



HAL
open science

Structure virtuelle pour une auto-organisation dans les réseaux ad hoc et hybrides

Fabrice Theoleyre, Fabrice Valois

► **To cite this version:**

Fabrice Theoleyre, Fabrice Valois. Structure virtuelle pour une auto-organisation dans les réseaux ad hoc et hybrides. *Annals of Telecommunications - annales des télécommunications*, 2007, 62 (1-2). <inria-00405970>

HAL Id: inria-00405970

<https://inria.hal.science/inria-00405970v1>

Submitted on 29 May 2020

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

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



HAL Authorization

Structure virtuelle pour une auto-organisation dans les réseaux ad hoc et hybrides

Fabrice Theoleyre — Fabrice Valois

*Laboratoire CITI - INSA Lyon - INRIA Rhône-Alpes
21, Avenue Jean Capelle, 69621 Villeurbanne Cedex, France
fabrice.theoleyre@imag.fr, fabrice.valois@insa-lyon.fr*

RÉSUMÉ. Les réseaux ad hoc sont des réseaux spontanés sans fil et sans infrastructure fixe de terminaux mobiles. Nous pensons que l'utilisation efficace d'un tel réseau doit passer par son auto-organisation. Nous proposons ici une topologie virtuelle d'auto-organisation constituée à la fois d'une dorsale et de zones de services. La première, véritable prolongation des dorsales des réseaux filaires, permet de collecter le trafic de contrôle. Les zones de services structurent le réseau en le découpant en zones homogènes possédant chacune un chef. Puisque les terminaux sont mobiles, nous proposons à la fois une procédure de construction et de maintenance afin de maintenir une structure virtuelle stable et efficace tout au long de la vie du réseau. Afin de construire une structure virtuelle stable, nous introduisons une métrique de stabilité, dépendant de plusieurs critères. Enfin, nous proposons une solution d'économie d'énergie afin d'optimiser la durée de vie du réseau.

ABSTRACT. Ad hoc networks are spontaneous wireless networks without any wired infrastructure of mobile terminals. We assume that an efficient network is achieved through a self-organization. Thus, we propose here a virtual structure allowing a self-organization, constituted by both a backbone and services areas. The backbone constitutes a connected structure to gather and disseminate the control traffic, representing a natural prolongation of backbone in wired networks. Services areas structure the network in separating the network in homogeneous zones managed by a clusterhead. Since terminals are mobile, we propose both construction and maintenance procedures in order to maintain an efficient and stable virtual structure, during all the network life. In order to create a stable virtual structure, we introduce a stability metric, depending on several criteria. Finally, a power-energy saving solution is proposed to optimize the network lifetime.

MOTS-CLÉS : réseaux ad hoc, structure virtuelle, auto-organisation, dorsale, zones de services

KEYWORDS: mobile ad hoc networks, virtual structure, self-organization, backbone, clusters

I. Introduction

Un réseau ad hoc (MANET) est littéralement un réseau *prêt à l'emploi* : des terminaux peuvent communiquer spontanément via des liaisons radio, sans infrastructure fixe préalable. Le réseau devant fonctionner de façon autonome, sans intervention humaine, les terminaux doivent donc collaborer pour attribuer des adresses, des préfixes réseau, s'entendre sur la fréquence radio à utiliser. . . De plus, un terminal, lorsqu'il souhaite envoyer un paquet de données, peut ne pas être à portée radio de la destination. D'autres nœuds intermédiaires doivent donc relayer le paquet jusqu'à la destination. Pour remplir un tel objectif, les terminaux doivent donc échanger des informations de contrôle pour créer des routes dans le réseau. Les réseaux ad hoc suppriment la distinction usuelle des réseaux classiques entre les clients et les équipements dédiés routant les informations. Par ailleurs, dans un réseau ad hoc, les terminaux sont mobiles, conduisant à des apparitions et disparitions de liens radio. Ainsi, la topologie du réseau est continuellement changeante. Le réseau doit donc s'auto-organiser pour prendre en compte une telle dynamique, et doit pouvoir s'adapter continuellement à ces changements de topologie. Les réseaux ad hoc, de par leur flexibilité, sont promis à un très large spectre d'utilisation. Ils peuvent être utilisés sur les champs de batailles des opérations militaires, permettant ainsi de déployer dynamiquement des liaisons entre fantassins et véhicules sans liaison satellite systématique. Ils sont également utiles dans les scénarios d'organisation des secours en cas de catastrophe naturelle ayant détruit les infrastructures de télécommunications.

Les réseaux ad hoc peuvent également être connectés à Internet via un équipement dédié, le point d'accès, jouant le rôle de passerelle entre le monde filaire et les terminaux radio. De tels réseaux sont appelés *réseaux hybrides* et constituent de véritables *réseaux cellulaires multisauts*. Nous pensons que les réseaux hybrides constituent une évolution naturelle des réseaux de périphérie. De tels réseaux pourraient être utilisés par les opérateurs pour étendre à moindre coût la couverture radio des réseaux cellulaires classiques. Ils permettraient également de connecter un réseau domotique à Internet.

Les réseaux ad hoc et hybrides constituent encore un vaste champ d'étude scientifique. Les solutions classiques doivent être entièrement repensées, les réseaux ad hoc présentant des contraintes fortes. L'utilisation de liaisons radio présentent des contraintes élevées en termes de bande passante disponible, d'interférences radio, d'instabilité des liens radio introduisant des changements de topologie brusques, une fiabilité dégradée et des pertes de paquets. Les réseaux ad hoc comportent de plus des terminaux embarqués : les contraintes en terme de capacité mémoire et de CPU sont donc élevées. De même, les terminaux possèdent des réserves en énergie très limitées. Le réseau doit donc s'accorder afin de suivre une politique efficace d'économie d'énergie. Il reste encore dans les réseaux ad hoc de nombreuses problématiques à aborder. Un schéma d'attribution robuste et distribuée d'adresses doit être proposé, de même que la sécurisation des échanges de données, l'interconnexion à Internet, la gestion de la mobilité au sein d'une zone ad hoc et inter-réseaux ad hoc, etc. Le routage est actuellement la problématique la plus étudiée dans les réseaux ad hoc. Le

coût en trafic de contrôle d'un protocole de routage doit être minimal tout en optimisant le taux de livraison et le délai de bout en bout. Il existe deux grandes classes de protocoles : les *réactifs* découvrent les routes à la demande, et les *proactifs* envoient périodiquement des paquets de topologie pour connaître les routes à l'avance. Cependant, les inondations dans les réseaux ad hoc connaissent des problèmes importants de fiabilité et de redondance, connus sous le problème de la tempête de broadcast [15]. Or, les protocoles de routage actuels utilisent massivement les inondations pour les paquets de topologie ou les requêtes de routes. De nouveaux protocoles de routage doivent donc être proposés.

Nous pensons qu'il est nécessaire de structurer, d'auto-organiser le réseau avant de pouvoir proposer des fonctionnalités évoluées telles que le routage, la localisation, etc. Les terminaux sont, dans un réseau ad hoc, hétérogènes par nature : les ordinateurs portables cohabitent avec les stations fixes, les organisateurs électroniques et les téléphones intelligents. Ainsi les capacités en terme d'énergie, de mémoire, de CPU sont donc foncièrement différentes. Nous pensons donc qu'il est nécessaire de proposer une organisation tirant partie d'une telle hétérogénéité. Nous proposons ici la création d'une structure virtuelle afin de structurer le réseau. Dans un premier temps, une dorsale permet de créer une prolongation naturelle des dorsales des réseaux filaires : certains nœuds du réseau ad hoc sont élus de façon distribuée afin de former un ensemble connexe de collecte de trafic. Cette dorsale permet notamment d'optimiser la diffusion d'information dans le réseau et permet de hiérarchiser les rôles entre des nœuds *forts*, jouant le rôle de routeurs, et les nœuds plus *faibles* jouant alors seulement le rôle de clients. Parallèlement, le réseau est découpé en zones homogènes possédant chacune un chef. Ces zones permettent de créer une vue macroscopique du réseau, s'affranchissant en partie de la topologie radio. Elles forment une topologie plus stable, créant l'équivalent des sous-réseaux filaires.

Nous allons dans une première partie exposer un état de l'art sur les solutions de création de structures virtuelles. Dans la partie III, nous présenterons la topologie virtuelle proposée, constituée d'une dorsale et de clusters. Les algorithmes de construction et de maintenance seront détaillés. Nous justifierons également l'introduction d'un nouveau poids, en accord avec notre objectif de stabilité et de persistance de la topologie virtuelle. Nous montrerons également comment le fait de bénéficier d'une structure d'auto-organisation permet d'introduire un mécanisme simple, mais efficace, d'économie d'énergie. Dans la partie IV, nous présenterons quelques résultats de simulation. Et nous finirons dans la partie V par conclure, en donnant quelques perspectives.

II. État de l'art

II.1. Modélisation

La théorie des graphes est un outil bien adapté à la modélisation des réseaux ad hoc. Un sommet du graphe représente un terminal, et il existe une arête dans le graphe

entre deux sommets si et seulement si les deux terminaux correspondant possèdent un lien radio les unissant. Si nous considérons les antennes radio comme omnidirectionnelles¹ et identiques pour tous les nœuds, alors un réseau ad hoc peut être modélisé à l'aide d'un *Graphe de Disque Unité* [14] : il existe une arête entre deux sommets si et seulement si les disques de rayon 1 et centrés sur ces deux sommets possèdent une intersection non vide. Nous utiliserons les notations suivantes :

- $G(V, E)$: graphe associé au réseau ad hoc, possédant l'ensemble des sommets E et l'ensemble des arêtes V ,
- $N_k(u)$: k -voisinage du sommet u , i.e. ensemble des nœuds à moins de k sauts de u . Par convention, $N_1(u) = N(u)$,
- $\Delta_k(u)$: k -degré de u , i.e. cardinalité de $N_k(u)$. Par convention, $\Delta_1(u) = \Delta(u)$,
- $chemin_{u_1 \rightarrow u_k}$: ensemble de sommets $\{u_1, \dots, u_k\}$ tel que les arêtes $(u_i, u_{i+1})_{i \in [1..k-1]}$ existent. $(k - 1)$ est appelée longueur du chemin.

II.2. Dorsales

Une épine dorsale dans un réseau ad hoc permet de collecter le trafic de contrôle pour un acheminement dans le réseau limitant le nombre de retransmissions. Une telle dorsale peut être modélisée en théorie des graphes à l'aide des *Connected Dominating Sets* (CDS). Par définition, un CDS est l'ensemble V' des sommets de $G(V, E)$ tels que :

$$\forall u \in V, \exists v \in V' \text{ où } v \in N(u) \quad [1]$$

$$\forall (u, v) \in V'^2, \exists c = chemin_{u \rightarrow v} / \forall w \in c, w \in V' \quad [2]$$

Un nœud appartenant au CDS est appelé *dominant*, les autres nœuds étant des *dominés*. Nous avons généralisé ce concept en proposant le concept de k -CDS, en modifiant la proposition 1 :

$$\forall u \in V, \exists v \in V' / v \in N_k(u) \quad [3]$$

On appelle MCDS un CDS de cardinalité minimale. La construction centralisée d'un MCDS est un problème NP-difficile [14]. Par conséquent, sa version distribuée l'est également. De nombreux algorithmes tentent donc d'approximer un MCDS. Certains créent un MCDS en 2 étapes. La première permet d'élire des nœuds *dominants*, tels que tout nœud non dominant, i.e. dominé, soit voisin d'un dominant. L'ensemble ainsi formé est un ensemble dominant total. Dans un deuxième temps, l'ensemble dominant est connecté, en minimisant la cardinalité de l'ensemble final. Un nœud peut prendre les états dominant, dominé, actif (en processus d'élection) ou isolé (en initialisation). Dans [5, 6, 10, 12], un leader se déclare dominant. Ses voisins deviennent

1. Une antenne est omnidirectionnelle si elle émet à la même puissance dans toutes les directions.

dominés, et les voisins des dominés deviennent actifs. L'actif possédant, à la fin de l'itération, le plus fort poids parmi ses voisins actifs devient dominant. Dans [5, 12], la métrique d'élection est le degré d'un nœud, dans [6] l'adresse, et dans [10] un poids générique, non explicite. La complexité en paquets est donc en $O(n)$ (chaque nœud envoyant au plus un message de changement d'état), et la complexité en temps en $O(n)$ (cas d'une chaîne linéaire où chaque itération ne colore qu'un nœud en actif). Puisqu'un dominé est voisin d'un dominant et qu'un actif élu dominant est voisin d'un dominé, un dominant est à au plus 2 sauts d'un autre dominant. La deuxième étape consiste à interconnecter ces dominants.

[6] propose un mécanisme d'exploration en partant du leader : un dominant choisit d'explorer un dominé voisin qui possède le plus de voisins dominants non connectés à la dorsale. Ce dominé exploré devient dominant et passe le message d'exploration à un de ses dominants voisins qui n'était pas auparavant connecté. L'exploration termine lorsqu'aucun dominé ne possède un dominant non connexe à la dorsale. L'exploration étant séquentielle, la complexité en temps est en $O(n)$, et celle en messages en $O(n)$ (un nœud reçoit au plus un message d'exploration). Une telle approche est peu adaptable à la construction d'un k-CDS, où un dominant possède un autre dominant à au plus $2k$ sauts. Chaque dominant doit optimiser la longueur de l'interconnexion aux autres dominants, donnée qu'il ne possède pas. Dans [5], les auteurs suivent une approche semblable en introduisant en plus un temporisateur afin d'obliger la fin de la première étape avant l'interconnexion. Aucune maintenance n'est proposée dans [5, 6] bien que les réseaux ad hoc présentent une topologie volatile.

[1] propose une méthode d'interconnexion moins performante en terme de cardinalité, mais plus simple. Chaque dominant envoie un message d'invitation à la connexion lorsque son voisinage ne comprend plus de voisins isolés ou actifs. Ces paquets possèdent un TTL (durée de vie ou *Time To Live* en anglais) de 2 sauts, distance maximale entre deux dominants. Un dominant sans ancêtre recevant un tel message va fixer l'expéditeur comme son nouveau père, et va forcer le dominé intermédiaire à devenir dominant. La complexité en temps d'un tel algorithme est donc en $O(n)$ et celle en message en $O(n)$ également : un dominé relaie au plus une invitation à la connexion avec un TTL t . Cependant, ni une étude de cardinalité en simulation ni une maintenance ne sont proposées.

Dans [12], les auteurs proposent la création de bases de données de localisation, se rapprochant de la construction d'un k-CDS. Un nœud peut avoir l'état : dominant, normal (un dominant est à au plus k sauts), samaritain (il possède un dominant et au moins un k-voisin est dans l'état panique), panique (aucun dominant n'est présent dans le k-voisinage). A chaque étape, le nœud possédant le plus de voisins *panique* devient dominant. Ces dominants jouent ensuite le rôle de bases de données et doivent s'interconnecter. Deux dominants étant espacés par construction d'au plus $2k + 1$ sauts, les dominants envoient périodiquement des paquets `hello`s avec un TTL de $2k + 1$. Lors de la maintenance, les auteurs proposent, afin de limiter la redondance, de fusionner les bases de données à moins de D sauts l'une de l'autre ($D < 2k + 1$). Nous estimons la complexité en temps de l'algorithme d'élection de dominants en

$O(n)$, et celle en messages en $O(n^2 \cdot \Delta_k)$ (le processus requiert au plus n étapes, durant chacune desquelles chaque nœud doit envoyer un message d'état à l'ensemble de ses k -voisins). Les `hello`s propagés à $2k + 1$ sauts peuvent présenter un volume de trafic de contrôle important. De plus, le paramètre D de fusion des dominants nous semble complexe à régler, proposant un compromis entre trafic de contrôle de fusion des bases, et trafic de contrôle de redondance des dominants. CEDAR [16] suit une approche similaire en élisant un ensemble dominant pour créer un CDS. Les dominants envoient ensuite des `hello`s à 3 sauts afin de créer des liens virtuels, i.e. des tunnels passant par des dominés, pour connecter les dominants entre eux. Cependant, la complexité reste élevée et la mobilité engendre rapidement des liens virtuels sous optimaux.

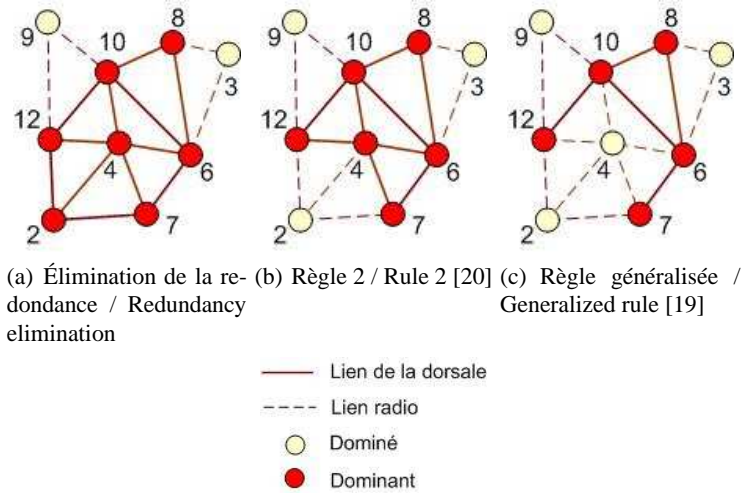


Figure 1. Construction d'un CDS utilisant les différentes règles de Dai & Wu
Construction of a CDS with the different rules of Dai & Wu

[20] présente un algorithme localisé de construction de CDS. Chaque nœud envoie ses `hello`s en broadcast pour découvrir son voisinage. Il exécute ensuite la règle suivante : *Un nœud est dominant si au moins deux de ses voisins ne sont pas connectés entre eux*. Prenons l'exemple de la figure 1(a). Le nœud 9 possède 2 voisins (12 et 10) connectés entre eux. Ainsi, 9 est dominé. Par contre, 8 possède 3 voisins (3, 6, 10) mais 3 et 10 n'ont pas de lien radio commun. Ainsi, 8 est dominant. Nous pouvons remarquer que la cardinalité du CDS ainsi construit peut être élevée, c'est donc pourquoi les auteurs ont proposé des extensions pour pallier un tel problème (connues sous le nom de règles 1 et 2). Dans la règle 1, un nœud u devient dominé s'il possède un voisin v d'identifiant supérieur et le couvrant, i.e. $N(u) \subseteq N(v)$ et si $id(u) < id(v)$. La règle 2 prend en compte le fait qu'un nœud u puisse être couvert par 2 nœuds v et w : $N(u) \subseteq N(v) \cup N(w)$, $id(u) < id(v)$, $id(v) < id(w)$, $v \in N(w)$. Les dominants forment finalement un ensemble connexe et dominant. De plus, les auteurs prouvent que les plus courtes routes empruntent forcément le CDS (les extrémités de la route

exceptées). Soit l'exemple de la figure 1(b) : le nœud 2 possède un voisin d'identifiant supérieur (4) couvrant tout son voisinage (7 et 12). D'après la règle 1, il est donc dominé.

Les auteurs étendent leur algorithme dans [19] en proposant une règle k pour éliminer plus de redondance. Enfin, cette règle a été généralisée par [7] : *un nœud est couvert si un ensemble connexe de ses voisins d'identité supérieure forme un ensemble dominant de son propre voisinage*. Soit l'exemple de la figure 1(c) le nœud 4 possède un ensemble de voisins d'id supérieur (6, 10, 12) qui forme un CDS de son voisinage (2, 6, 7, 10, 12). 4 devient donc dominé. Toutefois, le CDS de Dai & Wu a été conçu pour optimiser les mécanismes de *broadcast*, mais en aucun cas la persistance de la structure. Les simulations de la section IV mettent en exergue un nombre élevé de changements. Ainsi, par exemple un protocole de routage hiérarchique basé sur cette structure pourrait être moins efficace. De même, [18] propose d'élire dynamiquement des relayeurs pour réduire le trafic de contrôle d'une inondation. Mais, cette structure étant orientée source, sa persistance est nulle.

De tels CDS peuvent être utilisés pour le routage. OSR [17] propose d'utiliser le CDS construit afin d'optimiser les inondations des paquets de topologie d'un protocole de routage proactif. CEDAR [16] propose un protocole plus complexe, permettant de construire des routes selon une qualité de service donnée.

II.3. Clusters

Le découpage d'un réseau en zones homogènes est le *clustering*. Un cluster peut posséder une contrainte en terme de diamètre : la distance maximale entre 2 nœuds d'un même cluster est d'au plus 3 sauts. Lorsqu'un chef (*clusterhead*) est maintenu dans chaque cluster, la contrainte est usuellement formulée en terme de rayon :

$$\forall u \in V, \exists c \in C / c \in N_k(u) \quad (C = \bigcup \text{clusterheads}) \quad [4]$$

[13] propose un algorithme de clustering largement utilisé dans la littérature. Chaque nœud initie une découverte de voisinage. Puis tout nœud possédant un poids maximal dans son voisinage devient clusterhead, ses voisins devenant membres de son cluster. Lors de la maintenance, chaque nœud du cluster vérifie que la contrainte de diamètre n'est pas violée, sinon il décide des clusters à former en minimisant le nombre de nœuds devant changer de cluster. La maintenance est réalisée de façon distribuée, les clusterheads étant inutiles après la construction. De même, la fusion de 2 clusters est autorisée si le diamètre est inférieur à 3 après la fusion. La complexité en messages et en temps est en $O(n)$.

[2] propose la création de k -clusters. Durant la première des k itérations, chaque nœud propage l'identifiant le plus élevé entendu à l'itération précédente, propageant les plus grands identifiants à k sauts. Au cours de la deuxième itération, chaque nœud relaie le plus petit identifiant entendu au précédemment, permettant de créer des clus-

ters connexes. Un nœud entendant son identifiant durant cette phase devient clusterhead. Un nœud choisit comme clusterhead le dernier identifiant qu'il relaie. La complexité en temps est en $O(k)$ (le nombre d'itérations), et celle en messages en $O(n \cdot k)$.

Les contraintes de construction de clusters peuvent également être différentes. Dans [3], les k -clusters sont des clusters comportant au plus k membres. Les auteurs proposent de construire un arbre recouvrant dans le réseau et d'étayer les branches en partant des feuilles quand elles comportent k nœuds. La maintenance consiste à scinder les clusters trop importants ou au contraire fusionner de petits clusters en relâchant les contraintes de cardinalité. Cependant, cet algorithme requiert une connaissance du cluster et de ses voisins pour décider des fusions et scissions, le rendant sensible aux problèmes d'incohérences.

II.4. Poids

Les algorithmes de construction de clusters et de dorsales utilisent pour la plupart un processus d'élection distribuée. Les métriques de décision utilisées couramment sont principalement le degré et l'identifiant. Cependant, nous pensons que de telles métriques ne sont pas représentatives de l'aptitude d'un nœud à participer à la construction d'une structure stable et efficace en terme de cardinalité, d'optimisation de la diffusion, etc.

[4] propose d'utiliser un poids basé sur la mobilité absolue calculée à l'aide d'un GPS. Plus un nœud est mobile, plus son poids est faible. Cependant, un GPS est un équipement lourd et coûteux en énergie pour être systématiquement utilisé dans un terminal embarqué. Les auteurs proposent également un poids basé sur la puissance de transmission, un nœud à forte puissance pouvant servir de nombreux nœuds.

[8] propose d'utiliser une métrique combinée prenant en compte : la distance moyenne aux voisins (calculée à partir de la puissance en réception du signal radio), la différence de degré par rapport à un degré optimal, la durée d'un chef (un chef reste chef) et la mobilité absolue. Cependant, un signal radio subit des changements fréquents, rendant l'estimation de la distance erronée, d'autant plus en milieu *indoor*. De plus, dans les simulations, la métrique est basée à 70% sur le degré et 20% sur les distances, ce qui revient finalement à une élection basée sur le degré, les conflits étant départagés par les distances. [9] propose d'ajouter un critère basé sur l'énergie restante. Cependant, les simulations ne tiennent toujours pas compte significativement des métriques d'énergie et de mobilité. D'autre part, un GPS reste nécessaire.

III. Une proposition d'Auto-Organisation

III.1. Motivations

Nous souhaitons organiser le réseau à l'aide d'une structure virtuelle (fig. 2), permettant d'implémenter plus facilement de nouvelles fonctionnalités telles que le rou-

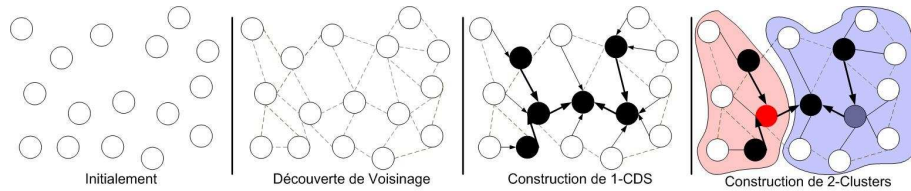


Figure 2. *Déroulement de la construction de la structure virtuelle*
Behavior of the construction of the virtual structure

tage, la localisation, l’adressage... Dans ce but, nous construisons dans un premier temps une dorsale. Elle est constituée de nœuds *forts*, et est chargée de collecter le trafic de contrôle afin d’en optimiser sa diffusion pour pallier le problème de la tempête d’inondation. Les nœuds de la dorsale sont choisis selon des critères reflétant leur aptitude à construire avec les autres nœuds élus une dorsale stable dans le temps et de cardinalité réduite. Parallèlement, nous proposons de découper le réseau ad hoc en entités logiques, introduisant une hiérarchie dans le réseau : un chef dirige une zone de services, organise le routage, attribue les adresses... La dorsale et les zones de services sont intégrées, combinant les fonctionnalités tout en limitant le trafic de contrôle requis pour la construction et la maintenance. Nous avons choisi d’optimiser la stabilité de ces structures plutôt que de nous attacher seulement à une minimisation de leur cardinalité.

III.2. Construction de la Structure Virtuelle

Nous avons choisi de construire la dorsale avant les zones de services (fig. 2) pour :

- optimiser le nombre de nœuds 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 nœud et son chef de zone via la dorsale.

III.2.1. Découverte de Voisinage

Pour construire la dorsale, un nœud doit connaître son k_{cds} -voisinage. De plus, pour détecter les liens unidirectionnels, chaque voisin doit inclure dans ses paquets de topologie ses 1-voisins. Ainsi, un nœud doit périodiquement envoyer en broadcast un paquet `hello` relayé sur $k_{cds}-1$ sauts. Chacun peut reconstruire son k_{cds} -voisinage, en distinguant les liens bidirectionnels de ceux unidirectionnels.

III.2.2. Dorsale

Nous avons choisi de construire une dorsale sous la forme d’un k_{cds} -CDS : la distance entre un nœud 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. Durant la construction, le point d'accès d'un réseau hybride joue le rôle de leader naturel du réseau. Dans un réseau ad hoc, un leader peut être élu de façon distribuée. Un nœud peut prendre l'un des états suivants :

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

Comme les algorithmes classiques de construction de dorsales, nous construisons dans un premier temps un ensemble dominant total. Le leader devient le premier dominant, engageant le processus de construction, engendrant une cascade de décisions localisées, selon les règles suivantes :

- un nœud isolé ou actif recevant un message d'un dominant à moins de k_{cds} sauts devient dominé et fixe la source comme père,
- un nœud isolé recevant un message d'un dominé à moins de $k_{cds}+1$ sauts devient actif et arme un temporisateur pour l'élection,
- un nœud actif pour lequel le temporisateur est écoulé, qui possède le poids le plus élevé parmi tous ses k_{cds} -voisins actifs devient dominant. Ce poids est une métrique permettant de discriminer les nœuds, et sera explicité plus loin dans la section III.6.

Schématiquement, l'algorithme de construction agit par vagues : à chaque vague un ou plusieurs nœuds actifs sont élus dominants, les voisins de ces dominants deviennent leurs dominés, puis les voisins des dominés deviennent actifs. Par ailleurs, nous pouvons noter qu'il est nécessaire dans cette phase de connaître l'état des nœuds à exactement $k_{cds}+1$ sauts. Ainsi, il suffit qu'un nœud changeant d'état envoie un hello avec un TTL de $k_{cds}+1$.

Nous avons choisi ensuite de connecter l'ensemble dominant total en nous inspirant de l'algorithme [1]. Initialement, seul le leader est considéré comme connecté. Un dominant connecté envoie un `cds-invite` avec un TTL de $2 \cdot k_{cds} + 1$. Un dominé relaie un `cds-invite` avec un TTL t , s'il a relayé au plus $x - 1$ `cds-invite` de TTL supérieur ou égal à t . Nous pouvons fixer $x = 1$ dans un environnement radio sans perte. Un dominant non connecté recevant un `cds-invite` devient connecté, fixe comme père le dominé lui ayant envoyé le paquet puis répond avec un `cds-accept`, suivant la route inverse. Chaque dominé recevant le `cds-accept` devient dominant, fixe comme père le prochain saut sur la route, puis relaie le paquet. Finalement, le nouveau dominant connecté envoie lui-même un `cds-invite` pour autoriser les autres dominants à se connecter à la dorsale par son intermédiaire.

La dorsale ainsi construite est un arbre avec le leader à la racine, les dominants formant les branches, et les dominés formant les feuilles (fig. 3 pour un exemple de

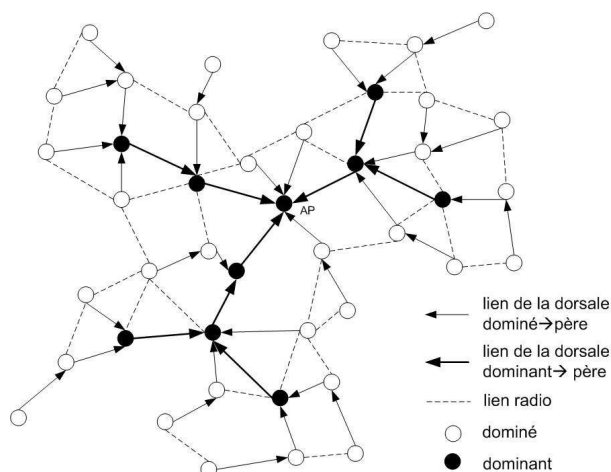


Figure 3. *Exemple d'un 2-CDS*
Example of a 2-CDS

2-CDS). La complexité en temps est en $O(n)$, et celle en messages en $O(n)$ (chaque nœud relayant ou envoyant un message d'état et au plus x cds-invite).

III.2.3. Zones de service

Seuls les dominants participent à la construction des zones. Un dominé rallie automatiquement le même cluster que son père, qu'il connaît grâce aux hellos venant de lui.

Le leader initie la construction des zones dès que son voisinage a fini de construire la dorsale. Le processus de construction se propage ensuite le long de la dorsale. Chaque dominant initie une découverte de voisinage de la topologie virtuelle : deux dominants sont voisins virtuels s'ils ont une relation de parenté père-fils dans la dorsale. Chaque dominant envoie donc périodiquement des `cluster-hellos`, propagés le long de la dorsale, avec un TTL $k_{cluster} - k_{cds}$. Le leader envoie le premier `cluster-hello`, déclenchant le processus de découverte de voisinage virtuel. Un dominant stoppe de tels envois lorsqu'il ne possède plus aucun $k_{cluster} - k_{cds}$ voisin virtuel sans chef de zone. Un dominant possédant le plus fort poids parmi tous ses $k_{cluster} - k_{cds}$ -voisins virtuels sans chef, s'élit chef de zone. Il envoie un `cluster-hello` gratuit² afin de prévenir de sa décision. Un dominant relayant un tel `cluster-hello` choisit la source comme chef si le précédent saut a lui-même choisi la source comme chef. Nous forçons ainsi la connexité des clusters. Il est important de noter que les paquets `cluster-hellos` ne sont envoyés que lors de la construc-

2. On dit d'un paquet qu'il est gratuit s'il est transmis immédiatement, avant que le temporisateur n'expire, afin d'accélérer la convergence du protocole.

tion, et non durant la maintenance. Nous pensons donc qu'un tel trafic de contrôle est négligeable.

Un dominé est à au plus k_{cds} sauts de son père, lui même à $k_{cluster} - k_{cds}$ sauts de son chef de zone. Ainsi, les clusters formés possèdent donc bien un rayon $k_{cluster}$. La complexité en temps est en $O(n)$ et celle en messages en $O(\Delta_{k_{cluster}-k_{cds}} \cdot n)$ (il existe au plus $O(n)$ itérations durant chacune desquelles un dominant envoie au plus un cluster-hello à une distance de $k_{cluster} - k_{cds}$ sauts dans la dorsale).

III.3. Maintenance Événementielle de la 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.

Afin de maintenir son k_{cds} -voisinage, chaque nœud envoie 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.

III.3.1. Dorsale

La dorsale doit rester connexe, et à portée de tous les dominés. Nous proposons donc les procédures de maintenance suivantes :

Dominé Un dominé vérifie seulement la validité de son père. Un père P est valide s'il est toujours dominant, à moins de k_{cds} sauts, et qu'il existe un voisin se déclarant à moins de $k_{cds} - 1$ sauts de P et possédant P comme père. Si un père est invalide, le dominé cherche un nouveau père dans sa table de voisinage. Nous maximisons ainsi dans notre algorithme la persistance : un dominé garde le même père tant qu'il le peut. Le dominé avertit son nouveau père en envoyant un hello gratuit.

Si un dominé ne trouve aucun père valide, alors il devient actif. Ultérieurement, au moins un nœud actif sera élu dominant et exécutera la procédure de maintenance réservée aux dominants.

Dominant déconnecté 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 :

- D génère un *cds-reconnect* avec le numéro de séquence du dernier *ap-hello* entendu. Il envoie le paquet en broadcast avec un TTL fixé à $2 \cdot k_{c_{ds}} + 1$,
- Les dominés de D relaient le paquet en broadcast,
- Les autres dominés le relaient en unicast vers leur dominant afin de minimiser le volume de trafic de contrôle généré,
- 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*, agissant comme 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 possible sans latence et sans trafic de contrôle. Cette procédure de reconnexion est illustrée sur la figure 4.

Cassure de branche 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.

Un dominant s'apercevant qu'il possède dans sa table de voisinage un $k_{c_{ds}}$ -voisin d'état isolé envoie un *cds-invite* avec un TTL de $k_{c_{ds}} + 1$. De même un dominé voisin de son dominant et qui possède un voisin isolé à exactement $k_{c_{ds}}$ 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.

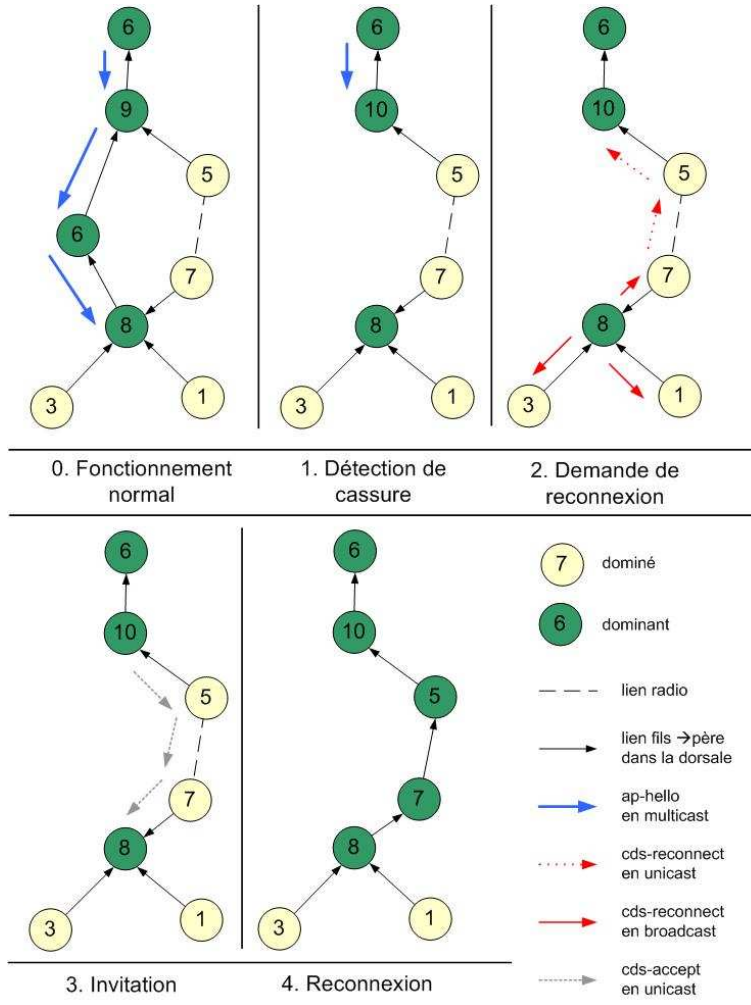


Figure 4. *Déroulement d'une reconnexion de la dorsale (cas d'un 1-CDS)*
Scheme of a backbone reconnection (case of a 1-CDS)

Cardinalité 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 aucun fils dominant, et seulement des dominés à au plus $k_{cds} - 1$ sauts. 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 inutile. D'autre part, le diamètre du CDS étant également plus réduit, un paquet diffusé dans la dorsale a moins de chance de subir une collision et d'être perdu.

III.3.2. Zones de service

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 chef via la dorsale et le prochain saut via la dorsale pour l'atteindre.

Un dominant D considère son chef perdu si le relais vers son chef n'est plus un voisin ou annonce un chef différent ou si sa distance annoncée au chef est supérieure strictement à $(k_{cluster} - k_{cds} - 1)$ sauts. Si son relais 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 prendre une nouvelle 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 :

- 1) 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,
- 2) 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 pour ses prochains hellos la distance qui l'en sépare 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. Un algorithme de type vecteur de distance converge rapidement car la distance au chef de cluster est faible. Par ailleurs, la dorsale représente un arbre lors de son fonctionnement normal. Ainsi, les incohérences dans le choix d'un chef de cluster sont rares.

Un chef de grappe 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 l'a choisi comme chef. Un tel nœud inutile cherche un cluster existant à rallier. Si un tel cluster existe, il s'y attache en devenant un nœud client.

Phase ³	Type de paquet	Rôle	Mode ⁴
C	cluster-hello	Découverte de la topologie du backbone pour élire les clusterheads	B
C&M	hello	Découverte du voisinage	B
	cds-invite	Invitation à se connecter à la dorsale	B/U
	cds-accept	Valide un chemin de connexion à la dorsale (colore les intermédiaires en dominants)	U
M	cds-reconnect	Demande de reconnexion à la dorsale	B/U
	ap-hello	Paquet inondé dans la dorsale pour la détection des déconnexions	M

3. C : Construction, M : Maintenance, C&M : Construction & Maintenance

4. U : Unicast, B : Broadcast, U/B : Unicast ou Broadcast, M : Multicast (i.e. tous les membres de la dorsale)

Tableau I. Ensemble des paquets de contrôle nécessaires à la construction et la maintenance de l'auto-organisation

Control packets required to construct and maintain the self-organization structure

III.4. Paquets de construction et maintenance de l'auto-organisation

Nous avons rassemblés dans le tableau I l'ensemble des paquets utiles à la construction et à la maintenance de l'auto-organisation. Nous avons notamment synthétisé les phases (construction, maintenance) dans lesquelles apparaît chaque type de paquets et quel mode de transmission est utilisé (unicast, broadcast).

III.5. Interconnexion des dorsales

III.5.1. Redondance des points d'accès

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. Afin qu'un nœud identifie l'AP qui le dessert, le point d'accès ajoute dans ses ap-hellos un champ *identifiant de dorsale*. Le nombre de dorsale étant paramétrable, mais probablement réduit, ce champ comporte peu de bits.

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, agissant donc comme un *handover* pour sa branche.

III.5.2. Interconnexion des dorsales

Les points d'accès sont interconnectés par le réseau filaire, constituant ainsi une racine commune de l'ensemble des dorsales. Cependant, il peut être intéressant d'interconnecter les dorsales créées lorsque le chemin suivi par le paquet de contrôle est important. Ainsi, lorsqu'une requête de routes est envoyée, il peut être intéressant de ne pas obligatoirement passer par le point d'accès lorsque la destination est desservie par un autre point d'accès (fig. 5).

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 d'optimiser 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. Ainsi, nous avons choisi que les dominants élisent leurs dominés-connecteurs à partir des `hello`s de leurs dominés, incluant les identifiants de dorsales entendus dans le 1-voisinage. Un dominant possède au plus un dominé-connecteur par dorsale voisine.

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 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.

III.5.3. Acquittements

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

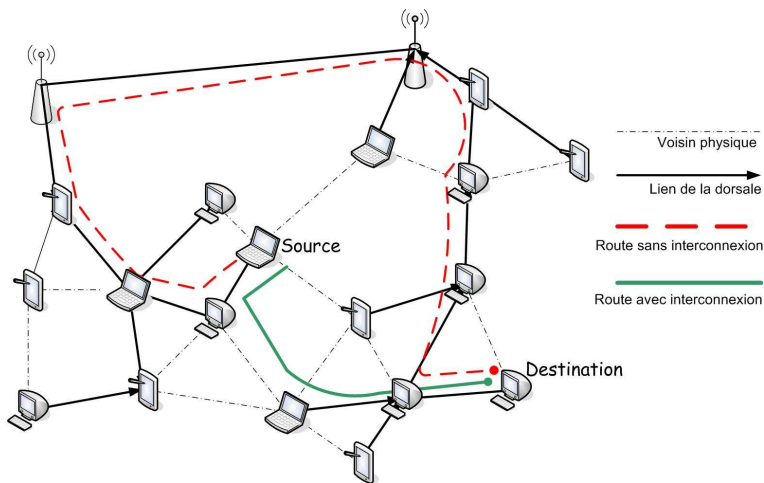


Figure 5. *Schéma d'interconnexion de dorsales*
Interconnection scheme of different backbones

III.6. 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 dominant. Il est évident que le choix de mauvais dominants conduira à une baisse significative de 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.

Comme nous l'avons rappelé dans la section II.4, de nombreux critères existent pour élire un dominant : l'identifiant le plus faible, le degré le plus important, la mobilité relative la plus faible ou encore le niveau d'énergie. Le choix de l'identifiant introduit une sélection aléatoire des dominants sans tirer parti des caractéristiques intrinsèques des nœuds, en particulier de leur hétérogénéité. 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. La mobilité et l'énergie ne peuvent, à elles seules, assurer la sélection pertinente d'un nœud.

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 dans le temps et l'espace les plus fortes. Nous proposons donc le poids ainsi défini :

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

où :

- D représente la différence du degré (nombre de voisins) par rapport à une valeur optimale. Ce degré optimal est fixé à 10 dans notre évaluation de performances, proposant un compromis entre congestions et nombre de dominants. Cependant, il représente d’une manière générale un paramètre de la métrique de stabilité

- M représente l’évaluation de la mobilité locale calculée à partir d’une moyenne glissante du nombre de changement de voisins

- δ est un critère discriminant lié au niveau d’énergie disponible. δ peut varier sous forme de paliers selon l’énergie restante. Un nœud avec peu d’énergie ne pourra donc être dominant

- α et β sont des facteurs de pondération suivant si le concepteur souhaite privilégier les nœuds avec un fort voisinage ou une faible mobilité. Dans nos évaluations de performances : $\beta \gg \alpha$ (section IV).

III.7. 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 [11], 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, en outre, d’introduire une meilleure gestion de l’énergie pour l’ensemble des nœuds. Effectivement, 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 créée.

Plus précisément, nous proposons qu’un dominé ne possédant pas suffisamment de voisins actifs ne puisse pas s’endormir : un dominé ne doit être autorisé à s’endormir que s’il possède au moins 6 voisins actifs. Afin de ne pas pénaliser la maintenance, nous interdisons à un nœud de dormir lorsqu’il est voisin d’une reconnexion de dorsale. Enfin, nous introduisons une pénalité (P_e) représentant le nombre de voisins 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 nœuds les moins importants dans le réseau. Bien entendu, cette métrique peut être modifiée suivant le modèle de consommation d’énergie des nœuds, l’objectif en terme de durée de vie du réseau, etc. Lorsqu’un

dominé possède le plus faible poids parmi son voisinage, il s'endormira de façon certaine. Au contraire, un dominé de poids élevé devra laisser la priorité aux nœuds de plus faible poids. Enfin, un dominant n'est pas autorisé à s'endormir : il est nécessaire à une gestion efficace du réseau.

IV. Résultats

Nous présentons dans cette partie les résultats de campagnes de simulation effectuées sous OPNET Modeler. Nous avons utilisé l'environnement radio 802.11b fourni par OPNET en mode DCF et sans RTS/CTS. Chaque nœud se déplace sur une surface de simulation suivant un modèle de mobilité de type *random waypoint*. Enfin, chaque nœud a 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 $5\text{m}\cdot\text{s}^{-1}$, 40 nœuds et un degré de 10.

Les résultats détaillés visent à montrer la pertinence et les *bonnes* propriétés de l'organisation proposée. En particulier, nous nous intéressons à l'influence des paramètres $k_{c ds}$ et $k_{cluster}$ sur le nombre de nœuds participant à la vie du réseau et nous montrons la résistance 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. Enfin, nous montrons que la solution d'économie d'énergie est pleinement efficace et justifiée.

Nous avons comparé la structure virtuelle présentée ici (dénommée CDCL pour CDS-Clusters) avec la dorsale proposée par Dai & Wu [19], utilisant la dernière formulation de l'algorithme car la plus efficace, décrite dans [7].

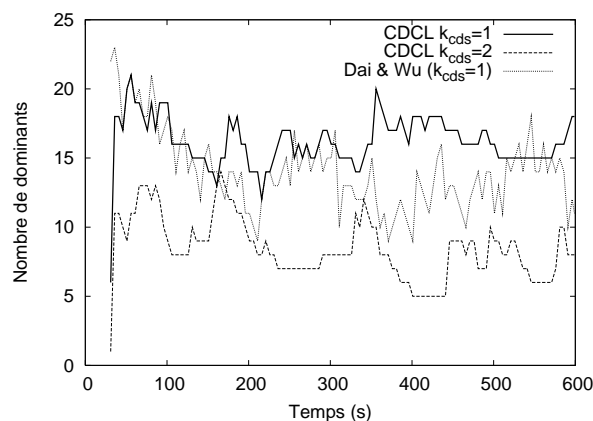


Figure 6. Évolution de la cardinalité de la dorsale au cours du temps
Variations in the backbone cardinality

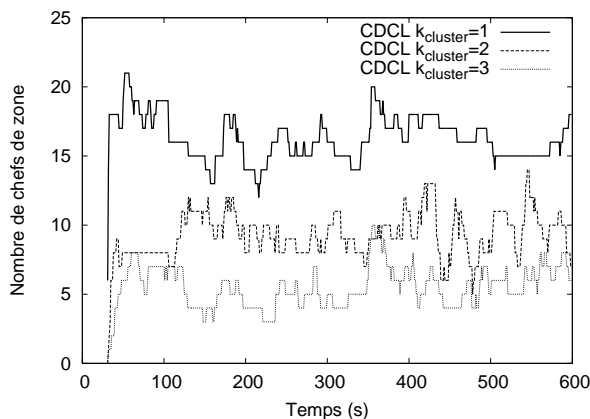


Figure 7. *Évolution du nombre de clusters au cours du temps*
Variations in the number of clusters

Paramétrage 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. 6). Nous pouvons également remarquer que notre algorithme demande sensiblement le même nombre de dominants que Dai & Wu lorsque $k_{cds}=1$, mais demande moins de dominant dès que $k_{cds}>2$. De même, plus le rayon de cluster ($k_{cluster}$) est grand, moins le nombre de chefs de clusters est important (fig. 7).

Degré Nous avons ensuite étudié l'impact du degré sur les performances des structures virtuelles créées. Nous pouvons remarquer que le nombre de dominants nécessaires à la dorsale diminue lorsque le degré augmente : un dominant peut couvrir plus de nœuds (fig. 8). Le nombre de dominants nécessaires lorsque $k_{cds}=2$ est toujours inférieur à celui de Dai & Wu pour le maintien de leur dorsale. La connexité de la dorsale est plus faible à faible densité : le réseau étant éparsé et étendu, les possibilités de reconnexion en cas de cassure de dorsale sont réduites. Nous pouvons par ailleurs remarquer que l'algorithme de Dai & Wu présente une connexité plus faible, visiblement plus sensible aux incohérences dans les tables de voisinage. Cependant, quel que soit l'algorithme, la connexité reste élevée. Nous pouvons noter les mêmes remarques sur les clusters (fig. 9).

Impact de la mobilité Les réseaux ad hoc présentant une topologie changeante, il est important de regarder également l'impact de la mobilité. Nous pouvons remarquer que la cardinalité est très peu influencée par la mobilité (fig. 10). Les mécanismes de suppression de redondance dans la dorsale sont efficaces, la cardinalité reste donc stable. De même, la connexité de la dorsale est assurée, même avec des vitesses éle-

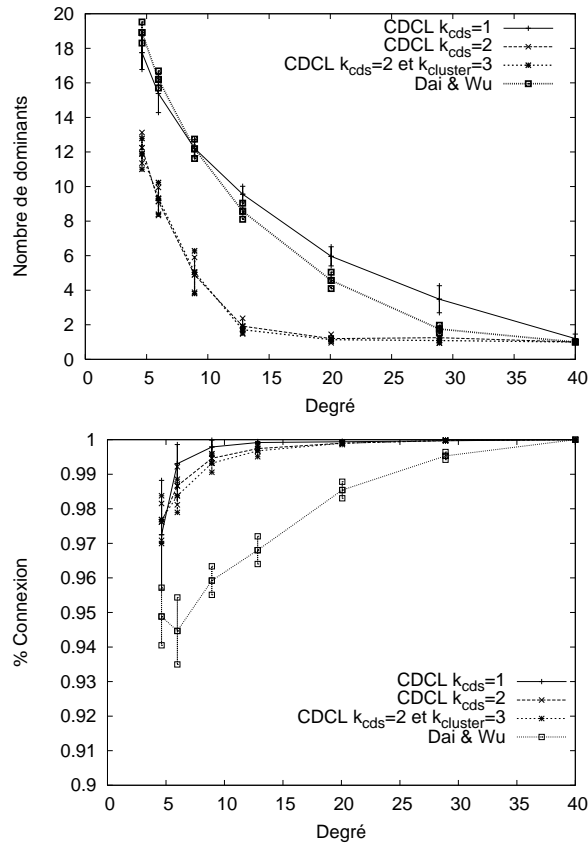


Figure 8. Cardinalité et connexité de la dorsale en fonction du degré
Backbone cardinality and connectivity according to the degree

vées, la dorsale restant à 30m.s^{-1} à 98% connexe, quelle que soit la valeur des paramètres de CDCL : l'inondation d'un paquet de contrôle dédié à la détection de déconnexion de la dorsale est efficace, et permet à un nœud de se reconnecter via un père secondaire immédiatement s'il en possède un. De plus, la reconnexion locale de la dorsale lorsqu'aucune redondance n'existe dans la dorsale est efficace, limitant le temps de convergence. Enfin, la dorsale élit plus de dominants lors d'une forte mobilité principalement à cause d'incohérences des vues de voisinage. Cet accroissement permet d'introduire plus de redondance dans la dorsale, et donc d'être plus robuste à la mobilité. Nous avons ensuite mesuré la persistance des dominants et clusterheads, i.e. le temps moyen pendant lequel ils gardent leur rôle de dominant/clusterhead (fig. 11). Nous pouvons voir que plus le rayon de dorsale est important, plus la persistance décroît : les structures sont plus complexes à maintenir, plus de changements dans la dorsale et les clusters sont requis. Cependant, les persistance des dominants et clus-

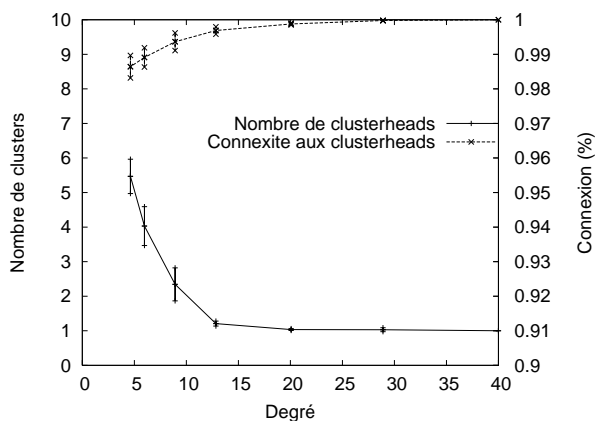


Figure 9. *Impact du degré sur les clusters*
Impact of the degree on the clusters

terheads restent acceptables, même avec des vitesses très élevées. Nous pouvons enfin remarquer que la dorsale de Dai & Wu présente une moins grande persistance, l'algorithme ne tentant pas d'optimiser un tel critère alors qu'il nous semble vital d'avoir une structure aussi stable que possible pour qu'elle puisse remplir son rôle de routage ou d'adressage. La cardinalité des clusters et leur connexité sont également très stables (fig. 12).

Passage à l'échelle horizontal Un réseau ad hoc pouvant comporter de nombreux terminaux, il est important de mesurer l'impact de la cardinalité du réseau (fig. 13). Nous pouvons remarquer que la connexité est très peu influencée par le nombre de nœuds dans le réseau. La cardinalité augmente par contre légèrement lorsque le nombre de nœuds est très grand : la distance entre le leader et un nœud devenant importante, plus d'ap-hellos sont perdus, demandant de nouvelles reconnections, limitant les possibilités de suppression de redondance, et nuisant ainsi à la cardinalité de la dorsale et donc du nombre de chefs de clusters. L'impact de la mobilité sur les clusters, semblable, n'a pas été reporté. Nous pouvons cependant noter que la connexité aux clusters est toujours de 99%, quel que soit le rayon de cluster.

Trafic de contrôle Nous avons également mesuré le trafic de contrôle des différents protocoles et observé leur passage à l'échelle (fig. 14). Nous pouvons vérifier la propriété de localisation parfaite de l'algorithme de [19] : seul l'envoi périodique par chaque nœud d'un hello est nécessaire. Ainsi, le trafic de contrôle de [19] passe parfaitement à l'échelle. Notre structure virtuelle présente un trafic de contrôle plus élevé : des hellos doivent être envoyés pour construire le k_{cds} -voisinage, et des paquets explicites sont requis pour maintenir la connexité de la dorsale et des clusters.

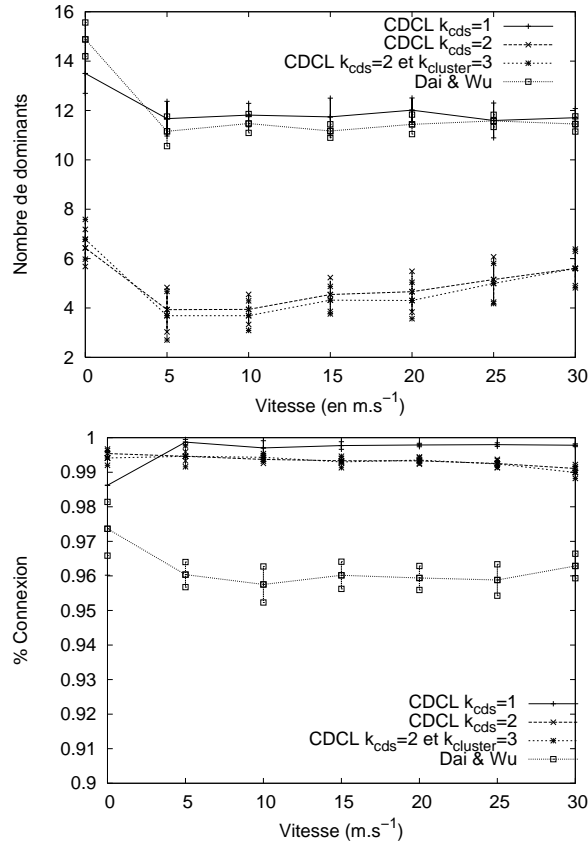


Figure 10. Cardinalité et connexité de la dorsale en fonction de la mobilité
Backbone cardinality and connectivity according to the mobility

Cependant, nous pouvons voir que CDCL passe bien à l'échelle, 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
Dai & Wu [19]	0.25	N/A	N/A	N/A	0.25

Tableau II. Trafic de contrôle par nœud par seconde dans un réseau de 40 nœuds
Overhead per node per second in a network of 40 nodes

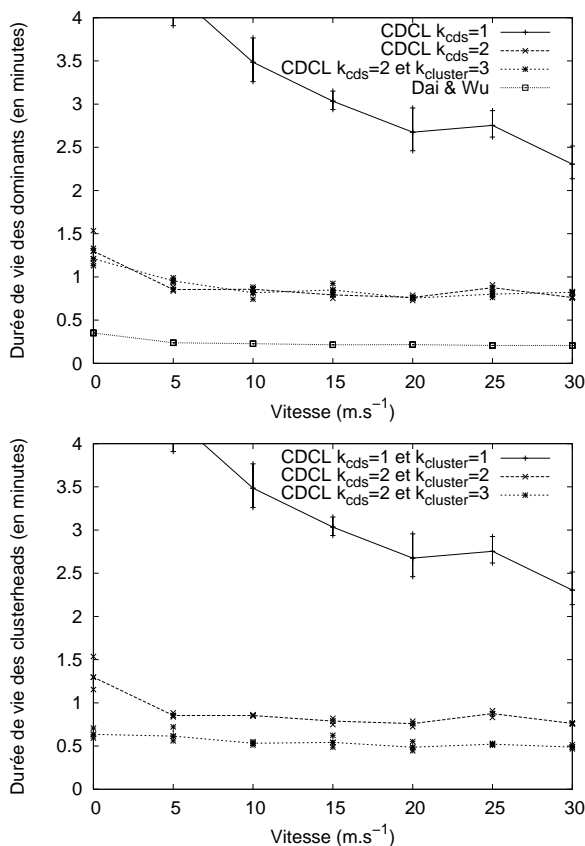


Figure 11. *Persistence des dominants et chefs de cluster selon la mobilité*
Dominators and clusterheads persistence according to the mobility

Le tableau II présente la répartition fine des sources du trafic de contrôle pour un réseau de 40 nœuds. Le trafic de contrôle de Dai & Wu provient uniquement des `hello`s, l'algorithme n'utilisant que des informations locales sur son voisinage. Le trafic de CDCL provient en grande partie lui aussi des `hello`s (près de 50%). Le trafic de contrôle est plus élevé quand $k_{c_{ds}}$ est plus grand : les paquets `hello`s doivent être propagés plus loin : à $k_{c_{ds}}$ sauts durant le déroulement *normal*, et $k_{c_{ds}}+1$ sauts lorsque la source change d'état. 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.

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 che-

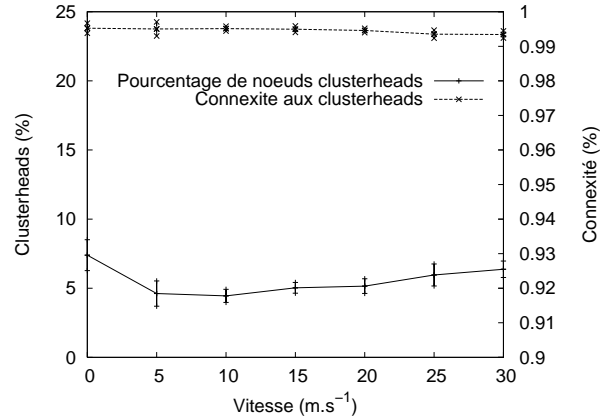


Figure 12. *Impact de la mobilité sur la cardinalité des clusters*
Impact of the mobility on the number of clusterheads

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

Tableau III. *Inondation de la dorsale - Efficacité*
Backbone flooding - Efficiency

mins de connexion entre dorsales. Le taux de livraison de ces paquets de contrôle est très élevé : en moyenne, plus de 97% des nœuds reçoivent un paquet d'inondation donné (tab. III). Le trafic de contrôle généré par une telle inondation est plus élevé avec un mécanisme d'acquiescement lorsque les paquets sont envoyés en unicast via les dominés-connecteurs (tab. IV). Cependant, le trafic de contrôle est bien pris en charge majoritairement par les dominants, ce qui est une des propriétés recherchées. Par exemple, avec $k_{cds}=2$ et sans acquiescement, seulement 16,7 paquets sont nécessaires en moyenne pour inonder la dorsale, alors qu'une inondation aveugle aurait demandé 40 paquets.

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. V). La connexité est maximale avec la métrique de stabilité, qui choisit plus de dominants pour connecter la dorsale. Le nombre de reconnections semble moins important avec une métrique basée sur le degré : un dominant

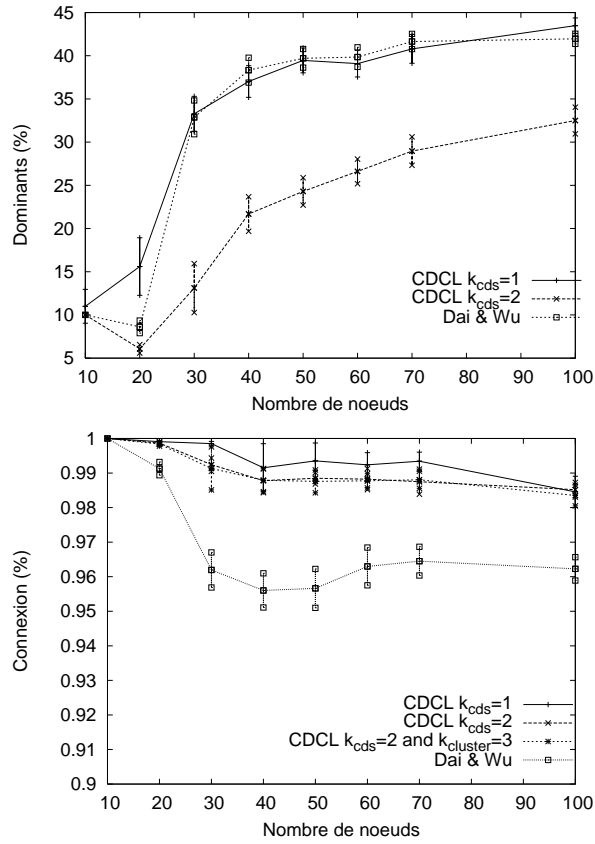


Figure 13. Cardinalité et connexité de la dorsale en fonction du nombre de nœuds
Backbone cardinality and connectivity according to the number of nodes

$k_{c_{ds}}$	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

Tableau IV. Inondation de la dorsale - Trafic de contrôle
Backbone flooding - Overhead

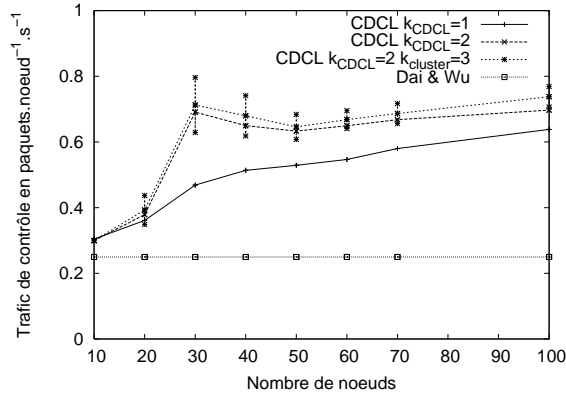


Figure 14. Trafic de contrôle
Overhead

Métrique	CDS				Clusters	
	Connexité	Cardinalité	Nb reconnections	Nb cassures	Connexité	Cardinalité
Identité	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

Tableau V. Performances de la dorsale et des clusters suivant la métrique utilisée
Backbone and clusters performances according to the chosen metric

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.

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. VI). Par contre, 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 élevé avec la métrique de stabilité. 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.

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

Tableau VI. *Performances de la solution d'économie d'énergie*
Performances of the power-energy saving solution

V. Conclusion

Dans cette étude, nous proposons la mise en place d'un mécanisme d'auto-organisation dans le contexte des réseaux ad hoc. L'objectif est de structurer le réseau par opposition aux approches *à plat*. Nous proposons la construction et la maintenance d'une structure virtuelle composée à la fois d'une dorsale permettant de fédérer le trafic de contrôle et de zones de services regroupant logiquement les nœuds proches géographiquement. Les algorithmes de maintenance et de construction étant basés sur un processus d'élection, nous proposons également une nouvelle métrique, dite de *stabilité*, afin de choisir les nœuds les plus pertinents pour jouer le rôle de dominant dans le réseau. L'étude de la structure virtuelle est étendue vers la notion de robustesse à travers l'introduction de plusieurs dorsales dans le réseau ad hoc et de leur interconnexion. Enfin, la séparation des nœuds en dominant/dominé favorise le développement d'une solution d'économie d'énergie très simple.

Les campagnes de simulation permettent de vérifier le bon comportement de cette auto-organisation. En particulier nous avons montré que le taux de connexité de la dorsale dépassait les 98% pour un nombre de dominants compris entre 20% et 40% du nombre de nœuds dans le réseau. Les performances font également apparaître une grande stabilité de la topologie : l'auto-organisation offre une vue hiérarchique du réseau, peu fluctuante au cours du temps. Enfin, l'auto-organisation présente un passage à l'échelle horizontal et vertical élevé.

Ayant montré, dans cet article, l'apport des mécanismes d'auto-organisation, nos travaux vont se tourner vers l'utilisation d'une telle organisation pour améliorer le comportement des réseaux ad hoc. En particulier, nous proposerons une solution de routage tirant partie de cette structure virtuelle pour acheminer les demandes de découvertes de route. De la même façon, l'épine dorsale étant une extension naturelle des réseaux filaires, nous proposerons une solution d'extension des réseaux cellulaires simple sans saut classiques, basée sur la structure virtuelle proposée ici.

Références

- [1] ALZOUBI (K. M.), WAN (P.-J.), FRIEDER (O.), Distributed heuristics for connected dominating set in wireless ad hoc networks, *IEEE ComSoc/KICS*

Journal of Communications and Networks, **4**, pp. 22–29, march 2002.

- [2] AMIS (A.), PRAKASH (R.), VUONG (T.), HUYNH (D.), Max-min d-cluster formation in wireless ad hoc networks, in *INFOCOM*, Tel-Aviv, Israel, pp. 32–41, IEEE, March 1999.
- [3] BANNERJEE (S.), KHULLER (S.), A clustering scheme for hierarchical control in wireless networks, in *INFOCOM*, Anchorage, Alaska, pp. 1028–1037, IEEE, April 2001.
- [4] BASAGNI (S.), Distributed and mobility-adaptive clustering for multimedia support in multi-hop wireless networks, in *Vehicular Technology Conference (VTC Fall)*, Amsterdam, The Netherlands, pp. 889–893, IEEE, September 1999.
- [5] BUTENKO (S.), CHENG (X.), DU (D.-Z.), PARDALOS (P. M.), On the construction of virtual backbone for ad hoc wireless networks, in *Cooperative Control : Models, Applications and Algorithms*, vol. 1 of *Cooperative Systems*, ch. 3, pp. 43–54, Boston, Kluwer Academic Publishers, January 2003.
- [6] CARDEI (M.), CHENG (X.), CHENG (X.), DU (D.-Z.), Connected domination in ad hoc wireless networks, in *International Conference on Computer Science and Informatics (CSI)*, North Carolina, USA, March 2002.
- [7] CARLE (J.), SIMPLOT-RYL (D.), Energy efficient area monitoring by sensor networks, *IEEE Computer Magazine*, **37**, pp. 40–46, February 2004.
- [8] CHATTERJEE (M.), DAS (S. K.), TURGUT (D.), A weight based distributed clustering algorithm for mobile ad hoc networks, in *International Conference on High Performance Computing (HiPC)*, Bangalore, India, pp. 511–521, IEEE, ACM, December 2000.
- [9] CHATTERJEE (M.), DAS (S. K.), TURGUT (D.), Wca : A weighted clustering algorithm for mobile ad hoc networks, *Journal of Cluster Computing (Special Issue on Mobile Ad hoc Networks)*, **5**, pp. 193–204, April 2002.
- [10] CHENG (X.), DU (D.-z.), Virtual backbone-based routing in multihop ad hoc wireless networks, Tech. Rep. 02-002, University of Minnesota, Minnesota, USA, January 2002.
- [11] FEENEY (L.), NILSON (M.), Investigating the energy consumption of a wireless network interface in an ad hoc networking environment, in *INFOCOM*, Anchorage, USA, pp. 1548–1557, IEEE, April 2001.
- [12] LIANG (B.), HAAS (Z. J.), Virtual backbone generation and maintenance in ad hoc network mobility management, in *INFOCOM*, Tel-Aviv, Israel, pp. 1293–1302, IEEE, March 2000.
- [13] LIN (C. R.), GERLA (M.), Adaptive clustering for mobile wireless networks, *IEEE Journal of Selected Areas in Communications*, **15**, pp. 1265–1275, 1997.
- [14] MARATHE (M. V.), BREU (H.), HUNT (S. S.), HARRY B. AND RAVI, ROSENKRANTZ (D. J.), Simple heuristics for unit disk graphs, *Networks*, **25**, pp. 59–68, December 1995.

- [15] NI (S.-Y.), TSENG (Y.-C.), CHEN (Y.-S.), SHEU (J.-P.), The broadcast storm problem in a mobile ad hoc network, in *International Conference on Mobile Computing and Networking (MOBICOM)*, Seattle, USA, pp. 151–162, ACM, August 1999.
- [16] SINHA (P.), SIVAKUMAR (R.), BHARGHAVAN (V.), CEDAR : a core-extraction distributed ad hoc routing algorithm, in *INFOCOM*, New York, USA, pp. 202–209, IEEE, March 1999.
- [17] SIVAKUMAR (R.), DAS (B.), BHARGHAVAN (V.), Spine routing in ad hoc networks, *Cluster Computing*, **1**, pp. 237–248, 1998.
- [18] STOJMENOVIC (I.), SEDDIGH (M.), ZUNIC (J.), Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks, *IEEE Transactions on Parallel and Distributed Systems*, **12**, pp. 14–25, December 2001.
- [19] WU (J.), DAI (F.), Distributed dominant pruning in ad hoc wireless networks, in *International Conference on Communications (ICC)*, Anchorage, USA, pp. 353–357, IEEE, May 2003.
- [20] WU (J.), LI (H.), Dominating-set-based routing in ad hoc wireless networks, in *International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM)*, Seattle, USA, pp. 7–14, ACM, August 1999.