



HAL
open science

Application de Concepts de Routage Géographique au Routage par Gradient: une Étude Qualitative

Thomas Watteyne, Isabelle Augé-Blum, Mischa Dohler

► **To cite this version:**

Thomas Watteyne, Isabelle Augé-Blum, Mischa Dohler. Application de Concepts de Routage Géographique au Routage par Gradient: une Étude Qualitative. 2006. inria-00113733

HAL Id: inria-00113733

<https://inria.hal.science/inria-00113733>

Submitted on 20 Nov 2006

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.

Application de Concepts de Routage Géographique au Routage par Gradient: une Étude Qualitative

Thomas Watteyne^{*†}, Isabelle Augé-Blum^{*} and Mischa Dohler[†]

^{*}CITI Laboratory, INSA Lyon, France.

prénom.nom@insa-lyon.fr

[†]France Télécom R&D, Meylan, France.

prénom.nom@orange-ft.com

Abstract—Les réseaux de capteurs sans fil connaissent un fort engouement ces dernières années, à la fois dans le monde académique et dans le monde industriel, grâce notamment, au grand nombre d'applications envisagées, et à l'apparition de solutions commerciales viables. A cause de leurs contraintes propres, des mécanismes de routage spécifiques ont été proposés.

Le routage géographique et le routage par gradient sont des candidats intéressants pour les réseaux de capteurs, puisqu'ils ne requièrent qu'une phase d'auto-organisation minimale. Dans cet article, nous montrons que les concepts du routage géographique sont applicables au routage par gradient. Nous mettons en évidence une faille dans la règle de la main droite, et nous proposons une solution sous la forme d'un protocole appliqué au routage géographique et par gradient.

I. INTRODUCTION

Les Réseaux de Capteurs sans fil (Wireless sensor networks – WSN) connaissent un fort engouement ces dernières années. Les applications de tels réseaux sont multiples, allant de la défense et la surveillance, à la santé et les maisons intelligentes [1].

Les WSN sont composés d'un nombre potentiellement très grand (plusieurs milliers) de ces petits objets communicants, déployés dans la zone à couvrir. Chaque capteur est capable d'effectuer trois tâches complémentaires: mesure d'une valeur physique, traitement de ces mesures, et communication par voie hertzienne.

Les capteurs sont des objets communicants limités en terme de bande passante, de puissance de calcul, de mémoire disponible et d'énergie embarquée. Le déploiement est la plupart du temps aléatoire et peut être réalisé dans des zones dangereuses (on ne peut alors pas compter sur un remplacement de batterie).

Dans les WSN, une structure peut être créée entre les capteurs. Des protocoles d'auto-organisation pour les réseaux ad-hoc [2]–[4] s'appuient sur une phase d'auto-organisation où une structure virtuelle est créée. Les noeuds sont regroupés en *clusters*, avec un chef pour chacun. Ces chefs sont interconnectés par un ensemble d'épines dorsales. Le routage se fait alors de manière hiérarchique en suivant le chemin *noeud source* → *chef de cluster* → *épine dorsale* → *chef de cluster* → *noeud destination*. Une fois la structure construite, un certain nombre de mécanismes permettent de prendre en compte le changement de topologie (par mouvement et apparition/disparition de noeuds) et l'endormissement de certains

noeuds pour éviter de sur-utiliser leurs batteries.

Ces formes d'auto-organisation explicite – *i.e.* avec une phase d'auto-organisation importante avant fonctionnement normal du réseau – semblent peu adaptées aux réseaux de capteurs. Dans ces protocoles, les mécanismes de maintien de la structure supposent l'échange *périodique* de messages de signalisation. Or ceux-ci sont coûteux en énergie, et réduisent la durée de vie des capteurs.

Pour certaines applications de réseaux de capteurs, une auto-organisation implicite est préférable. Dans celle-ci, le réseau s'organise en s'appuyant sur une métrique extérieure ou une phase d'organisation minimale. Nous nous intéressons dans ce document au routage géographique et au routage par gradient. Il s'agit des deux grandes classes de routage s'appuyant sur une phase d'auto-organisation implicite.

Dans les protocoles de routage utilisant le concept de routage géographique [5], un noeud élit parmi ses voisins le noeud qui est le plus proche géographiquement de la destination comme noeud de prochain saut. Dans une approche utilisant le routage par gradient [6]–[9], une métrique est rattachée à chaque noeud. Un noeud élit parmi ces voisins le noeud qui a la plus grande/petite métrique comme noeud de prochain saut, selon l'approche utilisée.

Dans la partie II, nous présentons l'état de l'art concernant le routage géographique et le routage par gradient. Dans la partie III nous montrons que les concepts de routage géographique peuvent s'appliquer au routage par gradient. Nous mettons ensuite en évidence une faille de la règle de la main droite, et nous proposons une solution. Enfin, nous concluons dans la partie III.

II. ETAT DE L'ART

Le routage dans les WSN pose de nombreux défis. Utiliser un plan d'adressage global semble peu efficace, à cause du grand nombre de noeuds. Celui-ci supposerait l'échange de messages de contrôle de conflit d'adresses, sur des liens de faible bande passante et entre des noeuds à énergie limitée. Nous sommes en présence d'un modèle de communication "sources multiples - destination unique". Un ensemble de noeuds situés à proximité l'un de l'autre peuvent simultanément détecter un même évènement (*e.g.* le départ d'un feu de forêt), un mécanisme d'agrégation de données est donc nécessaire.

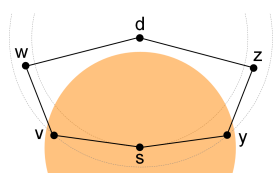


Fig. 1. Le routage géographique avide s'arrête à s . La règle de la main droite choisit le chemin s - y - z - d . Le disque plein représente la portée de communication de s . Les cercles en pointillés sont centrés en d , ils permettent de voir la distance relative des noeuds à d . Les segments relient des noeuds capables de communiquer entre eux.

A. Le routage géographique

Dans un protocole de routage géographique, un noeud est supposé connaître sa position géographique, celle de ses voisins et celle du noeud destination. La connaissance du voisinage est périodiquement mise à jour à l'aide de messages *HELLO* échangés entre noeuds voisins. Dans une approche avide (*greedy geographic forwarding*), le noeud le plus proche de la destination, parmi les noeuds voisins, est choisi comme prochain saut. Comme métrique de proximité, il est possible d'utiliser la distance euclidienne, ou la projection sur une ligne entre le noeud courant et la destination.

Cependant, [5] met en évidence une faille du routage géographique avide, illustrée Fig. 1. Lorsque le noeud s reçoit une alarme, il n'existe pas de noeud voisin de s plus proche que lui de d . Avec une approche avide, le routage s'arrête à s et le message est perdu. GPSR [5] propose la règle de la main droite comme solution. Dans celle-ci, lorsque s ne trouve pas de noeud voisin plus proche que lui de la destination, il trace une ligne virtuelle entre lui et la destination d , et fait tourner cette ligne vers la droite, centrée sur s . Le premier noeud rencontré (ici y) est élu noeud de prochain saut. Le chemin retenu est s - y - z - d . GPSR utilise le routage géographique avide, couplé à la règle de la main droite.

B. Le routage par gradient

Le routage géographique suppose que tous les noeuds connaissent leur position. Une solution basée sur le GPS peut être trop coûteuse, d'autant plus que le nombre de noeuds à équiper est très grand. Pour répondre à ce problème, le routage par gradient (ou Gradient Based Routing – GBR) [10] associe une *hauteur* à chaque noeud. Cette hauteur peut par exemple être le nombre de sauts qui sépare le noeud du puits. Pour obtenir une certaine efficacité en énergie, l'énergie résiduelle d'un noeud peut être couplée au nombre de sauts du puits comme hauteur. La description qui suit peut être généralisée à n puits. L'association noeud-hauteur se fait lors d'une phase d'initialisation: (1) le puits envoie un message contenant une *hauteur courante* de 0, (2) chaque noeud qui entend ce message initialise sa hauteur à *hauteur courante*+1 et retransmet le message avec cette nouvelle hauteur comme hauteur courante [8].

On suppose qu'à la fin de cette initialisation, tout noeud connaît sa hauteur, ainsi que celle de son 1-voisinage. Celle-ci pourra être apprise par un envoi périodique de messages de

type *HELLO*. Durant le fonctionnement, les noeuds envoient leur message au noeud voisin le plus "bas". Ce routage avide (*i.e.* le message sera transmis selon le chemin de plus forte pente) présente beaucoup de similitudes avec le routage géographique.

Pour conclure la présentation de l'état de l'art, les routages géographique et par gradient semblent bien adaptés aux réseaux de capteurs. En effet, ils ne requièrent qu'une information locale du réseau pour l'établissement de routes globales. Dans le routage géographique, il n'y a pas de phase d'auto-organisation explicite. Pour le routage par gradient, [8] propose un protocole localisé d'adaptation des hauteurs des noeuds en cas de changement de topologie (mobilité, arrivées/départ de noeuds).

III. PROPOSITIONS

Dans cette partie, nous montrons que les concepts de routage géographique peuvent facilement s'appliquer au routage par gradient. Nous mettons ensuite en évidence une faille de la règle de la main droite, aussi bien dans le routage géographique que dans le routage par gradient. Nous proposerons une solution générale, et nous l'appliquons au routage géographique et au routage par gradient.

A. Application de concepts de routage géographique au routage par gradient

Le routage par gradient présente beaucoup de similitudes avec le routage géographique. On peut en effet directement appliquer le protocole de routage géographique avide au routage par gradient car, dans les deux mécanismes de routage, chaque noeud contient une métrique de proximité à la source. Par contre, la notion de droite n'a pas de sens dans le cas de routage par gradient. On ne peut donc appliquer directement la règle de la main droite au routage par gradient.

En routage géographique, la règle de la main droite consiste à contourner une zone sans noeud. Cette règle permet une avancée transversale lorsqu'une avancée en direction de la destination n'est pas possible. Le choix de ce prochain noeud n'est pas fortuit: en faisant tourner une ligne imaginaire vers la droite, on choisit le noeud qui s'éloigne le moins de la direction initiale. Nous proposons pour l'approche par gradient d'*élire le noeud de plus proche de la destination parmi les noeuds candidats*. Une fois ce changement effectué, les concepts que nous présentons pour le routage géographique peuvent être appliqués au routage par gradient.

B. Mise en évidence d'une faille dans la règle de la main droite

Nous présentons la faille de la règle de la main droite dans le cadre d'un routage géographique, mais cette faille est aussi présente dans l'adaptation au gradient présentée dans le paragraphe précédent. Comme représenté dans la Fig. 2, lorsque le message arrive au noeud s , il sera transmis au noeud x puisqu'il s'agit du noeud le plus proche de d parmi les voisins de x . x n'ayant pas de voisins plus proche de d que lui enclenche la règle de la main droite; s est élu comme

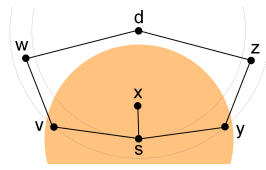


Fig. 2. Le chemin parcouru par le message avec le routage géographique et la règle de la main droite est $s-x-s-x-\dots$, il y a donc création d'une boucle.

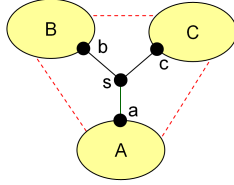


Fig. 3. Topologie abstraite du point de vue du noeud s . A , B et C représentent les sous réseaux connectés ou non entre eux.

noeud suivant. Nous mettons ici en évidence que la règle de la main droite couplée au routage géographique peut créer des boucles.

Deux problèmes se posent avec cette faille:

- des paquets sont échangés indéfiniment dans le réseau, épuisant rapidement les batteries des noeuds émetteurs, mais également de tous les noeuds récepteurs aux alentours. Nous appellerons ces paquets des paquets fantômes, parce que même s'il transitent sur le réseau, ils n'ont aucune utilité.
- ces paquets ne trouvent pas de chemin vers la destination, et sont donc perdus.

C. Enregistrement du chemin parcouru dans les noeuds

Nous supposons la stabilité du réseau (stabilité des liens, donc du graphe de connectivité). Nous supposons également qu'une seule copie du paquet est transmise à travers le réseau. Nous présentons le concept d'enregistrement du chemin de manière assez générale pour qu'il soit appliqué au routage géographique ou au routage par gradient.

L'idée est de retenir le trajet parcouru par un message. Si un message repasse par le même noeud, celui-ci choisira un chemin différent de tous ceux déjà choisis. Nous nous appuyons dans cette description sur la Fig. 3. Dans celle-ci, s représente le noeud courant. Les noeuds a , b et c relient s aux sous-réseaux A , B et C respectivement. Notons que les sous réseaux peuvent ou non communiquer entre eux par un chemin ne passant pas par le noeud s (les liens en pointillés).

Nous supposons dans cette partie un protocole de routage qui donne à chaque noeud voisin une métrique. Plus la métrique est haute, plus le protocole de routage considère que la probabilité d'atteindre la destination à travers ce voisin est grande. Nous supposons pour l'instant que le noeud a la capacité de retenir l'identifiant d'un paquet, et les voisins vers lesquels il a déjà transmis le paquet. Pour cet exemple, nous considérons que $\text{métrique}_B > \text{métrique}_C > \text{métrique}_A$. Lorsque s reçoit un paquet, il le transmet au sous réseau B . Si ce paquet

revient à s , alors s ne devra plus le renvoyer au sous réseau B . En effet, comme nous supposons la stabilité du réseau, le paquet suivra le même chemin et reviendra à s . s renvoie donc le paquet à C . Si le paquet revient à s , il est retransmis à A . Si le paquet repasse encore une fois par s , alors il est détruit, car cela signifie qu'aucun chemin n'existe, puisque nous supposons la stabilité du réseau.

D'une manière plus générale, à la réception d'un paquet, s le renvoie au voisin de plus haute métrique parmi les voisins auxquels s ne l'a pas encore renvoyé. Il est facile d'utiliser cet algorithme dans le cas d'un protocole de routage géographique averse, en prenant comme métrique pour chaque voisin l'inverse de sa distance au puits. Dans le cas du routage par gradient, la métrique sera directement la hauteur du noeud.

D. Enregistrement du chemin parcouru dans le paquet

L'algorithme présenté jusque là suppose néanmoins que tout noeud retienne les identifiants des paquets qu'il a déjà traités, et les voisins vers lesquels il les a envoyés. Il faut donc prévoir un espace mémoire dans chaque noeud. Déterminer la taille de cette mémoire semble difficile. Pour ne pas que cette taille soit trop grande, il est possible d'affecter une durée de vie à chaque entrée. Plus cette durée de vie est grande, plus il faudra prévoir de place mémoire; avec une durée de vie trop courte, il est possible d'effacer des entrées trop tôt, *i.e.* lorsque le message n'a pas encore atteint la destination, donc peut repasser par le noeud.

Une solution est donc d'enregistrer le chemin parcouru par un paquet non pas dans les noeuds traversés mais dans le paquet. Chaque noeud traversé rajoutera dans l'entête du paquet son identifiant. Lorsqu'un noeud reçoit un paquet, il peut connaître le chemin déjà parcouru, donc savoir s'il a déjà été traversé. Si oui, l'identifiant du noeud se situant après le sien est le voisin vers lequel il l'a envoyé. En parcourant tout le chemin enregistré dans l'entête du paquet, il construit la liste des voisins vers lesquels le paquet n'a pas encore été transmis, et il peut appliquer les résultats de la partie III-C.

Ajouter son identifiant à l'entête d'un paquet fait grossir la taille de celui-ci. Donc il faudra dans les prochains sauts dépenser plus d'énergie à l'émission et la réception. Dans l'équation (1), nous exprimons $l_{overhead}$, le nombre d'octets supplémentaires transmis au cours de n_{hops} sauts, lorsqu'à chaque saut un noeud rajoute son adresse codée sur $l_{address}$ octets.

$$l_{overhead} = \frac{n_{hops}(n_{hops} + 1)}{2} \cdot l_{address} \quad (1)$$

Il ne s'agit pas ici d'un problème similaire aux adresses IP. En effet, nous supposons que chaque noeud a un identifiant unique – assimilable à une adresse MAC –, mais il ne s'agit pas d'un adressage utilisable pour le routage comme les adresses IP. Si on considère un protocole de routage optimal – donc n_{hops} optimal –, pour réduire $l_{overhead}$, on ne peut que réduire $l_{address}$. Si le nombre de noeuds est très grand, et que les identifiants uniques sont codés sur un trop grand nombre de bits, on peut demander à chaque noeud de tirer un identifiant

entier aléatoirement dans $[0, \dots, id_{max}-1]$. Comme id_{max} est plus petit que N le nombre de noeuds, des noeuds choisiront forcément la même adresse. Il est donc possible de retrouver plusieurs noeuds ayant choisi la même adresse sur le chemin parcouru par un paquet. Nous pouvons réduire cette probabilité en regardant l'adresse suivante: lorsqu'un noeud trouve son identifiant a dans le chemin déjà parcouru par un paquet, il regarde $a+1$ l'adresse suivante. Si $a+1$ est l'identifiant d'un de ses voisins, alors il fait l'hypothèse que a l'identifie lui.

E. Enregistrement du chemin parcouru dans le paquet, avec règle de la main droite

Le protocole présenté jusqu'à maintenant (le chemin déjà parcouru est enregistré dans le paquet, le noeud choisit le prochain saut uniquement parmi les voisins vers lesquels le paquet n'a pas encore été envoyé) ne suffit pas dans le cas de la règle de la main droite. En effet, en prenant la topologie de la Fig. 2, nous obtenons avec ce protocole incomplet le chemin $s-x-s-y-s-v-s$, et le paquet est perdu. Il faut donc différencier un envoi par routage avide d'un envoi par règle de la main droite. Uniquement dans le premier cas, le paquet peut faire "demi tour".

Pour faire cette distinction, nous rajouterons dans le chemin jusque là parcouru un astérisque (*) à tout noeud qui a choisi le prochain saut par la règle de la main droite; un noeud sans astérisque a choisi le prochain saut par routage géographique avide.

F. Proposition de protocole

Dans Fig. 4, nous présentons en pseudo-code le protocole de routage associant le routage avide, la règle de la main droite, et l'enregistrement du chemin parcouru dans les paquets. Ce code se déroule à chaque réception de paquet. On suppose que chaque noeud a tiré aléatoirement un identifiant, et que chaque noeud connaît la liste de ses voisins, avec leurs identifiant et leur métrique (hauteur dans la cas de routage par gradient, position dans la cas de routage géographique). Ce protocole est utilisable tel quel dans le cas du routage géographique et par gradient. Ce qui change est l'implémentation de la fonction `voisin_main_droite` et la définition de proche (proximité géographique au puits ou hauteur d'un noeud).

IV. CONCLUSION ET PERSPECTIVES

Dans ce papier, nous mettons en évidence une faille dans la règle de la main droite, qui a été proposée pour le routage géographique comme une solution à la faille du routage géographique avide. Pour répondre à cette faille, ce papier propose l'enregistrement de la route déjà parcourue dans le paquet. La route ainsi enregistrée permet à chaque saut d'éliminer des voisins potentiels. Nous utilisons le protocole présenté au routage géographique et au routage par gradient.

Comme perspectives, nous voulons développer une plateforme de validation formelle de nos protocoles. De plus, nous souhaitons mener une campagne de simulation des protocoles. En particulier, nous voudrions regarder l'efficacité énergétique de tels protocoles qui présentent une phase d'auto-organisation

```

voisins_possibles = voisins
tant que (pas fin entête_chemin)
  si (id_courant==mon_id &&
      id_courant+1 in voisins &&
      id_courant-1 in voisins)
    voisins_possibles -= id_courant+1
    si (asterisque(id_courant-1) == vrai)
      voisins_possibles -= id_courant-1
    fin si
  fin si
  indentifiant_courant++
fintantque

si (voisins_pas_utilisés==NULL)
  détruire_paquet
FIN
finsi

voisin_prochain_saut =
voisin_plus_proche(voisins_pas_utilisés)
si (plus_proche(voisin_prochain_saut)==VRAI)
  rajouter(mon_id, entête_chemin)
  envoi_paquet_à(voisin_prochain_saut)
sinon
  voisin_prochain_saut =
  voisin_main_droite(voisins_pas_utilisés)
  rajouter(mon_id*, entête_chemin)
  envoi_paquet_à(voisin_prochain_saut)
finsi

```

Fig. 4. Pseudo-code du protocole de routage associant routage avide, règle de la main droite et enregistrement de la route.

simple avec d'autres protocoles de routage qui s'appuient sur une phase d'auto-organisation plus lourde.

REFERENCES

- [1] I. F. Akyildiz and I. Kasimoglu, "Wireless sensor and actor networks: research challenges," in *International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. Fort Lauderdale, Florida, USA: IEEE, December 2004.
- [2] N. Mitton, "Large-scale self-organization in wireless multihop networks [in french: Auto-organisation des réseaux sans-fil multi-sauts à grande échelle]," Ph.D. dissertation, INSA de Lyon, March, 28 2006.
- [3] F. Theoleyre and F. Valois, "About the self-stabilization of a virtual topology for self-organization in ad hoc networks," in *7th International Symposium on Self-Stabilizing Systems (SSS)*, Barcelona, Spain, October 2005.
- [4] I. Stojmenovic, M. Seddigh, and J. Zunic, "Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 1, pp. 14–25, January 2002.
- [5] B. Karp and H. Kung, "Gpsr: Greedy perimeter stateless routing for wireless networks," in *Annual International Conference on Mobile Computing and Networking (Mobicom)*. ACM, August 2000, pp. 243–254.
- [6] C. E. Perkins and E. M. Royer, "Ad hoc on-demand distance vector routing," in *Workshop on Mobile Computing Systems and Applications*. New Orleans, USA: IEEE, February 1999, pp. 90–100.
- [7] D. B. Johnson, D. A. Maltz, and J. Broch, *Ad Hoc Networking*. Charles E. Perkins, 2001, ch. DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks, pp. 139–172.
- [8] K.-H. Han, Y.-B. Ko, and J.-H. Kim, "A novel gradient approach for efficient data dissemination in wireless sensor networks," in *IEEE 2004 International Conference on Vehicular Technology Conference (VTC)*, Fall 2004. [Online]. Available: http://dmc.ajou.ac.kr/paper/hkh_vtc04f.pdf
- [9] J. Faruque, K. Psounis, and A. Helmy, "Analysis of gradient-based routing protocols in sensor networks," in *International Conference on Distributed Computing in Sensor Systems (DCOSS)*. Marina del Rey, CA, USA: IEEE/ACM, June 2005, pp. 258–275.
- [10] J. Faruque and A. Helmy, "Gradient-based routing in sensor networks," *ACM SIGMOBILE Mobile Computing and Communications Review, Special feature on MOBICOM 2003 posters*, vol. 4, no. 4, pp. 50–52, october 2003.