

Regroupement économe en énergie par vote de capteurs

Michel Chilowicz et Ibrahim El Otmani

1 Introduction

Diverses approches peuvent être envisagées pour réaliser la dissémination d'informations dans un réseau de capteurs :

- Le stockage des données sur le réseau de capteurs, chaque donnée pouvant être conservée sur le capteur qui l'a produite ou bien hébergée par des capteurs tiers (avec ou sans réplique) et accessible au moyen d'une table de hachage distribuée. La récupération de la donnée nécessite une requête de demande auprès du capteur hébergeur.
- L'envoi de données vers l'extérieur du réseau de capteurs (un puits) : une réponse naïve consiste à établir une communication unidirectionnelle directe entre chaque capteur et le puits. Cette solution de communication directe avec un serveur central induit un coût énergétique de transmission excessif ; de plus la portée limitée de communication des capteurs et l'éloignement du puits peut rendre cette solution impossible. Pour résoudre ces problèmes, on s'intéresse alors au regroupement des capteurs autour de routeurs.

Les algorithmes de regroupement (*clustering*) pour un réseau de capteurs permettent la constitution de grappes (*clusters*) de capteurs, chacun étant dominé par un routeur élu. Le regroupement autour d'un routeur s'impose naturellement afin de limiter la consommation d'énergie des capteurs : les capteurs affiliés au routeur transmettent leurs informations et celles-ci sont relayées par le routeur vers l'extérieur. Le nombre de messages envoyés (par fusion et compression¹ [5] des informations des capteurs) ainsi que les portées de transmission sont ainsi réduits.

Nous nous intéresserons tout particulièrement ici à un article de *Qin et Zimmermann* [8] proposant un algorithme de regroupement par vote de capteurs avec pour principal souci l'économie d'énergie.

2 Regroupement avec minimisation d'énergie consommée

2.1 Données et introduction au problème

Soit un réseau de capteurs c_1, \dots, c_n : on caractérise ce réseau par les données suivantes :

- Une matrice de coût de transmission entre capteurs C où $c_{i,j}$ est le coût de transmission du capteur i vers le j (pouvant s'exprimer en J/bit). On considère le coût de transmission $c_{i,j}$ infini si le capteur i est hors de portée du capteur j . On note que la matrice de coûts est idéalement symétrique ; cependant dans la pratique, il est courant de rencontrer des cas de dissymétries de communication entre noeuds.

Le coût de communication entre deux capteurs i et j est déduit de la distance $d_{i,j}$ séparant

¹Nous ne nous intéressons pas ici à la problématique de la fusion d'informations provenant de capteur qui est très dépendante de l'application envisagée. Par exemple, si un réseau de capteurs a pour tâche de mesurer la température de différents points d'un bâtiment, l'agrégation et la fusion peuvent consister à ne retenir que les températures minimales, maximales et la moyenne des capteurs voisins.

les capteurs : il est possible d'utiliser un modèle d'amplification du signal en d^2 ².

- Un vecteur de débit d'émission \vec{d} indiquant pour chaque capteur la quantité d'information émise par unité de temps (en bit/s). Par souci de simplification, on considère souvent que la fréquence et la quantité d'information émise par tous les capteurs sont similaires : ceci est cependant dans la pratique rarement vrai.
- Un coût énergétique de réception de données E_{Rx} (exprimable physiquement en J/bit).
- Un vecteur coût de transmission $\overleftarrow{e}_{puits}$ d'un capteur (noeud maître) vers le puits. Ces coûts de transmission sont généralement plus élevés que les coûts de communication entre capteurs.
- Un vecteur d'énergie résiduelle des capteurs \overleftarrow{e} : e_i est l'énergie résiduelle du capteur i (s'exprimant en joules).

On cherche à réaliser un regroupement des capteurs autour de noeuds maîtres communiquant avec le puits afin de maximiser la durée de vie du réseau, i.e. la durée restante avant l'épuisement en énergie d'un capteur du réseau. Il existe d'autres critères afin d'évaluer un algorithme de regroupement comme la minimisation du nombre de grappes (et donc des routeurs) ou l'équilibrage de la charge entre routeurs et donc de la taille des grappes.

Nous noterons que, dans le cas général, trouver un regroupement optimal de capteurs autour de noeuds maîtres est un problème combinatoire NP-complet : la recherche de la solution optimale nécessite l'évaluation de toutes les possibilités de grappes possibles, en l'occurrence le nombre de partitions de capteurs³ associé à un choix de noeud maître. Parmi lesquelles certaines configurations sont trivialement non-optimales (et peuvent ainsi ne pas être examinées). La recherche d'une solution optimale n'est donc possible que pour des réseaux de très faible effectif. Toutefois, l'utilisation d'heuristiques classiques d'optimisation combinatoire telles que le recuit simulé ou les algorithmes génétiques restent envisageables mais demeurent difficilement adaptables pour une utilisation distribuée : un serveur central omniscient (avec connaissance de la position des capteurs — matrice de coût — et des énergies résiduelles) doit réaliser le calcul des grappes.

2.2 Problèmes connexes

Nous évoquons ici quelques problèmes connexes au problème de regroupement étudié.

2.2.1 Routage par recherche du chemin de coût minimal

Fonctions de coût Le routage par recherche d'un chemin de coût minimal consiste à trouver un itinéraire minimisant une fonction de coût entre deux noeuds du réseau considéré. En pratique, la fonction de coût considérée est le nombre de sauts pour le routage du paquet ; il est cependant possible de considérer toute autre fonction, comme par exemple, le temps exact de transfert (prenant en compte le temps de traitement pour chaque saut — lié à la congestion des routeurs —). On pourrait également envisager l'utilisation d'une fonction de coût énergétique⁴.

²À titre d'illustration, une puce émettrice Bluetooth typique fonctionnant sous une tension de 2,7V et une intensité d'émission de 30mA consomme 115 nJ/bit.

³Le nombre de partitions de capteurs (sans considération du noeud maître) pour un réseau de n capteurs est $B(n)$, le n ème nombre de Bell, nombres de progression exponentielle.

⁴Une telle fonction prendrait en compte l'énergie nécessaire au traitement d'un message par un routeur (coût énergétique par saut) ainsi que l'énergie de transmission radio.

Algorithmes et protocoles Le maintien statique des routes pour la minimisation d'une fonction de coût peut être réalisée par un serveur centralisé en utilisant un algorithme de calcul de chemins optimaux tel que l'algorithme de Dijkstra [4]. Cette approche est impossible pour des réseaux ad-hoc dynamiques. On utilise alors des protocoles de calcul de routes et de routage distribués.

Routage pro-actif par états de lien

Le protocole OLSR Parmi ces protocoles, on compte le protocole de routage pro-actif OLSR (*Optimised Link State Routing*) [3], où chaque noeud diffuse sa connectivité avec son voisinage : ces informations reçues par les nœuds du réseau permettent de mettre à jour leur table de routage locale.

Relais multi-points La principale optimisation apportée par OLSR, par rapport à un algorithme classique d'inondation d'états de lien, pour limiter le trafic d'informations de routage, réside dans la présence de relais multi-points : chaque noeud élit dans son voisinage un nombre minimal de relais multi-points pouvant atteindre l'ensemble de son voisinage à deux sauts ; seuls les relais multi-points assurent la transmission des informations d'états de liens. Des nœuds peuvent indiquer leur refus d'assurer le rôle de relais multi-points : par exemple, dans le cadre du problème que nous étudions, des capteurs limités en énergie.

Inconvénients pour un réseau statique L'usage d'un protocole pro-actif tel que OLSR nécessite une mémoire importante ainsi que la transmission et réception fréquentes d'informations sur les liens : il semble intéressant pour assurer un routage de bonne réactivité sur des réseaux dynamiques. Il ne s'avère cependant pas adapté à des réseaux relativement statiques, où le trafic de communication est faible et l'énergie et la mémoire des nœuds limitées comme c'est souvent le cas pour des réseaux de capteurs.

Routage réactif par vecteurs de distance

Le protocole AODV L'autre famille de protocoles utilisées pour les réseaux ad-hoc utilise des vecteurs de distance. Un de ses représentants est AODV (*Adhoc On-demand Distance Vector*) [7] qui utilise un routage réactif : la route est calculée lorsque sa connaissance est nécessaire.

Routage par AODV Un noeud nécessitant une route vers un autre noeud diffuse sa demande sur le réseau : cette diffusion multi-sauts permet aux noeuds la réceptionnant de connaître une route vers le noeud demandeur ; ils peuvent répondre à ce noeud s'ils connaissent une route vers le noeud destinataire. Ce protocole présente l'avantage de ne pas nécessiter la transmission périodiques de messages de routage sur un réseau statique. D'autre part, il n'est pas nécessaire de disposer de ressources de calcul et de mémoire importantes ce qui le rend plus adapté que OLSR pour des réseaux de capteurs.

2.2.2 Arbres de capteurs

Constatations Les solutions de routage abordées précédemment sont adaptées aux situations nécessitant une communication entre deux nœuds arbitraires du réseau. Pour le problème de réseaux de capteurs avec puits, les communications sont essentiellement unidirectionnelles vers un nombre limité de destinations, voire un unique puits. La recherche d'économie d'énergie incite à reléguer le rôle de routeurs à des noeuds proches d'un nombre important de capteurs et mieux dotés en énergie que lesdits capteurs. De plus, sachant que le coût de communication entre un capteur et le puits n'est pas constant, il est avantageux de réaliser un routage des informations par l'établissement de regroupements hiérarchiques de noeuds.

Structure d'arbre de capteurs Ces constatations conduisent à l'adoption d'une structure d'arbres de capteurs ayant pour racine le puits agrégateur d'informations, pour noeuds interne les capteurs jouant le rôle de routeurs et pour feuilles des capteurs limités en énergie se contentant d'émettre leurs informations. Généralement, chaque saut est très consommateur d'énergie : minimiser le nombre de sauts est essentiel. Cela se traduit alors par la recherche d'un arbre recouvrant le réseau avec une durée de vie du réseau maximale qui présente une faible hauteur.

Arbres de hauteur 2 Dans le cas étudié, on simplifie le problème considéré en posant pour hypothèse l'accessibilité du puits en deux sauts au maximum⁵ : le coût énergétique de transmission d'information au puits depuis n'importe quel noeud est élevé mais constant. Le problème consiste alors à constituer des grappes de capteurs et à élire pour chacune de ces grappes un noeud maître assurant le rôle de routeur vers le puits, ce qui revient à trouver un arbre recouvrant de hauteur 2 pour le réseau.

3 Quelques protocoles de regroupements existants

Nous présentons ici brièvement quelques protocoles existants pour réaliser le regroupement de capteurs.

3.1 Le protocole LEACH (*Low Energy Adaptative Clustering Hierarchy*)

3.1.1 Principe général de fonctionnement de LEACH

Le protocole LEACH [6] prend pour hypothèse l'égalité des énergies résiduelles des capteurs lors du démarrage de fonctionnement du réseau. La vie du réseau est alors segmentée en rondes caractérisées par un choix de routeurs de grappes différent : pour chacune de ces rondes, les routeurs sont sélectionnés aléatoirement parmi les capteurs qui n'ont pas été auparavant élus routeurs. Lors d'une ronde, les capteurs n'ayant pas été élus routeurs se rattachent au capteur le plus proche (ou du moins celui dont le signal reçu est le plus puissant).

3.1.2 Élection probabiliste des routeurs

On choisit un ratio P de routeurs parmi tous les capteurs du réseau. Lors de chaque ronde r , la phase d'initialisation permet l'élection du routeur : chaque capteur génère un réel pseudo-aléatoire a compris entre 0 et 1 et se détermine routeur si et seulement si a est inférieur à un seuil $T(i)$ défini par la formule 1.

$$T(i) = \begin{cases} \frac{P}{1 - P(r \bmod \frac{1}{P})} & \text{si } i \text{ n'a pas été élu routeur lors des } 1/P \text{ dernières rondes} \\ 0 & \text{sinon} \end{cases} \quad (1)$$

On note que le seuil de choix augmente à chaque ronde : ainsi pour $r = 0$ (première ronde), $P(i) = P$ et pour la dernière ronde de la séquence ($r = \frac{1}{P} - 1$), $P(i) = 1$ si le capteur n'a pas été routeur auparavant : un capteur est donc routeur toutes les $\frac{1}{P}$ dernières rondes.

La particularité de LEACH réside donc dans la non-nécessité de communication pour l'élection des routeurs : le choix est réalisé indépendamment lors de chaque ronde par une méthode probabiliste. Il est cependant nécessaire de définir manuellement un ratio de routeur P parmi les capteurs.

⁵On peut également noter qu'un protocole utilisé pour résoudre le problème du regroupement peut également être adapté pour réaliser un regroupement hiérarchique des grappes.

3.1.3 Choix du routeur d'un capteur

Lors de chaque phase, les routeurs qui se sont élus diffusent leur résultat en utilisant un protocole de gestion de collisions. En supposant la puissance d'émission de l'annonce constante pour tous les routeurs, un capteur sélectionne pour routeur celui dont le signal lui parvient avec le plus de puissance. Heuristiquement, en supposant la topologie du réseau stationnaire durant la ronde, et les communications de nature symétrique, cela permet de limiter les coûts énergétiques de transmission du capteur vers son routeur.

Le capteur ayant choisi son routeur, il lui communique son choix. Au terme de la phase d'initialisation de la ronde, chaque routeur auto-élu a donc connaissance de l'ensemble des capteurs qu'il gère. On remarque qu'il est possible qu'un capteur ne soit couvert par aucun capteur (notamment pour des topologies de réseau particulières ou lorsque P est trop faible), le capteur doit alors constituer une grappe unitaire.

3.1.4 Élaboration d'un plan de communication

Plan de communication Le routeur d'une grappe élabore un plan de communication qui consiste à éviter les collisions intra-grappe par allocation de quanta de temps de transmission à chaque capteur (selon ses besoins). Les capteurs sont informés du plan de communication élaboré par le routeur : ils ne peuvent émettre leur données que lors des quanta qui leur sont réservés. Ainsi aucun protocole de détection ou d'évitement de collision n'est nécessaire et le routeur peut contenter d'activer son dispositif de réception uniquement lors d'une réception prévue par le plan de communication : ceci permet d'assurer une économie d'énergie. Il est cependant nécessaire de convenir d'un mécanisme de synchronisation d'horloge entre le routeurs et les capteurs de la grappe.

Utilisation de fréquences différentes pour les grappes adjacentes Cette méthode est néanmoins compromise par la présence des grappes adjacentes : une solution proposée est l'utilisation de fréquences différentes pour des grappes adjacentes : le problème d'assignation des fréquences est cependant équivalent à celui du coloriage d'un graphe et donc NP-complet.

On peut noter que le mécanisme de mise au point de plan de communication est d'utilisation indépendante de l'algorithme de regroupement utilisé : le seul pré-requis reste la connaissance par le routeur des capteurs qui lui sont rattachés ainsi que leur caractéristiques (fréquence de communication). Ce mécanisme peut être particulièrement utile pour des applications temps-réel.

3.1.5 Considération de l'énergie résiduelle : le protocole HEED

Fonction de coût pour le choix des routeurs HEED [9] est un protocole basé sur LEACH avec quelques différences. En particulier pour HEED, une fonction de coût est introduite pour chaque capteur et est annoncée aux voisins. Le choix d'un routeur ne s'effectue donc plus par la sélection du routeur dont la puissance de réception du signal est la plus forte : on choisit le routeur dont la fonction de coût est minimale. La fonction de coût peut par exemple être le nombre de voisins d'un noeud pour distribuer la charge sur les capteurs (ou l'inverse du nombre de voisins pour obtenir des grappes denses).

Une méthode probabiliste avec itérations est également utilisée pour l'auto-élection des routeurs, mais avec considération de l'énergie résiduelle de chaque capteur. À la première itération, pour chaque capteur i , le seuil d'éligibilité est fixé à :

$$CH_{prob}^i = \max\left(C_{prob} \frac{E_i}{E_{max}}, p_{min}\right) \quad (2)$$

où C_{prob} est le ratio recherché de routeurs sur le réseau, E_i l'énergie résiduelle du capteur i et p_{min} une probabilité minimale d'éligibilité pour assurer la terminaison de l'algorithme (afin d'éviter un

nombre d'itérations infini). À chaque itération CH_{prob} est multiplié par 2 (pour atteindre un maximum de 1 où le capteur est garanti d'être élu routeur).

Performances Des tests réalisés par les concepteurs de HEED [9] montrent que pour certaines situations, l'autonomie d'un réseau de capteurs (temps avant l'indisponibilité du premier capteur par insuffisance d'énergie) est plus que doublée par rapport à LEACH. À première vue, HEED s'avère plus adapté, par la prise en compte de l'énergie résiduelle des capteurs, pour des réseaux de capteur d'énergie résiduelle hétérogène.

3.2 Le protocole WCA (*Weighted Clustering Algorithm*)

3.2.1 Principe

WCA [2] contrairement à LEACH procède au choix des routeurs par la diffusion de scores d'éligibilité pour chaque capteur : les capteurs doivent donc communiquer pour pouvoir aboutir au choix des routeurs. Ce score est calculé suivant divers critères pondérables.

3.2.2 Calcul du score d'éligibilité

Le protocole WCA introduit les critères suivants afin d'établir un score d'éligibilité pour chaque capteur du réseau :

- L'optimisation du nombre de capteurs (degré) géré par chaque routeur. Ainsi chaque capteur est capable de gérer un nombre limité δ de capteurs en tant que routeur. On calcule donc pour chaque capteur i la valeur $\Delta_i = |D_i - \delta|$ où V_i est le nombre de capteurs dans le voisinage de i V_i : on cherche à minimiser Δ_i pour un routeur.
- Le temps cumulatif d'occupation P_i de la charge de routeur pour chaque capteur i . En effet l'algorithme WCA fait l'objet de plusieurs exécutions au cours de la vie du réseau : on comptabilise ainsi pour chaque capteur le temps passé à assumer la charge de routeur depuis l'installation du réseau. On cherche à minimiser ce temps lors du choix d'un routeur lors de l'exécution de l'algorithme : on suppose alors que chaque capteur dispose au démarrage du réseau d'énergies résiduelles similaires et que la puissance énergétique nécessaire à l'occupation de la charge de routage est constante. Il aurait sans doute été plus judicieux de considérer l'énergie résiduelle de chaque capteur.
- La somme des distances $D_i = \sum_{j \in V_i} distance(i, j)$ du capteur avec tous ses voisins. La minimisation de cette valeur est recherché afin de diminuer les coûts de communication du routeur avec les capteurs qu'il gère⁶. Ceci nécessite de pouvoir calculer la distance entre deux capteurs (utilisation d'un système GPS ou mesure de la puissance du signal reçu).
- La vitesse moyenne M_i du capteur i sur les T dernières secondes à titre de quantification du mouvement du capteur. La minimisation de la vitesse moyenne du capteur est recherchée afin de privilégier des capteurs relativement fixes à la désignation de routeur. L'obtention de la vitesse moyenne nécessite l'intégration soit d'un système de relevé de position (GPS, ...), soit d'un accéléromètre.

Toutes les valeurs citées précédemment doivent être minimisées : on calcule à partir de celles-ci un score d'éligibilité⁷ W_i pour chaque capteur i par pondération de ces valeurs⁸ :

$$W_i = w_1 \Delta_i + w_2 D_i + w_3 M_i + w_4 P_i \quad (3)$$

⁶On notera qu'il aurait probablement été plus pertinent de considérer la somme des carrés des distances dans la mesure où le coût variable de transmission est généralement proportionnel au carré de la distance.

⁷Ici le score d'éligibilité doit être minimisé ; il serait néanmoins possible de considérer des paramètres différents (par exemple leur inverse) afin de créer un nouveau score d'éligibilité qui devrait alors être maximisé.

⁸On procédera préalablement à une normalisation des valeurs.

3.2.3 Diffusion des scores d'éligibilité et élection des routeurs

Diffusion par inondation : une élection par itération L'élection des routeurs est assurée par un processus itératif : dans un premier temps, l'ensemble des capteurs communiquent leur score d'éligibilité. Le score d'éligibilité de chaque capteur est diffusé sur tout le réseau (par inondation) : chaque capteur constitue un vecteur \vec{W} de scores d'éligibilité. À l'issue de cette itération initiale, chaque capteur connaît le capteur de score d'éligibilité minimal : celui-ci est élu comme routeur et ses voisins s'y rattachent (dans la limite de δ). Le processus est répété pour les itérations suivantes avec les capteurs non-rattachés : W_i est mis à jour en ignorant les capteurs attachés, les scores mises à jours sont diffusés sur le réseau. À chaque itération un routeur est donc élu.

Diffusion aux capteurs voisins Une variante nécessitant moins d'itérations et de messages échangés consiste à ne diffuser le score d'éligibilité d'un capteur que dans son voisinage. Les capteurs réceptionnent alors les scores d'éligibilité de tous les capteurs voisins et s'auto-désignent routeur si leur score d'éligibilité est plus faible que celui de leur voisins. Les capteurs non-routeurs et non-rattachés écoutent alors les annonces des capteurs devenus routeurs et se rattachent à l'un deux (en prenant en compte le degré du routeur voire la puissance de son signal). Cette procédure est itérativement répétée jusqu'à ce que tous les capteurs soient soit attachés à un routeur, soit routeur.

4 Regroupement et élection de routeur par VCA

Le protocole VCA proposé par *Qin* et *Zimmermann* [8] réalise une sélection de routeurs de grappes de manière distribuée : seule la connaissance des énergies résiduelles des capteurs du voisinage est nécessaire.

4.1 Calcul du score énergétique

Score énergétique La fonction de poids utilisée pour la sélection d'un routeur de grappe est l'énergie résiduelle du capteur. Chaque capteur i diffuse à son voisinage sa valeur d'énergie résiduelle e_i : tous les capteurs à sa portée reçoivent cette information. Le capteur i collecte les valeurs d'énergie résiduelle de ses voisins V_i (V_i étant l'ensemble des capteurs dans la portée radio de i , y compris i) et leur attribue la note $n_{i,j}$ suivante, proportionnelle à leur énergie résiduelle (la somme des notes attribuées par un noeud étant normalisée à 1) :

$$n_{i,j} = \frac{e_j}{\sum_{k \in V_i} e_k} \quad (4)$$

Ensuite chaque capteur diffuse les notes qu'il a attribué aux capteurs de son voisinage : ceux-ci réceptionnent ces notes ainsi que celles attribuées par les autres capteurs. Chaque capteur i détermine donc son score s_i par la somme des notes attribuées par ses voisins V_i :

$$s_i = \sum_{k \in V_i} n_{k,i} \quad (5)$$

Auto-désignation du routeur Le principe de VCA consiste alors pour chaque capteur à comparer son score avec celui de ses voisins : si celui-ci est le plus élevé, alors il s'auto-désigne comme capteur routeur (tête de grappe). Ainsi au sein d'une grappe, le capteur assurant le rôle de routeur est nécessairement celui disposant de la plus haute énergie résiduelle.

Influence du degré du capteur On remarque qu'un capteur disposant d'un voisinage important bénéficie d'un nombre de votes élevé et donc d'un score important. En particulier, si

l'on considère que tous les capteurs ont une énergie résiduelle identique (hypothèse réaliste à l'installation du réseau), un capteur i obtient le score suivant :

$$s_i = \sum_{k \in V_i} \frac{1}{|V_k| + 1} \quad (6)$$

Vote attribué pour une distribution uniforme d'énergie résiduelle *Qin et Zimmermann* montrent dans leur article [8] que pour une distribution uniformément distribuée d'énergie résiduelle de capteurs sur le réseau — entre 0 et 1J —, le vote attribué par un capteur i à un de ses voisins $j \in V_i$ ($|V_i| = d$) est :

$$v_{i,j} = s \frac{1}{d} \left(1 - \int_0^d \frac{\sum_{k=0}^d (-1)^k \binom{d}{k} (u-k)^{d-1} \text{sgn}(u-k)}{2(u+e)(d-1)!} du \right) \quad (7)$$

Utilisation d'autres scores Le principe décrit pour le calcul du score énergétique pourrait être généralisé à la prise en compte d'autres fonctions d'évaluation de pertinence d'utilisation d'un capteur comme routeur. On pourrait par exemple proposer la capacité du traitement du capteur, ses ressources en mémoire, ... La possibilité d'utiliser un score pondéré de divers critères comme pour WCA (voir 3.2) peut également être évoquée. Cependant, l'objectif ici fixé de maximiser la durée de vie avant défaut de tension du réseau conduit naturellement à la considération de la seule énergie résiduelle.

4.2 Choix du routeur

Problématique du choix du routeur Un capteur peut se situer dans la portée de plusieurs routeurs de grappes : il doit alors choisir de rejoindre l'une des grappes. On cherche par cette décision à réaliser un équilibrage de charge : le capteur non-attaché privilégiera son rattachement à la grappe dont le routeur est le moins chargé ou le plus doté en énergie au regard de l'effectif géré. À cet effet, on associe à chaque routeur une valeur de *fitness* : un capteur choisit le routeur présentant une valeur de *fitness* maximale.

4.2.1 Équilibrage par charge de routeur

Une première approche consiste à minimiser l'effectif de capteur constituant une grappe et donc géré par un unique routeur. On utilise donc la fonction de *fitness* suivante :

$$fitness(i) = \frac{1}{deg_i} \quad (8)$$

Un capteur non-attaché choisit donc de rejoindre la grappe d'effectif le plus faible. La prise en compte de l'effectif comme quantification de charge du routeur suppose un débit de transmission similaire pour tous les capteurs. Dans le cadre de l'utilisation de capteurs hétérogènes, la prise en compte d'une fonction du débit moyen de transmission du capteur s'avère plus adaptée.

4.2.2 Équilibrage par énergie résiduelle

Énergie résiduelle L'objectif fixé par l'algorithme VCA est la maximalisation de la durée de vie du réseau de capteurs, durée débutant par la mise en service du réseau et s'achevant par le premier défaut de tension d'un capteur. En l'occurrence, il apparaît intéressant qu'un capteur non-attaché privilégie son rattachement au routeur disposant de la durée de vie avant défaut de tension la plus importante, i.e. l'énergie résiduelle par capteur attaché (ou plus précisément par débit géré) la plus importante. On permet ainsi d'équilibrer la consommation d'énergie entre les différents routeurs du réseau. On propose alors la fonction de *fitness* suivante :

$$fitness(i) = \frac{e_i}{deg_i} \quad (9)$$

De la limitation de l'effectif géré par les routeurs La considération de l'énergie résiduelle d'un capteur n'est dans la pratique pas suffisante pour le choix du routeur. En effet, certains capteurs-routeurs peuvent disposer d'énergie résiduelle importante mais de ressources calculatoires ou mémorielles faibles : cela entraîne nécessairement une limitation de la taille de l'effectif gérable⁹, en particulier lorsque le capteur est chargé de réaliser des tâches d'agrégation de données. Un capteur non-attaché ne se connectera donc pas à un routeur saturé.

4.2.3 Économie d'énergie d'un capteur par choix du routeur le plus proche

Dans les approches de considération de valeur de *fitness* des routeurs pour le choix de routeur de capteur non-rattachés, nous cherchons à distribuer la consommation d'énergie sur les routeurs. Une autre approche (se rapprochant de la méthode de choix pour le protocole *LEACH*) pourrait consister à rechercher une économie d'énergie sur le capteur non-routeur par le choix du routeur le plus proche : ce choix peut être réalisé soit par la mesure de la puissance du signal reçu¹⁰, soit par la connaissance de la position géographique des capteurs¹¹.

4.3 Algorithme de regroupement

L'algorithme de regroupement permet à chaque capteur de choisir son routeur : chaque routeur et ses capteurs rattachés forment alors une grappe.

4.3.1 Phase d'initialisation

La phase d'initialisation consiste à assigner pour chaque capteur un score d'éligibilité à devenir routeur en fonction de son énergie résiduelle en suivant la méthode décrite en 4 : chaque capteur calcule son score d'éligibilité en sommant toutes les notes attribuées par ses voisins qui lui ont été communiquées. D'autre part, il calcule sa valeur de *fitness* (cf 4.2). Le score d'éligibilité et la valeur de *fitness* sont diffusés à tous les voisins.

4.3.2 Procédure de regroupement

Auto-proclamation des routeurs La procédure de regroupement est réalisée en plusieurs itérations : à chaque itération, chaque capteur compare son score d'éligibilité avec les scores de ses voisins non encore rattachés à un routeur ; si ce score est le plus élevé, alors il s'auto-proclame routeur. Des cas d'égalité de score d'éligibilité peuvent survenir : dans de telles situations, certaines données telles que le degré du capteur (nombre de ses voisins) ou son identificateur peuvent être utilisés pour départager les ex-æquos. La procédure d'auto-proclamation de routeur interdit la présence de deux routeurs de même voisinage radio.

Rattachement aux routeurs

Les capteurs auto-désignés routeurs informent les autres capteurs de leur état par la diffusion d'un message. Les capteurs non-rattachés réceptionnent les annonces des routeurs nouvellement proclamés et réalisent leur choix en comparant les valeurs de *fitness* de chaque routeur. Leur choix réalisé, ils diffusent un message de retraitement de la procédure de regroupement. Au terme de l'itération, certains capteurs se seront désignés routeurs et d'autres auront choisi de se rattacher à un des nouveaux routeurs. Il est possible que certains capteurs ne se soit ni désignés routeurs, ni rattaché à un routeur au cours de l'itération (se reporter au cas présenté en 4.6.1).

⁹La limitation peut également être d'ordre protocolaire : on notera par exemple que le protocole Bluetooth [1] peut gérer une grappe (*piconet*) comportant un maître qui contrôle au maximum 7 esclaves.

¹⁰En supposant les puissances d'émission de chaque capteur connues afin de pouvoir déduire la distance entre le capteur et le routeur.

¹¹Par l'utilisation d'un système de positionnement par satellite ou par la présence de capteurs de positions connues et l'utilisation d'un procédé de triangulation.

Mise à jour des scores d'éligibilité Au début de chaque itération, il est nécessaire de réactualiser les scores d'éligibilité des capteurs non-rattachés (ainsi que les valeurs de *fitness*) en ne considérant que les notes attribuées par les capteurs non-rattachés. Un capteur peut avoir connaissance des capteurs qui se sont rattachés récemment en surveillant la présence de messages de retraitement sur le médium. Il met à jour son score global en ignorant les notes des capteurs attachés lors de l'itération précédente : si la différence entre son ancien et nouveau score est importante (supérieure à un seuil fixé), la diffusion du nouveau score est décidée afin d'informer les capteurs non encore rattachés.

4.4 Mobilité des capteurs et périodicité du regroupement

Théoriquement, si le réseau de capteurs est totalement statique, mener l'algorithme de regroupement une unique fois à l'installation du réseau est suffisant. Cependant, dans la pratique, il est possible que certains routeurs puissent être mobiles. Il peut donc être nécessaire de renouveler l'exécution de l'algorithme VCA pour déduire un nouveau regroupement de capteurs. On note toutefois que la mobilité des capteurs, pour les hypothèses fixées (possibilité de joindre le puits depuis n'importe quel capteur), n'introduit pas de capteurs non-entendus par le puits ; cependant, l'éloignement d'un capteur peut le séparer du routeur auquel il s'est rattaché. On peut néanmoins palier ce problème en imposant à tous les routeurs la transmission d'un signal de vie (*heartbeat* comprenant son énergie résiduelle et sa valeur de *fitness*) à intervalle régulier δ . Un capteur orphelin de son routeur écoute alors le médium pendant une durée $\delta + \epsilon$: il connaît tous les routeurs à sa portée et choisit le routeur avec la valeur de *fitness* la plus faible. Si aucun routeur n'est à sa portée, il devient lui-même routeur.

La période d'exécution de l'algorithme de regroupement doit dépendre de la mobilité des capteurs. D'autre part, relancer une exécution peut être utile lorsque les énergies résiduelles des capteurs ont beaucoup évolué.

Qin et Zimmermann ne discutent pas spécifiquement du problème de la mobilité des capteurs dans l'article [8] : ils prévoient néanmoins d'étendre VCA afin de prendre en compte dans les notations des capteurs de la mobilité des voisins.

4.5 Résistance aux fautes byzantines

On notera que l'algorithme de regroupement proposé repose entièrement sur l'honnêteté et le bon fonctionnement des capteurs participants au protocole¹². Ainsi un capteur peut s'auto-attribuer un score d'éligibilité fantaisiste et le diffuser aux autres capteurs. Cette situation peut être provoquée soit par une erreur du capteur, soit par le contrôle du capteur par un tiers malhonnête dont l'objectif serait :

- soit d'introduire un routeur sous son contrôle au sein du réseau,
- soit de limiter l'autonomie du réseau.

Éviction des capteurs intrus L'éviction de capteurs intrus peut être aisément réalisée par l'utilisation d'une méthode de signature des messages échangés. À cet effet, on peut doter chaque capteur d'une paire de clé publique/clé privée, la clé publique faisant l'objet d'une certification par une autorité. Les capteurs ne considèrent alors que les messages correctement signés par une clé privée dont la clé publique fait l'objet d'une certification par l'autorité.

Erreur de capteurs Il pourrait être possible de limiter des erreurs de calcul de scores d'éligibilité par l'exécution multiple de la phase de calcul de scores puis par l'utilisation d'une moyenne sur les scores.

¹²Il en est de même pour la grande majorité des protocoles de regroupement.

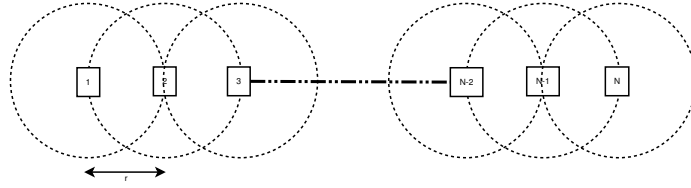


Figure 1: Chaîne de N capteurs.

4.6 Complexité de l'algorithme de regroupement VCA

Nous discutons ici de la complexité de l'algorithme VCA en temps et en messages échangés.

4.6.1 Complexité temporelle

Dans le pire des cas, N itérations de l'algorithme VCA sont nécessaires afin d'achever la procédure de regroupement pour un réseau de N capteurs.

Chaîne de capteurs Le pire cas se produit pour une chaîne de capteurs (voir figure 1). Soient N capteurs distincts c_1, c_2, \dots, c_N de portée r alignés avec une distance entre deux capteurs consécutifs c_i et c_{i+1} de $d_{i,j} = r - \epsilon$. On pose $e_1 \ll e_2 \ll \dots \ll e_n$ (de sorte que $s_1^i < \dots < s_n^i$ pour toutes les itérations i de l'algorithme). Lors de la première itération de l'algorithme, tous les capteurs diffusent leur score d'éligibilité : le capteur c_i réceptionne les scores de c_{i-1} et c_{i+1} (pour $2 \leq i \leq N - 1$) ; comme $s_{i+1} > s_i > s_{i-1}$, c_i ne s'auto-désigne pas comme routeur. Pour cette première itération, seul c_N devient routeur : il diffuse son auto-désignation en tant que routeur. Deux possibilités peuvent alors être envisagées :

- La communication est symétrique : c_{N-1} reçoit l'annonce de c_N et se rattache à c_N . Il reste une chaîne de $N - 2$ capteurs à regrouper.
- La communication n'est pas symétrique : c_{N-1} ne reçoit pas l'annonce et reste non-attaché. Il reste une chaîne de $N - 1$ capteurs à regrouper.

Lors de l'itération suivante, le problème est envisagé récursivement pour une chaîne de $N - 1$ ou $N - 2$ capteurs. Ainsi dans le pire des cas, seul un capteur du réseau devient routeur et n'entraîne pas l'attachement d'un autre capteur par itération : N itérations sont nécessaires.

4.6.2 Complexité en nombre de messages

Durant la phase d'initialisation Pour la phase d'initialisation, chaque capteur i émet un message notifiant son énergie résiduelle e_i , puis transmet un unique message spécifiant les notes attribuées à ses voisins, et enfin diffuse son score d'éligibilité calculé en sommant les notes reçues.

Durant la phase de regroupement

Pendant la phase de regroupement, un capteur ne diffuse un message que lors de trois occasions :

1. Lorsqu'il s'auto-désigne routeur.
2. Lorsqu'il choisit un routeur pour s'y attacher (émission d'un message de retirement).
3. Lorsque son score d'éligibilité change (à cause de l'attachement de capteurs de son voisinage lors de l'itération).

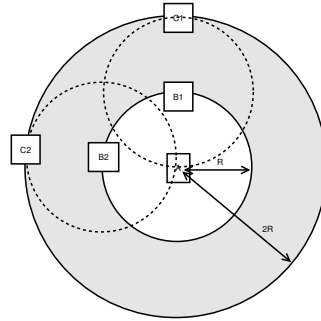


Figure 2: Illustration du cas de mise à jour du score d'éligibilité de A (par attachement de B_1 , puis B_2 , ...).

Les deux premières situations ne se produisent qu'une seule fois par capteur : en effet, au terme des itérations de l'algorithme, tous les capteurs sont soit devenus routeur (émission d'un message 1), soit se sont rattachés à un routeur (émission d'un message 2).

Nous cherchons maintenant à trouver une borne supérieure pour le changement de scores d'éligibilité pour chaque routeur et donc pour le nombre de messages de type 3. Soit un capteur A non-rattaché, m capteurs B_1, \dots, B_m dans le rayon de portée r de A tels que pour tout B_i et B_j , $d(A, B_i) = r - \epsilon$ et $d(i, j) > r$, et m capteurs C_1, \dots, C_m tels que pour tout capteur C_i , $\overrightarrow{AB_i} = \overrightarrow{B_i C_i}$ (voir figure 2). On suppose que, à tout moment, pour tout i , on a l'inégalité suivante sur les scores d'éligibilité : $s(A) < s(B_i) < s(C_i)$. Lors de l'itération $iter_i$, C_i s'auto-proclame routeur : B_i écoute l'annonce d'auto-proclamation de C_i et se rattache à C_i . Alors B_i se retire : A obtient le message de retraitement de B_i et doit alors procéder à la mise à jour de son score et en informer les capteurs C_{i+1}, \dots, C_m non encore rattachés. Ainsi m messages de mise à jour de score (un pour chaque itération) sont nécessaires pour la mise à jour du score de A . Il reste à quantifier la borne maximale de m : ceci revient à déterminer, dans le cadre bidimensionnel, le nombre de sommets que comporte un polygone régulier dont les sommets sont de longueur r inscrit dans un cercle de rayon r . Le périmètre d'un tel polygone est majoré par $2\pi r$: m est donc majoré par 2π . Ainsi, dans un espace à 2 dimensions, au plus 6 messages de mise à jour sont à prévoir pour chaque capteur.

Finalement, on montre donc que la complexité en nombre de messages est $O(N)$.

5 Tests et évaluation de performances

Nous présentons ici les résultats des tests réalisés par *Qin* et *Zimmermann* sur l'algorithme VCA. Ceux-ci cherchent à comparer les performances de leur algorithme par rapport à d'autres algorithmes de regroupement :

- HEED (voir 3.1.5) qui est un algorithme de sélection de routeurs probabiliste utilisant l'énergie résiduelle (la fonction de coût utilisée est le nombre de voisins — degré — du capteur : un capteur joint le routeur de degré minimal),
- et GCA (*Generic Clustering Algorithm*), un algorithme de regroupement générique (similaire au principe de fonctionnement de WCA décrit en 3.2) sélectionnant les routeurs par leurs poids (on cherche ici à maximiser le poids qui est l'énergie résiduelle).

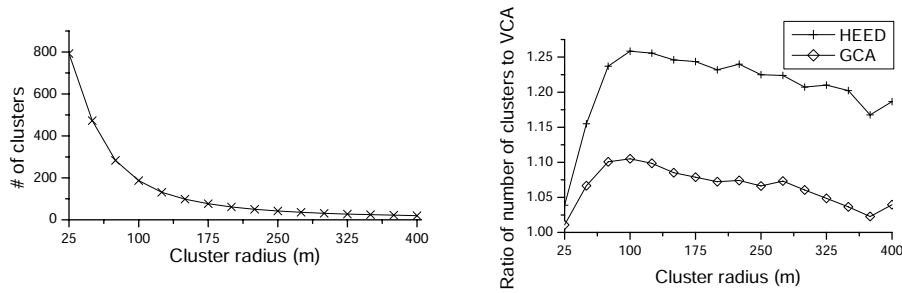


Figure 3: Nombre de grappes selon la portée des capteurs (à gauche : nombre de grappes créées par VCA, à droite : nombre de grappes créés par HEED et GCA par rapport à VCA) [8].

Les simulations réalisées utilisent 1000 capteurs immobiles aléatoirement dispersés sur une surface carrée de $c^2\text{m}^2$ (ici $c = 2000\text{m}$). L'énergie résiduelle des capteurs est attribuée aléatoirement dans l'intervalle $[0, 1\text{J}]$. Pour chaque test, 100 exécutions indépendantes sont réalisées : les résultats utilisés moyennent les résultats de toutes les exécutions. Pour les simulations réalisées, le paramètre variable utilisé est la portée de chaque capteur : augmenter la portée d'un capteur est alors équivalent à augmenter la densité du réseau ; il aurait donc été équivalent d'utiliser la surface du réseau pour variable.

5.1 Nombre de clusters

On s'intéresse au nombre de clusters générés par les algorithmes de regroupement étudiés. On recherche particulièrement la minimisation du nombre de clusters, la bonne répartition des capteurs sur les routeurs (par minimisation de l'écart type de l'effectif d'une grappe) et en particulier la minimisation du nombre de grappes constitué d'un unique capteur-routeur.

On constate que le nombre de grappes n générées par VCA par rapport à la portée r de chaque capteur (1^{er} diagramme de la figure 3) suit une loi inverse ce qui est conforme aux attentes théoriques : une fonction $r \rightarrow c/r$ est en effet attendue en raison de la dispersion aléatoire des capteurs (ici $c = 2000\text{m}$: 800 grappes sont nécessaires pour couvrir 2000m^2). Les routeurs sont donc homogènement dispersés sur la surface.

VCA permet de générer un nombre de grappes sensiblement inférieur à celui obtenu par l'utilisation de GCA ou de HEED d'après le 2^{ème} diagramme de la figure 3. Le ratio du nombre de grappes de GCA ou HEED par rapport à VCA atteint son maximum pour une portée de 100m : supérieur à 1,25 pour HEED et approximativement 1,10 pour GCA. L'augmentation de la portée au-delà de 100m (équivalente à la densification du réseau), entraîne une diminution du ratio. Il est possible d'expliquer la limitation du nombre de grappes (et donc de routeurs) par VCA par le favoritisme pour l'élection de routeur accordé aux capteurs disposant d'un important voisinage : en effet le score d'éligibilité d'un capteur est la somme de notes *énergétiques* accordées par ses voisins.

Le pourcentage de grappes formées uniquement d'un routeur (diagramme 1 de la figure 4) diminue avec l'augmentation de la portée des capteurs (densification du réseau). L'utilisation de VCA avec une fonction d'équilibrage par charge de routeur (cf 4.2.1) permet de limiter les grappes composées de capteurs solitaires : un capteur non-attaché cherche alors à rejoindre la grappe d'effectif le plus faible dans sa portée.

HEED assure l'écart-type (diagramme 2 de la figure 4) en effectif de grappe le plus faible (avec des résultats quasiment similaires à VCA avec équilibrage par charge). GCA génère des grappes

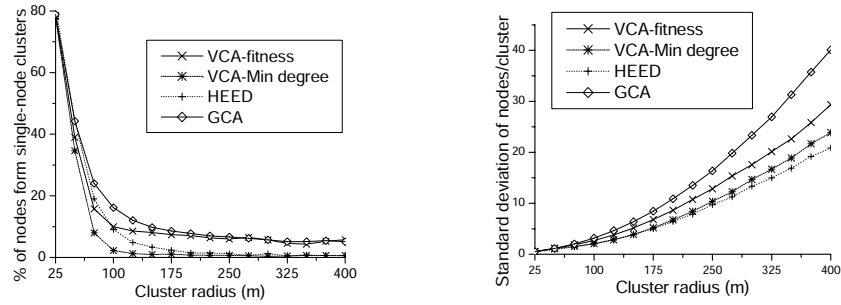


Figure 4: Pourcentage de routeurs solitaires et écart type du nombre de capteurs par grappe [8].

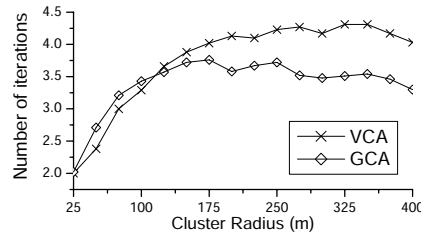


Figure 5: Nombre moyen d'itérations réalisées pour l'exécution de l'algorithme de regroupement [8].

d'effectifs plus disparates ce qui s'explique ici par la non-prise en compte du degré des routeurs¹³ lors du choix d'un capteur.

5.2 Nombre d'itérations

L'étude du nombre d'itérations de VCA permet d'étudier la complexité temporelle de cet algorithme de regroupement. Nous avons montré en 4.6.1 que le nombre d'itérations nécessaires pouvait atteindre $N/2^{14}$ (où N est la taille en capteurs du réseau), donc 500 itérations pour un réseau de 1000 capteurs. En pratique, avec un réseau de capteurs uniformément répartis, on constate sur la figure 5 (où l'on notera l'absence de HEED¹⁵) que le nombre d'itérations nécessaires pour VCA est inférieur à 5 (inférieur à 4 pour GCA avec diffusion de scores aux voisins). Globalement, le nombre d'itérations augmente avec la densification du réseau mais reste raisonnable. Il serait néanmoins intéressant de réaliser des tests avec un nombre de capteurs plus important.

5.3 Autonomie du réseau

L'autonomie du réseau peut être mesurée par le temps avant la défaillance du premier capteur et du dernier capteur par défaut d'énergie (figure 6). L'usage de la fonction de *fitness* définie en 4.2.2 permet de maximiser avec VCA le nombre de rondes avant la défaillance du premier capteur : le choix d'un routeur par un capteur est réalisé en optant pour celui dont l'énergie

¹³La version originale de WCA considère, en plus de l'énergie résiduelle du capteur, la vitesse de celui-ci et son degré (non utilisés ici).

¹⁴Dans le cas de communications symétriques.

¹⁵HEED réalise la couverture du réseau en une itération ; les itérations suivantes permettent de réaliser un regroupement différent, nécessaire pour répartir la charge de routage entre capteurs.

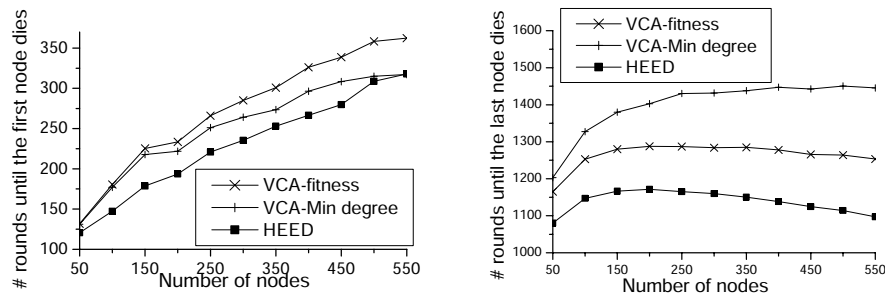


Figure 6: Nombre de rondes avant défaillance par défaut d'énergie du premier et du dernier capteur du réseau [8].

résiduelle par capteur attaché est la plus élevée. Cette stratégie permet d'équilibrer la sollicitation énergétique des routeurs : tous les capteurs expérimentent alors une rupture énergétique dans un laps de temps rapproché. L'utilisation d'un choix de routeur par degré minimal (comme vu en 4.2.1) entraîne une disparité de durée de vie des capteurs : la durée de vie du dernier capteur est supérieure. Globalement VCA permet d'obtenir une autonomie du réseau supérieure de 25 à 20% par rapport à HEED. On pourra toutefois regretter que des tests n'aient pas été réalisés pour comparer l'autonomie induite par VCA par rapport à GCA.

6 Conclusions

D'après les premiers tests réalisés, le protocole VCA proposé par *Qin* et *Zimmermann* est efficace pour réduire le nombre de routeurs et augmenter l'autonomie d'un réseau de capteurs par rapport à d'autres protocoles existants tels que HEED ou GCA. Cela s'explique sans doute par l'utilisation d'un protocole de vote induisant une dépendance des scores d'éligibilité des capteurs au voisinage au degré 2 (voisinage du voisinage) du capteur considéré. Cependant, les auteurs ne se sont intéressés qu'au cas de réseaux quasi-stationnaires : il pourrait être intéressant d'adapter VCA pour des réseaux composés de capteurs mobiles, par exemple en s'inspirant des scores d'éligibilités pondérés de WCA. D'autre part, l'extension de VCA pour un routage en plusieurs sauts pourrait être étudié¹⁶ : nous avons en effet supposé que le coût de communication avec le puits était constant pour tous les capteurs, ce qui n'est en pratique jamais le cas (le puits pouvant même ne pas être accessible de certains capteurs).

References

- [1] Spécifications du protocole bluetooth. URL <https://www.bluetooth.org/spec/>.
- [2] M. Chatterjee, S. Das, and D. Turgut. Wca: A weighted clustering algorithm for mobile ad hoc networks, 2002. URL <http://citeseer.ist.psu.edu/chatterjee02wca.html>.
- [3] T. Clausen and P. Jacquet. RFC 3626 – optimized link state routing protocol (OLSR), 2003. URL <http://hipercom.inria.fr/olsr/rfc3626.txt>.
- [4] Edsger. W. Dijkstra. A note on two problems in connexion with graphs. In *Numerische Mathematik*, volume 1, pages 269–271. Mathematisch Centrum, Amsterdam, The Netherlands, 1959. URL <http://gdz-srv3.sub.uni-goettingen.de/sub/digbib/loader?ht=VIEW&did=D196313>.

¹⁶En premier lieu, il serait possible d'appliquer récursivement l'algorithme VCA sur les capteurs, les routeurs, les routeurs des routeurs, ...

- [5] David L. Hall and Sonya A. H. McMullen. *Mathematical Techniques in Multisensor Data Fusion*. Artech House, 1992. ISBN 1580533353.
- [6] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *HICSS*, 2000. URL <http://citeseer.ist.psu.edu/rabinerheinzelman00energyefficient.html>.
- [7] Charles E. Perkins, Elizabeth M. Belding-Royer, and Ian Chakeres. Ad hoc on demand distance vector (AODV) routing draft, 2003. URL <http://moment.cs.ucsb.edu/pub/draft-perkins-manet-aodvbis-00.txt>.
- [8] Min Qin and Roger Zimmermann. An energy efficient voting-based clustering algorithm for sensor networks. 2005. URL <http://dmrl.usc.edu/pubs/sawn05.pdf>.
- [9] Ossama Younis and Sonia Fahmy. HEED: A hybrid, energy-efficient, distributed clustering approach for ad-hoc sensor networks. URL <http://citeseer.ist.psu.edu/698974.html>.