD avec un processus de découverte de voisinage afin qu'il distingue les liens radio unidirectionnels de ceux bidirectionnels. Nous avons mesuré le passage à l'échelle horizontal et vertical, et l'impact de la mobilité. De même, la solution d'économie d'énergie décrite dans la section 4.7.2 page 75 a été également testée.

6.5.1 Passage à l'échelle horizontal

Nous avons dans un premier temps étudié l'impact de la cardinalité du réseau sur les performances des deux protocoles. Nous voyons que **le taux de livraison de SOMOM reste quasiment constant quand le nombre de nœuds augmente** (fig. 6.6(a) page suivante). De plus, les paquets de données subissent des pertes quasiment similaires en upload (du mobile vers l'AP) et en download. Nous voyons par contre que MEWLANA-RD subit plus de pertes de paquets. Les `AP Hellos` sont périodiques et causent des collisions avec les paquets de données. De plus, nous avons pu voir que de nombreux `AP Registration` n'arrivent pas jusqu'au point d'accès. Le point d'accès ne possède donc pas l'intégralité des routes du réseau. Ceci explique l'asymétrie de performances dans les sens upload et download. Le mobile a plus de chance de recevoir un `AP Hello`, car le flooding est redondant, la route vers l'AP a donc plus de chance d'exister.

Nous avons ensuite étudié l'impact du nombre de nœuds sur le délai (fig. 6.6(b) page suivante). Nous pouvons voir que tous les protocoles présentent un délai très similaire, le délai en download de SOMOM étant légèrement plus élevé. Ainsi, **SOMOM qui combine le proactif et le réactif permet de proposer un délai très faible, tout en réduisant l'overhead généré**. Une telle approche semble donc efficace.

Nous avons finalement mesuré la longueur moyenne des routes générées par les deux protocoles avec un réseau de 40 nœuds. La longueur moyenne des routes de SOMOM (2.7 sauts)



(a) Délai de bout en bout (b) Taux de livraison

FIG. 6.6 – Passage à l'échelle horizontal

est légèrement supérieure à celle de MEWLANA-RD (2.64 sauts). Notre protocole utilisant la topologie de la dorsale, les chemins peuvent être allongés. Mais, une différence de seulement 3% nous semble avoir un impact négligeable.

6.5.2 Passage à l'échelle vertical

Nous avons ensuite observé l'impact de la charge réseau (fig. 6.7). Nous pouvons remarquer que le taux de livraison de SOMOM et de MEWLANA-RD reste stable. Cependant, SOMOM continue à présenter un taux de perte très inférieur à celui de MEWLANA-RD. Alors que SOMOM arrive à délivrer plus de 98% des paquets, MEWLANA-RD n'en délivre que 90%. **SOMOM paraît donc beaucoup plus robuste**.

Par contre, le délai de bout en bout augmente avec le nombre de connexions : plus de paquets doivent être relayées, augmentant le temps passé dans les files d'attente. Pour un nombre limité de connexions, MEWLANA-RD et SOMOM se comportent similairement. Pour une forte charge, le délai de SOMOM est plus important que celui de MEWLANA-RD : SOMOM occasionne moins de pertes de paquets, notamment pour les routes longues. Les routes longues présentant le délai le plus élevé, le délai moyen de SOMOM est donc supérieur à celui de MEWLANA-RD.

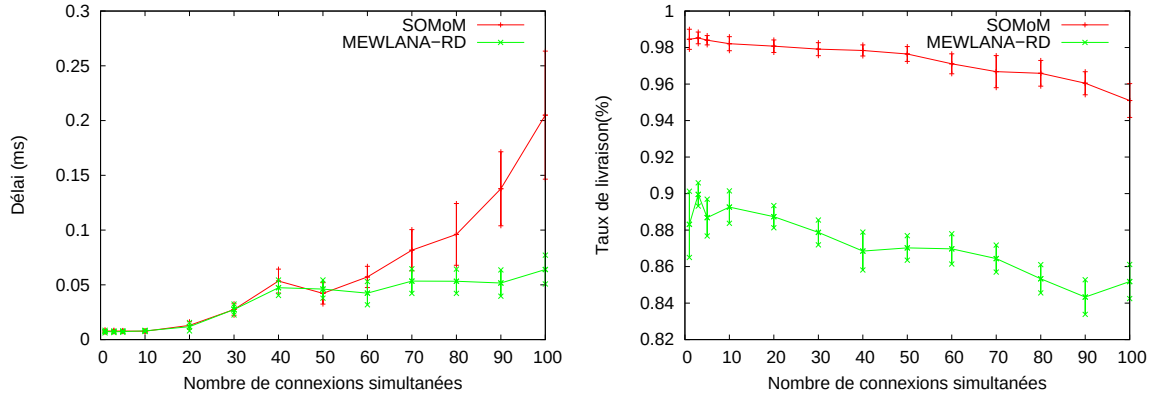
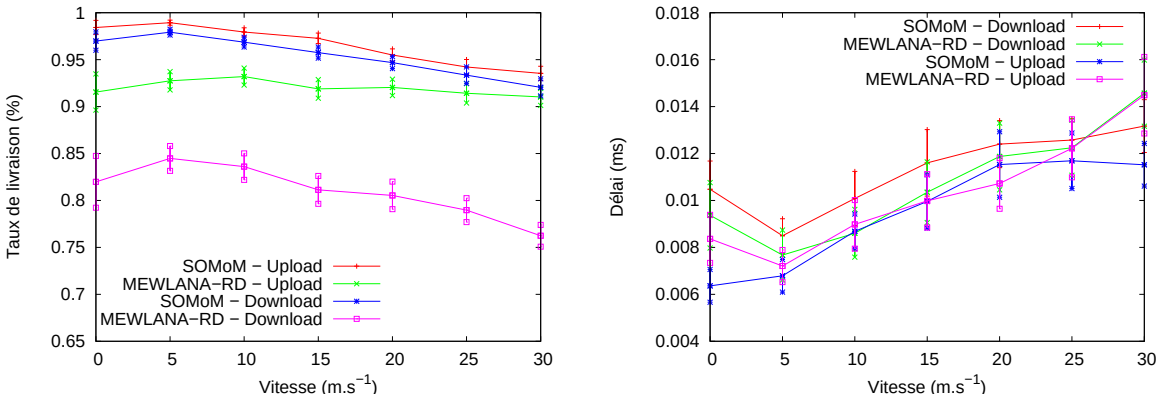


FIG. 6.7 – passage à l'échelle vertical

6.5.3 Impact de la mobilité

Nous avons ensuite mesuré l'impact de la mobilité en augmentant la vitesse maximale du modèle de mobilité. Si les nœuds sont plus mobiles, les pertes de paquets augmentent (fig. 6.8(a)) :

des liens disparaissent, cassant potentiellement les routes utilisées. Cependant, ces pertes restent limitées pour les deux protocoles. SOMoM et MEWLANA-RD arrivent à maintenir efficacement leurs routes. MEWLANA-RD qui reconstruit l'intégralité de ses routes périodiquement réagit bien à la mobilité, de façon logique. SOMoM qui propose un mécanisme de mise à jour gratuit des routes utilisées et un mécanisme de suppression des routes obsolètes **permet de maintenir des routes efficaces à un moindre coût**. Enfin, MEWLANA-RD présente toujours un taux de livraison très inférieur à celui de SOMoM.



(a) Taux de livraison

(b) Délai de bout en bout

FIG. 6.8 – Impact de la mobilité

Nous pouvons remarquer que le délai de bout en bout des deux protocoles augmente dans un environnement plus mobile (fig. 6.8(b)) : plus de routes cassent, des retransmissions sont obligatoires, augmentant le délai. De plus, une cassure de route dans SOMoM oblige à réinitier une découverte de route dans le sens descendant. Cependant, ce délai reste inférieur à 15 ms, même à une vitesse maximale de 30 m.s^{-1}.

6.5.4 Overhead

	SOMoM	MEWLANA-RD
Hellos	0.32	0.25
Retransmissions de données	0.01	0
Acquittements	0.26	0.35
AP Hellos	-	0.25
AP Registrations	-	0.58
Topologie virtuelle	0.165	-
Route Request	0.01	-
Route Reply	0.0043	-
Route Delete	0.005	-
Overhead Total	0.461	0.794

TAB. 6.1 – Trafic de contrôle (en paquets par nœud par seconde)

Le tableau 6.1 présente le trafic de contrôle généré par les deux protocoles, en détaillant tous les types de paquets générés. Nous pouvons voir que la structure virtuelle génère un overhead proactif, mais permet de réduire de façon importante l'overhead global, notamment de localisation d'un nœud. MEWLANA-RD qui reconstruit périodiquement sa dorsale présente un overhead important. De plus, l'overhead généré par les `AP Hellos` et `AP Registrations` augmente lorsque plus de nœuds sont présents dans le réseau. Au contraire, **SOMoM permet de**

réduire ce trafic de contrôle global, inondant juste les paquets de contrôle dans la dorsale. Ainsi, SOMOM permet un meilleur passage à l'échelle horizontal.

6.5.5 Mode ad-hoc

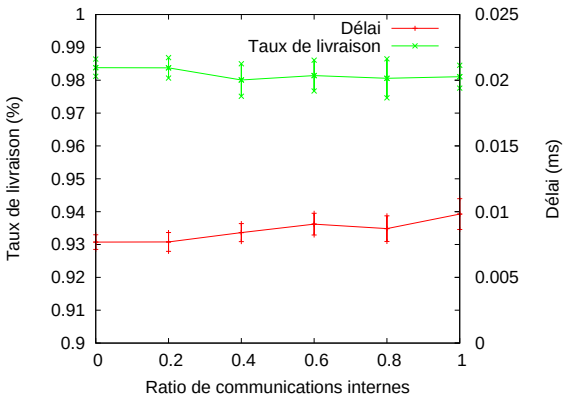


FIG. 6.9 – Performances de SOMOM en mode ad-hoc

Nous avons ensuite observé les performances du mode ad-hoc, i.e. autorisant des communications entre nœuds de la bulle ad-hoc. Nous pouvons voir que si la proportion de communications destinées aux nœuds internes augmente, ni le délai ni le taux de livraison ne sont impactés. **SOMOM permet d'autoriser des routes internes sans pour autant nuire aux performances globales**. Une telle extension nous parait donc parfaitement fonctionnelle.

6.5.6 Économie d'énergie

Enfin, nous avons implémenté la solution d'économie d'énergie décrite dans la section 4.7.2 page 75. Nous pouvons voir que le délai avec ou sans la solution d'économie d'énergie reste semblable (fig. 6.10). Par contre, le taux de livraison de SOMOM avec la solution d'économie d'énergie baisse de 2%. Lorsque la dorsale casse, la maintenance nécessite quelquefois de faire participer des nœuds endormis. Le temps de convergence de la maintenance prend donc en compte le temps de réveil. La convergence étant plus lente, les routes sont reconstruites moins rapidement, créant des paquets perdus. Cependant, un taux de livraison de 95% à une vitesse de 10m.s^{-1} dans un réseau ou les nœuds peuvent économiser leur énergie nous parait largement acceptable.

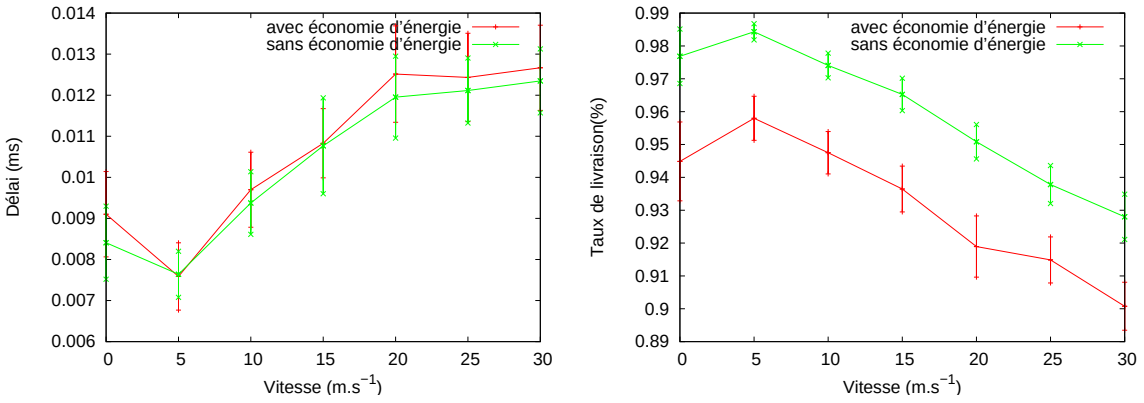


FIG. 6.10 – Impact du mécanisme d'économie d'énergie

6.6 Conclusion

Nous avons dans ce chapitre présenté une solution de gestion de la mobilité dans un réseau sans-fil multisauts connecté à Internet. L'originalité de cette solution réside dans son exploitation très simple de la dorsale d'auto-organisation présentée auparavant. Inspiré de Cellular IP, SOMoM tire parti de la dorsale, constituant le cache réactif pour les fonctions de localisation des terminaux. Pour les routes inverses, un terminal, grâce aux informations tirées de la dorsale, possède une route proactive vers le point d'accès. Ici encore, la structure d'auto-organisation démontre sa flexibilité en permettant la création d'un protocole de localisation efficace.

Par ailleurs, tous les terminaux d'un réseau hybride étant mobiles, nous avons proposé des fonctions de *cross-layer* entre le protocole de localisation et de maintenance de la dorsale : lorsqu'un changement dans la topologie de la dorsale survient, des actions sont enclenchées par SOMoM afin de mettre à jour les caches de localisation en conséquence. La dorsale se réparant de façon autonome lorsque des changements dans la topologie radio se produisent, les fonctions de SOMoM ne sont pas impactées : le protocole présente une grande robustesse à la mobilité.

Dans un futur proche, il serait intéressant des créer une suite de protocoles inter-agissant afin d'offrir un service d'accès à Internet de façon transparente et auto-organisée. Ainsi, un protocole permettant l'auto-configuration des terminaux (adresse IP, paramètres Mobile IP...) devrait être étudié. De même, un réseau hybride, pour des raisons de fiabilité, devrait comprendre dans le futur plusieurs passerelles vers Internet. Il faudrait donc proposer un mécanisme permettant de rétablir la charge des points d'accès, permettant aux terminaux de faire des handovers transparents. A terme, SOMoM pourrait constituer un équivalent des réseaux d'accès cellulaire, tout en proposant une couverture multisauts, diminuant les contraintes de dimensionnement et de déploiement.

Par ailleurs, SOMoM s'appuie sur la dorsale pour diffuser son trafic de contrôle mais également de données. Ainsi, la dorsale pourrait à forte charge constituer un goulet d'étranglement, créant des pertes dans les paquets de données. De manière générale, une structure d'auto-organisation permet de créer une vue logique logique : certains liens radio ne sont jamais utilisés bien qu'ils continuent à impacter sur les interférences. De même, certains nœuds concentrant les fonctions de routage peuvent excéder leur capacité en bande passante. Nous nous proposons donc dans le chapitre suivant de quantifier l'impact d'une auto-organisation sur la capacité, i.e. le débit disponible pour les applications. Nous validerons ainsi les approches hiérarchiques telles que VSR et SOMoM.

Bibliographie

[1] E. Belding-Royer and Y. Sun. Connectivity for ipv4 mobile ad hoc networks. Internet-Draft Version 00, IETF, November 2001.

[2] M. Benzaid, P. Minet, K. Al Agha, C. Adjih, and G. Allard. Integration of mobile-ip and olsr for a universal mobility. *Wireless Networks*, 10(4) :377–388, July 2004.

[3] B. Bhargava, X. Wu, Y. Lu, and W. Wang. Integrating heterogeneous wireless technologies : A cellular aided mobile ad hoc network (cama). *Mobile Networks and Applications*, 9(4) :393–408, August 2004.

[4] A. Campbell, J. Gomez, S. Kim, and C.-Y. Wan. Comparison of ip micro-mobility protocols. *Wireless Communications Magazine*, 9(1) :72–82, February 2002.

[5] A. Campbell, J. Gomez, C.-Y. Wan, and S. Kim. Cellular ip. Internet-Draft Version 01, IETF, december 2000.

[6] A. T. Campbell, J. Gomez, S. Kim, Z. Turanyi, C.-Y. Wan, and A. G. Valko. Internet micromobility. *Journal of High Speed Networks, Special Issue on Multimedia in Wired and Wireless Environment*, 11(3-4) :177–198, September 2002.

[7] M. Carli, A. Neri, and A. R. Picci. Mobile IP and cellular IP integration for inter access networks handoff. In *International Conference on Communications (ICC)*, Helsinki, Finland, June 2001. IEEE.

[8] C. Castelluccia. HMIPv6 : A hierarchical mobile ipv6 proposal. *ACM Mobile Computing and Communication Review (MC2R)*, 4(1) :48–59, April 2000.

[9] D. Cavalcanti, C. Cordeiro, D. Agrawal, B. Xie, and A. Kumar. Issues in integrating cellular networks, wlans, and manets : A futuristic heterogeneous wireless network. *IEEE Wireless Communications*, 12(3) :30–41, June 2005.

[10] M. Chiussi, D. A. Khotimsky, and S. Krishnan. Mobility management in third generation all-ip networks. *IEEE Communications Magazine*, 40(9) :124–135, September 2002.

[11] R. Droms. Dynamic host configuration protocol DHCP. RFC 2131, IETF, March 1997.

[12] 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.

[13] T. Ernst. *Network Mobility Support in IPv6*. Mathematics and computer science, University Joseph Fournier, Grenoble, France, 2001.

[14] M. Ghassemian, V. Friderikos, and A. H. Aghvami. Scalability analysis of internet gateway discovery algorithms for ad hoc networks. In *International Workshop on Wireless Ad-hoc Networks (IWWAN)*, London, UK, May 2005.

[15] S. Ghosh, R. G. Melhem, D. Mosse, and J. S. Sarma. Fault-tolerant mobile ip. Technical Report 11, Washington University, 1998.

[16] U. Jonsson, F. Alriksson, T. Larsson, P. Johansson, and G. Q. Maguire. Mipmanet - mobile ip for mobile ad hoc networks. In *Interational Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pages 75–85, Boston, USA, May 2000. ACM, IEEE Press.

[17] M. J. Miller, W. D. List, and N. H. Vaidy. A hybrid network implementation to extend infrastructure reach. Technical report, University of Illinois at Urbana-Champaign, January 2003.

[18] E. Nordstrom, P. Gunningberg, and C. Tschudin. Poster : Comparison of forwarding strategies in internet connected manets. In *Interational Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, Tokyo, Japan, May 2004. ACM.

[19] OPNET Modeler. http://www.opnet.com (v8.1).

[20] C. E. Perkins. IP mobility support for IPv4. RFC 3344, IETF, August 2002.

[21] P. Ratanchandani and R. Kravets. A hybrid approach to internet connectivity for mobile ad hoc networks. In *Wireless Communications and Networking Conference (WCNC)*, New Orleans, USA, March 2003. IEEE.

[22] P. M. Ruiz, F. J. Ros, and A. Gomez-Skarmeta. Internet connectivity for mobile ad hoc networks : Solutions and challenges. *IEEE Communications Magazine*, 43(10) :118–125, October 2005.

[23] V. Typpo. Micro mobility within wireless ad hoc networks : Towards hybrid wireless multihop networks. Thesis, University of Oulu, Finland, 2001.

[24] K. Xu, X. Hong, and M. Gerla. Landmark routing in ad hoc networks with mobile backbones. *Journal of Parallel and Distributed Computing*, 63(2) :110–122, 2003.

[25] X. Zhang, J. G. Castellanos, and A. T. Campbell. P-mip : paging extensions for mobile ip. *Mobile Networks and Applications*, 7(2) :127–141, 2002.

Publications

Conférence internationale

[1] Fabrice Theoleyre and Fabrice Valois. Mobility management in multihops wireless access networks. In World Scientific, editor, *Personal Wireless Communications (PWC)*, pages 146–153, Colmar, France, August 2005. IEEE, IFIP.

Rapport de recherche

[2] Fabrice Theoleyre and Fabrice Valois. Localization and routing in multihops wireless access networks. Research Report 5431, INRIA, January 2005.

Chapitre 7

Compromis entre capacité et auto-organisation

7.1 Introduction

Une caractéristique fondamentale des réseaux ad-hoc est l'utilisation de la radio comme médium de transmission. Ainsi, les liens peuvent présenter des instabilités à cause par exemple de la variation temporelle du canal radio ou des évanouissements de signal. De plus, la radio ne permet d'obtenir que des débits très inférieurs aux débits actuellement disponibles dans un réseau filaire. La dernière norme de IEEE 802.11 [1] fournit un débit radio théorique de 54Mbps, et ne fournit dans bien des cas qu'une bande passante disponible très inférieure. A l'opposé, il n'est pas rare de connecter des stations de travail en 1Gbps. De plus, les réseaux filaires classiques fournissent maintenant un médium de transmission dont la bande passante est entièrement disponible pour une paire en communication. A l'opposé, les réseaux radio ne peuvent fournir qu'un médium partagé, i.e. chaque terminal qui communique ne peut prendre qu'une partie de la bande passante globale. En outre, lorsqu'un terminal émet un signal, il va entrer en interférences avec d'autres nœuds émetteurs et récepteurs, diminuant le débit global offert aux autres nœuds du réseau. Bien que les réseaux radio présentent un atout évident de facilité d'utilisation et de déploiement, les performances proposées ne sont pas du tout du même ordre de grandeur. Si le réseau ad-hoc utilise de plus IEEE 802.11 comme couche MAC, des problèmes d'équité et de baisse de performances surviennent [5]. C'est pourquoi un nouveau protocole de niveau MAC devrait être proposé afin de juguler de tels problèmes.

Ainsi, de nombreux articles traitent de la problématique de la *capacité* dans les réseaux ad-hoc. La capacité est souvent définie comme le débit maximum agrégé que les terminaux peuvent envoyer. Évaluer la capacité permet de définir les applications éventuelles de tels réseaux et d'en déterminer les limites. Certains se sont attachés à l'étude de cette capacité pour des réseaux ad-hoc utilisant des interfaces IEEE 802.11. D'autres se sont au contraire affranchis de la couche MAC utilisée, supposée quelconque (IEEE 802.11, Bluetooth...), et ont étudié la capacité asymptotique du réseau.

Nous avons proposé dans le chapitre 3 page 31 une structure virtuelle dans laquelle certains nœuds sont élus leaders et certains liens radio privilégiés tandis que d'autres sont sous-utilisés. Puis, nous avons proposé un protocole de routage et de localisation s'appuyant sur cette topologie virtuelle. Cependant, il vient tout naturellement à l'esprit qu'une *simplification*[1] de la topologie radio peut amener à une baisse de la capacité offerte par le réseau, ce qui pourrait présenter un inconvénient majeur dans un réseau ad-hoc. Nous proposons donc dans ce chapitre d'étudier la capacité d'un réseau ad-hoc dans le but de quantifier l'impact d'une auto-organisation. Cependant, nous ne nous sommes pas attachés à une étude asymptotique : nous avons au contraire

[1] par exemple, certains nœuds sont sous exploités, des arêtes peuvent ne convoyer ni trafic de contrôle ni paquets de données

souhaité évaluer la capacité du réseau pour un protocole de routage et une topologie donnés. Nous définissons donc la capacité comme le débit maximum agrégé d'un ensemble de flots au niveau 3, de bout en bout. Notre évaluation prend donc en compte toute la pile protocolaire jusqu'à la couche de routage. En effet, le protocole de routage influera forcément sur la capacité : en utilisant certaines routes, il chargera plus certains liens radio, et donc impactera sur les interférences entre nœuds. Enfin, nous avons souhaité évaluer une telle capacité pour un réseau dans lequel une équité est introduite, afin de ne pas désavantager certains nœuds ou liens radio.

Nous allons dans un premier temps présenter un état de l'art de l'étude de la capacité dans un réseau ad-hoc. Ensuite, nous exposerons les hypothèses sur le réseau modélisé. La section 7.5 détaillera une modélisation des interférences radio en introduisant deux modèles d'équité différents. La section 7.6 présentera plusieurs méthodes permettant d'évaluer quantitativement la capacité d'un réseau ad-hoc. La section 7.7 étudie le cas particulier de la ligne afin d'exemplifier la modélisation proposée sur un cas simple. Enfin, nous présenterons dans la section 7.8 quelques résultats quantitatifs afin de pouvoir comparer plusieurs protocoles de routage à un protocole de routage s'appuyant sur une structure virtuelle.

7.2 La problématique de la capacité dans les réseaux ad-hoc

7.2.1 Modélisation des interférences

La modélisation des interférences radio représente une problématique clé de l'évaluation de la capacité. Lorsqu'une paire de nœuds entre en communication, les interférences vont interdire certaines autres communications concomitantes. Ainsi, les interférences régissent la distribution de la bande passante parmi l'ensemble des terminaux du réseau. [12] présente 3 modélisations des interférences radio existant dans la littérature (fig. 7.1 page ci-contre), auxquelles nous ajoutons la dernière[1] :

- Basé sur l'émetteur : une émission du nœud s est réussie si et seulement si tout autre émetteur s_2 est à une distance de s supérieure à la portée radio de s plus celle de s_2. Plus formellement :

$$dist(s, s_2) \quad > \quad r(s) + r(s_2) \tag{7.1}$$

- Le modèle protocolaire : une communication de s à d est réussie si d est à portée radio de s et s'il n'existe aucun autre émetteur à une distance de d inférieure à $1 + \Delta$ de celle de s.

$$\forall \text{ source } s_2, \ dist(d, s_2) \quad > \quad (1 + \Delta) \cdot dist(s, d) \tag{7.2}$$

- Le modèle émetteur / récepteur : une émission de s à d est réussie si et seulement si aucun des voisins de s ou d n'est en train de transmettre ou recevoir.

$$\forall u \in N(s) \cup N(d), \{u \notin source \wedge \notin destination\} \tag{7.3}$$

- SNR : une communication de s vers d est réussie si le ratio entre le signal reçu de s par d sur le bruit plus l'ensemble des signaux des autres émetteurs reçus par d est supérieur à un seuil. Une telle modélisation tient compte de l'ensemble des émetteurs du réseau et fait intervenir l'affaiblissement physique des signaux radio. Bien qu'une telle modélisation soit la plus fidèle, sa complexité la rend difficilement exploitable de façon analytique.

$$\frac{P_{\text{signal reçu}}}{\text{bruit} + P_{\text{autres signaux}}} > \text{seuil} \tag{7.4}$$

[1] Nous rappelons les notations introduites dans le paragraphe 2.4.2 page 14 :
- $dist(A, B)$ est la distance euclidienne séparant A de B
- $r(A)$ est la portée radio de A (en distance euclidienne également)
- P_A est la puissance en réception d'un signal A

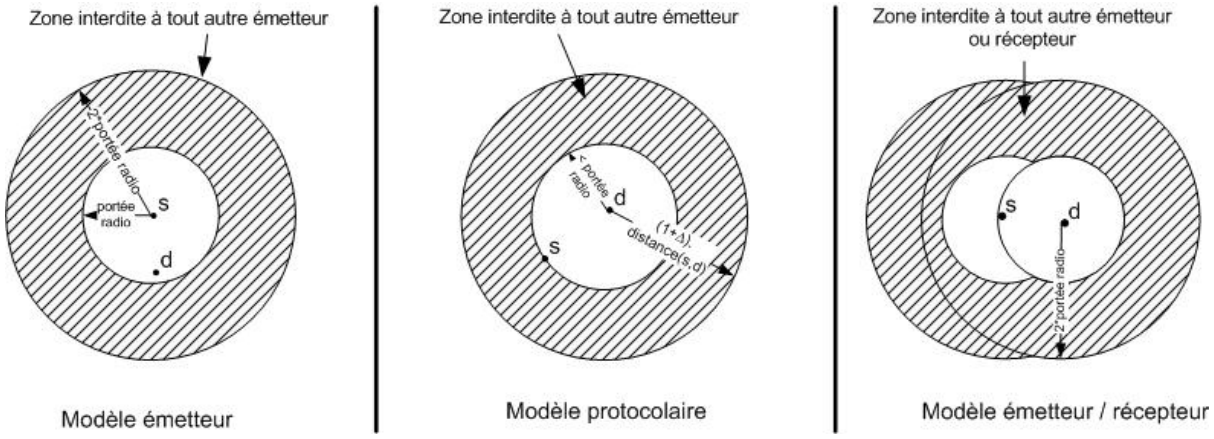


FIG. 7.1 – Modélisation possible des interférences radio

7.2.2 La capacité asymptotique des réseaux ad-hoc

Gupta & Kumar [7] ont présenté un travail pionnier dans l'étude de la capacité asymptotique des réseaux ad-hoc. Ils s'appuient dans leur modélisation sur le modèle d'interférences basé sur le récepteur (modèle protocolaire). Les auteurs définissent la capacité comme le débit maximum atteignable par le réseau, un débit étant atteignable s'il satisfait à toutes les contraintes en terme d'interférences et d'ordonnancement. De plus, chaque nœud choisit une destination aléatoire, l'équité entre tous les flux étant idéale. A partir d'un réseau contenu sur une sphère, les auteurs construisent des cellules de Voronoï, deux cellules pouvant communiquer directement si elles partagent une arête commune. De la même manière, deux cellules sont en interférence si les deux points qui leur correspondent sont en interférence. Grâce au modèle d'interférences, le nombre de cellule interférant avec une cellule donnée est donc borné. Partant d'un tel constat, les auteurs proposent un scheduling et un routage proche du routage par le plus court chemin. Soit n le nombre de nœuds dans le réseau. Si les nœuds sont distribués aléatoirement sur la sphère et si la portée radio est identique pour tous les nœuds, ils obtiennent une capacité par nœud de :

$$\Theta\left(\frac{1}{\sqrt{n \cdot log(n)}}\right) \tag{7.5}$$

De plus, si m nœuds chargés seulement de relayer sont ajoutés, la capacité devient :

$$\Theta\left(\frac{n+m}{n\sqrt{(n+m) \cdot log(n+m)}}\right) \tag{7.6}$$

Ainsi, la capacité n'est pas grandement augmentée, même avec des routeurs sans-fil dédiés à aider les autres nœuds du réseau. Par ailleurs, si le placement est optimal en terme de capacité et la portée radio ajustable par chaque nœud, la capacité devient :

$$O\left(\frac{1}{\sqrt{n}}\right) \tag{7.7}$$

Intuitivement, la réutilisation spatiale du médium radio permet à plusieurs paires d'envoyer simultanément leurs informations sans interférences. Cependant, le diamètre du réseau augmentant en $O(\sqrt{n})$, la capacité globale n'est pas une constante.

Les auteurs présentent donc dans leur article une étude asymptotique de la capacité d'un réseau ad-hoc lorsque tous les nœuds envoient une même quantité d'information. Cette étude est indépendante de tout protocole MAC sous-jacent, ce qui constitue sa force de généricité

mais sa faiblesse également : la capacité offerte par exemple par un réseau IEEE 802.11 a peu de chance de refléter un tel comportement. De plus, le modèle d'interférences est centré sur le récepteur. Cependant, les couches MAC dans les réseaux radio utilisent fréquemment un mécanisme d'acquittement. Si l'émetteur ne reçoit pas cet acquittement à cause d'un autre nœud interférant, il considérera le paquet comme perdu. Finalement, nous pensons que la capacité est extrêmement liée au protocole de routage utilisé puisque nous évaluons la capacité au niveau 3, et non au niveau radio. Ainsi, un protocole découvrant les routes les plus courtes en nombre de sauts ne maximisera pas la capacité du réseau, créant rapidement des goulots d'étranglement [4]. Finalement, bien qu'une étude de la capacité offerte par certaines topologies, protocoles de routage, méthodes d'accès au médium puisse être intéressante, la modélisation de Gupta & Kumar ne nous le permet pas.

7.2.3 Comment augmenter une telle capacité ?

Après une conclusion si dramatique, [6] propose d'augmenter la capacité en tirant parti de la mobilité des nœuds d'un réseau ad-hoc. Le problème de la capacité venant de l'allongement de la longeur moyenne d'une route, les auteurs proposent qu'une source envoie ses paquets à des nœuds relais dédiés. Ces nœuds relais se déplaçant aléatoirement, ils délivreront le paquet à la destination finale lorsqu'elle sera à portée radio directe, utilisant ainsi une route de longueur bornée. En conséquence, la capacité par nœud évolue en $O(1)$. Cependant, le délai peut être arbitrairement grand avant que le paquet n'arrive à la destination. Ainsi, une telle approche est peu applicable pour la plupart des applications. [2] suit la même approche, mais en bornant toutefois le délai. Cependant, les auteurs supposent que la destination est statique et que la vitesse et la direction de tous nœuds relais sont connues à priori par tous. Enfin, un routage GPS en sachant que la localisation de la destination est connue a priori est également requis. Ainsi, nous jugeons une telle approche difficilement applicable dans de nombreuses applications réalistes.

7.2.4 La capacité asymptotique des réseaux hybrides

Plusieurs articles [10, 14, 19] proposent d'étendre cette étude de la capacité aux réseaux hybrides, permettant de déployer des points d'accès interconnectés par un réseau filaire considéré de coût nul. Les auteurs proposent de garder la modélisation des interférences décrite dans [7]. La question dans un tel réseau est : "les AP vont-ils constituer un goulot d'étranglement, ou vont-ils au contraire permettre d'améliorer la capacité ?".

Dans [10], les nœuds sont uniformément distribués sur un disque. De plus, il existe en moyenne α points d'accès par nœud dans le réseau. Les auteurs décomposent le réseau en cellules de Voronoï d'une manière similaire à [7]. S'il est requis que le réseau soit connexe sans l'aide des AP la capacité devient :

$$O\left(\frac{1}{log(n)}\right) \tag{7.8}$$

Ceci constitue donc une amélioration en terme de capacité, mais elle n'atteint toujours pas une constante : les AP forment rapidement des goulots d'étranglement.

Dans [14], m AP sont placés régulièrement sur une grille, et n nœuds sont eux placés aléatoirement sur un disque, la portée radio étant uniforme pour tous les nœuds. Les auteurs utilisent ici aussi un découpage en cellules de Voronoï. Ils proposent par contre de changer de stratégie de routage afin de tirer pleinement parti de l'infrastructure : une source choisit d'envoyer ses paquets directement à la destination si elle appartient à l'une des k cellules les plus proches, et sinon l'envoie au point d'accès le plus proche. Cet AP relaiera lui même le paquet au point d'accès le plus proche de la destination. Ainsi, k permet de concentrer le trafic ou non dans l'infrastructure. Par exemple, si k est choisi arbitrairement grand, le réseau se comporte en mode ad-hoc pur. Cependant, contrairement aux autres travaux, les auteurs choisissent de maximiser

le trafic global agrégé, sans contrainte d'équité. Ainsi, un nœud peut ne jamais pouvoir communiquer car il crée trop d'interférences. Son trafic serait donc nul. Une telle iniquité nous semble un point faible d'une telle proposition, un tel cas de figure pouvant poser des problèmes dramatiques pour de nombreuses applications. Soit $T(m,n)$ le débit atteignable avec n nœuds et m points d'accès. Les auteurs, contrairement à [10] identifient plusieurs types de facteur d'échelle entre n et m, et obtiennent les résultats suivants :

$$m = o(\sqrt{n}) \quad \Rightarrow \quad T(m,n) = \Theta\left(\sqrt{\frac{n}{log\left(\frac{n}{m^2}\right)}}\right) \tag{7.9}$$

$$m = \Omega(\sqrt{n}) \quad \Rightarrow \quad T(m,n) = \Theta(m) \tag{7.10}$$

Dans [19], les auteurs étendent le travail de [14] en étudiant d'autres facteurs d'échelle entre n et m. Les auteurs calculent par contre ici la capacité par nœud, en réintroduisant donc l'équité. De plus, les nœuds mais également les AP sont placés de façon aléatoire, les AP pouvant en outre ajuster leur portée. Les capacités $C(n,m)$ obtenues sont donc :

$$m \leq \sqrt{\frac{n}{log\ n}} \quad \Rightarrow \quad C(n,m) = \Theta\left(\frac{1}{\sqrt{n\ log\ n}}\right) \tag{7.11}$$

$$\sqrt{\frac{n}{log\ n}} \leq m \leq \frac{n}{log\ n} \quad \Rightarrow \quad C(n,m) = \Theta\left(\frac{m}{n}\right) \tag{7.12}$$

$$m \leq n/log\ n \quad \Rightarrow \quad C(n,m) = \Theta\left(\frac{1}{log\ n}\right) \tag{7.13}$$

Les auteurs concluent donc en remarquant que dans le premier régime (eq. 7.11), seules les communications ad-hoc sont utilisées, le nombre d'AP étant trop faible. Au contraire, dans le $3^{ième}$ régime (eq. 7.13), la capacité atteint rapidement un maximum asymptotique. Ainsi, la capacité atteindra un maximum même si le nombre de points d'accès devient arbitrairement grand. Ceci corrobore donc les résultats de [10].

7.2.5 Évaluation de la capacité à travers des simulations

Les simulations sont également un moyen d'évaluer la capacité des réseaux ad-hoc. [13] étudie la topologie de la ligne et de la grille. En comparant les résultats analytiques atteignables avec les débits mesurés à l'aide de simulation, les auteurs concluent que IEEE 802.11 ne permet pas d'atteindre la capacité théorique maximale. Finalement, les auteurs envisagent de changer le type de trafic généré par les réseaux ad-hoc afin de rendre la capacité constante. En diminuant la longueur de la route, les interférences sont réduites. Ainsi, les auteurs proposent qu'un nœud communique avec un autre nœud à une distance x avec une probabilité :

$$p(x) = \frac{x^\alpha}{\int_\epsilon^{\sqrt{A}} t^\alpha dt}$$

Avec A étant l'aire de la surface du réseau, ϵ la distance minimale d'une source à une destination, α un paramètre du pattern de trafic. Ainsi, la capacité par nœud devient constante si $\alpha < -2$.

[11] propose d'étudier lui aussi la capacité par le biais de simulations. Un simulateur est implémenté, en utilisant le modèle d'interférences émetteur / récepteur. Chaque nœud génère des paquets suivant un taux μ en choisissant une destination aléatoirement, le paquet étant relayé sur la route la plus courte. Les auteurs modélisent une couche MAC idéale, découpant le temps en slots. Durant un slot, un nœud est choisi aléatoirement. S'il n'est pas en interférence avec une paire de nœuds déjà en communication, il choisit de relayer le premier paquet de sa

file. Si le prochain saut est déjà en interférence avec des nœuds communicant, le nœud choisit un autre paquet à relayer. Enfin, un slot est considéré alloué lorsqu'aucune communication supplémentaire ne peut être activée. La capacité simulée dépend donc du routage (de type plus court chemin) et de la topologie. Cependant, nous pensons qu'un tel schéma ne permet pas de refléter la capacité réellement atteignable d'un réseau, aucune équité dans la répartition de la bande passante n'étant introduite.

7.2.6 Optimisation du placement des points d'accès

[17] traite de la problématique de la capacité en étudiant le dimensionnement d'un réseau sans-fil. Les auteurs tentent de minimiser le nombre de points d'accès permettant d'atteindre une certaine capacité. Ils introduisent une formulation en programmation linéaire (LP), et un algorithme glouton de placement des AP. Ils proposent 3 modèles d'interférences. Cependant, le modèle le plus complexe propose de modéliser les interférences comme une diminution linéaire du débit en fonction de la longueur de la route. Un tel modèle, trop macroscopique, ne nous semble absolument pas modéliser le comportement réel des interférences radio, les inter-blocages entre émetteurs/récepteurs, faussant ainsi les résultats.

7.2.7 Formulation en programmation linéaire du problème de la capacité

Dans [9], les auteurs proposent d'utiliser une formulation en programmation linéaire en nombre entiers pour traiter le problème de la capacité. Ils souhaitent se focaliser sur l'étude de la capacité lorsque la topologie et la charge de trafic sont connues. Les auteurs formulent la capacité comme un problème de multi-flots. Cependant, contrairement à un réseau filaire, la complexité d'une telle étude résulte dans l'estimation de la capacité inhérente à chaque arête, puisque des interférences radio créent des boucles de contrôle entre les débits de chaque arête. Ainsi, les auteurs construisent une borne supérieure et inférieure de la capacité. Dans un premier temps, le graphe des conflits entre les arêtes est construit : un sommet est associé à chaque arête, et il existe une arête entre deux sommets dans le graphe des conflits si les liens radio correspondant étaient en interférence dans le graphe d'origine. Les auteurs proposent de construire le graphe des conflits découlant du modèle des interférences radio de type SNR. Les auteurs estiment que, dans la borne inférieure, un débit est atteignable lorsque l'ensemble des arêtes activables simultanément constitue un ensemble indépendant (IS). Ainsi, une combinaison linéaire de ces ensembles d'arête fournit un scheduling faisable. Les auteurs proposent donc de trouver plusieurs ensembles indépendants, et à chaque slot de temps t, de n'en activer qu'un seul. En conclusion, la capacité allouée à une arête e dépend du nombre d'ensembles indépendants dans lesquels elle apparaît. Ainsi, les auteurs modélisent une couche MAC inéquitable : tous les ensembles indépendants possèdent la même probabilité d'activation. Nous pensons au contraire qu'il faut modéliser une couche MAC idéale, afin d'en tirer la capacité. Nous verrons en quoi notre approche nous permet de suivre une telle propriété.

[12] propose de maximiser le débit d'un groupe de sources vers un groupe de destination. Les auteurs proposent un ordonnancement d'arêtes tel que deux arêtes ordonnancées ne peuvent être en interférences : un algorithme glouton permet de partager la capacité entre les arêtes en interférences, en allouant des slots aux arêtes dans un ordre décroissant de leur longueur. En effet, les auteurs montrent l'existence de la condition suffisante suivante : un ordonnancement est faisable si la capacité allouée à une arête et à l'ensemble des arêtes plus longues et en interférence est inférieure à la capacité radio. Cependant, ils ont tendance ainsi à sous-estimer la capacité atteignable : si deux arêtes e_1 et e_2 sont en interférence avec une même troisième, e_3, la capacité est partagée entre notamment e_1, e_2 et e_3. Pourtant, e_1 et e_2 pourraient parfaitement envoyer des paquets simultanément. Notre modélisation, comme nous le verrons plus loin, permet de partager plus finement la capacité entre les arêtes en interférences en étudiant plus précisément les interactions dans le 2-voisinage d'un nœud. Finalement, les auteurs proposent une métrique

permettant d'ajuster l'équité entre les flux d'un réseau : ils bornent par une constante le ratio entre le flux de débit minimal et celui de débit maximal. La discrimination entre les différents flux est donc limitée.

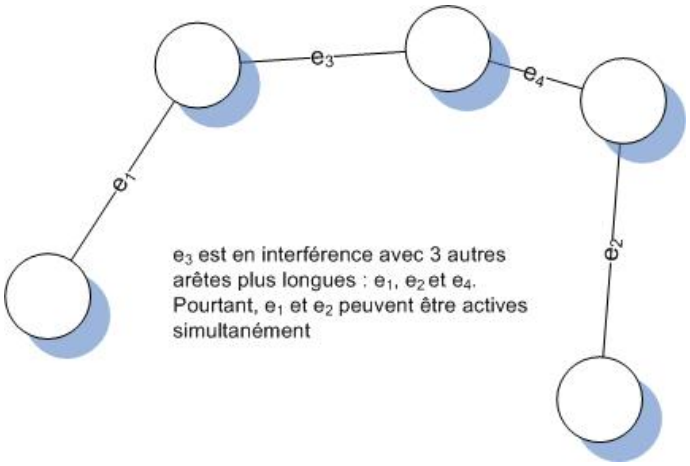


FIG. 7.2 – Configuration dans laquelle [12] sur-estime les interférences

7.3 Démarche proposée

Nous souhaitons évaluer la capacité inhérente à une topologie et à un protocole de routage donnés dans un réseau de type IEEE 802.11. Ainsi, nous souhaitons modéliser le comportement d'une pile protocolaire de la couche physique à la couche réseau. Nous pourrons notamment évaluer l'impact d'une structure virtuelle sur la capacité : la topologie virtuelle et le routage l'exploitant présentent-ils une capacité amoindrie ?

Nous avons donc choisi de suivre la démarche générale suivante :

- Nous établissons des règles modélisant le partage de la bande passante entre un nœud et ses voisins proches. La bande passante disponible pour un nœud dépend des voisins avec lesquels il est en interférence et de leur activité. Nous nous focalisons sur un trafic bidirectionnel comprenant des acquittements de type *saut par saut*.
- Ces interactions radio sont traduites en une liste de contraintes locales au voisinage d'un nœud. Nous présentons un modèle optimiste et un modèle pessimiste du partage de la bande passante, présentant donc des contraintes différentes. La topologie radio étant une entrée du problème, les contraintes d'interférences sont donc clairement déterminées.
- Le trafic est modélisé de bout en bout, en multisauts. Ainsi, nous utilisons le problème des multiflots : la charge d'un flux se répercute individuellement sur chaque lien radio faisant partie de la route entre la source et la destination données. Nous rappelons qu'une telle route est, par hypothèse, extraite du protocole de routage lui-même.
- En combinant les contraintes liées aux multiflots et les contraintes de partage de la bande passante, nous avons une série de contraintes globales nous donnant les flux, faisables ou non, avec la topologie et le protocole de routage donnés en entrée.
- Enfin, nous définissons formellement la capacité afin d'extraire des contraintes définissant la capacité maximale atteignable par le réseau.

Finalement, des résultats quantitatifs de capacité maximale d'une topologie et d'un protocole de routage donnés peuvent être obtenus. Nous pouvons donc comparer les capacités inhérentes à différents types de protocoles de routage.

7.4 Hypothèses

Cette section présente la formulation générale du programme linéaire ainsi que les hypothèses et notations utilisées dans notre modélisation.

7.4.1 Programmation linéaire

7.4.1.1 But de la programmation linéaire

Les problèmes de programmation linéaire sont des problèmes d'optimisation dans lesquels les contraintes et l'objectif à atteindre sont représentés sous la forme d'équations linéaires. De nombreux programmes tels que Cplex [8] ou lpSolve [15] permettent de maximiser la fonction objective tout en ne violant aucune des contraintes formulées. La programmation linéaire est par exemple très utilisée dans le domaine des réseaux ou de l'économie.

Les contraintes étant données sous forme linéaires, elles forment un polytope[1] convexe. La fonction objectif étant également linéaire, les optima locaux sont également des optima globaux. Ainsi, le solveur permet de trouver une solution permettant de maximiser l'objectif sans violer les contraintes. Des algorithmes génétiques, ou de recuit-simulés sont donc inutiles dans de tels problèmes. Cependant, deux cas de figure ne donnent aucune solution : si certaines contraintes sont en contradiction, le polytope formé par les contraintes est vide. Ainsi, il n'existe pas de solution. Inversement, si les contraintes ne forment pas un polytope borné dans une direction, la fonction objectif peut tendre vers l'infini sans violer les contraintes. Cependant, de tels cas de figure sont assez rares dans les problèmes rencontrés.

Cependant, nous ne donnons ici qu'un bref aperçu de la programmation linéaire. Le lecteur pourra se référer à [18] pour plus de détails et d'explications.

7.4.1.2 Démarche entreprise

Notre programme linéaire suit la forme générale décrite dans LP 1. Les interférences radio sont traduites en une première série de contraintes. Ensuite, les flux de bout en bout sont éclatés en une somme de charges sur chaque lien radio qui composent la route p empruntée par le flot.

L'ensemble des routes est défini par $\mathcal{P} = \{p =< s, u_1, u_2, \ldots, u_k, d >\}$, utilisant la même notation que définie dans le chapitre 4.2 page 55. Un chemin p est donné comme la suite des sommets que doivent emprunter les paquets pour aller d'une source s à une destination d. La fonction objectif est définie en maximisant la quantité de trafic envoyé sur l'ensemble de ces routes comme nous le verrons plus loin dans la section 7.6 page 141.

Programme Linéaire 1 (Modèle générique)

Maximiser	*fonction objective sur* $\mathcal{P}$
	Soumise à :
Contraintes de partage de ressources autour de u	$\forall$ *nœud u*
Répartition du flux envoyé sur p	$\forall$ *route p*

7.4.2 Hypothèses et notations

Comme toute modélisation, nous avons effectué un certains nombres d'hypothèses sur le comportement de la couche MAC et de la couche radio :

[1] polygone généralisé à n dimensions

- La couche radio est modélisée comme parfaite. Ainsi, le signal radio ne fluctue pas au cours du temps, et la bande passante peut être pleinement utilisée sans qu'aucun paquet de données n'arrive corrompu à destination. Un tel médium est donc irréaliste dans un environnement radio réel. Cependant, de telles hypothèses sont nécessaires afin de proposer une modélisation utilisable de la couche radio.
- Les antennes radio sont supposées quelconques : nous n'utilisons que la topologie radio découlant de la couche radio et des antennes données. Des antennes directionnelles modifiant la topologie radio peuvent être ainsi parfaitement utilisées.
- La couche MAC est également idéale : la couche MAC régente l'accès aux médium tout en évitant les collisions et en partageant équitablement la bande passante. Nous verrons plus loin ce que nous entendons par *équitable*
- Nous supposons l'existence d'un ordonnancement global utilisant un mode synchrone et sans faille dans tout le réseau (permettant ainsi d'atteindre la capacité maximale voulue).
- Le trafic de données est bidirectionnel. Un trafic f_{sd} de la source s vers la destination d engendre un trafic de réponse de quantité βf_{sd} sur la route inverse (de d vers s). Une telle hypothèse permet par exemple de modéliser le trafic http de type requête/réponse.
- Un envoi au niveau MAC s'accomplit avec l'envoi d'une trame de données, et par le retour d'un acquittement envoyé par la destination MAC. Le temps séparant la trame de données de l'acquittement est considéré nul.

Nous utilisons par ailleurs les notations suivantes (en sus de celles introduites dans la section 2.4.2 page 14 :

- BW est la bande passante disponible au niveau radio. Ainsi, un terminal s'il est seul peut envoyer à un débit maximum de BW
- n est le nombre de nœuds du réseau
- $f(p)$ est le débit de données envoyé sur la route p.
- Soit u un nœud. $T(u)$ est la quantité totale de trafic envoyée par u :
  $T(u) = \sum_{v \in N(u)} T(u,v) + T_c(u)$ avec :
  - $N_k(u)$ est le k-voisinage de u, i.e. l'ensemble des nœuds à au plus k sauts radio de u. $N_1(u)$ est noté $N(u)$ pour plus de simplicité. Nous pouvons remarquer que nous considérons le voisinage fermé, i.e. $\forall k, u \in N_k(u)$
  - $\Delta_k(u)$ est la taille du k-voisinage de u. Ainsi, $\Delta_k(u) = |N_k(u)|$.
  - $T(u,v)$ est le trafic en unicast transitant sur le lien (u,v). Ce trafic est lui même la somme de :
    * flux de données envoyés sur les routes comprenant (u,v) : $\sum_{p \ni (u,v)} f(p)$,
    * flux de données en réponse aux flux envoyés sur les routes comprenant (v,u) : $\sum_{p \ni (v,u)} \beta f(p)$,
    * acquittements pour les paquets envoyés sur le lien radio (v,u) : $\alpha \left(\sum_{p \ni (v,u)} f(p) + \sum_{p \ni (u,v)} \beta f(p) \right)$.
  - $T_c(u)$ est le trafic de contrôle envoyé en broadcast par u. Nous pouvons noter que ce trafic n'est pas propre à une arête puisqu'avec un médium radio, il suffit d'une transmission pour couvrir l'intégralité de ses voisins.

Nous pouvons noter que grâce à une telle notation, la décomposition du trafic $f(p)$ en la somme des charges sur chaque lien radio $T(u,v)$, $((u,v) \in p)$ apparaît triviale.

7.4.3 Modèle d'interférences utilisé

Nous avons supposé un modèle d'interférences de type émetteur / récepteur, se rapprochant de celui utilisé pour modéliser IEEE 802.11. Ainsi, un envoi d'une source à une destination bloque l'intégralité des voisins de la destination mais également de la source. Sinon, la source pourrait manquer l'acquittement de la destination et donc être obligée de retransmettre inutilement ses paquets. Ce modèle nous permet d'expliquer facilement les interactions en cours dans le voisinage

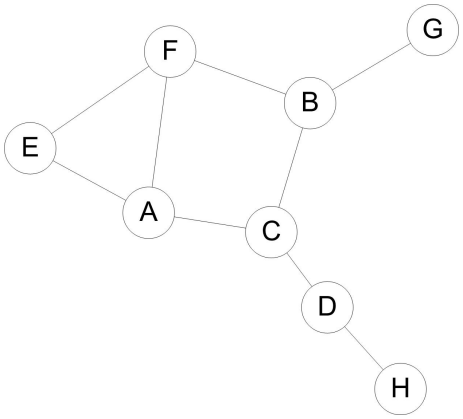


FIG. 7.3 – 2-voisinage d'un nœud C

d'un nœud pour le partage de la bande passante.

Cependant, notre modèle d'expression de la capacité est adaptable à tout autre modèle d'interférence. Il est uniquement basé, comme le lecteur pourra le vérifier au cours de ce chapitre, sur le graphe d'interférences. Cependant, nos explications se cantonnent au modèle d'interférence sus-nommé pour des raisons de simplicité.

7.5 Modélisation du partage de la ressource radio

Nous avons proposé un modèle permettant de déterminer les règles de partage de la bande passante dans un réseau radio. Cependant, nous avons souhaité introduire de l'équité dans un tel partage afin d'obtenir des résultats réalistes. Comment se déroule un tel partage dans IEEE 802.11 ? Chaque nœud va initialiser un timer avec une valeur aléatoire comprise entre 0 et un maximum (son *backoff*). Le nœud peut décrémenter son backoff si aucune activité radio n'est détectée sur le médium. Sinon, il l'interrompt temporairement. Finalement, lorsque le backoff devient nul, le nœud peut envoyer son paquet. Ainsi, chaque nœud tirant un backoff dans les mêmes bornes, le partage des ressources se réalise de façon équitable. Cependant, il existe des cas plus complexes, ou dans un milieu avec interférences, un tel partage est inéquitable [5], mais nous ne rentrerons pas dans ces détails.

Nous pensons que pour être réaliste, nous devons modéliser une équité de la couche MAC. Cependant, nous pouvons définir deux types différents d'équité : entre les nœuds en contention, ou entre les arêtes radio. La première approche est celle utilisée dans de nombreux protocoles : chaque nœud qui rentre en interférence avec d'autres nœuds reçoit une part égale de la bande passante. L'équité entre liens radio est rencontrée plus rarement. Bien que les liens radio n'aient pas d'existence physique[1], une telle équité permet, selon nous, d'éviter la formation de goulots d'étranglement. Un nœud qui possède beaucoup de voisins et donc peut-être plus de trafic à relayer se voit offrir une bande passante identique aux autres nœuds. Il nous semble peut être plus efficace de donner à un nœud une bande passante proportionnelle au nombre de ses voisins. Ainsi, dans un réseau en ligne à 3 nœuds, le nœud du milieu recevra 2 fois de plus de bande passante que les extrémités, lui permettant de relayer plus de paquets.

Enfin, pour chaque modèle d'équité, nous présentons un modèle pessimiste du partage de la bande passante, et un modèle optimiste.

[1] Un lien filaire est dédié aux deux terminaux dont il est l'extrémité. Au contraire, un lien radio est en fait un symbolisme permettant de définir une possibilité de communication entre deux nœuds. Par contre, le médium radio doit être partagé entre tous les nœuds en contention et utilisant la même fréquence. Par ailleurs, une réutilisation spatiale du spectre radio étant possible, nous voyons bien que le terme de *lien radio* constitue un raccourci forcément simplificateur.

7.5.1 Équité orientée nœuds

7.5.1.1 Partage pessimiste des ressources radio

Dans cette partie, nous allons présenter un modèle pessimiste de partage des ressources radio, tout en maintenant une équité dans la répartition de bande passante entre les nœuds. Selon le modèle d'interférences émetteur / récepteur, lorsqu'une paire est en communication, aucun voisin ni de la source ni de la destination n'est autorisé à émettre ou recevoir. Ainsi, les deux communications (A → B) et (C → D) sont possibles si et seulement si aucun des liens radio (A,C), (A,D), (B,C) et (B,D) n'existe. Ainsi, les contentions entrant en compte au niveau d'un nœud ne font intervenir que son 2-voisinage.

Ainsi, nous pouvons proposer un modèle pessimiste de partage de la ressource radio si les activités dans le 2-voisinage d'un nœud se font séquentiellement. En d'autres termes, lorsqu'un nœud parle, aucun de ses 2-voisins n'est autorisé à parler en même temps. Afin de modéliser une certaine équité, un même pourcentage de la bande passante radio est distribué à tous les nœuds faisant partie d'un 2-voisinage commun. Ceci constitue un cas pire : si nous prenons l'exemple de la topologie illustrée sur la figure 7.5 page ci-contre, (A,E) / (D,H) ou (E,F) / (C,D) représentent des paires de communications simultanées possibles qui ne seraient pas autorisées à transmettre simultanément dans notre modèle.

Il est important de noter que nous traduisons par activité tout mécanisme soit de réception soit d'émission. En effet, un acquittement étant nécessaire, le récepteur utilise donc le lien radio dans le sens inverse du paquet de données. D'autre part, tous les ensembles possibles de 2-voisins sont examinés. Il en existe n : le 2-voisinage centré sur chaque nœud du réseau. Ainsi, les communications vers E provenant d'un nœud ne faisant pas partie du 2-voisinage de C seront prises en compte par exemple par les contraintes de partage centrées sur le nœud A, E, ou le nœud émetteur lui-même.

Nous pouvons également observer que dans ce type de partage, nous ne distinguons pas le trafic broadcast de celui unicast : l'un comme l'autre bloquent l'intégralité du 2-voisinage d'un nœud.

Soit c un nœud du réseau (appelé le *centre* de son 2-voisinage). Cette modélisation peut être traduite formellement de la manière suivante :

- La bande passante est distribuée uniformément entre tous les nœuds en contention potentielle, i.e. le centre c et tous ses 2-voisins, $N_2(c)$:

$$\forall c,\ \forall u \in N_2(c),\ T(u) \leq \frac{BW}{\Delta_2(c)} \tag{7.14}$$

  Nous pouvons noter que pour chaque centre c (il en existe n), un ensemble de $\Delta_2(c)$ équations est donné. Ainsi, la capacité $T(u)$ allouée à un nœud u est bornée par exactement $\Delta_2(u)$ équations, une centrée sur chacun de ses 2-voisins. Sa capacité finale sera donc contrainte par l'équation la plus restrictive, i.e. le 2-voisin qui possède le plus de 2-voisins.
- Cependant, il est important de noter que dans la capacité allouée à un nœud, celui-ci doit envoyer à la fois son trafic de données, mais également son trafic de contrôle. De plus, un nœud partage équitablement la bande passante qu'il obtient entre tous les liens radio qu'il possède :

$$\forall u,\ \forall v \in N(u) - \{u\},\ T(u,v) \leq \frac{T(u) - T_c(u)}{\Delta(u) - 1} \tag{7.15}$$

Ainsi, l'équation 7.14 modélise le partage de la bande passante équitablement entre les nœuds tandis que l'équation 7.15 régit le partage au sein même d'un nœud entre tous ses liens radio.

Nous pouvons remarquer qu'un nœud très chargé recevra la même quantité de bande passante qu'un autre nœud peu chargé. Nous obtenons donc le programme linéaire LP 2 page suivante.

Programme Linéaire 2 (Partage pessimiste équitable entre nœuds)

Maximiser	*fonction objective sur* $\mathcal{P}$
	Soumise à :
Ensemble d'équations du type 7.14	$\forall c \in V$
Ensemble d'équations du type 7.15	$\forall u \in N_2(c)$
Répartition du flux envoyé sur p	$\forall$ *route* p

7.5.1.2 Partage optimiste des ressources radio

Dans le précédent modèle, nous avons établi des contraintes fortes dans le partage de la bande passante : certaines interférences sont sur-estimées afin d'être assurés que des partages ne rentrent pas en conflit. Comme nous l'avions fait remarquer, certaines communications sont pourtant possibles sans interférences au sein du 2-voisinage d'un nœud. Si nous reprenons la figure 7.5 page 134, les communications (A → E) et (B → G) sont faisables puisque ces paires ne possèdent pas d'arête commune. IEEE 802.11 autoriserait la transmission simultanée de ces deux paires.

Ainsi, nous proposons ici un modèle permettant de prendre en compte de telles transmissions simultanées. Cependant, un tel modèle est optimiste dans le sens où nous ne traduisons des contraintes que localement à un 2-voisinage. Ainsi, nous supposons l'existence d'un schéma d'ordonnancement global qui permet de satisfaire à l'intégralité des contraintes locales. Cependant, un tel ordonnancement est dans certains cas impossible. Ce modèle constitue donc une borne supérieure de la bande passante attribuable au réseau.

Supposons qu'initialement le canal est libre. Un nœud u prend le médium et communique avec un de ses voisins v. Si un nouveau nœud u' souhaite envoyer des paquets et donc prendre le médium, il ne doit être voisin ni de u ni de v. De plus, il doit envoyer des paquets vers un nœud v' voisin ni de u ni de v afin d'éviter l'apparition de collisions. Le mécanisme de RTS/CTS de IEEE 802.11 est justement conçu pour suivre un tel schéma. Nous modélisons donc le médium radio comme une entité centralisée qui distribue la bande passante aux paires qui souhaitent communiquer, tout en maintenant une certaine équité. Une nouvelle communication est autorisée si elle n'entre pas en interférence avec une paire déjà active. Un tel schéma est donc très proche de la structure des *ensembles indépendants* dans la théorie des graphes (cf. chapitre 2.4 page 13).

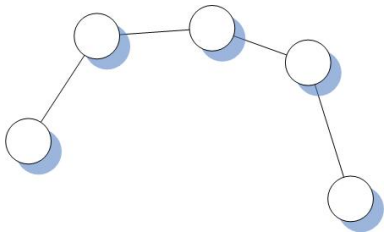
Prenons le graphe des contentions $L_{1,2}(\mathcal{LG})$ (cf. fig. 7.4 page ci-contre) défini comme :

- G_c est le graphe du 2-voisinage de c.
- $\mathcal{LG} = \mathcal{L}(G_c)$ est le *linegraph* de G_c : un sommet est associé à chaque arête de G_c, et une arête entre deux sommets existent si les deux arêtes correspondantes dans G_c sont adjacentes dans le graphe d'origine.
- $L_{1,2}(\mathcal{LG})$ est la 2-fermeture de $\mathcal{LG}$: des arêtes entre 2-voisins dans $\mathcal{LG}$ sont ajoutées.

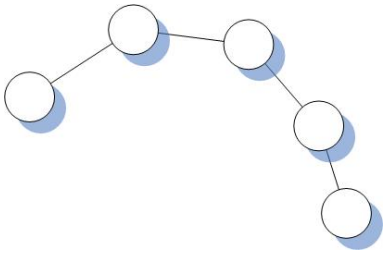
Les liens radio activables simultanément forment un ensemble de sommets indépendant dans le $L_{1,2}(\mathcal{LG})$. Ainsi, si un ensemble de sommets dans $L_{1,2}(\mathcal{LG})$ forme un ensemble indépendant total (i.e. maximal pour l'inclusion cf. chapitre 2.4 page 13), l'ensemble des liens radio correspondants forme un ensemble maximal de liens activables simultanément.

Nous utilisons ici directement le modèle d'interférences émetteurs / récepteurs. Cependant, rien n'empêche d'utiliser ici un autre modèle d'interférences. Ainsi, au lieu de construire le $L_{1,2}(\mathcal{LG})$, nous relirions ensemble les liens radio en interférence dans le $\mathcal{LG}$: nous utiliserions donc

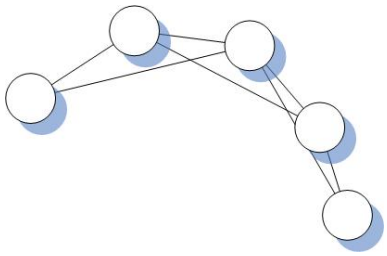
[0] La k-fermeture d'un graphe correspond à ajouter des arêtes entre toutes les paires de sommets à une distance inférieure à k sauts



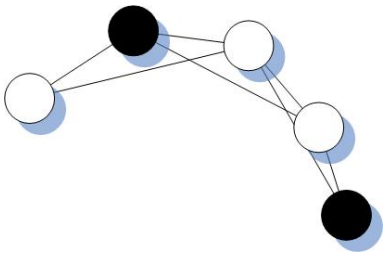
(a) Topologie radio correspondant au réseau ad hoc



(b) Linegraph correspondant : à chaque arête du graphe radio correspond un sommet du linegraph, et deux sommets du linegraph possèdent une arête commune si les deux arêtes étaient adjacentes dans le graphe d'origine



(c) Graphe de contention : ensemble des arêtes en contention. Dans le modèle d'interférences que nous utilisons, le graphe de contention correspond à une 2 fermeture du linegraph (note 0 page précédente)



(d) Exemple d'ensemble indépendant dans le graphe de contention, correspondant à des arêtes activables simultanément.

FIG. 7.4 – Construction du graphe de contention

directement le *graphe de contention*. Par contre, notre modélisation ne fonctionne pas avec un modèle d'interférences non *binaire*. Or, de telles interférences sont plus réalistes : une arête plutôt que de bloquer totalement une autre arête va plutôt diminuer son débit, dans une proportion d'autant plus importante que ces deux arêtes sont proches. Cependant, les interférences *binaires* représentent une bonne première approximation, comme nous avons pu le remarquer avec des simulations. Par ailleurs, une modélisation plus complexe des interférences n'a, au mieux de notre connaissance, jamais été utilisée pour évaluer la capacité des réseaux radio, sans nul doute à cause de la complexité de son utilisation dans des modèles analytiques.

Le médium radio modélisé de façon centralisée distribuera donc la bande passante entre les ensembles indépendants en maintenant une équité entre les nœuds. Soit $BW(I)$ la bande passante allouée à l'ensemble indépendant $I \in \mathcal{I}$, I étant l'ensemble de tous les ensembles indépendants maximaux possibles dans $L_{1,2}(\mathcal{L}(G_c))$. $BW(I)$ est proportionnel à $P(I)$, la probabilité que I soit sélectionné par le médium radio centralisé. $P(I)$ dépend des arêtes qui le constituent et de leur ordre d'activation. La bande passante est distribuée dans le 2-voisinage de c de la manière suivante :

$$BW(I) = P(I) \cdot \left(BW - T(c) - \sum_{u \in N(c) - \{c\}} T_c(u) \right) \tag{7.16}$$

$$\Rightarrow BW \geq T(c) + \sum_{I \in \mathcal{I}} BW(I) + \sum_{u \in N(c) - \{c\}} T_c(u) \tag{7.17}$$

La bande passante totale d'un lien communicant (u, v) est la somme de la bande passante allouée à tous les ensembles indépendants auxquels il appartient :

$$T(u,v) \leq \sum_{I \ni (u,v)} BW(I) \tag{7.18}$$

$$T(u,v) \leq \left(BW - T(c) - \sum_{x \in N(c)-\{c\}} T_c(x) \right) \cdot \sum_{I \ni (u,v)} P(I) \tag{7.19}$$

De plus, $\sum_{I \ni (u,v)} P(I)$ est exactement la probabilité que le lien radio (u,v) soit activé par le médium. Cette quantité est dénotée par la suite $P(u,v)$.

$$\forall (u,v) \in N_2(c) - \{c\}, T(u,v) \leq \left(BW - T(c) - \sum_{x \in N(c)-\{c\}} T_c(x) \right) \cdot P(u,v) \tag{7.20}$$

Cependant, la quantité $P(I)$ et donc $P(u,v)$, dans la plupart des cas, n'est pas calculable sans trouver l'ensemble $\mathcal{I}$ de façon exhaustive. Or $\mathcal{I}$ présente une taille exponentielle. Ainsi, le calcul exact devient rapidement impossible dans un réseau de taille même raisonnable (quelques dizaines de nœuds). Nous proposons une approche statistique afin de proposer une estimation de $P(u,v)$, dénotée $freq(u,v)$ par la suite.

Les fréquences $freq(u,v)$ doivent prendre en compte un partage équitable entre les nœuds. Nous proposons donc le schéma suivant construisant les ensembles indépendants :

- Initialement, tous les nœuds sont non bloqués, i.e. ils peuvent potentiellement être activés par le médium pour entrer en communication
- Tant qu'un nœud reste encore non bloqué, en choisir un aléatoirement. Soit u ce nœud.
- Choisir aléatoirement un voisin v de u, tel que v est non bloqué
  - Si un tel nœud v existe, activer la communication (u,v) et marquer tous les voisins de u et v comme bloqués
  - sinon, marquer u comme bloqué

Cet algorithme est répété n fois. Ainsi, $freq(u,v)$ est égal à la proportion des tirages dans lesquels l'arête (u,v) a été activée. Nous pouvons remarquer que chaque lien est dirigé : le lien (u,v) a une probabilité faible de recevoir la même quantité de trafic que le lien (v,u).

Maintenant, il est également important de prendre en charge le trafic de contrôle. L'équation 7.20 permet de modéliser une telle répartition : lorsque le nœud c envoie du trafic de contrôle, il est nécessaire que l'ensemble de son 2-voisinage stoppe toute activité radio. En effet, les 1-voisins de c doivent recevoir le paquet de contrôle, bloquant ainsi toute émission des 2-voisins. D'un autre côté, les 1-voisins et 2-voisins de c incluront leur trafic de contrôle dans la bande passante qui leur a été allouée. Ceci constitue également une autre hypothèse optimiste : il est possible que ce nœud ne puisse envoyer tout son trafic de contrôle dans la bande passante qui lui a été allouée sans toutefois entrer en interférences avec d'autres liens radio.

La bande passante est distribuée par lien : un nœud ne peut ré-allouer la bande passante localement d'une arête qui n'utilise pas toute la bande passante. En effet, le médium radio centralisé alloue la capacité en tenant compte des interférences potentielles de chaque arête, et non de chaque nœud. Ainsi, une redistribution locale à un nœud de de la bande passante pourrait violer les contraintes d'interférences. Afin de proposer une borne réaliste, nous souhaitons éviter un tel cas de figure.

Comme nous l'énoncions précédemment, nous supposons que les ordonnancements locaux peuvent se combiner en un mécanisme global ne violant pas les règles d'interférences. Cependant, nous voyons bien ici que l'union des ensembles indépendants, définis sur chacun des centres possibles, peut ne pas former globalement un ensemble indépendant. Notre formulation LP 3 page suivante négligeant un tel effet, elle constitue une borne supérieure de notre estimation de la capacité.

Programme Linéaire 3 (Partage optimiste équitable entre nœuds)

Maximiser	*fonction objective sur* $\mathcal{P}$
	Soumise à :
Ensemble d'équations du type 7.20 page ci-contre	$\forall c \in V$
Répartition du flux envoyé sur p	$\forall$ *route p*

7.5.2 Équité orientée lien radio

Nous avons présenté dans la section précédente une modélisation du partage de la capacité entre nœuds en concurrence en suivant une certaine équité entre les nœuds. Afin de distribuer à un nœud une bande passante proportionnelle au nombre de ses voisins, nous proposons un modèle modifiant le mode de distribution de la bande passante radio en respectant une équité entre liens radio. Ce modèle étant construit de manière similaire au précédent, nous n'en présentons ici que les différences.

Nous introduisons ici les notations utiles pour ce nouveau modèle d'équité :

- $n_k(e)$: le k-voisinage dans $\mathcal{LG}$[1] d'une arête e dans G. Dans cet ensemble, chaque arête est orientée.
- $\delta'_k(e)$: $|N_k(e)|$

7.5.2.1 Partage pessimiste des ressources radio

Nous réutilisons le modèle pessimiste de partage des ressources radio. Cependant, au lieu de partager la bande passante entre les nœuds en conflit, nous nous focalisons sur la distribution de la bande passante entre *liens radio* en interférence potentielle. Nous souhaitons continuer à établir des contraintes locales. Ainsi, la bande passante radio est partagée entre un lien radio e et tous les autres liens radio en possible interférence. Par ailleurs, nous utilisons un modèle d'interférences de type émetteur / récepteur. Ainsi, l'ensemble des liens interférant avec e est l'ensemble des 2-voisins de e dans le linegraph de G. Il en existe exactement $\delta_2(e)$.

Nous pouvons remarquer que notre modèle alloue la capacité à un lien radio orienté, i.e. à un nœud particulier pour ses émissions vers un de ses voisins donné. Nous aurions pu également considérer une arête comme non orientée, et répartir la bande passante de manière équitable aux deux extrémités. La bande passante allouée aurait été au final identique. Nous avons donc choisi d'utiliser des liens radio orientés pour plus de clarté.

La même capacité est allouée à chaque arête. De plus, le trafic de contrôle doit également être pris en compte. Ainsi, nous obtenons la contrainte suivante :

$$\forall e \in E, \quad \forall f \in N_2(e), \qquad T(f) \leq \frac{BW - \sum_{(u,x)\in n_2(e)} T_c(u)}{\delta_2(e)} \tag{7.21}$$

Nous obtenons donc pour le partage pessimiste des ressources radio avec une équité entre liens le programme linéaire LP 4 page suivante. Il est important de noter que les capacités obtenues avec des modèles d'équité orientés nœuds et liens radio ne sont pas comparables : un modèle n'est pas *meilleur* qu'un autre, il modélise seulement des comportements différents de la couche MAC. Cependant, nous pouvons éventuellement conclure au vu des résultats quantitatifs qu'un modèle d'équité permet d'atteindre un débit maximal plus important.

[1] pour rappel, $\mathcal{LG}$ est le linegraph de G

Programme Linéaire 4 (Partage pessimiste équitable entre liens radio)

Maximiser	*fonction objective sur* $\mathcal{P}$
	Soumise à :
Ensemble d'équations du type 7.21 page précédente	$\forall$ *lien radio* $e \in E$
Répartition du flux envoyé sur p	$\forall$ *route* p

7.5.2.2 Partage optimiste des ressources radio

Nous avons également choisi de réutiliser le modèle de partage optimiste des ressources radio. Afin d'introduire une équité différente, il est seulement nécessaire de changer le mécanisme d'attribution de la bande passante par le médium radio centralisé.

Ainsi, seul l'algorithme statistique calculant les fréquences associées à chaque lien (u, v) doit être modifié de la manière suivante :

- Soit une arête e et l'ensemble des arêtes voisines de e dans le graphe des conflits radio, i.e. les 2-voisines dans le linegraph de G
- Marquer toutes ces arêtes comme non bloquées
- Tant qu'il existe une arête non bloquée, en choisir aléatoirement une, notée f
- Marquer toutes les arêtes voisines de f dans le graphe des conflits comme bloquées.

Ainsi, nous répétons cet algorithme n fois (n étant donc la précision de l'algorithme). Au final, le médium radio centralisé allouera à cette arête une bande passante proportionnelle à sa fréquence de sélection. Nous pouvons noter que les arêtes sont ici aussi considérées comme orientées. Ainsi, une arête (u, v) recevra éventuellement une capacité différente de l'arête (v, u).

Nous pouvons remarquer que des liens radio ayant plus de voisins interférants recevront une capacité moindre. Cependant, notre algorithme permet de considérer une probabilité uniforme d'accès au médium de toutes les arêtes. Ainsi, les arêtes interférant peu auront une probabilité plus élevée d'être activées simultanément avec d'autres arêtes. De manière figurée, *consommant* peu du médium, elles prendront de la bande passante restante, pouvant être considérée comme *gratuite*.

Le reste du modèle reste inchangé. La formulation LP 3 page précédente reste en particulier analogue.

7.5.3 Remarques sur le modèle d'interférences utilisé

Comme nous le faisions remarquer, le modèle d'interférences influe peu sur nos modèles de partage de la ressource radio. Ainsi, les quelques points à modifier seraient donc :

- Dans le modèle du partage pessimiste avec une équité orientée nœuds, il serait nécessaire de partager la bande passante équitablement entre un nœud V et l'ensemble de ses voisins qui ont au moins un lien radio voisin dans le graphe des conflits d'un lien sortant de V.
- Dans le modèle du partage optimiste, il serait nécessaire de construire l'ensemble des $freq(u, v)$ pour l'ensemble des arêtes voisines du sommet ou du nœud considéré dans le graphe des conflits radio

Nous pouvons donc remarquer que de tels changements sont mineurs, nos modèles présentant une flexibilité élevée.

7.6 Quelle définition de la capacité ?

Nous avons donné au sein de la section précédente une série de contraintes locales définissant le partage des ressources radio. Nous devons donc maintenant définir une fonction objective linéaire traduisant le problème de la capacité, i.e. une définition formelle de la capacité. Nous avons choisi d'introduire deux définitions de la capacité, remplissant selon nous deux objectifs différents.

7.6.1 Max-Sum

Si nous définissons comme dans beaucoup d'articles la capacité comme le débit maximal transportable par le réseau, nous pouvons introduire la définition formelle suivante :

$$Max\left(\sum_{p\in\mathcal{P}} f(p)\right) \tag{7.22}$$

Ainsi, dans une telle approche, nous maximisons la quantité globale de trafic acheminée dans le réseau, créant donc un objectif global et non individuel. Ainsi, il est fort probable que certains flux présentent une bande passante allouée nulle car ils entrent en interférence avec trop d'autres flux. Les flux multisauts recevront notamment peu de bande passante : chaque relais engendre des contraintes supplémentaires en terme d'interférences, baissant la capacité allouée aux autres flux. Nous obtiendrons donc vraisemblablement un ensemble de flux activés simple-saut formant un ensemble indépendant dans le réseau. Les flux possédant le moins d'interférents seront sélectionnés. Cette définition nous semble donc en désaccord avec la définition usuelle des réseaux ad-hoc. Cependant, *max-sum* nous permet de donner un aperçu du débit maximum atteignable dans le réseau.

7.6.2 Max-Min

Il est communément acquis qu'un réseau ad-hoc doit présenter une certaine équité. C'est cette propriété que nous avons introduite dans nos modèles de partage de la ressource radio. Nous l'introduisons donc également dans notre définition de la capacité, la fonction objectif devenant ainsi :

$$Max\left(Min_{p\in\mathcal{P}} f(p)\right) \tag{7.23}$$

Dans un tel schéma, nous garantissons une certaine quantité de bande passante à chaque flux possible du réseau. Aucun flux n'est désavantagé car possédant trop d'interférents ou car étant multisauts. Le débit maximum agrégé sera donc inférieur à celui d'un objectif tel que le *max-sum*. Mais, nous trouvons une telle définition de la capacité plus représentative. Par extension, nous pourrions introduire un ratio d'équité tel que décrit dans [12]. Cependant, nous avons choisi de ne présenter que des résultats basés sur une équité *dure* car ils nous semblent représentatifs.

7.7 Application de cette étude à un réseau en ligne

Nous proposons dans cette section d'étudier le cas de la ligne afin d'illustrer nos modèles. La capacité est évaluée via les deux fonctions d'objectif comme définies précédemment. Afin de simplifier les explications, le trafic de contrôle est considéré nul. De la même façon, la capacité est égale à une unité. Dans le cas du *max-min*, x représente la bande passante allouée à chaque flux.

Soit la topologie telle qu'illustrée sur la figure 7.5 page suivante. Le point d'accès est placé sur l'extrémité gauche, et nous étudions l'ensemble des flux menant vers ce point d'accès. La ligne contient exactement n nœuds plus l'AP.

7.7.1 Équité orientée nœuds

7.7.1.1 Modèle pessimiste

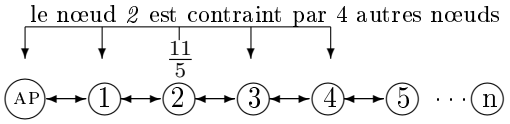


FIG. 7.5 – Le cas de la ligne (partage pessimiste de ressources)

Le nœud 1 constituera le goulot d'étranglement. Par ailleurs, puisque le nœud 2 possède quatre 2-voisins et que le nœud 1 est un 2-voisin, la capacité $\frac{1}{4+1}$ est allouée au nœud 1, qui doit la répartir entre les liens (AP, 1) et (1, 2). Ainsi, 1 reçoit $\frac{1}{5}$ de la capacité du médium, et plus particulièrement $\frac{1}{10}$ pour son arête (1,AP).

- *max-sum* : 1 doit relayer tout son trafic vers l'AP. Finalement, $max\text{-}sum = \frac{1}{10}$.
- *max-min* : 1 doit recevoir et relayer le trafic de $(n-1)X$ autres nœuds, et envoyer son propre trafic vers l'AP. Finalement, la capacité allouée au nœud 1 est utilisée pour le trafic d'exactement n noeuds : $\frac{1}{10} \leq (n-1)x + x$. Ainsi, $max\text{-}min = \frac{1}{10n}$

7.7.1.2 Modèle optimiste

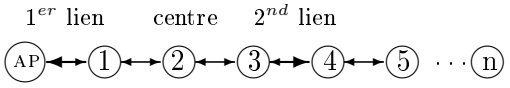


FIG. 7.6 – Le cas de la ligne (partage optimiste de ressources) : exemple d'un ensemble indépendant de 2 liens

- *max-sum* : le nœud 1 envoie le trafic de n nœuds vers l'AP, représentant ainsi le goulot d'étranglement. Si nous considérons le 2-voisinage centré sur le nœud 2, il existe les ensembles indépendants suivant :
  - (AP, 1) & (3, 4)
  - (AP, 1) & (4, 3)
  - (1, AP) & (3, 4)
  - (1, AP) & (4, 3)

  Ainsi, $freq$(1,AP) $= \frac{1}{2}$. Si le nœud 2 envoie du trafic, il devra être relayé par 1, créant ainsi plus de contraintes. Le nœud 1 va donc être naturellement le seul émetteur du réseau car il se retrouve moins contraint. Ainsi, Le nœud 1 peut obtenir la moitié de la bande passante, conduisant finalement à $max\text{-}sum = \frac{1}{2}$
- *max-min* : le nœud 1 constitue toujours le goulot d'étranglement, envoyant x trafic et dévant en relayer $x(n-1)$. L'ensemble de nœuds le plus contraint est le 2-voisinage de 2. Nous avons donc :
  - Le trafic $x(n)$ doit être envoyé via le lien (AP, 1).
  - Le trafic $x(n-3)$ doit être envoyé via le lien $(4, 3)$.
  - Le nœud 2 doit en plus envoyer le trafic $x(n-1)$.

  Les fréquences calculées pour l'objectif *max-sum* restent inchangées. Ainsi, nous obtenons les contraintes suivantes :

$$\begin{aligned} x(n) &\leq [1 - x(n-1)] \cdot \frac{1}{2} \\ x(n-3) &\leq [1 - x(n-1)] \cdot \frac{1}{2} \end{aligned}$$

Finalement, $max\text{-}min = \frac{1}{3n-1}$

7.7.2 Équité orientée liens radio

7.7.2.1 Modèle pessimiste

Le nœud 1 continue à constituer le goulot d'étranglement. Le 2-voisinage du lien $(2,3)$ constitue l'ensemble le plus contraint. La capacité $\frac{1}{10}$ est allouée à chaque lien de cet ensemble puisque le lien $(2,3)$ possède également 10 voisins dans le graphe de contention.

- $max\text{-}sum$: le nœud 1 envoie tout son trafic vers l'AP via le lien $(\text{AP}, 1)$. Ainsi, $max\text{-}sum = \frac{1}{10}$
- $max\text{-}min$: le nœud 1 doit envoyer son trafic et relayer celui des $n-1$ autres nœuds de la ligne. En conséquence, $max\text{-}min = \frac{1}{10n}$.

Nous pouvons remarquer que dans le cas de la ligne, $max\text{-}min$ reste inchangé quelle que soit l'équité. Cependant, un tel cas de figure est rare dans une topologie plus commune.

7.7.2.2 Modèle optimiste

Les fréquences restent dans le cas de la ligne inchangées. Ainsi, les capacité avec une équité orientée nœuds ou liens radio sont identiques.

7.8 Résultats quantitatifs

7.8.1 Démarche adoptée

Nous souhaitons estimer quantitativement la capacité du réseau associée à une certaine topologie et un certain protocole de routage. Nous souhaitons en particulier comparer quantitativement l'impact de l'utilisation d'une topologie virtuelle qui n'utilise qu'une sous-partie des liens radio possibles.

Nous avons donc simulé plusieurs protocoles de routage dans OPNET [16]. Plus précisément, nous avons créé une topologie dans laquelle nous avons placé x nœuds répartis aléatoirement sur une surface carrée de telle sorte que le degré[1] moyen du réseau soit de 10. Ensuite, les protocoles de routage étudiés ont été exécutés in vitro afin d'en extraire les paramètres qui nous intéressent. Nous avons choisi de simuler 2 types de réseau :

1. un réseau ad-hoc dans lequel le pattern de trafic est quelconque : tout nœud possédant une route vers tout autre nœud du réseau. Il existe donc $n(n-1)$ routes.
2. un réseau hybride dans lequel un nœud ne communique qu'avec le point d'accès. Il existe donc exactement $n-1$ routes. Plus précisément, puisque nous utilisons un trafic de type requête/réponse, $2(n-1)$ routes sont utilisées (de et vers le point d'accès).

Nous avons choisi d'utiliser 3 protocoles différents afin d'étudier l'impact de plusieurs structures :

- OLSR : ce protocole proactif (section 5.3.1.1 page 83) permet de donner une représentation du comportement d'un protocole de routage dit *à plat*. Plus précisément, les routes utilisées sont les plus courtes en nombre de sauts. Ainsi, nous aurions pu simuler un protocole tel qu'AODV, seule la quantité de trafic de contrôle ayant alors changé.
- WU & Li : seule la topologie en backbone est utilisée, les liens radio entre dominés étant ignorés (cf. chapitre 2.5 page 15 pour plus de détails sur une telle topologie). Les plus courtes routes sont alors calculées uniquement sur la topologie du backbone. Il est important de noter que même si un sous-ensemble des liens radio est utilisé, même ceux non utilisés continuent à créer des interférences, pouvant potentiellement réduire la capacité. De plus, la route via le backbone pouvant être plus longue, un flux créera en moyenne plus d'interférences. Nous avons utilisé une version étendue de l'algorithme, décrite dans [3]

[1] nombre de voisins radio d'un nœud

- VSR & SOMoM : en mode ad-hoc, les routes sont constituées de routes de clusters, potentiellement plus longues. En mode hybride, les routes passent uniquement via le backbone, présentant donc potentiellement les mêmes problèmes que Wu & Li.

Nous adoptons la démarche suivante :

1. Une topologie de x nœuds est générée ($x \in [10..60]$).
2. Un protocole de routage est exécuté au sein de la simulation, dont nous extrayons les routes et le trafic de contrôle par nœud
3. Les contraintes de partage de la ressource radio sont extraites de la topologie. Les fréquences du partage optimiste sont notamment calculées dans cette phase
4. Les contraintes des flux sont reportées individuellement sur les liens radio correspondant
5. Le solveur CPLEX [8] nous permet de donner la capacité en fonction de la fonction d'objectif et des contraintes formulées précédemment.

Afin de rendre le calcul des fréquences plus efficace, l'algorithme modélisant un médium radio centralisé se base sur un nombre dynamique de tirages. Nous stoppons l'algorithme lorsqu'il a effectué au minimum *min* tirages, et que la différence des fréquences calculées avant et après un tirage est inférieure à un seuil x. Nous avons vérifié qu'une telle démarche nous permet d'obtenir des résultats aussi précis qu'un nombre élevé mais invariant de tirages, tout en présentant un délai de calcul plus faible.

7.8.2 Résultats

Dans cette section, nous proposons d'étudier la capacité de protocoles de routage utilisant le réseau à plat ou au contraire une structure virtuelle. Nous supposons que la bande passante radio est normalisée à 1. Dans un premier temps, nous exposons des remarques générales sur l'évolution de la capacité avec un nombre croissant de nœuds dans le cas de l'objectif *max-min*. Ensuite, nous proposons de comparer les capacités des différents protocoles de routage.

7.8.2.1 Évolution générale de la capacité

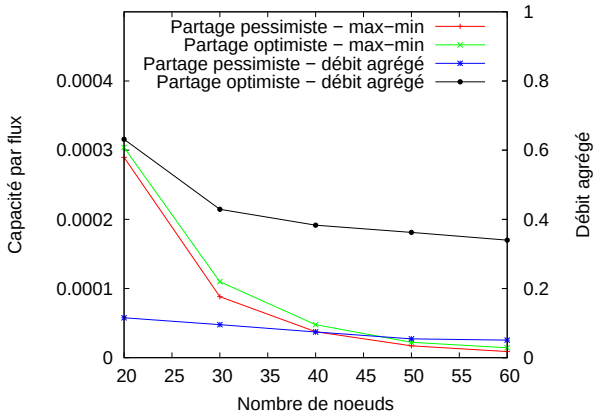


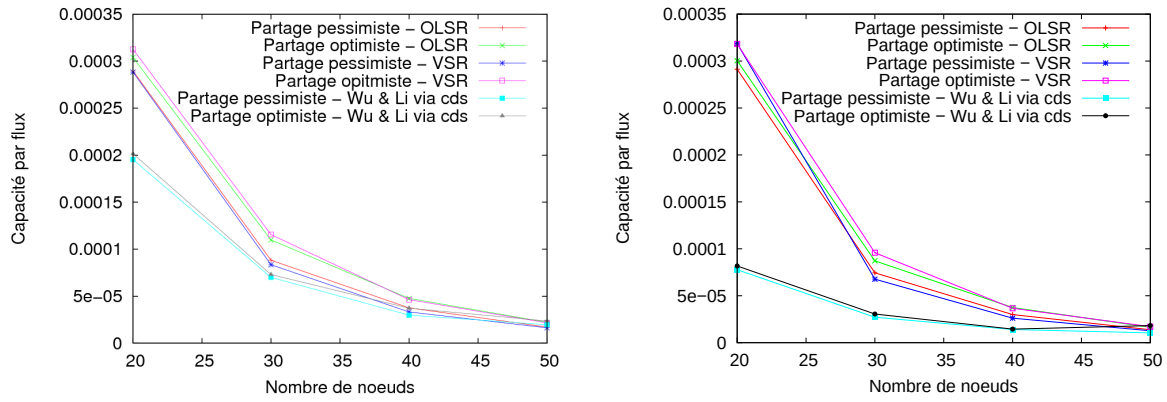
FIG. 7.7 – Capacité d'un réseau ad-hoc avec OLSR, objectif max-min (équité orientée lien radio)

Dans un premier temps, nous évaluons la capacité d'un MANET avec une équité orientée lien radio (fig. 7.7). Avec un protocole de routage à plat, nous pouvons observer que la capacité par flux diminue lorsque le nombre de nœuds augmente. En effet, un réseau ad-hoc comprenant, comme expliqué précédemment, $n(n-1)$ routes, le nombre de flux possibles augmente, créant ainsi potentiellement plus d'interférences. Ainsi, la bande passante allouée par flux décroît, corroborant les résultats de [7]. Par contre, à l'opposé, **le débit agrégé sur tous les flux reste**

constant. Il n'augmente pas car de nombreux flux passent par le centre du réseau, dans lequel beaucoup d'interférences sont créées. La réutilisation spatiale du médium radio est donc limitée. Nous verrons que ce n'est pas le cas avec une capacité de type *max-sum*. Finalement, nous pouvons remarquer que **les partages optimiste et pessimiste des ressources radio présentent des résultats très proches, permettant d'encadrer finement la capacité** possible d'un protocole.

7.8.2.2 Mode ad-hoc

Nous évaluons ensuite la capacité d'un réseau ad-hoc suivant différents protocoles de routage. Dans un premier temps, nous utilisons la fonction d'objectif *max-min* avec une équité orientée lien radio (fig. 7.8(a)). Nous pouvons remarquer que la capacité décroît avec plus de nœuds pour les mêmes raisons que précédemment. **Un protocole de routage à plat présente la capacité la plus élevée** : les routes les plus courtes présentent par définition une longueur de route minimale, limitant les interférences au niveau des nœuds relais. Le protocole VSR qui utilise le backbone seulement pour les inondations permet de diminuer le trafic de contrôle. De plus, VSR utilise la topologie en clusters pour calculer ses plus courtes routes, mais n'augmente que d'un très faible pourcentage la longueur moyenne des routes. Ainsi, **la capacité offerte par VSR est très proche de celle d'un protocole à plat**. Par contre, Wu & Li calcule ses routes les plus courtes seulement via le backbone, tendant à en augmenter significativement la longueur. De plus, seuls les nœuds du backbone relaient les paquets, pouvant créer potentiellement un goulot d'étranglement dans le réseau. Ainsi, Wu & Li présente une capacité minimale.



(a) équité orientée lien radio

(b) équité orientée nœuds

FIG. 7.8 – Capacité d'un réseau ad-hoc avec l'objectif max-min

Dans un deuxième temps, nous étudions la même capacité avec une équité orientée nœuds afin de comparer l'impact d'une équité différente (fig. 7.8(b)). **Les capacités de VSR et OLSR restent très proches**, peu impactées par le type d'équité. Apparemment, les nœuds possédant un grand nombre de voisins ne semblent pas concentrer les fonctions de relais de paquets de données. une équité entre liens radio n'apporte donc aucune différence. Par contre, la capacité découlant du protocole de Wu & Li diminue significativement. En effet, dans une équité orientée lien, la bande passante est distribuée équitablement entre tous les liens radio pouvant supporter un flux. Or, les dominés dans le protocole de Wu & Li ne possèdent qu'un seul lien radio actif : celui vers leur dominant. Ainsi, ils ont quantitativement besoin d'une bande passante plus faible que les nœuds du backbone qui possèdent plus de liens actifs. Cependant, la bande passante étant distribuée uniformément entre les nœuds, les nœuds non membres du backbone reçoivent une bande passante disproportionnée par rapport à leurs besoins, *gaspillant* des ressources radio.

Finalement, nous étudions la capacité en terme de *max-sum* (fig. 7.9 page suivante). Avec une telle maximisation, **les routes les plus courtes, voire simple saut, seront avantagées**

puisqu'elles créent moins d'interférences radio. Ainsi, la capacité globale ne diminue pas lorsque le nombre de nœuds augmente. Nous pouvons même remarquer que **la capacité augmente avec un partage optimiste de ressources** : le degré étant constant, le diamètre du réseau augmente, permettant de maximiser la réutilisation spatiale du spectre radio. Au contraire, le modèle de partage pessimiste a tendance à surestimer les interférences, limitant la réutilisation dans les réseaux de petite taille. Par ailleurs, nous pouvons remarquer qu'OLSR et VSR présentent une capacité très proche, quelle que soit la fonction d'objectif. La capacité de Wu & Li reste bien en deçà. De plus, la capacité avec un partage optimiste n'augmente pas aussi vite qu'avec OLSR ou VSR. En conclusion, une auto-organisation à travers une structure virtuelle n'impacte pas de façon drastique la capacité, OLSR et VSR offrant une capacité très proche. Bien que des routes soient quelquefois plus longues, que certains nœuds soient plus sollicités, le réseau ne présente pas de goulot d'étranglement.

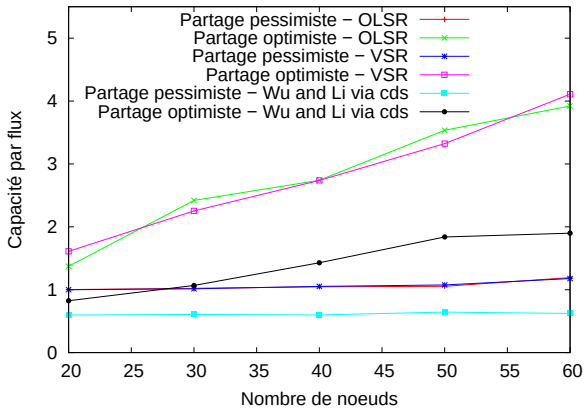


FIG. 7.9 – Capacité d'un réseau ad-hoc avec l'objectif max-sum (équité orientée lien radio)

7.8.2.3 Mode hybride

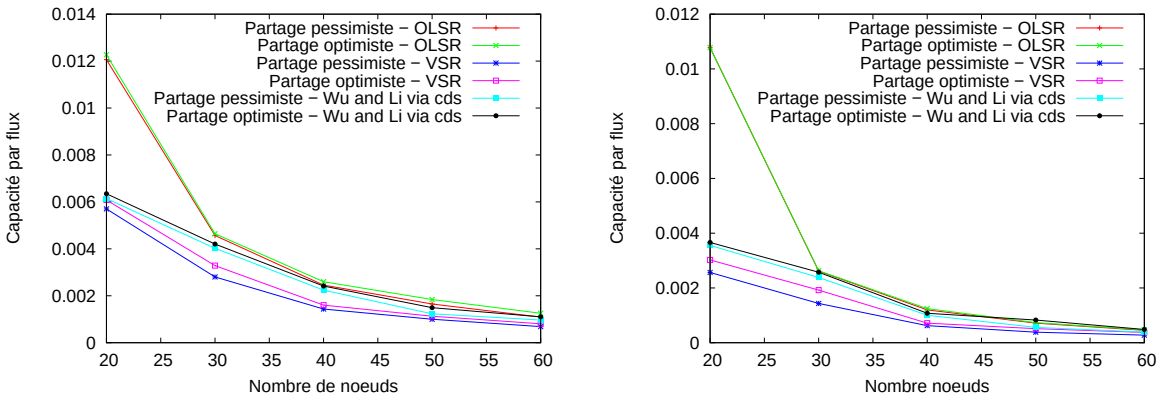
Dans un deuxième temps, nous étudions la capacité avec l'objectif *max-min* et une équité orientée lien radio dans un réseau hybride (fig. 7.10(a) page suivante). Dans un tel réseau, le point d'accès constitue l'unique destination. Ainsi, la capacité par flux est bien supérieure au mode ad-hoc. OLSR présente une capacité très supérieure à celle offerte par SOMoM et Wu & Li. En effet, dans un réseau hybride, un goulot d'étranglement apparaîtra au niveau du point d'accès. Cependant, un algorithme de routage à plat permet de distribuer efficacement la charge entre toutes les voisins du points d'accès. Par contre, **les solutions reposant sur un backbone semblent moins efficaces** dans une telle répartition : certains liens radio se retrouvent trop chargés. Nous pouvons remarquer que Wu & Li présente une capacité supérieure à celle offerte par SOMoM : le premier backbone semble plus efficace pour la distribution équilibrée de la charge.

De plus, nous pouvons remarquer que le modèle d'équité ne présente aucun impact dans un réseau hybride : la capacité reste inchangée ((fig. 7.10(b) page ci-contre).

Dans un réseau hybride, un protocole basé sur une structure virtuelle semble offrir un débit dégradé en comparaison d'une approche à plat. En conséquence, **un backbone efficace et plus équilibré autour de l'AP reste à proposer**.

7.9 Conclusion

Nous avons présenté dans ce chapitre une solution permettant de quantifier l'impact en terme de capacité d'une solution de routage sur structure d'auto-organisation. La force de notre



(a) équité orientée liens

(b) équité orientée nœuds

FIG. 7.10 – Capacité d'un réseau hybride avec l'objectif max-min

proposition réside sa généricité : elle permet d'évaluer la capacité de toute solution d'auto-organisation mais également d'une topologie ou d'un protocole de routage quelconques. Nous avons finement analysé et modélisé les interactions radio et le partage de bande passante comme dans IEEE 802.11. Une telle approche possède donc des atouts surpassant largement le cadre de l'étude des structures d'auto-organisation.

Nous avons modélisé ce partage de la bande passante en introduisant deux types d'équité au niveau de l'accès au médium. La première répartit équitablement la bande passante entre les nœuds en compétition pour accéder au médium, tandis que la deuxième va au contraire partager la bande passante entre les liens radio en interférences, permettant ainsi de limiter potentiellement l'apparition de goulots d'étranglement au niveau des nœuds à fort degré. Pour chaque équité, nous avons détaillé deux scénarios de partage des ressources : un modèle pessimiste surestime les interférences, tandis qu'un modèle optimiste sous-estime les possibles incompatibilités d'ordonnancements définis seulement localement. Ces partages de ressources et équités sont traduits sous la forme d'ensemble d'équations linéaires, prenant en entrée les routes calculées par les protocoles de routage.

Nous avons évalué quantitativement la capacité offerte par trois protocoles de routage que sont OLSR, Wu & Li et VSR/SOMoM dans des réseaux ad-hoc mais également hybrides. Nous avons pu vérifier que l'utilisation d'une structure virtuelle peut avoir un faible impact sur la capacité, OLSR et VSR présentant des débits atteignables très similaires. Si au contraire la structure virtuelle est mal exploitée, telle que dans Wu & Li, la capacité peut par contre se trouver réduite. Dans un réseau hybride, nous avons pu vérifier que les structures virtuelles actuelles ne permettent pas de répartir équitablement la charge entre les voisins du point d'accès, créant ainsi un goulot d'étranglement. Conséquemment, la capacité offerte par OLSR est très supérieure à celle offerte par Wu & Li ou SOMoM. La création d'un backbone possédant un moindre impact au niveau de la capacité doit donc être proposé. En particulier, la charge doit être répartie sur plus de liens radio.

Il est connu que les protocoles réactifs semblent adapter les routes découvertes à la charge du réseau, et donc répartir plus uniformément le trafic dans le réseau. Il serait intéressant d'adapter notre framework afin d'utiliser des routes calculées par des protocoles réactifs en forte charge et vérifier un tel constat. De la même façon, il pourrait être intéressant d'étudier l'impact sur la capacité de l'utilisation simultanée de plusieurs routes pour un même couple source/destination. Enfin, une méthode de partage des ressources moins idéale pourrait être proposée, plus proche des mécanismes régissant IEEE 802.11. Nous pourrions ainsi étudier l'impact des iniquités de IEEE 802.11 sur la capacité globale.

Nous avons proposé un protocole d'auto-organisation, étudié ses propriétés via une étude analytique et par simulations. Nous avons également proposé des applications de cette auto-

organisation. Ce chapitre valide l'impact limité d'une telle solution sur la capacité. Cependant, il reste à nos yeux à valider dans des conditions *réelles* nos protocoles. Les simulations et les études analytiques ne peuvent que simplifier l'environnement radio. Nous proposons donc dans le chapitre suivant d'étudier notre structure d'auto-organisation dans une plate-forme d'expérimentations en environnement réel.

Bibliographie

[1] 802.11g, IEEE standard for 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 : Further higher data rate extension in the 2.4 ghz band, 2003.

[2] N. Bansal and Z. Liu. Capacity, delay and mobility in wireless ad-hoc networks. In *INFOCOM*, San Francisco, USA, April 2003. IEEE.

[3] J. Carle and D. Simplot-Ryl. Energy efficient area monitoring by sensor networks. *IEEE Computer Magazine*, 37(2) :40–46, February 2004.

[4] D. S. J. De Couto, D. Aguayo, B. A. Chambers, and R. Morris. Performance of multihop wireless networks : shortest path is not enough. *ACM SIGCOMM Computer Communication Review*, 33(1) :83–88, january 2003.

[5] D. Dhoutaut. *Etude du standard IEEE 802.11 dans le cadre des réseaux ad hoc : de la simulation à l'expérimentation.* PhD thesis, INSA Lyon, December 2003.

[6] M. Grossglauser and D. Tse. Mobility increases the capacity of ad-hoc wireless networks. In *INFOCOM*, Anchorage, Alaska, USA, April 2001. IEEE.

[7] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2) :388–404, March 2000.

[8] ILOG CPLEX. http://www.ilog.com/products/cplex/index.cfm (v7.5).

[9] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. In *International Conference on Mobile Computing and Networking (MOBICOM)*, pages 66–80, San Diego, USA, September 2003. ACM.

[10] U. C. Kozat and L. Tassiulas. Throughput capacity of random ad hoc networks with infrastructure support. In *International Conference on Mobile Computing and Networking (MOBICOM)*, pages 55–65, San Diego, USA, September 2003. ACM.

[11] W. Krause, R. Sollacher, and M. Greiner. Self-* topology control in wireless multihop ad hoc communication networks. In LNCS, editor, *SELF-STAR*, volume 3460, pages 49–62, Bertinoro, Italy, May 2004.

[12] A. Kumar, M. Marathe, S. Parthasarathy, and A. Srinivasan. Algorithmic aspects of capacity in wireless networks. *ACM SIGMETRICS Performance Evaluation Review*, 33(1) :133–144, June 2005.

[13] J. Li, C. Blake, D. S. J. de Decouto, H. I. Lee, and R. Morris. Capacity of ad hoc wireless networks. In *International Conference on Mobile Computing and Networking (MOBICOM)*, Roma, Italy, July 2001. ACM.

[14] B. Liu, Z. Liu, and D. Towsley. On the capacity of hybrid wireless networks. In *INFOCOM*, volume 3, pages 1543–1552, San Francisco, USA, April 2003. IEEE.

[15] lpSolve. http://sourceforge.net/projects/lpsolve/.

[16] OPNET Modeler. http://www.opnet.com (v8.1a).

[17] L. Qiu, R. Chandra, K. Jain, and M. Mahdian. Optimizing the placement of integration points in multi-hop wireless networks. In *International Conference on Network Protocols (ICNP)*, Berlin, Germany, October 2004. IEEE.

[18] H. Rivano. *Algorithmique et télécommunications : Coloration et multiflot approchés et applications aux réseaux d'infrastructure.* PhD thesis, Université de Nice-Sophia Antipolis (UNSA), November 2003.

[19] A. Zemlianov and G. De Veciana. Capacity of ad hoc wireless networks with infrastructure support. *IEEE Journal on Selected Areas in Comunications*, 23(3) :657–667, March 2005.

Publications

Conférence internationale

[1] Herve Rivano, Fabrice Theoleyre, and Fabrice Valois. Capacity Evaluation Framework and Validation of Self-Organized Routing Schemes. In *International Workshop on Wireless Ad Hoc & Sensor Networks (IWWAN)*, New York, USA, Juin 2006.

Conférence francophone

[2] Herve Rivano, Fabrice Theoleyre, and Fabrice Valois. Influence de l'auto-organisation sur la capacité des réseaux ad hoc. In *Rencontres Francophones sur les aspects Algorithmiques des Télécommunications (ALGOTEL)*, Hyeres, France, Mai 2005.

Chapitre 8

De la conception d'un protocole à son expérimentation

8.1 Introduction

Les réseaux ad-hoc représentent un vaste champs d'étude, exploré depuis déjà quelques années. Ainsi, de nombreuses propositions ont déjà été réalisées, présentant des protocoles pouvant remplir dans un réseau à plat les fonctions de routage, de localisation, d'adressage... Cependant, les réseaux ad-hoc restent encore un problème ouvert : aucun protocole ne propose des performances optimales tenant compte de toutes les contraintes inhérentes aux réseaux ad-hoc. Ainsi, l'utilisation de terminaux embarqués et l'emploi de liaisons radio pour communiquer continuent à constituer des contraintes majeures. C'est pourquoi les chercheurs comparent leur proposition avec les protocoles déjà existant dans le domaine. La première méthode de comparaison, la plus simple à mettre en œuvre se base sur la simulation [11] : elle permet de donner une évaluation quantitative des performances lorsque l'environnement étudié est trop complexe pour en tirer des performances de façon analytique. Or, nous verrons que les simulateurs s'appuyant sur une modélisation de l'environnement, n'en offrent qu'une vision partielle, pouvant fausser potentiellement une évaluation de performances. Deux algorithmes peuvent également être comparés par le biais d'une étude analytique de leur comportement. Nous en verrons également les limites.

Pour répondre à de telles carences, de nombreux chercheurs se sont astreints à déployer expérimentalement des réseaux ad-hoc afin de mesurer les performances de leurs protocoles in vivo. Un environnement radio réel permet d'éprouver un protocole dans des conditions réalistes. Nous avons donc choisi d'étendre l'évaluation de performances du protocole de localisation que nous avions présenté dans le chapitre 6 page 107 en déployant une plate-forme de test. Cette plate-forme servira de preuve de faisabilité et de fonctionnement d'une telle solution, créant un véritable réseau sans-fil multisauts. De plus, des performances en environnement réaliste pourront corroborer les résultats encourageants des simulations. Nous allons donc dans un premier temps expliquer plus en profondeur l'intérêt de la simulation et des études analytiques, mais également leurs limites. Ensuite, nous présenterons quelques testbeds déployés actuellement et décrits dans la littérature. Nous décrirons également l'architecture logicielle et matérielle de la plate-forme expérimentale que nous avons utilisée. Finalement, nous donnerons une évaluation de performances du protocole de localisation SOMoM dans un environnement radio réel. Puis nous exposerons quelques perspectives dans ce domaine de recherche.

8.2 De la complémentarité des simulations, des études analytiques et des expérimentations en environnement réel

La grande majorité des protocoles conçus pour les réseaux ad-hoc ont été évalués par le biais de simulations. Un simulateur permet après une modélisation du sujet étudié de reproduire un phénomène in vitro. Ainsi, un simulateur pour réseaux ad-hoc offre une modélisation logicielle du médium radio, de la mobilité, des transmissions de paquets... Les simulations sont intensivement utilisées pour les nombreuses qualités qu'elles présentent :

- Reproductibilité : la modélisation de l'environnement est fixe et déterministe. Bien souvent, le caractère variable du comportement ou du médium radio est modélisé via un générateur pseudo aléatoire. Cependant, une même graine donnera de façon déterministe les mêmes valeurs dans la suite de nombres générée par un générateur pseudo aléatoire. Ainsi, les résultats sont reproductibles de façon parfaite par tout un chacun : un chercheur peut utiliser des simulations existantes, et rejouant les scénarios dans son laboratoire, obtiendra exactement les mêmes valeurs. De la même manière, si un protocole exhibe un comportement étrange, il suffit de rejouer le scénario pour que le phénomène particulier se reproduise au même instant de la simulation.
- Ré-utilisation : le code de simulation peut être partagé par la communauté, et réutilisé facilement. Le déploiement sur des systèmes embarqués hétérogènes présente plus de difficultés
- Facilité d'implémentation : un simulateur offre souvent une facilité accrue d'implémentation, grâce à un haut niveau d'abstraction, des primitives conçues pour la simulation. L'implémentation directe sur un terminal embarqué demande des adaptations pour chaque type de terminal, de système d'exploitation, et requiert une connaissance approfondie du système hôte. De plus, un simulateur implémente de façon centralisée l'environnement distribué entier à tester. Ainsi, la collecte de statistiques par exemple devient triviale
- Flexibilité du code : la couche protocolaire classique est modifiable à l'envi. Ainsi, un concepteur peut collecter des statistiques au niveau de la couche MAC et les faire partager aux couches supérieures. Une telle fonctionnalité n'est bien souvent pas offerte par de nombreux systèmes
- Passage à l'échelle : un simulateur permet souvent de modéliser le comportement de dizaines à quelques milliers de nœuds différents.
- Coût : de nombreux simulateurs sont open-source. De plus, il est seulement nécessaire d'investir dans une machine de calcul, même peu puissante. Un testbed requiert l'achat d'un grand nombre de terminaux embarqués.

A la vue de telles remarques, nous pourrions conclure de façon abrupte que les simulations constituent la panacée de l'évaluation de performances. Cependant, tel n'est pas le cas : un simulateur se base comme nous le disions sur une modélisation. Or, une modélisation passe obligatoirement par une formalisation de règles régissant le réel. L'établissement de ces règles est d'une telle complexité que bien souvent, la modélisation passe obligatoirement par une simplification. Cependant, quels sont les paramètres de l'environnement que nous sommes en droit de simplifier ? Ainsi, la modélisation du médium radio doit tenir compte de l'affaiblissement, des évanouissement, des réflexions, de l'hétérogénéité de l'environnement... Or, les performances peuvent varier de façon appréciable entre un environnement simulé et un environnent réel.

Nous pouvons citer le cas emblématique de la modélisation de la mobilité. De nombreux modèles ont été proposés afin de simuler des mobilités individuelles ou de groupe pour les nœuds d'un réseau ad-hoc (cf. [4] pour un aperçu). Cependant, deux conclusions peuvent selon nous être tirées des travaux existant dans la littérature :

- Les performances d'un protocole peuvent varier significativement selon le modèle de mobilité utilisé.
- Nous ne savons toujours pas quel modèle de mobilité est plus représentatif.

Ainsi, des évaluations de performances en simulations et en environnement réel ne peuvent être considérées sur le même plan. Nous pensons qu'un protocole doit auparavant être évalué par le biais de simulations pour des raisons de coût en ressource humaines et temps. Cependant, une expérimentation réelle doit accompagner une telle étude avant de penser à une utilisation réelle.

Une étude analytique permet souvent d'observer le comportement asymptotique d'un algorithme, et prouver de façon formelle sa convergence vers un état valide. Une telle analyse correspond donc à un premier pas dans la validation. Cependant, là encore, une analyse théorique requiert une modélisation. De plus, la classe des problèmes solubles analytiquement étant très restreinte, cette modélisation doit être très simple afin d'être exploitable. En conséquence, les études analytiques simplifient encore plus l'environnement que les simulations. Par exemple, il est extrêmement difficile, et dans bien des cas impossible, de se baser sur un modèle radio additif de type SNR. Une étude analytique ne permet donc de valider le comportement de l'algorithme que dans un environnement idéal. C'est la raison pour laquelle de nombreux articles corroborent leurs résultats analytiques par des simulations. Nous pensons donc qu'étude analytique, simulation et expérimentation réelle constituent un triplé de solutions complémentaires toutes requises pour la validation la plus fine possible d'un algorithme ou protocole.

8.3 Panorama des testbeds existants

[16] présente un travail pionnier dans la conception d'une plate-forme expérimentale adaptée aux réseaux ad-hoc. Les auteurs proposent d'évaluer expérimentalement les performances du protocole DSR. Le testbed comprend 2 nœuds fixes constituant les extrémités du réseau, et 5 nœuds mobiles se déplaçant entre. Un des nœuds statiques agit en tant que passerelle vers Internet et implémente les fonctionnalités de Foreign Agent de Mobile IP. Les véhicules se déplacent à la même vitesse, sur une trajectoire circulaire. Les équipements embarqués comprennent une antenne directionnelle à fort gain et une carte sans-fil IEEE 802.11. Chaque terminal comprend en outre un récepteur GPS afin de pouvoir tracer les mouvements et positions des nœuds mobiles. Les auteurs proposent d'évaluer le protocole de routage via la mesure du délai grâce aux *ping*, et les débits TCP grâce à l'outil Netperf [17]. Le protocole a été implémenté sur un système Linux, intégré comme un module noyau. Dans les résultats, les flux TCP permettent notamment de mettre en exergue les cassures de routes engendrées par la mobilité des nœuds.

Dans le même article, [16], les auteurs proposent d'implémenter un filtrage au niveau de la couche MAC afin de pouvoir contrôler statiquement les nœuds voisins. Si tous les nœuds sont à une distance radio de 1 saut, le filtrage permet d'implémenter facilement une topologie quelconque. Une telle fonctionnalité peut être utile durant l'implémentation : le concepteur peut tester une topologie particulière tout en maintenant les nœuds dans un espace limité. Naturellement, les performance réelles sont faussées, ainsi, un tel outil ne doit être utilisé que dans la phase de conception et de test, en aucun cas durant l'évaluation de performances. Suivant le même principe, [24] présente MobiEmu : n nœuds du réseau ad-hoc sont représentés par n PC. Par contre, les nœuds sont interconnectés via Ethernet, et un nœud central va donner à chacun des membres ses voisins *virtuels*. [12] propose également un mécanisme permettant de contrôler facilement la topologie créée. Les auteurs proposent de déployer des dispositifs matériels permettant de câbler les cartes réseaux et donc de contrôler la topologie via des atténuateurs, séparateurs et combineurs de signal... Cependant, une telle approche nous semble biaisée : un tel environnement ne reproduit en aucun cas le comportement d'ondes radio : les évanouissements ou réflexions sont même littéralement absents. Ainsi, nous ne pensons pas qu'un tel environnement offre une plus value par rapport à des simulations. [21] propose de réduire la portée radio en utilisant des câbles BNC pour antennes et en plaçant des atténuateurs pour réduire le signal et donc réduire la superficie du testbed. Bien que le médium radio soit utilisé pour la propagation, ces antennes et la réduction de portée peuvent présenter des modifications importantes dans les protocoles utilisant un tel testbed. [20] décrit le fonctionnement d'un testbed en grille, déployé

à l'université de Rutgers, pour tester le bon fonctionnement de protocoles pour réseaux sans-fil. Les liaisons radio sont *contrôlées* afin de proposer une reproductibilité : l'environnement ne permet donc que de valider un protocole, mais dans un environnement artificiel. Dans un futur proche, le projet Orbit Lab [19] fournira également un environnement radio réel pour tester finement les protocoles, utilisant de multiples technologies sans-fil.

[22] propose le déploiement d'un testbed pour tester le protocole de routage ABR. La plate-forme est constituée de quatre PC classiques placés en milieu urbain, formant une chaîne. Cependant, les auteurs ne détaillent pas l'architecture logicielle de leur protocole. Ils mesurent le débit atteint par un flux TCP et le délai pour différentes longueurs de routes et de tailles de paquets. [9] compare les performances de quatre protocoles de routage (APRL, AODV, ODMRP et STARA en environnement réel. Le réseau maillé (donc statique) est constitué de 40 nœuds. Les auteurs comparent les performances indoor et outdoor des protocoles. AODV qui présente les meilleures performances en outdoor ne permet d'atteindre qu'un taux de livraison de 50% : seulement un paquet sur deux arrive à destination. Ces performances très médiocres nous semblent étranges, les résulats que nous obtenons étant bien meilleurs. Par ailleurs, ODMRP permet un taux de livraison de 60%, mais ODMRP étant un protocole de routage multicast, nous ne pensons pas qu'une comparaison avec les protocoles unicast soit judicieuse. Enfin, les expérimentations indoor consistent à reproduire articifiellement la topologie observée en environnement extérieur à l'aide d'une des méthodes décrites dans le précédent paragraphe. Ainsi, certains liens radio, bien qu'existant et générant des interférences, ne sont pas utilisés. De telles mesures nous semblent donc biaisées.

[2] présente la plate-forme MIT Roofnet, comprenant 37 nœuds dans un environnement urbain. Ce testbed présente la grande spécificité d'être opérationnel et en production, i.e. des utilisateurs l'utilisent librement. Les traces issues de l'utilisation d'un tel testbed, bien que ne faisant pas partie de l'article, pourraient être d'une grande aide pour la communauté. La plate-forme utilise un protocole de routage Srcr, inspiré de DSR mais permettant de favoriser l'utilisation de liens radio à bande passante élevée. Les auteurs présentent une évaluation fine de la qualité des liens radio de la plate-forme (débit TCP versus distance), de la distribution du degré, de la robustesse de la topologie. Les auteurs mesurent également le débit moyen TCP de bout en bout en fonction de la longueur de la route. Les performances présentées valident selon nous l'utilité des réseaux ad-hoc. [23] présente un testbed de 12 nœuds statiques répartis sur une route de Dublin. Des nœuds mobiles peuvent se connecter à Internet via les nœuds fixes, en passant potentiellement par des nœuds mobiles intermédiaires. Les auteurs ont principalement mesuré le débit UDP via des routes découvertes par AODV[1]. Récemment, [6, 14] présentent un rapide tour d'horizon des plate-formes expérimentales actuellement déployées par la communauté.

8.4 Architecture logicielle et matérielle

Nous allons décrire ici l'architecture matérielle et logicielle de l'environnement de tests que nous avons déployé.

8.4.1 Architecture logicielle

Nous avons choisi de déployer un système Linux sur l'ensemble des nœuds actuels de la plate-forme. Ce système d'exploitation présente des avantages en terme de :

- Ouverture du code
- Facilité de programmation réseau
- Système d'exploitation largement utilisé par la communauté, le code pouvant être réutilisé sur d'autres testbeds

- Nombreux outils existant pour la mesure des performances réseau (débit des flux TCP ou UDP...)

Nous avons implémenté un démon permettant de construire le backbone virtuel décrit dans le chapitre 3 page 31, ainsi que le protocole de routage et de localisation décrit dans le chapitre 6 page 107. Ce démon est fonctionnel, la topologie virtuelle construite de façon valide, et le protocole de localisation tirant parti du backbone virtuel fonctionne parfaitement. La mise à jour des systèmes d'exploitation sur les nœuds clients se faisant par l'intermédiaire de la connexion sans-fil multisauts se déroule normalement.

Nous allons dans la suite de cette section détailler certains points de l'implémentation que nous pensons présenter des spécificités nécessitant d'être détaillées.

8.4.1.1 Niveau d'exécution

La philospophie de Linux est d'implémenter en mode noyau le minimum de fonctionnalités. Les programmes doivent, tant que faire se peut, résider dans l'espace d'exécution *normal*. Cette propriété est peu suivie par les concepteurs de protocoles de routage pour les réseaux ad-hoc. Ainsi, Ad-Hoc Support Library (ASL) [13] permet à un démon d'être implémenté dans l'espace utilisateur mais contient lui même des modules noyau, par exemple pour la gestion d'une table de routage modifiée. Les auteurs de [5] présentent trois méthodes pour coder un protocole de routage :

- Utilisation de Netfilter [18]. Cette fonctionnalité du noyau permet d'activer des filtres sur les paquets IP et de passer les paquets à d'autres programmes pendant la traversée de la couche IP
- Le démon peut être codé dans l'espace noyau. Ainsi, Kernel-AODV [15] est une implémentation d'AODV entièrement contenue dans l'espace noyau.
- Le programme écoute les trames passant au niveau MAC. Si une requête ARP est générée, alors une découverte de route est initiée. Cependant, une telle méthode repose sur l'utilisation d'ARP pour un réseau ad-hoc.

Comme nous le verrons, nous n'avons utilisé aucune des trois méthodes décrites ici, car trop *intrusives* selon nous. En effet, une implémentation en mode noyau présente plusieurs inconvénients :

- Une erreur dans le code du démon peut induire une instabilité du système entier
- Le code est spécifique à un noyau. Une modification du noyau nécessite quelquefois des modifications dans le code source du protocole de routage

Notre démon est entièrement implémenté dans l'espace utilisateur et a été testée sur les noyaux 2.6.14 et 2.6.12. Nous verrons un peu plus loin comment nous avons réussi à supprimer l'utilisation de modules noyau.

8.4.1.2 Programmation multi-threads

Le démon nécessite de surveiller plusieurs tables (table de voisinage, table de routage...) et d'éliminer les données obsolètes. Bien que la maintenance soit événementielle, il n'existe aucun moyen pour obtenir en temps réel des informations exactes sur le réseau. Un nœud ne peut qu'implémenter des temporisateurs pour les entrées obsolètes, étant donné que le médium radio n'est pas fiable, ni pour les transmissions en *broadcast*, ni pour celles en *unicast*. De même, les procédures de maintenance de la structure virtuelle se déclenchent sur certains évènements eux mêmes découlant de temporisateurs[1]. Nous avons donc choisi de développer le démon en multi-threads, chaque thread possédant une action spécifique. Un multi-processus aurait demandé plus de mémoire et rendrait le partage de variable moins flexible. Naturellement, le concepteur

[1] La non réception d'une série d'`ap-hellos` constitue par exemple un évènement déclencheur d'une procédure de reconnexion. Nous voyons bien qu'une telle occurrence constitue un *non-évènement*, qu'il faut donc déclencher sur un temporisateur réarmé à chaque réception d'un `ap-hello`

doit gérer des processus concurrents accédant aux mêmes variables. Des IPC telles que des sémaphores [3] doivent donc gérer les accès aux variables partagées.

8.4.1.3 Table de routage

La table de routage classique du noyau Linux a été conçue pour les réseaux filaires : les routes présentent une grande stabilité et les changements constituent des exceptions. Au contraire, dans un réseau ad-hoc, la mise à jour des tables de routage est commune : les nœuds sont mobiles, nécessitant de mettre à jour continuellement sa vue de la topologie. Cependant, nous avons préféré ne pas modifier la table du routage du noyau pour des raisons de portabilité. Ainsi, un thread est chargé de maintenir une table de routage interne au processus, gérer les timeouts, et synchroniser lui-même la table de routage du noyau.

8.4.1.4 Pile protocolaire

Le modèle OSI stipule que les couches protocolaires doivent être indépendantes afin de pouvoir favoriser l'inter-changeabilité entre plusieurs couches. Cependant, nous pensons personnellement qu'une telle architecture est beaucoup trop contraignante pour les réseaux ad-hoc. En effet, l'indépendance des couches permet une plus grande flexibilité au prix d'une légère baisse de performances : aucune information n'est partagée entre couches. Or, dans les réseaux ad-hoc, les ressources en bande passante sont rares. De plus, la mutualisation des informations permet de réduire la quantité de trafic de contrôle, et donc d'améliorer les performances globales. Ainsi, certains protocoles de routage, par exemple DSR, présupposent l'existence d'une API de notification de livraison de paquets par la couche MAC. Une telle fonctionnalité permet de réduire les retransmissions et autorise les reconstructions locales de routes.

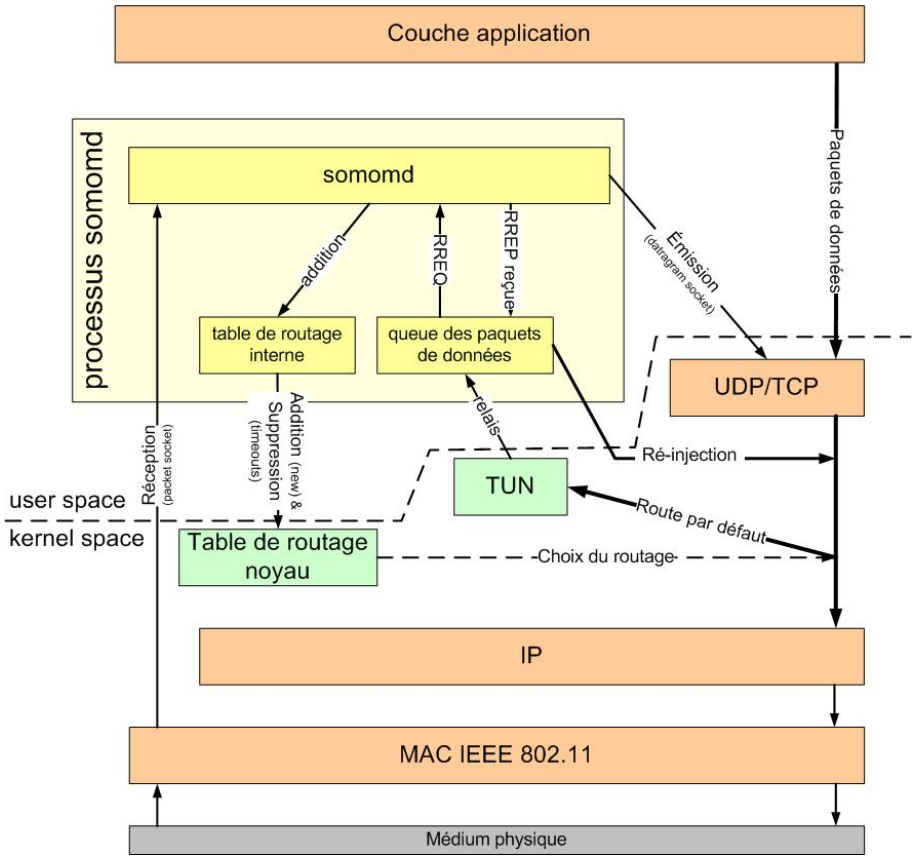


Fig. 8.1 – Architecture logicielle générale de somom

Nous avons donc décidé de *casser* la pile protocolaire, approche souvent dénommée sous le terme de *cross-layer*. L'architecture retenue est représentée sur la figure 8.1 page ci-contre. Le démon s'exécute au niveau logique au dessus de la couche réseau, et emprunte un port UDP pour l'envoi des paquets de contrôle. Par contre, le protocole ayant besoin d'informations sur tous les paquets transitant via la couche IP, le démon implémente une *packet socket*. Cette fonctionnalité offerte par Linux permet de capturer l'ensemble des paquets venant de la couche MAC. Le démon extrait les paquets IP, en extrait l'information qui l'intéresse, et passe éventuellement le paquet aux threads chargés du routage et de la structure virtuelle si les paquets sont destinés au port UDP sur lequel le démon est enregistré. Ainsi, lorsqu'un paquet est relayé par un nœud, SOMOM peut rafraîchir gratuitement la route inverse menant vers la source du paquet.

8.4.1.5 Adressage / Configuration

Dans la première étape de ce déploiement, nous avons choisi d'assigner statiquement les adresses aux nœuds du réseau. Un même préfixe IP est assigné à tout le réseau ad-hoc, mais chaque nœud voit sa propre adresse IP avec un masque réseau de 32 bits, puisque certaines destinations ne peuvent être jointes directement. Ainsi, une entrée de la table de routage est spécifique à un hôte particulier, avec un masque de réseau de 32 bits.

8.4.1.6 Routage

Le comportement réactif n'est pas intégrable de façon triviale dans un démon Linux en mode utilisateur. En effet, le noyau devrait relayer tous les paquets sans route vers le processus réactif afin qu'il initie une découverte de route. Nous avons choisi d'utiliser la fonctionnalité TUN/TAP de Linux. Une interface virtuelle du type `/dev/tunX` est créée derrière laquelle un processus utilisateur peut s'enregistrer afin que le noyau lui relaie les paquets envoyés via cette interface. En créant au sein de la passerelle une route pour le préfixe réseau correspondant à la bulle ad-hoc menant vers cette interface virtuelle, le noyau relaie de façon transparente tous les paquets sans route vers le démon `somomd`. De plus, lorsque plusieurs routes possibles sont présentes, Linux choisit, comme la norme le stipule, la route avec le préfixe le plus long. Ainsi, les routes spécifiques, avec un préfixe de 32 bits sont choisies en priorité, la route menant vers l'interface *tun* n'étant choisi qu'en dernier recours.

Le démon récupère les paquets, les stocke dans une file d'attente temporaire en espace utilisateur, dans un segment mémoire alloué à `somomd`. Puis il génère une découverte de route. Si aucune réponse de route n'arrive au bout d'un timeout (une seconde dans notre cas), le démon renvoie un nouveau `RREQ`. Dans notre implémentation, si un paquet n'obtient aucune réponse de route, il est supprimé de la queue au bout de cinq secondes. Si la passerelle reçoit une réponse de route, elle enregistre la route dans le noyau, extrait dans sa queue locale les paquets destinés à cette entrée, et réinjecte normalement les paquets dans la couche IP. La nouvelle entrée dans la table de routage sera utilisée pour router les paquets vers la bonne destination.

Parallèlement, chaque client possède une route par défaut menant vers son père dans le backbone. Si une route spécifique avec un préfixe de 32 bits existe, elle sera utilisée, sinon, le paquet remontera dans le backbone jusqu'à atteindre la passerelle. Celle-ci sera ensuite chargée de délivrer le paquet vers Internet après avoir fait de la translation d'adresse (NAT). Éventuellement, si la destination est en fait présente au sein de la bulle ad-hoc, la route via l'interface *tun* sera empruntée.

8.4.1.7 Logs

Durant la phase de tests, le comportement du démon doit être finement étudié. Nous avons choisi d'utiliser les fonctions de `syslog`, offertes par Linux : un message de `log`, possédant un certain niveau de sévérité est stocké dans les fichiers de logs système. `Syslog` permet ensuite de

rediriger suivant des filtres les informations pertinentes, tout en les classant. `Syslog` nous permet donc de loguer les comportements du programme, mais également tous les évènements protocolaires, permettant de collecter des statistiques (trafic de contrôle, changement de voisinage, paquets de données relayés, pertes des paquets...).

8.4.2 Équipements des nœuds

Les terminaux utilisés sont des mini-PC silencieux, sans ventilateur pour processeur. Cependant, les capacités sont assez puissantes pour stocker les logs, déployer un système d'exploitation Linux complet, installer un analyseur réseau... Nous avons choisi d'utiliser dans un premier temps des nœuds statiques pour des raisons de facilités de tests, et de configuration. Cependant, la topologie radio peut changer sous l'influence d'obstacles tels que des personnes dans le couloir ou une porte fermée.

La bande de fréquences des 2,4 GHz est intensivement utilisée dans le laboratoire et sur le campus. Nous avons donc choisi d'utiliser des cartes sans-fil Atheros configurées en IEEE 802.11a. De plus, la bande des 5 GHz réduit la portée radio, permettant de déployer dans un espace restreint un réseau multi-sauts. Afin que les cartes présentent un débit identique en broadcast et unicast, nous les avons bloquées à un débit de 6 Mbps [10]. Afin de simplifier l'administration, tous les mini-PC possèdent une carte réseau filaire additionnelle. Le trafic de management ne vient donc pas perturber les expérimentations.

8.4.3 Testbed

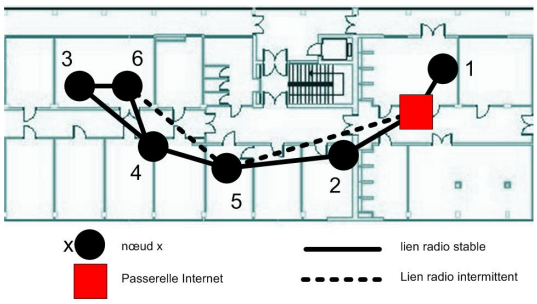


FIG. 8.2 – Topologie du réseau maillé déployé

Le testbed est constitué de 7 mini-PC déployés dans le laboratoire CITI, en environnement indoor. La topologie est quasiment linéaire (fig. 8.2) afin de tester les limites du protocole. Un nœud (un carré rouge sur la figure) agit en tant que passerelle pour Internet implémentant les fonctions de pare-feu, NAT... Certains des liens radio présentent une grande stabilité (représentés en trait plein) : le débit est peu variable au cours du temps, le SNR restant stable et acceptable. Par contre, d'autres liens radio (représentés en pointillés) sont pus fluctuants : ils laissent périodiquement passer des paquets, puis rapidement stoppent tout envoi fiable.

8.5 Évaluation de performances

Le testbed déployé, nous avons mené des expérimentations afin d'en apprécier les performances. Les délai et débits ont notamment été mesurés.

8.5.1 Ping

Dans un premier temps, nous avons mesuré le délai de bout en bout avec un ping de chaque client vers la passerelle. Nous générons donc un ICMP echo request, auquel répond un ICMP

echo reply. La route utilisée existe donc obligatoirement dans les deux sens (upload et download). Par ailleurs, un ping mesure le délai d'aller/retour d'un paquet. Nous avons généré une série de 3 tests de 50 pings de 100 octets, dont nous avons extrait le délai aller/retour moyen, minimum et maximum.

Nous avons dans un premier temps mesuré le délai suivant leur distance à la passerelle (fig. 8.3). Plus la route s'allonge, plus le délai augmente : le paquet nécessite d'être relayé, requérant un délai de transmission dans chaque nœud relais puisque IEEE 802.11 suit une stratégie de type *store and forward*. **Le délai augmente linéairement avec la route**, ce qui nous parait une propriété intéressante. Par ailleurs, nous pouvons voir que les délais maximum, minimum et moyens sont très proches : la gigue est donc réduite (différence de délai entre deux paquets consécutifs). Nous pouvons voir que lorsque la route est de 1 saut, le délai maximum ne suit pas la tendance générale. En effet, le délai maximum représentant le pire cas, sa valeur peut changer de façon importante avec un seul *mauvais* délai.

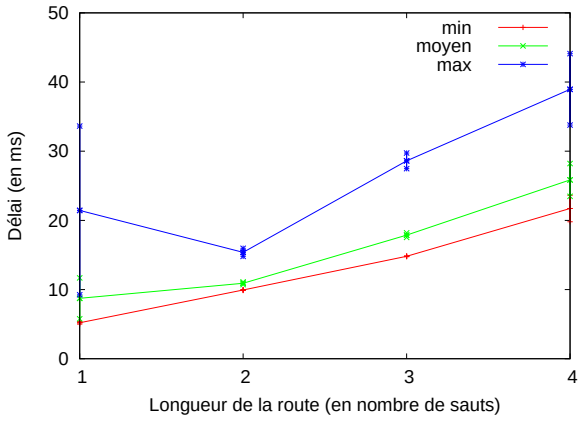


FIG. 8.3 – Délai aller/retour suivant la longueur de la route

Pour mettre ces valeurs en relation, nous avons comparé les résultats de ces mesures avec les mesures sur le testbed dans lequel les routes ont été fixées statiquement (tab. 8.1). Les routes statiques sont les plus courts chemins sur l'ensemble des liens radio stables. Nous pouvons remarquer que **les mesures présentent des résultats similaires**, quelles que soient la taille du paquet et la longueur de la route. Par ailleurs, nous pouvons remarquer que le délai augmente lorsque la taille d'un paquet augmente : le temps requis pour la transmission à un débit constant augmente.

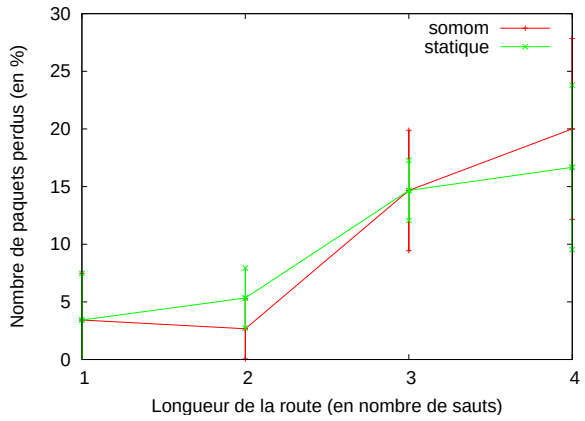
Pk size	Protocol / Source	1	2	3	5	4	6
100	static	1.26	0.8	4	1.9	3.2	5.4
	SOMOM	1.5	0.9	3.3	1.7	2.7	4.2
1500	static	11.5	5.3	28.8	12.9	21.7	32.6
	SOMOM	12	5.3	23.2	10.9	17.9	28.5

TAB. 8.1 – Délai aller/retour d'un ping (en ms)

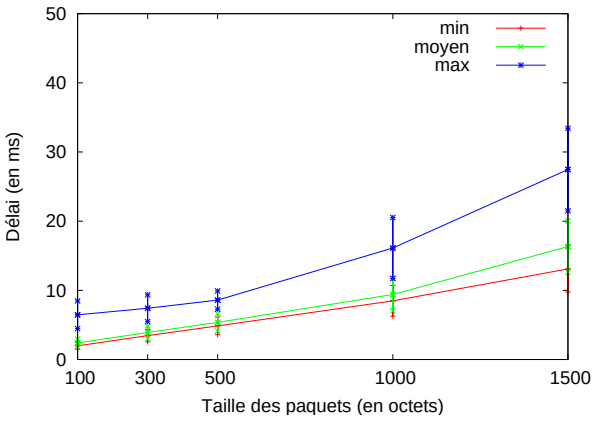
Nous avons également observé le pourcentage de ping ne présentant aucune réponse (fig. 8.4(a) page suivante). Lorsque la route s'allonge, plus de paquets sont perdus, à cause par exemple des collisions entre les différents relais. Les pertes de paquets peuvent atteindre 15% pour une route de 4 sauts. Par ailleurs, une route statique présente des pertes de paquets très similaires. **L'aspect dynamique de SOMOM ne semble donc pas foncièrement nuire aux performances du routage**.

Enfin, nous avons mesuré l'impact de la taille des paquets sur le délai (fig. 8.4(b) page suivante). Un paquet long requerra un délai plus important. Mais là encore, la gigue reste

réduite.



(a) Nombre de paquets perdus suivant la longueur de la route (paquets de 1000 octets)



(b) Délai de bout en bout suivant la taille des paquets

8.5.2 Débit UDP

Nous avons ensuite mesuré le débit maximum offert par un flux UDP avec l'outil netperf [17]. Nous considérons qu'un flux UDP est *faisable* s'il présente une perte inférieure à 5% des paquets. Un flux dure 5 secondes. Si la durée d'un flux est trop importante, la charge réseau provoque la perte de nombreux paquets de contrôle, occasionnant des dysfonctionnements dans les protocoles proactifs. Le débit UDP présente des différences importantes entre les paires de nœuds. Par exemple, le nœud 1 arrive à envoyer un débit de 1 Mbps vers la passerelle, alors que le nœud 2, qui est également à un saut de la passerelle, atteint un débit de 4 Mbps. **La qualité du lien radio possède une conséquence non négligeable sur les performances**. Nous pouvons remarquer qu'une taille de paquets de 1000 octets environ représente une taille optimale vis à vis du débit, présentant un compromis entre la minimisation des collisions et la maximisation du temps de parole. Le débit diminue avec la distance, mais continue à présenter une valeur acceptable (environ 700 kbps).

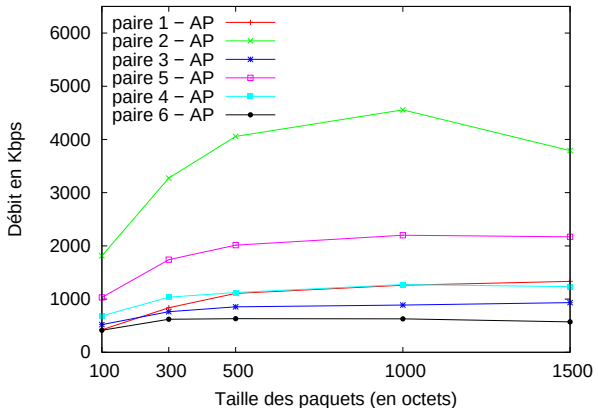


FIG. 8.4 – Débit d'un flux UDP suivant la taille des paquets

8.5.3 Débit TCP

Puisqu'une grande partie des flux actuels sont TCP, nous avons également mesuré le débit TCP maximum atteignable (tab. 8.2 page ci-contre). **Les débits offerts par TCP et UDP**

présentent des valeurs très similaires. Cependant, sur les longues routes, UDP présente un gain appréciable : TCP semble plus souffrir des retransmissions et collisions de paquets, réadaptant son flux dynamiquement.

Type	nœud 1	nœud 2	nœud 3	nœud 5	nœud 4	nœud 6
TCP	1226.6	4563.3	870.6	2113.3	1223.3	431.6
UDP	1260	4556.6	887.3	2200	1273.3	628.3

TAB. 8.2 – Débit maximum atteignable pour un flux TCP / UDP vers la passerelle (paquets de 1000 octets)

8.5.4 Découverte de routes

Enfin, nous avons mesuré le délai nécessaire à la découverte d'une route initiée par la passerelle (fig. 8.3). Pour la route la plus longue (4 sauts), 1,5 secondes sont nécessaires pour un ping (initiation de la découverte de route, retransmissions éventuelles, réception de la réponse, aller/retour du ping). Cependant, une découverte de route est rare : lorsque la connexion est initiée par le client, une route inverse est automatiquement créée dans les caches des routeurs intermédiaires. De plus, ce délai est nécessaire seulement pour le premier paquet d'un flux, la route étant maintenue ensuite dynamiquement.

Type	nœud 3	nœud 4	nœud 6
Délai	1.6	1.8	0.6
Écart-type	1.6	1.1	0.23

TAB. 8.3 – Délai aller/retour (en s) lors d'une découverte d'une route initiée par la passerelle

8.6 Un testbed : avantages, limites et écueils

Ce testbed constitue une première étape dans l'évaluation expérimentale des performances d'un protocole de routage et d'auto-organisation pour réseaux hybrides. Cependant, certaines fonctionnalités restent à développer :

- Plus de nœuds doivent être déployés dans le réseau
- L'impact de nœuds mobiles doit être évalué
- Des nœuds embarqués à plus faible capacité doivent pouvoir être installés afin de mesurer les contraintes de CPU, mémoire, d'énergie...

Nous avons d'ores et déjà rencontrés de nombreux problèmes dans l'utilisation d'un médium radio réel :

- Certains liens radio sont faibles et instables. Une porte fermée suffit à changer la topologie radio. De même, la portée radio n'est pas binaire : certains liens longs ne permettent d'acheminer que x % des paquets. Si un `hello` passe, le nœud considère que le lien radio est utilisable, alors que ses pertes sont en réalité importantes. Nous avons donc modifié SOMoM : un lien radio est considéré valide si plus de 2 paquets `hello` consécutifs sont reçus. Cependant, une métrique d'efficacité de lien radio telle que [7] devrait être implémentée.
- Bien que le testbed soit constitué de nœuds homogènes, des liens unidirectionnels peuvent apparaître, à cause d'antennes non omnidirectionnelles, ou de puissances d'émission différentes. Le protocole doit donc clairement distinguer les liens bidirectionnels de ceux unidirectionnels, comme SOMOM le fait.
- L'environnement radio est très hétérogène. En d'autres termes, les graphes de disque unité représentent une mauvaise représentation des réseaux ad-hoc. Deux mini-PC peuvent être

proches sans qu'ils possèdent un lien radio les interconnectant. Le dimensionnement des réseaux maillés doit donc être minutieusement étudié.

Par ailleurs, nous avons listé quelques problèmes posés actuellement par les testbed :

- La communauté n'a pas encore développé des scénarios types pour l'étude des performances. Ainsi, il est difficile de comparer différents protocoles.
- Le protocole IEEE 802.11 présente des performances médiocres en environnement multisauts, posant des problèmes de gaspillage de bande passante et d'équité [8]. Un autre protocole MAC adapté aux réseaux ad-hoc devrait être proposé.
- IEEE 802.11 ne gère pas la priorité des paquets. Un trafic important véhiculé par le testbed va en conséquence entraîner la perte de nombreux paquets de contrôle. Des routes vont donc se casser. Les flux subissent ainsi des fluctuations de débits importantes.
- IEEE 802.11 ne présente pas le même débit en unicast et en broadcast. Ainsi, les paquets de contrôle envoyés en broadcast portent plus loin. Un nœud peut donc posséder un voisin en mode broadcast sans qu'aucun paquet de données ne puisse être acheminé en unicast.
- L'adaptation de la topologie radio est coûteuse en temps, requérant souvent une approche *essai, erreur*. Une méthode générique de création de topologie serait utile pour les tests.
- Le lancement de tests est coûteux sur un tel environnement distribué. Des outils d'automatisation doivent être obligatoirement être déployés, tout en synchronisant l'horloge de tous les nœuds, sans pour autant générer de trafic de contrôle qui fausserait les expérimentations.

Il reste donc encore de nombreux points à développer avant d'obtenir un testbed flexible et performant.

8.7 Conclusion

Nous avons dans ce chapitre présenté le testbed que nous avons déployé au sein du laboratoire afin de tester les performances de nos protocoles pour réseaux hybrides. Nous avons notamment déployé le protocole de construction et de maintenance de la structure virtuelle, ainsi que SOMoM. Les deux protocoles prouvent leur efficacité, démontrant s'il en était besoin l'utilité des réseaux sans-fil multisauts. SOMoM affiche des performances proches d'un routage statique lorsque les flux sont courts. Si au contraire les flux de données occupent trop le médium, les paquets de contrôle subissent des collisions, nuisant aux performances de SOMoM. Ainsi, si trop de paquets `hellos` subissent des collisions, un nœud peut considérer un lien radio comme rompu, cassant par la même occasion certaines routes.

Une évaluation de performances via des simulations ne constitue qu'une première étape, des expérimentations réelles étant nécessaires afin de valider un protocole dans un environnement radio réel. Ainsi, nous développerons prochainement le protocole maintenant la structure virtuelle en greffant les fonctions de routage de VSR. Des comparaisons de performances entre VSR et SOMoM pourront notamment être réalisées.

Il reste néanmoins à développer certaines fonctionnalités sur notre testbed. Nous devons notamment prendre en charge dans le futur la mobilité, déployer un service de supervision et tester les protocoles sur une autre couche MAC (IEEE 802.11 modifié, ou Bluetooth). A terme, cette plate-forme servira d'accès radio multisauts à Internet pour tous les terminaux du laboratoire. Elle permettra notamment d'étudier des cas d'utilisation réelle d'un réseau hybride ou d'obtenir des traces de mobilité des terminaux.

Bibliographie

[1] P. Barron, S. Weber, S. Clarke, and V. Cahill. Experiences deploying an ad-hoc network in an urban environment. In *Workshop on Multi-hop Ad hoc Networks : from theory to reality (REALMAN)*, Santorini, Greece, July 2005. IEEE.

[2] J. Bicket, D. Aguayo, S. Biswas, and R. Morris. Architecture and evaluation of an unplanned 802.11b mesh network. In *International Conference on Mobile Computing and Networking (MOBICOM)*, Cologne, Germany, August 2005. IEEE.

[3] C. Blaess. *Programmation Système en C sous Linux : signaux, processus, threads, IPC et Sockets*. Eyrolles, 2° édition edition, 2005.

[4] 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.

[5] I. D. Chakeres and E. M. Belding-Royer. AODV routing protocol implementation design. In *International Workshop on Wireless Ad hoc Networking (IWWAN)*, pages 698–703, Tokyo, Japan, March 2004.

[6] P. De, A. Raniwala, S. Sharma, and T.-c. Chiueh. Design considerations for a multihop wireless network testbed. *IEEE Communications Magzaine*, 43(10) :102–109, October 2005.

[7] D. S. J. De Couto, D. Aguayo, B. A. Chambers, and R. Morris. Performance of multihop wireless networks : shortest path is not enough. *ACM SIGCOMM Computer Communication Review*, 33(1) :83–88, january 2003.

[8] D. Dhoutaut. *Etude du standard IEEE 802.11 dans le cadre des réseaux ad hoc : de la simulation à l'expérimentation*. PhD thesis, INSA Lyon, December 2003.

[9] D. Gray, Robert S.and Kotz, C. Newport, N. Dubrovsky, A. Fiske, J. Liu, C. Masone, S. McGrath, and Y. Yuan. Outdoor experimental comparison of four ad hoc routing algorithms. In *International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM)*, pages 220–229, Venice, Italy, October 2004. ACM.

[10] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. Performance anomaly of 802.11b. In *INFOCOM*, San Francisco, USA, April 2003. IEEE.

[11] R. Jain. *The art of computer systems performance analysis*. Wiley, 1991.

[12] J. T. Kaba and D. R. Raichle. Testbed on a desktop : strategies and techniques to support multi-hop manet routing protocol development. In *International Symposium on Mobile Ad Hoc Networking & Computing (MOBIHOC)*, pages 164–172, Long Beach, USA, October 2001. IEEE.

[13] V. Kawadia and P. R. Kumar. Power control and clustering in ad hoc networks. In *INFOCOM*, San Francisco, USA, March-April 2003. IEEE.

[14] W. Kiess and M. Mauve. A survey on real-world implementations of mobile ad-hoc networksnext term. *Ad Hoc Networks*.

[15] L. Klein-Berndt and et.al. Kernel AODV implementation. http ://w3.antd.nist.gov/wctg/aodv_kernel/.

[16] D. A. Maltz, J. Broch, and D. B. Johnson. Experiences designing and building a multi-hop wireless ad hoc network testbed. Technical Report CMU-CS-99-116, School of Computer Science, Carnegie Mellon University, March 1999.

[17] nerperf. http://www.netperf.org/netperf/NetperfPage.html.

[18] netfilter. http://www.netfilter.org/.

[19] Orbit Lab. http://www.orbit-lab.org/.

[20] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, and M. Singh. Overview of the orbit radio grid testbed for evaluation of next-generation wireless network protocols. In *Wireless Communications and Networking Conference (WCNC)*, New Orleans, USA, March 2005. IEEE.

[21] S. Sanghani, T. X. Brown, S. Bhandare, and S. Doshi. EWANT : The emulated wireless ad hoc network testbed. In *Wireless Communications and Networking Conference (WCNC)*, New Orleans, USA, March 2003. IEEE.

[22] C. K. Toh, R. Chen, M. Delwar, and D. Allen. Experimenting with an ad hoc wireless network on campus : insights and experiences. *ACM SIGMETRICS Performance Evaluation Review*, 28(3) :21–29, december 2000.

[23] S. Weber, V. Cahill, S. Clarke, and M. Haahr. Wireless ad hoc network for dublin : A large-scale ad hoc network test-bed. *ERCIM News*, 54 :34–35, July 2003.

[24] Y. Zhang and W. Li. An integrated environment for testing mobile ad-hoc networks. In *International Symposium on Mobile Ad Hoc Networking & Computing (MOBICOM)*, pages 104–111, Lausanne, Switzerland, June 2002. IEEE.

Publications

Logiciel

[1] Self-Organized Mobility Management Daemon (`somomd`), http://sourceforge.net/projects/somom

Chapitre 9

Conclusion

9.1 Apports de la thèse

Nous avons dans cette thèse étudié la problématique de l'auto-organisation des réseaux ad hoc et hybrides et proposé des applications pour de telles structures, par exemple pour le routage et la localisation. Pour nous, une auto-organisation structure le réseau, i.e. elle crée une vue logique, différente de la topologie radio, flexible, pouvant être utilisée par tous les protocoles de niveau 3.

Nous avons commencé par proposer une structure d'auto-organisation, hiérarchisant le réseau, et rendant son exploitation plus aisée, et plus performante. Cette structure d'auto-organisation est constituée à la fois d'une dorsale servant de structure de collecte pour le trafic de contrôle, et d'un découpage du réseau en zones, introduisant un deuxième niveau de hiérarchie. L'originalité de cette proposition est d'abord d'intégrer finement les deux types de structures virtuelles afin qu'elles créent une hiérarchie unifiée. Ensuite, nous avons également proposé des algorithmes tant de construction que de maintenance, aucun algorithme maintenant une dorsale en arbre n'existant dans la littérature. De plus, cette structure virtuelle est maintenue de telle sorte que la persistance en soit optimisée : la hiérarchie est stable, utilisable de façon plus efficace par les protocoles des couches supérieures. Par ailleurs, nous avons finement évalué les performances de ce protocole d'auto-organisation tant grâce à des simulations qu'à une étude analytique. Nous avons notamment démontré les propriétés auto-stabilisantes des algorithmes : ils convergent vers un état stable en étant initialisés dans un état quelconque. La structure virtuelle est donc robuste aux changements de topologies et aux défaillances éventuelles ce certains nœuds.

Cette auto-organisation ne constituant pas un but en soi, nous avons exploré des applications de cette structure d'auto-organisation, notamment pour les fonctions de routage et de localisation. La problématique consistait à concevoir un protocole s'intégrant à l'auto-organisation créée, pour l'exploiter efficacement. En d'autres termes, nous souhaitions démontrer qu'une auto-organisation permet bien d'améliorer les performances du protocole qui l'exploite. Dans notre démarche, nous avons souhaité largement nous inspirer des propositions existantes, tout en les adaptant pour qu'ils puissent agir en symbiose avec la couche d'auto-organisation et pleinement en tirer parti. En effet, pour que la structure virtuelle démontre son intérêt, il est par exemple nécessaire d'adapter le routage pour qu'il tire parti de la stabilité de la structure virtuelle. Nous avons donc présenté un protocole de routage pour les réseaux ad hoc et un protocole de localisation pour les réseaux hybrides. La dorsale permet d'optimiser la diffusion du trafic de contrôle, en limitant sa charge sur le médium radio. De plus, la hiérarchie des clusters nous permet d'optimiser la stabilité des routes. Ces deux protocoles exhibent des performances surpassant celles des protocoles dits à plat, prouvant ainsi l'intérêt d'une structure d'auto-organisation. Cette hiérarchie, cette structure permet bien de servir de support générique à des protocoles l'exploitant, leur permettant d'optimiser leurs performances tout en simplifiant leur conception.

Parallèlement, nous avons montré qu'un tel schéma d'auto-organisation ne possède qu'un impact faible sur la capacité en termes de flots d'un réseau radio multisauts : la hiérarchie introduite ne nuit pas au débit global offert par le réseau. Cependant, pour aboutir à une telle conclusion, nous avons créé un modèle très générique permettant d'évaluer la capacité d'un protocole de routage quelconque associé à une topologie donnée. De par sa généricité, ce travail dépasse donc largement l'objectif initial. Nous avons introduit un modèle des interférences radio et de la répartition de la bande passante entre nœuds en compétition pour accéder au médium radio. Pour cela, deux schémas d'équité ont été introduits : une équité entre nœuds souhaitant communiquer, et une entre liens radio actifs. Cette évaluation de la capacité vient compléter l'étude par simulations des protocoles de routage et de localisation basés sur une auto-organisation proposés précédemment.

Enfin, les performances d'un réseau ad-hoc sont très contraintes par l'utilisation de communications radio. Or les simulations ne permettent que de donner un aperçu des propriétés inhérentes à un tel environnement. Seules des expérimentations dans un environnement réel permettent réellement d'apprécier les performances d'un protocole donné. Nous avons donc souhaité compléter l'évaluation de performances analytique et par simulations menée précédemment. Nous avons déployé un testbed sur lequel le protocole d'auto-organisation s'exécute au sein de chaque nœud. Le protocole de localisation tirant parti de cette auto-organisation a également été implémenté. Nous avons pu ainsi confirmer que ces protocoles présentent des performances intéressantes, même dans un environnement radio réel, corroborant le fait qu'une auto-organisation permet de simplifier les protocoles tout en améliorant leurs performances.

9.2 Perspectives

9.2.1 Manque d'un référentiel commun

Les réseaux ad-hoc forment encore un domaine de recherche jeune, attractif pour beaucoup de chercheurs, très dynamique. Ainsi, de nombreuses conférences et journaux sont dédiés aux réseaux ad-hoc, engendrant un grand nombre de publications. La multiplication de ces travaux présente un avantage certain pour la rapidité de progression dans ce domaine. Cependant, je vois deux inconvénients majeurs, par ailleurs souvent présent dans le domaine de la recherche en réseaux. Tout d'abord, le grand nombre de publications peut avoir tendance à parasiter les grandes propositions : il est extrêmement coûteux en temps de maintenir sa connaissance sur les avancées en réseau ad-hoc. D'autre part, la croissance très rapide de ce domaine a nuit à la création d'un référentiel commun, aucune proposition pivot ne se détachant distinctement du lot. Ainsi, l'évaluation de performances, la comparaison et la mise en perspective des différentes propositions sont délicates : sur quels critères et scénarios se baser pour ne pas biaiser les évaluations ? Nous avons tenté dans notre approche de juguler un tel problème en combinant des méthodes complémentaires d'évaluation de performances. Nous avons ainsi utilisé tant des études analytiques que des simulations et des expérimentations réelles. Nous pouvons ainsi limiter les incertitudes sur les performances réelles de nos propositions.

Si nous prenons le problème de la modélisation de la mobilité, de nombreux modèles ont été, et continuent à être proposés. Cependant, il est impossible de juger quantitativement ces modèles et distinguer les modèles réalistes : un référentiel d'applications types des réseaux ad-hoc n'existe pas encore. Récemment, J.P. Hubaux [1] expliquait que l'évolution de la recherche en réseaux ad hoc passe par la recherche d'applications. Ainsi, ses recherches se focalisent actuellement sur les communications inter-véhiculaires. Selon nous, de telles applications permettront de dégager des applications types et les grands défis de demain.

Enfin, une plate-forme commune pour la validation des protocoles pourrait passer par une approche de type Planet-Lab [2] adaptée aux réseaux ad-hoc. Il reste à définir quelles sont les caractéristiques requises pour une telle plate-forme de tests, devant être assez générique pour

que tout chercheur puisse valider son approche.

9.2.2 Revenir au réel

Les réseaux informatiques sont par essence même une science appliquée, i.e. tournée et inspirée de l'application. Ainsi, dans une telle discipline, il est requis de rester guidé par l'application. Une modélisation du réel est bien entendue nécessaire à une étude poussée. Cependant, une modélisation floue, voire simpliste peut fausser le problème initial, s'écartant alors du domaine des réseaux. Une modélisation peu représentative permet d'amener à un problème plus générique, mais peut également amener si nous n'y prenons pas garde, à un problème factice, sur-contraint.

Je pense donc que le domaine des réseaux ad-hoc devrait repartir de l'expérience. En s'approchant du réel, le chercheur peut continuellement modéliser, proposer et éprouver ses propositions dans un environnement réel. Je pense que cet aller et retour entre ces différents domaines est vital. Bien souvent, les expérimentations nous permettent de détecter des anomalies, qui, une fois modélisées, fournissent des problèmes théoriques. Ainsi, la communauté s'intéresse actuellement au problème de la couche MAC dans les réseaux sans-fil. Or, tout concepteur de testbed remarque obligatoirement les faiblesses de IEEE 802.11, qui le mène tout naturellement à la recherche de nouveaux protocoles dans cette voie là.

9.2.3 Une approche unifiée

Les réseaux ad-hoc sont actuellement fragmentés en de multiples sous-domaines. Ainsi, le routage est différencié de la problématique d'adressage, ou des protocoles de routage différents permettent de prendre en charge l'unicast et le multicast. Pourtant il me semble qu'une approche unifiée permettrait de proposer des solutions efficaces. En considérant le problème des réseaux ad-hoc dans sa globalité, une même solution nous permettrait de résoudre simultanément de nombreuses écueils. Durant cette étude, nous avons tenté d'avoir une telle approche : nous nous sommes intéressés à la problématique de l'auto-organisation, avons proposé des solutions de routage et de localisation, une solution très simple d'économie d'énergie. Naturellement, cette étude doit être étendue pour prendre réellement en compte toutes les fonctions qui doivent pouvoir être proposées par un réseau ad-hoc. La problématique du cross-layer afin d'optimiser les performances entre couches protocolaires connaît actuellement un fort essor. Je pense que nous devrions étendre un tel concept à la conception d'un réseau ad-hoc unifié, sans préjuger de la séparation en couche protocolaire et fonctions différentes. Nous pourrions étudier le problème dans sa globalité, l'adressage, le routage, la gestion de la mobilité, l'économie d'énergie constituant des modules partageant un socle commun. Nous avons proposé dans cette étude qu'un tel socle fédérateur soit constitué par une structure virtuelle d'auto-organisation. Cependant, de nombreuses fonction tirant parti d'une auto-organisation restent à développer (multicast, adressage, auto-configuration...). De plus, nous avons implicitement placé une couche d'auto-organisation au dessus de la couche MAC et juste au dessous des fonctions de routage et localisation. Cependant, il reste encore une question ouverte : une auto-organisation efficace est-elle commune à toutes les couches protocolaires de la couche MAC à l'intergiciel ? Doit-on au contraire segmenter l'auto-organisation pour l'adapter aux contraintes de chaque couche protocolaire et n'autoriser des interactions que par exception ?

9.2.4 Le contrôle de topologie

La problématique du contrôle de topologie représente actuellement une question clé dans les réseaux ad-hoc : comment chaque terminal peut adapter sa puissance de transmission afin de réduire la consommation en énergie et les interférences radio ? Nous avions expliqué que, pour nous, le contrôle de topologie et une couche d'auto-organisation doivent se trouver intimement liés pour être efficaces. Cependant, nous n'avons pas encore exploré une telle problématique :

quelle topologie radio pouvons nous créer afin d'avoir une hiérarchie efficace ? Comment remplir les deux rôles sans perdre la connexité du réseau ? Nous pensons qu'une optimisation conjointe est obligatoire, rendant là encore une approche de type *cross-layer* obligatoire.

9.2.5 Multi-disciplinarité

Le domaine des réseaux possède, selon moi, une inclination naturelle à segmenter les domaines de recherche. Ainsi, des domaines comme la théorie des graphes, la programmation linéaire, la conception d'algorithmes distribués, l'ingénierie des protocoles, les expérimentations, la modélisation mathématiques ont trop souvent tendance à constituer des domaines étanches. Les domaines théoriques sont trop souvent opposés aux domaines d'études pratiques. Je pense qu'au contraire, un domaine de recherche comme les réseaux ad-hoc se nourrit pour la conception de protocoles efficaces des travaux théoriques en informatique fondamentale et théorie des graphes. Nous devons là encore multiplier les interactions entre modélisation, théorie et pratique. De la même manière, le chercheur doit évaluer ses propositions avec tout l'éventail de solutions à sa disposition : analyse théorique en établissant des bornes supérieures et inférieures, simulations, expérimentations... J'ai essayé durant ma thèse de considérer la théorie des graphes, la programmation linéaire, l'algorithmie distribuée ou l'auto-stabilisation comme des outils. J'ai tenté des combiner les approches pour essayer d'apercevoir toutes les facettes d'un problème. Cependant, il reste encore beaucoup de travaux théoriques à tenter d'exploiter. De nombreux travaux se déclarant du domaine des réseaux ad-hoc proposent en réalité des algorithmes centralisés ou distribués mais peu adaptables. De telles propositions n'ont pas été assez approfondies afin de pouvoir conclure sur leur utilité réelle. Il reste donc un travail important d'adaptation de la multitude de propositions théoriques et de leur amélioration pour leur mise en réelle adéquation aux besoins des réseaux ad-hoc.

9.2.6 L'auto-organisation pour les réseaux ad-hoc, mais ailleurs ?

Enfin, une auto-organisation parait présenter un avantage certain dans l'exploitation d'un réseau ad-hoc. Cependant qu'en est-il des autres domaines ? Un réseau de senseurs possède des contraintes spécifiques au regard des réseaux ad-hoc : la redondance de la topologie est plus élevée, l'objectif est collaboratif, le pattern de trafic peut être plus limité, la longévité du réseau représente un objectif primordial... Ainsi, comment une auto-organisation conçue pour les réseaux ad-hoc peut elle être adaptée aux réseaux de capteurs ? Allons même plus loin : un réseau de capteurs nécessite-t-il une auto-organisation ?

De même, les réseaux filaires d'accès se tournent de plus en plus vers la problématique de l'auto-$*$ (auto-configuration, auto-réparation...). Une structure d'auto-organisation peut-elle apporter une solution à de tels problèmes ? Les solutions d'auto-organisation pour les réseaux autonomes peuvent-elles être inspirées de celles conçues pour les réseaux ad-hoc, ou doivent-elle être au contraire entièrement repensées ?

Bibliographie

[1] J.P. Hubaux. Securing vehicular communications. *Summer School RESCOM 2006*, Porquerolles, France. June 2006.

[2] Planet Lab. http://www.planet-lab.org/

Glossaire

ANS Ad hoc Network System, 3

AODV Ad Hoc On-Demand Distance Vector Routing, 84, 94, 111, 154

AP Point d'accès (Access Point), 37, 128

APRL Any Path Routing without Loop, 154

ASL Ad-Hoc Support Library, 155

BCN Backbone Capable Node, 20

BGP Border Gateway Protocol, 81

BNC British Naval Connector, 153

BSC Base Station Controller, 9

CBRP Cluster Based Routing Protocol, 85, 94

CDS Connected Dominating Set, 17, 56

CEDAR Core Extraction Distributed Ad Hoc Routing, 86

CIP Cellular IP, 110

CTS Clear To Send, 42

DCF Distributed Coordination Function, 42

DDR Distributed Dynamic Routing, 86

DS Dominating Set, 34

DSDV Destination Sequence Distance Vector Routing Protocol, 83, 94

DSR Dynamic Source Routing, 83, 85

DSSS Direct Sequence Spread Spectrum, 42

FA Foreign Agent, 109

GPS Global Position System, 82, 84

GPSR Greedy Perimeter Stateless Routing, 84

HA Home Agent, 109

HMIP Hierarchical Mobile IP, 109

ICMP Internet Control Message Protocol, 158

IDMP Intra-Domain Mobility Management Protocol, 110

IEEE Institute of Electrical and Electronics Engineers, 1, 42

IETF Internet Engineering Task Force, 3

IP Internet Protocol, 4

IRTF Internet Research Task Force, 3

IS Independent Set (ensemble indépendant), 14, 130, 136

LANMAR Landmark Routing, 84

LAR Location-Aided Routing, 84

LMST Localized Minimum Spanning Tree, 24

MAC Medium Access Control, 4, 21, 23, 42, 128, 152, 156, 162

MANET Mobile Ad Hoc Network, 2, 110, 144

MEWLANA-RD Mobile Enriched Wireless Local Area Network Architecture-Root Driven, 112

MEWLANA-TD Mobile Enriched Wireless Local Area Network Architecture-Table Driven, 112

MIP Mobile IP, 109

MIPMANET Mobile IP MANET, 112

MIS Maximal Independent Set, 14

MPR Multi Points Relais, 83

MSC Mobile Switching Center, 9

MST Minimum Spanning Tree, 15, 24

NAT Network Address Translation, 157

ODMRP On-Demand Multicast Routing Protocol, 154

OLSR Optimized Link State Routing, 83, 94, 96, 111

OSPF Open Shortest Path First, 81, 83

RREP Route Reply, 83

RREQ Route Request, 84, 157

RTS Request To Send, 42

SNR Signal to Noise Ratio, 42, 126, 130, 153

SOMoM Self-Organized Mobility Management Protocol, 113, 117, 151, 156, 157

ST Spanning Tree, 15

STARA System and Traffic Dependent Adaptive Routing Algorithm, 154

TCP Transport Control Protocol, 153

TTL Time To Live, 34, 36, 39, 88

UDP User Datagram Protocol, 155

VSR Virtual Structure Routing, 87, 91, 94, 96

WCDS Weakly Connected Dominating Set, 19

ZRP Zone Routing Protocol, 84