



HAL
open science

Auto-organisation avec canaux et puits multiples de réseaux de capteurs sans fil de grande taille

Cheick-Tidjane Kone, Francis Lepage, Michael David

► **To cite this version:**

Cheick-Tidjane Kone, Francis Lepage, Michael David. Auto-organisation avec canaux et puits multiples de réseaux de capteurs sans fil de grande taille. Sixième Conférence Internationale Francophone d'Automatique, CIFA 2010, Jun 2010, Nancy, France. pp.CDROM. hal-00543084

HAL Id: hal-00543084

<https://hal.science/hal-00543084>

Submitted on 5 Dec 2010

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.

Auto-organisation avec canaux et puits multiples de réseaux de capteurs sans fil de grande taille

Cheick-Tidjane KONE, Francis LEPAGE, Michael DAVID

Centre de Recherche en Automatique de Nancy,
UMR 7039 – Nancy-Université, CNRS
Campus sciences – BP 70239 – 54506 Vandœuvre Cedex, France.

Cheick-Tidjane.Kone@cran.uhp-nancy.fr, Francis.Lepage@cran.uhp-nancy.fr, Michael.David@cran.uhp-nancy.fr

Résumé— Ce papier présente un protocole de *clustering* simple et distribué, approprié pour les réseaux de capteurs sans fil (RCSFs) de grande taille composés de plusieurs milliers de nœuds capteurs. Une architecture à deux niveaux est utilisée pour assurer performance, passage à l'échelle et durabilité : N puits (bien) répartis dans l'espace sont connectés par Wi-Fi en réseau de niveau 2 appelé réseau de puits. Un algorithme d'auto-organisation est proposé pour créer une organisation hiérarchique du réseau de niveau 1 appelé réseau de capteurs. Il partitionne le réseau de capteurs en plusieurs sous-réseaux ou clusters disjoints ayant sensiblement la même profondeur en nombre de sauts. La disponibilité de canaux fréquentiels multiples est ensuite mise à profit pour créer une structure cellulaire en allouant un canal par sous-réseau. L'évaluation de la proposition, effectuée à l'aide de simulations sur un réseau de grande taille, jusqu'à 10 000 nœuds, porte sur le taux de d'intégration ou de connectivité des nœuds dans la structure, sur la charge du trafic, sur les délais et sur le taux de livraison des paquets. Une comparaison avec un réseau de capteurs à un seul canal sur la même topologie montre l'intérêt des canaux multiples.

Mots-clés— Réseaux de capteurs sans fil, grand réseau, auto-organisation, multi-canal, multi-uits, structure cellulaire

I. INTRODUCTION

L'utilisation des réseaux de capteurs sans fil (RCSFs) présente l'intérêt de ne pas nécessiter d'infrastructure. Par exemple, dans des applications de surveillance environnementale après un accident, un grand nombre de capteurs (de l'ordre de quelques centaines de milliers de nœuds) peuvent être déployés rapidement et aléatoirement sur des étendues géographiques de plusieurs kilomètres carrés. Il se pose une problématique de passage à l'échelle mais aussi de préservation énergétique et de connectivité au cours du temps [2], [15], [18]. Pour une utilisation efficace de l'énergie et de la capacité du réseau, plusieurs recherches se sont focalisées sur l'auto-organisation (organisation hiérarchique) [1]. De plus, l'utilisation de plusieurs puits dans les grands RCSFs a été perçue comme une voie possible pour faciliter la gestion et le passage à l'échelle du réseau, raccourcir le délai de transmission en nombre de sauts, améliorer la connectivité et la durée de vie du réseau [?]. De tels RCSFs avec plusieurs puits peuvent être partitionnés en plusieurs sous-réseaux ou clusters avec un puits par cluster. Le nombre et/ou la position des puits maximisant la capacité et la durée de vie du réseau peuvent être optimisés selon les besoins applicatifs [17], [9].

Les canaux multiples peuvent aussi être utilisés pour réduire les collisions et par conséquent accroître la capacité et la durée de vie du réseau. On peut ainsi créer une structure cellu-

laire du RCSF en allouant un canal par cluster tout en respectant le principe de réutilisation de fréquences [16]. Tous les nœuds d'un cluster doivent communiquer sur le même canal.

La contribution principale de ce papier est de proposer une architecture appropriée pour les grands RCSFs (de plusieurs milliers de nœuds) et une méthode pour la construire.. L'architecture utilise plusieurs puits et plusieurs canaux pour améliorer la performance du réseau et faciliter sa gestion et son utilisation. La méthode de construction s'appuie sur un positionnement déterministe des puits, une auto-organisation en clusters à base d'un nouvel algorithme combinée à une allocation de canal par cluster et un basculement des nœuds capteurs sur ce canal. Ensuite, nous évaluons son impact sur la capacité du réseau par rapport à la même architecture mais à canal unique.

Le papier est organisé de la façon suivante. La section II présente brièvement les travaux menés sur la problématique des RCSFs hétérogènes et situe notre approche par rapport à ces travaux. La section III présente le modèle réseau utilisé dans ce papier. La section IV décrit et explique notre algorithme de clustering exploitant plusieurs puits et canaux. La section V évalue par simulation la performance de notre architecture sur les deux approches (mono-canal et multi-canal). Enfin nous concluons ce papier dans la section VI.

II. ETAT DE L'ART ET OBJECTIFS

Cette section résume les travaux menés dans la littérature sur la problématique des RCSFs hétérogènes et situe nos travaux par rapport à ces précédentes recherches.

Les travaux de [12], [7] et [13] se focalisent sur un RCSF multi-uits. Comme le nombre de puits représente le nombre de clusters, ces auteurs étudient la problématique liée au nombre optimal et/ou à la position des puits pour maximiser la couverture et la durée de vie de ce réseau. [12] propose une solution hors-ligne et centralisée pour la détermination du nombre optimal de puits et de leur localisation, tout en ayant une connaissance préalable sur la contrainte de durée de vie du réseau et l'emplacement des nœuds capteurs dans le réseau. Une autre solution basée sur une formulation de programmation linéaire est proposée dans [7]. En utilisant un algorithme centralisé de partitionnement de graphes, les auteurs dans [13] proposent de partitionner le réseau en sous-réseaux de taille relativement comparable en nombre de nœuds, en ayant ici aussi une connaissance préalable sur la position des nœuds capteurs. Ensuite, ils proposent de déployer de façon aléatoire

un puits dans chaque sous-réseau. Chaque puits peut se déplacer dans toute la surface de son cluster.

Pour améliorer aussi la durée de vie du réseau et la couverture radio, un autre type de réseaux de capteurs hétérogènes appelé "Wireless Sensor and Actor Networks (WSAN)" incluant la présence de nœuds "actors" est utilisé dans [1]. Tout comme les puits, les nœuds "actors" collectent les données de nœuds de capteurs. Contrairement aux puits, ils peuvent exécuter certaines tâches spécifiques comme des tâches mécaniques. Les auteurs de ce papier proposent de déterminer le nombre de chefs de cluster ("actor") et leur position grâce à un mécanisme basé sur un "Ensemble dominant indépendant à k sauts (k-IDS)". Ils supposent que les nœuds capteurs et "actors" connaissent leur position grâce à un mécanisme de localisation comme le GPS. Après avoir construit un k-IDS du réseau, chaque nœud "actor" doit bouger pour se placer à côté d'un nœud dominant du k-IDS. Notons que ce processus est très coûteux en termes d'échange de messages, de bande passante et d'énergie.

Nous remarquons que les principaux objectifs des approches énumérées ci-dessus sont de trouver le nombre optimal de puits et/ou leur localisation qui maximisent la durée de vie du réseau. De plus, ces approches sont centralisées et un système de localisation comme le GPS consomme beaucoup d'énergie. Un tel système est incompatible financièrement avec des réseaux de capteurs étendus composés de quelques milliers de nœuds capteurs. Notons que la plupart de ces approches sont testées sur des RCSFs de petites ou de moyennes tailles (moins de 1000 nœuds capteurs). Notons aussi qu'aucun papier à notre connaissance ne décrit un test pour des RCSFs de grandes tailles (au moins de 10000 nœuds capteurs).

Notre approche est différente des précédentes, car notre objectif est de renforcer de manière distribuée la capacité et la durée de vie du réseau indépendamment de la position des nœuds capteurs et puits. Ainsi, nous proposons un algorithme de clustering qui réduit considérablement la charge du trafic dans les grands RCSFs (de l'ordre de quelques dizaines de milliers de nœuds). Comme les collisions et les retransmissions sont également responsables de la perte d'énergie dans un RCSF, tout comme la périodicité et la taille des messages, nous proposons de combiner à notre algorithme de clustering l'utilisation de plusieurs canaux radios pour renforcer significativement la capacité du réseau [14]. On pourra ainsi créer un système cellulaire qui alloue un canal de transmission par cluster, tout en respectant le principe des motifs cellulaires [5], [16]. Tous les nœuds d'un cluster doivent communiquer sur le même canal. Notons qu'un seul émetteur-récepteur radio peut suffire pour utiliser dynamiquement plusieurs canaux radios [4] et que le délai pour changer de canal n'a pas d'incidence sur la capacité du réseau [8]. Par exemple, la latence d'un émetteur-récepteur radio CC2420 pour basculer d'un canal à un autre est d'environ 200 μ s.

III. MODELE RESEAU ET HYPOTHESES

Nous considérons un RCSF hétérogène à 2 niveaux comme illustré dans la figure 1. Le réseau de niveau 1 appelé réseau de capteurs fonctionne sur des canaux de fréquences du standard 802.15.4. Notons que les nœuds du niveau 1 utilisent seulement la couche physique de ce standard. Pour le mécanisme de transmission (couche MAC), nous utilisons un

CSMA/CA classique c'est-à-dire "non slotté". Le réseau de niveau 2 appelé réseau de puits fonctionne sur un unique canal du standard 802.11. Les nœuds du niveau 2 implémentent toute la pile protocolaire du standard 802.11. Tous les émetteurs radios du réseau de niveau 1 ont la même portée radio. Contrairement aux nœuds capteurs, nous supposons que les nœuds de puits ne sont pas limités en ressources énergétiques et possèdent deux couples émetteurs-récepteurs radios : un pour communiquer avec le réseau de niveau 1 et l'autre pour communiquer avec le réseau de niveau 2. Nous supposons que les nœuds capteurs sont déployés aléatoirement dans une zone de surveillance pour détecter certains événements physiques et les transmettre à un utilisateur final à travers le réseau de niveau 2. Pour des raisons de simplicité, les puits sont supposés être placés manuellement sur la zone de surveillance tout en formant un réseau de puits connectés. Nous supposons que le réseau de puits ou de niveau 2 utilise un protocole de routage MANET (par exemple AODV, OLSR, DSR, ...) pour router les événements physiques détectés par des nœuds capteurs du réseau de niveau 1 vers un utilisateur final. Nous supposons aussi que tous les nœuds dans le réseau global sont statiques. Chaque puits peut construire un cluster d'au plus k sauts. Notons que le canal Wi-Fi utilisé par le réseau de niveau 2 est choisi de tel sorte qu'il n'interfère pas avec les canaux 802.15.4 utilisés dans le réseau de niveau 1 [3]. Pour pouvoir utiliser plusieurs canaux radios, nous supposons qu'au déploiement des nœuds, seuls les puits connaissent le canal C_x qu'ils devront allouer à leur cluster juste après leur formation. Ainsi, on aura des communications intra-clusters dissociées (voir Figure 1). Notons que 4 canaux 802.15.4 (C_x) suffisent pour créer pour un motif et faciliter la réutilisation des canaux de fréquences.

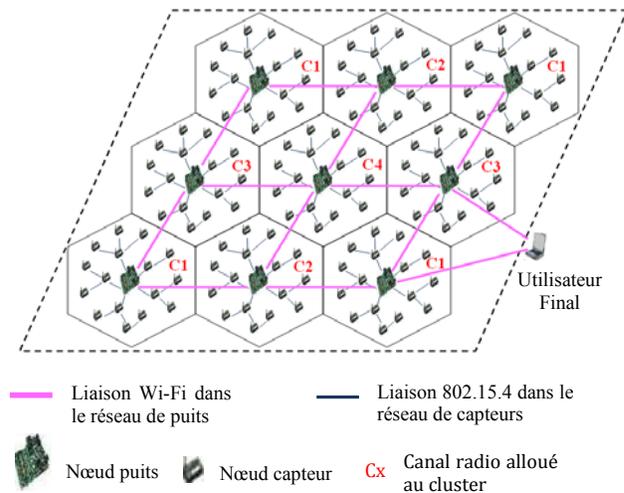


Fig. 1. Topologie proposée dans un RCSF à plusieurs puits exploitant plusieurs canaux (un motif de 4 canaux).

IV. NOTRE CONTRIBUTION

A. Définition des règles de sélection du parent

Considérons les notations suivantes. $1-Densité(u)$ ou $Densité(u)$ représente la densité de liens du nœud u , i.e., le ratio du nombre de liens par le nombre de nœuds dans le voisinage à 1 saut du nœud u . Nous avons choisi cette métrique de densité de liens parce que les auteurs dans [10] ont montré

qu'elle offre une meilleure stabilité de la structure virtuelle que d'autres métriques (degré, identifiant) face aux petits changements topologiques dans le voisinage d'un nœud. $Energie(u)$ représente le taux d'énergie consommée du nœud u , i.e., le ratio de l'énergie consommée par l'énergie initiale du nœud u . Le paramètre λ , définissant un seuil pour le taux d'énergie consommée du nœud, a une valeur comprise entre 0 et 1. $Hops(u)$ désigne la distance en nombre de sauts entre le nœud u et son chef de cluster (un puits). $Poids(u)$ définit le poids de stabilité du nœud u . La fonction $Poids$ garantit que deux nœuds ont différents poids. Supposons que deux nœuds v et w soient les voisins à un saut d'un nœud u et que l'un des deux est actuellement le père du nœud u . Le nœud u choisira comme nouveau père entre les deux nœuds v et w , celui ayant le poids le plus élevé. Par exemple le nœud u choisira le nœud v comme père si $Poids(v) > Poids(w)$. C'est-à-dire si :

- $Energie(v) \leq \lambda \wedge Energie(w) > \lambda, \vee$
- $Energie(v) \leq \lambda \wedge Energie(w) \leq \lambda \wedge Hops(v) < Hops(w), \vee$
- $Energie(v) \leq \lambda \wedge Energie(w) \leq \lambda \wedge Hops(v) = Hops(w) \wedge Densité(v) > Densité(w), \vee$
- $Energie(v) \leq \lambda \wedge Energie(w) \leq \lambda \wedge Hops(v) = Hops(w) \wedge Densité(v) = Densité(w) \wedge$ (le nœud v a été le père du nœud u tout juste avant que le nœud u exécute cette procédure de sélection de père), \vee
- $Energie(v) > \lambda \wedge Energie(w) > \lambda \wedge$ (le nœud v a été le père du nœud u tout juste avant que le nœud u exécute cette procédure de sélection du père)

B. Construction d'une structure virtuelle

Nous proposons un algorithme de clustering adapté et efficace pour les RCSF de grande taille avec puits et des canaux multiples. Pour construire les clusters du réseau, nous définissons quatre états pour un nœud :

- *Ordinaire* : c'est l'état initial d'un nœud en attente d'un signal ou d'un nœud déconnecté d'un cluster.
- *Leader* : C'est l'état d'un puits initiant la construction de son cluster. Il s'agit de la racine ou du chef d'un cluster.
- *Membre* : nœud intermédiaire entre la racine et les feuilles d'un arbre de cluster.
- *Gateway* : nœud intermédiaire entre les clusters.

L'algorithme de clustering proposé construit des arbres de cluster à k sauts (la distance en nombre de sauts entre un nœud et son chef dans un cluster est d'au plus k sauts). L'algorithme procède par vagues successives à travers des messages spécifiques. Nous utilisons seulement quatre types de messages de diffusion : *CLUSTER_INITIATE*, *CLUSTER_ACCEPT*, *CLUSTER_FINISH* et *HELLO*. Notre heuristique fonctionne comme suit :

Premièrement, chaque nœud doit découvrir son voisinage afin de calculer sa propre métrique (comme la densité de liens [10]) et maintenir dans sa table de voisinage les informations (identifiant, métriques, etc.) de ses voisins à un saut nécessaire à la construction du cluster. Cette découverte est réalisée grâce à un envoi périodique de messages *HELLO*. Le message *HELLO* a un entête de 8 octets.

Après un délai, chaque puits ou leader initie simultanément la construction de son cluster en diffusant à un saut un mes-

sage *CLUSTER_INITIATE* servant à inviter ses voisins à le rejoindre dans le réseau dans lequel il est le chef (voir Fig. 2.1). Ensuite, il arme un temporisateur pour une retransmission de ce message si nécessaire. Notons que ce message *CLUSTER_INITIATE* a une taille de 11 octets et contient les informations sur le puits (son identifiant représentant aussi l'identifiant du cluster, son état, le canal alloué au cluster).

Dès la réception d'un message *CLUSTER_INITIATE* en provenance d'un puits, un nœud capteur *ordinaire* stocke ou met à jour les informations de ce puits dans sa table de voisinage. Il le choisit comme son père. Il devient un nœud *membre* et diffuse ensuite un message *CLUSTER_ACCEPT* à un saut pour avertir d'une part le puits de son adhésion au cluster et d'autre part pour inviter ses voisins à le rejoindre dans le cluster s'ils ne sont pas déjà intégrés (voir Fig. 2.2). Notons qu'une retransmission du message *CLUSTER_INITIATE* est faite par le puits lorsque celui-ci n'a reçu aucun message *CLUSTER_ACCEPT* au bout d'un temporisateur qu'il a armé au préalable. Le message *CLUSTER_ACCEPT* a une taille de 18 octets, il contient les informations sur la source de ce message (identifiant, état, identifiant du cluster, canal alloué au cluster, distance en nombre de saut du chef de cluster, taux d'énergie consommée).

A la réception d'un message *CLUSTER_ACCEPT*, un nœud (puits ou capteur) exécute une procédure selon son état :

- Si c'est un puits alors il met à jour dans sa table de voisinage les informations du nœud transmettant ce message. Ensuite, il stoppe la retransmission du message *CLUSTER_INITIATE* et affecte physiquement à son émetteur-récepteur le canal radio alloué à son cluster s'il ne l'a pas encore fait (voir Fig. 2.3).
- Si c'est un nœud capteur *membre* ou *gateway* alors :
 - il met à jour dans sa table de voisinage les informations du nœud transmettant ce message et choisit comme père le nœud possédant le meilleur poids entre ce nœud et son actuel père ;
 - il stoppe la retransmission de son message *CLUSTER_ACCEPT* et affecte physiquement à son émetteur-récepteur le canal radio alloué au cluster s'il ne l'a pas encore fait et s'il est plus proche du chef de cluster que le nœud transmettant ce message (voir Fig. 2.4).
- Si c'est un nœud capteur *ordinaire* alors :
 - il met à jour dans sa table de voisinage les informations du nœud transmettant ce message et après un temporisateur il choisit comme père parmi ses nœuds voisins *membres*, le nœud possédant le meilleur poids ;
 - si il avait reçu des messages *CLUSTER_ACCEPT* d'au moins deux nœuds *membres* de clusters différents alors il devient un nœud *gateway* et diffuse un message *CLUSTER_FINISH* (voir Fig. 2.5) sinon il devient un nœud *membre* et diffuse un message *CLUSTER_ACCEPT* pour notifier son adhésion à un cluster et inviter par ailleurs ses nœuds voisins ordinaires à le rejoindre dans ce cluster (voir Fig. 2.3).

Le message *CLUSTER_FINISH* contient les informations similaires au message *CLUSTER_ACCEPT*. Il est seulement transmis par un nœud *gateway* pour notifier son adhésion à un cluster et finaliser une partie de l'arbre du cluster.

A la réception d'un message *CLUSTER_FINISH*, un nœud exécute une procédure spécifique à son état. Par exemple :

- Si c'est un puits ou leader alors il met à jour dans sa table de voisinage les informations du nœud transmettant ce message.
- Si c'est un nœud capteur *membre* ou *gateway* alors :
 - il met à jour dans sa table de voisinage les informations du nœud transmettant ce message et stoppe selon son état la retransmission du message *CLUSTER_ACCEPT* ou *CLUSTER_FINISH* qu'il avait armé au préalable ;
 - il affecte physiquement à son émetteur-récepteur le canal radio alloué à son cluster s'il ne l'a pas encore fait (voir Fig. 2.3).

Nous pouvons remarquer sur la figure 2.7 que quelques nœuds appartenant à des clusters n'ont pas pu affecter physiquement à leur émetteur-récepteur radio le canal alloué au cluster auquel ils appartiennent par faute de réception de messages leur permettant de le faire. Donc ces nœuds se déconnecteront du cluster par la suite car ils ne pourront plus recevoir d'informations de leur père qui communique désormais sur le canal du cluster. Comme ces nœuds connaissent leur cluster et le canal du cluster, nous proposons que ces nœuds avant de se déconnecter de la structure affectent physiquement à leur émetteur-récepteur le canal du cluster (voir Fig. 2.8). Ainsi, ces nœuds pourront se reconnecter au cluster suite à une réception de messages *HELLO* (voir Fig. 2.9).

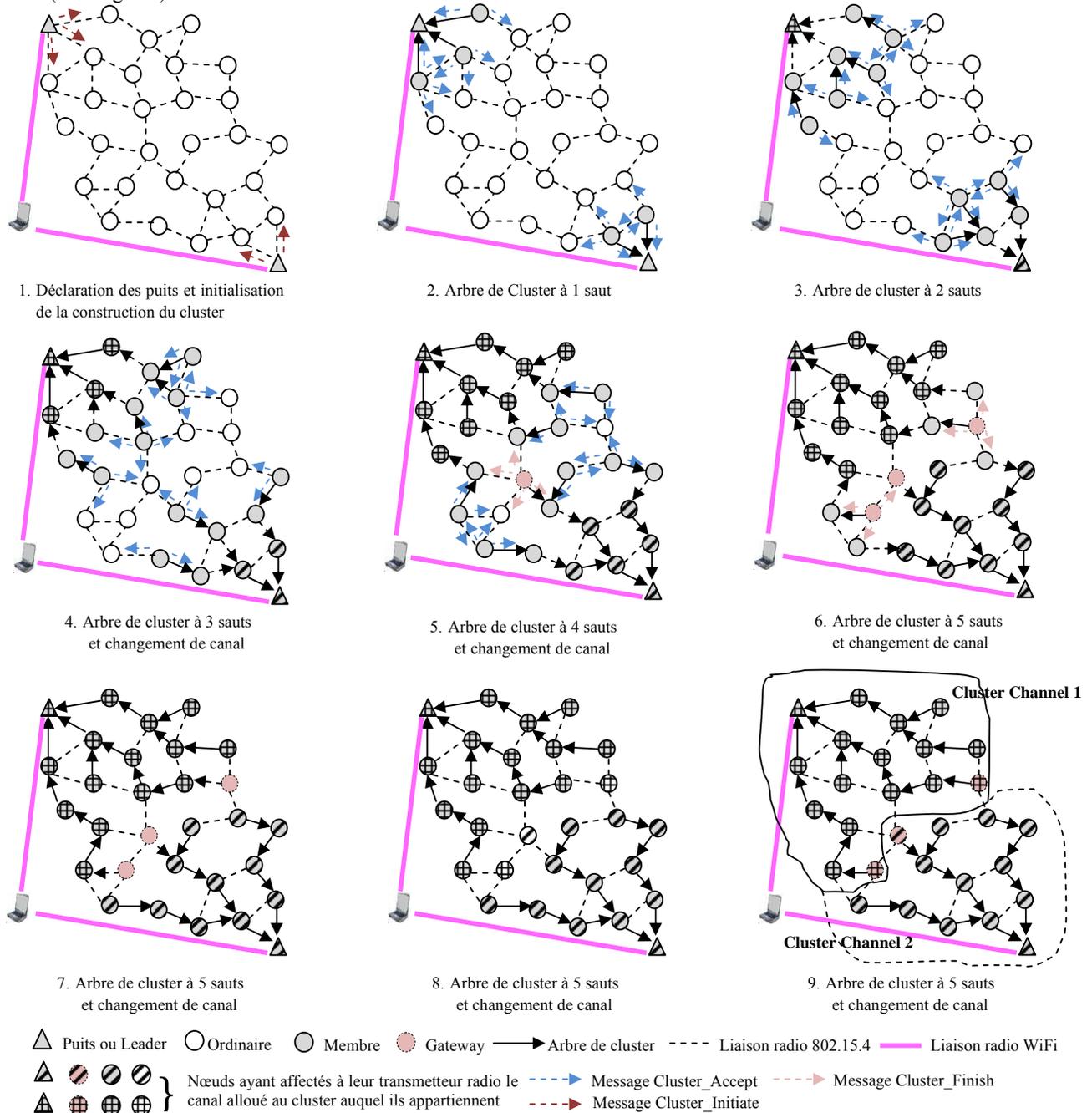


Fig. 2. Construction de clusters avec plusieurs canaux (exemple avec 2 puits et $k = 5$)

A. Paramètres de simulation

Nous évaluons les performances de notre algorithme sur les deux stratégies (mono-canal et multi-canal) en fonction de la taille du réseau dans un simulateur appelé OPNET Modeler Wireless Suite [11]. Le réseau général est composé, comme indiqué sur la figure 1: du réseau de niveau 1 ou réseau de capteurs et du réseau de niveau 2 ou réseau puits. Dans le réseau de niveau 1, nous utilisons une couche physique définie dans le standard IEEE 802.15.4-2003. Cette couche opère sur une bande de fréquences autour de 2.4 GHz avec un débit de 250 kbit/s et une portée radio de 50 mètres. Rappelons que la couche MAC dans le réseau de niveau 1 implémente le protocole CSMA/CA classique (i.e. CSMA/CA non slotté) et non celui du standard IEEE 802.15.4-2003. Nous utilisons un modèle d'estimation de la consommation énergétique proposé dans [6] et basé sur les spécifications du capteur MicaZ de la compagnie Crossbow. Le réseau de niveau 2 implémente toute la pile protocolaire du standard IEEE 802.11g-2003 avec un débit de 1 Mbit/s en mode DCF, une portée de 500 mètres et OLSR comme protocole de routage. Notons que nous utilisons seulement 4 canaux de fréquences 802.15.4 pour créer un motif et faciliter la réutilisation de ces canaux pour des communications intra-cluster dissociées.

Le réseau de niveau 1 est composé de nœuds capteurs statiques (entre 1000 et 10000) aléatoirement déployés sur une surface de 1000 x 1000 mètres. Le réseau de niveau 2 est composé d'un utilisateur final et de 9 puits judicieusement placés sur cette même surface. Nous supposons qu'au déploiement, seuls les puits connaissent le canal alloué à leur cluster sur lequel devront basculer tous les nœuds lui appartenant (voir Figure 1). Chaque puits peut construire un cluster jusqu'à 5 sauts. Pour réduire la congestion du réseau, nous supposons un temps de traversée maximal des paquets de 2s dans la file d'attente de la couche MAC des nœuds capteurs. Ceci permet de supprimer les paquets qui pourraient être retardés. Le temporisateur pour qu'un nœud passe d'un état à un autre est de 5s. Le temps d'exécution et la superficie de déploiement sont respectivement de 7200s et de 1000 m x 1000 m pour toutes les simulations. Les messages HELLO sont diffusés périodiquement dans un intervalle de temps $[f_{HELLO} - 5, f_{HELLO} + 5]$ s avec $f_{HELLO} = 60$ s. La durée de vie maximale d'un nœud dans la table de voisinage d'un autre nœud est 180s. La table de voisinage d'un nœud est vérifiée chaque 35s pour supprimer les voisins invalides ou morts. Les résultats sont obtenus dans un intervalle de confiance de 95%.

B. Résultats

Nous étudions la performance de notre algorithme sur les deux approches : mono-canal et multi-canal. Rappelons que dans l'approche mono-canal, les transmetteurs radios de tous les nœuds capteurs restent sur le même canal de fréquence initial. Ce n'est pas le cas de l'approche multi-canal où tous les nœuds capteurs sont capables de changer physiquement de canaux après une affectation initiale dans le but de communiquer sur le canal alloué au cluster auxquels ils appartiennent.

La figure 3 représente le trafic de contrôle moyen en bits/s reçu par la couche réseau du nœud capteur en fonction de la taille du réseau de niveau 1. Le trafic de contrôle est composé de messages *CLUSTER_INITIATE*, *CLUSTER_ACCEPT*,

CLUSTER_FINISH et *HELLO*. Nous observons sur cette figure que le trafic de contrôle reçu par chaque nœud capteur reste acceptable dans les deux stratégies même avec une augmentation de taille du réseau. Néanmoins, ce trafic reçu par chaque nœud capteur est plus faible (environ 18% de moins) dans l'approche multi-canal que dans celui du mono-canal. La segmentation en plusieurs canaux fait que de manière générale un nœud possède moins de voisins (en moyenne 12%) de moins qu'un nœud dans une approche mono-canal selon nos simulations et donc reçoit moins de paquets.

La figure 4 représente l'énergie dépensée en moyenne par heure par un nœud capteur dans les deux approches. Nous pouvons observer que l'approche multi-canal contribue à étendre la durée de vie du réseau car un nœud de cette approche consomme en moyenne 7% d'énergie de moins que celui de l'approche mono-canal. Cela est dû au fait qu'un nœud dans l'approche multi-canal reçoit moins de paquets comme indiqué sur la figure 3.

Nous pouvons constater sur la figure 5 que le temps d'accès au médium est approximativement similaire dans les deux approches lorsque la taille du réseau est petite. Par contre, une différence est observée pour de plus grandes tailles à cause des mêmes raisons mentionnées ci-dessus comme le trafic de contrôle reçu (Figure 3) et le degré moyen d'un nœud. En effet, plus le degré moyen d'un nœud est élevé, plus le temps pour accéder au médium l'est aussi. Le temps pour accéder au médium varie très peu dans l'approche multi-canal même avec une augmentation de la taille du réseau, tandis qu'il croît rapidement dans l'approche mono-canal. Ces résultats montrent clairement le bénéfice d'utiliser un système multi-canal dans des réseaux étendus d'autant plus qu'il réduit le nombre de messages échangés sur le réseau, mais aussi la consommation énergétique et l'occupation de la bande passante.

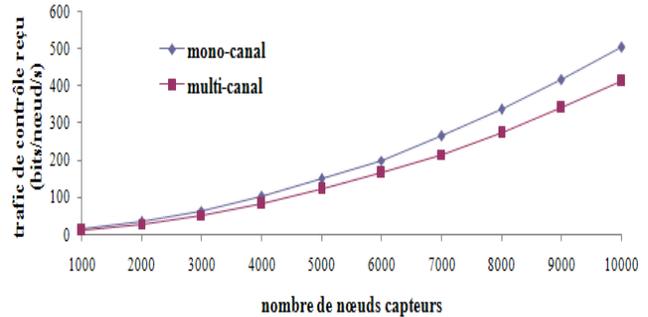


Fig. 3. Trafic de contrôle moyen reçu par nœud capteurs par seconde en fonction de la taille du réseau : mono-canal vs multi-canal

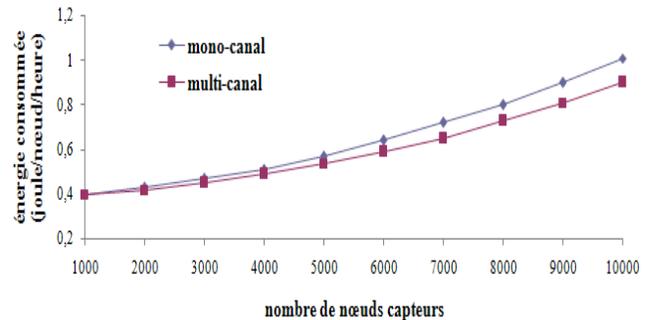


Fig. 4. Energie consommée en moyenne par nœud capteur par heure en fonction de la taille de réseau : mono-canal vs multi-canal

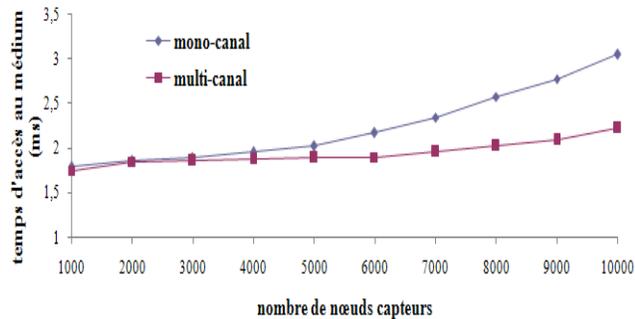


Fig. 5. Temps d'accès au médium en fonction de la taille du réseau : mono-canal vs multi-canal

Pour estimer le délai de bout-en-bout et le taux de livraison de paquets, nous considérons uniquement le réseau composé de 10000 nœuds capteurs, 9 nœuds puits et un utilisateur final. Nous choisissons aléatoirement des nœuds sources (entre 200 et 1000) parmi les 10000 nœuds capteurs, afin de transmettre indépendamment leurs données applicatives. Comme nous supposons être dans une application de surveillance environnementale, le processus de génération de paquets implémenté indépendamment sur chaque nœud source suit une loi exponentielle de moyenne 300 secondes. Nous avons opté pour une taille de donnée applicative de 4 octets adaptée à la majorité des phénomènes physiques (température, pression, accélération, etc.) détectés par les capteurs. La figure 6 illustre le délai de bout en bout moyen mesuré dans les deux approches en fonction du nombre de nœuds sources. Ce délai représente la durée moyenne de l'ensemble du trafic applicatif transmis à partir des nœuds sources jusqu'à un utilisateur final via les réseaux de niveau 1 et 2. Nous pouvons clairement voir sur cette figure que l'approche multi-canal garantit un plus faible délai de bout en bout (environ 2 ms de moins quelque soit le nombre de nœuds sources) que celui du mono-canal parce qu'il assure un plus faible temps d'accès au médium (Figure 5). Cela montre une fois de plus les avantages d'une approche multi-canal par rapport au mono-canal lorsque la taille du réseau est élevée. Toutefois, les performances globales observées dans les deux approches sont acceptables pour un tel RCSF (10000 nœuds capteurs), car ils garantissent un faible délai de bout en bout (en dessous de 12ms) et un taux de livraison de donnée de plus de 93%.

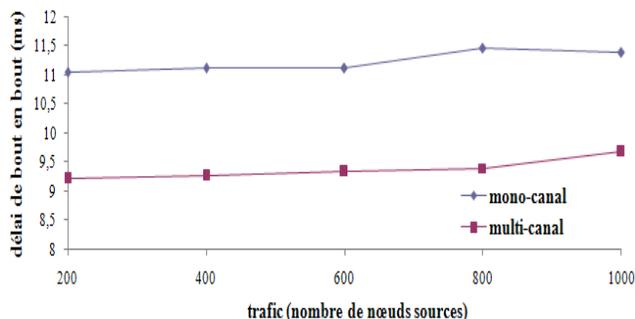


Fig. 6. Délai de bout-en-bout moyen dans un RCSF de 10000 capteurs, 9 puits et un utilisateur final : mono-canal vs multi-canal

VI. CONCLUSION ET PERSPECTIVES

Dans ce papier, nous avons proposé une organisation en clusters simple, distribuée et appropriée pour les RCSFs de grandes tailles (notre cible était de 10000 nœuds capteurs).

Cette organisation utilise plusieurs canaux radios pour créer une structure cellulaire en affectant un seul canal par cluster. Nous avons évalué l'impact des deux stratégies (mono-canal et multi-canal) sur la capacité du réseau. Les résultats de simulation ont montré que le schéma multi-canal améliore les performances du réseau en terme de charge du réseau, de consommation énergétique, de délais de transmission, etc.

Comme perspectives, nous allons réaliser une étude théorique pour déterminer les limites de fonctionnement de tels grands réseaux suivant le mode et la fréquence de génération des messages. Nous planifions aussi de proposer un algorithme d'allocation dynamique du canal aux puits, tout en respectant le principe du motif cellulaire, lorsque ceux-ci sont déployés aléatoirement.

REMERCIEMENTS

Ce travail a été soutenu par OPNET Technologies, Inc. à travers le programme "Teaching with OPNET".

RÉFÉRENCES

- [1] Akkaya K., Senel F. et McLaughlan B. Clustering of wireless sensor and actor networks based sensor distribution and connectivity. *J. Parallel Distrib. Comput.* 69, 6, pp. 573-587, June 2009.
- [2] Akyildiz I. F., Y. Sankarasubramaniam W. Su. et Cayirici E. Wireless sensor networks : a survey. *Computer Networks (Elsevier)*, vol. 38(4), pp. 393-422, March. 2002.
- [3] Crossbow Technology, Inc. MICAz-Based Zigbee and WiFi Coexistence. Techreport, 2004.
- [4] Darties B., Theoleyre F. et Duda A. A Divide-and-Conquer Scheme for Assigning Roles in Multi-Channel Wireless Mesh Networks. In : Proc. of IEEE LCN (Local Computer Networks), Zurich, Switzerland, October 23-23, 2009.
- [5] Katzela I. et Naghshineh M. Channel assignment schemes for cellular mobile telecommunication systems : a comprehensive survey. *IEEE Personal Communications*, 3, pp. 10-31, 1996.
- [6] IEEE 802.15.4 OPNET Simulation Model, <http://www.open-zb.net>
- [7] Kim H., Seok Y., Choi Y. and Kwon T. Optimal Multi-sink Positioning and Energy-efficient Routing in Wireless Sensor Networks. In : Kim, C. (ed.) *ICOIN 2005. LNCS*, vol. 3391, pp. 264-274. Springer, Heidelberg, 2005.
- [8] Kyasanur P. et Vaidya N. H. Capacity of multichannel wireless networks under the protocol model. *IEEE/ACM Trans. Netw., IEEE Press*, 17, pp. 515-527, 2009.
- [9] McLaughlan B. et Akkaya K. Coverage-based clustering of wireless sensor and actor networks. In Proc. of IEEE International Conference on Pervasive Services, ICPS'07, Istanbul, Turkey, July 2007.
- [10] Mitton N., Busson A. et Fleury E. Self-organization in large scale ad hoc networks. In Third Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), Brodum, Turkey, June 2004.
- [11] OPNET Technologies, Inc., *Opnet Modeler Wireless Suite* - ver. 15.0.A PL1, <http://www.opnet.com>
- [12] Oyman E. I. et Ersoy C. (2004). Multiple sink network design problem in large scale wireless sensor network networks. In : Proc. Of ICC, Paris, June 2004.
- [13] Slama I., Jouaber B. et Zeghlache D. Multiple Mobile Sinks Deployment for Energy Efficiency in Large Scale Wireless Sensor Network. International joint Conference on e-Business and Telecommunications, ICETE, Porto, Portugal, July 26-29, 2008.
- [14] Vassis D., Kormentzas G. et Skianis C. Performance evaluation of single and multi-channel actor communication for wireless sensor actor networks. *Ad Hoc Networks*, 4, pp. 487-498, 2006.
- [15] Verdore R., Dardari D., Mazzini G. et Conti A. *Wireless Sensor and Actuator Networks : Technologies, Analysis and Design*. Academic Press.
- [16] Wang X. et Berger T. Spatial channel reuse in wireless sensor networks. *Wirel. Netw.*, 14, 2, pp. 133-146, Mar. 2008.
- [17] Yarvis M., Kushalnagar N., Singh H., Rangarajan A., Liu Y. and Singh S. Exploiting Heterogeneity in Sensor Networks. *IEEE INFOCOM*, 2005.
- [18] Yick J., Mukherjee B. et Ghosal D. Wireless sensor network survey. *Computer Networks (Elsevier)*, vol. 52, n° 12, pp. 2292-2330, Aug. 2008.