

# Un Nouvel Algorithme de Routage Géographique dans les Réseaux de Capteurs.

Julien Champ, Clément Saad

► **To cite this version:**

Julien Champ, Clément Saad. Un Nouvel Algorithme de Routage Géographique dans les Réseaux de Capteurs.. MANifestation des JEunes Chercheurs en Sciences et Technologies de l'Information et de la Communication, Oct 2007, pp.9, 2007, <<http://majestic07.info.unicaen.fr/>>. <lirmm-00194362>

**HAL Id: lirmm-00194362**

**<https://hal-lirmm.ccsd.cnrs.fr/lirmm-00194362>**

Submitted on 6 Dec 2007

**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.

# Un Nouvel Algorithme de Routage Géographique dans les Réseaux de Capteurs.

Julien Champ<sup>1</sup>, Clément Saad<sup>1,2</sup>

<sup>1</sup>Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier <sup>2</sup>Laboratoire d'Informatique d'Avignon

---

julien.champ@lirmm.fr, saad@lirmm.fr

## Résumé:

Ce papier présente un nouvel algorithme de routage géographique pour les réseaux de capteurs. On suppose que chaque capteur connaît la position de l'ensemble des capteurs, y compris la sienne, à une erreur de position près. Un nœud n'est donc pas certain de l'existence d'un lien entre deux capteurs du réseau, sauf s'il est l'un des deux capteurs. On propose un algorithme de routage capable d'optimiser deux critères simultanément : la consommation d'énergie du système et le taux d'acheminement des messages. Les performances sont validées par des simulations confrontant notre méthode à un algorithme optimal du point de vue énergétique.

Mot-clés : Réseaux de capteurs sans fil, erreur de positionnement, dépense d'énergie

## Abstract:

This paper presents a new geographic routing algorithm for wireless sensor networks. It assumes that every sensors know the position of all the sensors, including its, with a positioning error. A node is not sure about the existence of a link between two sensors, excepted if he is one of them. We propose a routing algorithm which simultaneously optimizes two criteria : the energy consumption and the delivery rate. Performances are validated thanks to simulations which compare our method with an optimal algorithm in term of energy consumption.

Keywords : Wireless sensor networks, positioning error, energy consumption

## 1 Introduction

Les progrès réalisés ces dernières années dans les domaines des microsystèmes électromécaniques ainsi que des techniques de communication sans fil ont permis de voir apparaître un nouveau type de réseau : **le réseau de capteurs sans fil**. Ces réseaux sont composés de capteurs qui ont la capacité de récolter diverses données environnementales telles que la température ou la présence d'un gaz. Une fois ces informations récoltées les données sont transmises grâce à un routage multi-sauts vers un «puits» ou «point de col-

lecte» qui traite alors l'information. Les domaines d'applications des réseaux de capteurs sont nombreux [Akyildiz 02] : par exemple la surveillance de départs de feu de forêt ou de mouvements de troupes ennemies. C'est pour cette raison que le MIT's Technology Review [MIT 03] considère les réseaux de capteurs comme l'une des dix technologies qui vont changer le monde.

La problématique du routage est essentielle dans ce type de réseau. Elle doit prendre en compte les caractéristiques des capteurs : chaque capteur est doté d'une unité de détection d'événements, de calcul et de communication. Tous ces composants sont alimentés par une batterie, il est donc nécessaire que les algorithmes de routage proposés utilisent des chemins permettant une dépense d'énergie minimale. Plusieurs solutions ont été proposées pour résoudre ce problème, par exemple [Jacquet 01, Intanagonwiwat 00, Bose 01, Stojmenovic 01, Kuruvila 06]. Parmi ces solutions, certaines utilisent les informations de localisation des capteurs pour effectuer le routage [Bose 01, Stojmenovic 01, Kuruvila 06]. Ces méthodes supposent que chaque capteur a la connaissance exacte de sa position, soit grâce à la technologie GPS (Global Positioning System), soit grâce à des méthodes de localisation telles que [Saad 07a, Saad 07b, Niculescu 01]. De telles méthodes de localisation utilisent soit une estimation de la distance séparant deux capteurs, en fonction des propriétés du signal reçu (temps de propagation, atténuation du signal . . .), soit en fonction de l'angle d'arrivée du message. Les méthodes de localisation utilisent ces connaissances afin d'en déduire la position estimée des nœuds du réseau.

Dans la pratique, les systèmes de localisation ne sont pas forcément précis. Par exemple, dans [Saad 07a, Saad 07b], les auteurs proposent des méthodes de localisation où les capteurs déterminent leurs positions ainsi que leurs bornes d'erreur de positionnement.

Dans ce papier, nous proposons une technique de routage géographique (en utilisant le positionnement des capteurs) prenant en compte l'incertitude des positions des capteurs. Nous supposons qu'un capteur connaît la position ainsi que l'erreur de positionnement de tous les capteurs. À cause des erreurs de positionnement, les liens de communication ne sont pas sûrs. Chaque capteur possède la même vision du réseau, sauf en ce qui concerne son voisinage : la probabilité d'existence de liens avec ses voisins est 1. Les voisins d'un capteur sont les capteurs situés à portée de communication de celui-ci. L'algorithme de routage que nous proposons choisit un chemin composé d'arcs possédant à la fois une bonne probabilité d'existence et un bon rapport entre la dépense d'énergie et le progrès réalisé. La notion de progrès sera définie dans la section 3.

## 2 Hypothèses et modélisation du problème

Dans certaines applications, les capteurs peuvent être mobiles. Ce papier considère uniquement des réseaux de capteurs statiques. Nous supposons que tous les capteurs ont le même rayon d'émission  $r$ , et que les émissions sur le lien radio sont idéales<sup>1</sup>. Si deux capteurs sont séparés par une distance inférieure ou égale à  $r$ , ils peuvent alors communiquer. Cependant, tous les capteurs ont la capacité d'ajuster leur rayon d'émission afin d'émettre sur une portée plus courte et ainsi économiser de l'énergie. On suppose qu'il y a un seul point de collecte dans le réseau et que chaque nœud sait où il est positionné.

Imaginons que les capteurs soient déployés par hélicoptère, il est alors possible de connaître à l'avance leur position finale. Cependant plus les capteurs sont jetés de haut, plus l'erreur de positionnement est grande. Il est possible d'enregistrer avant le déploiement, dans chaque capteur, la position de tous les autres capteurs du réseau ainsi que l'erreur de positionnement associée à chacun d'entre eux.

---

<sup>1</sup>Une émission radio est considérée comme idéale s'il n'y a pas de brouillages dus par exemple à la présence d'obstacles

Considérons un réseau de capteurs. Soit  $V$  l'ensemble des capteurs du réseau. On associe à chaque capteur  $u$  une position estimée  $(X_u, Y_u)$  dans le plan cartésien et une borne d'erreur de positionnement  $\epsilon_u$ . La position réelle du capteur modélisé par le nœud  $u$  est incluse dans le disque de centre  $(X_u, Y_u)$  et de rayon  $\epsilon_u$ . On considère une distribution uniforme : le capteur a autant de chance d'être en chaque point de ce disque. Nous modélisons le réseau par un graphe orienté  $G$  dont l'ensemble des sommets est l'ensemble  $V$ . Il y a un arc entre deux nœuds  $A$  et  $B$  si et seulement si  $dist_{AB} - \epsilon_A - \epsilon_B \leq r$  où  $dist_{AB}$  est la distance euclidienne entre les points de coordonnées  $(X_A, Y_A)$  et  $(X_B, Y_B)$ . Du fait des hypothèses de départ, le graphe  $G$  est symétrique. Notons  $E$  l'ensemble des arcs du graphe  $G$ .

La figure 1 représente un réseau de capteurs sans fil que nous considérons dans cet article, le graphe étant symétrique nous représentons deux arcs  $(A, B)$  et  $(B, A)$  par un segment.

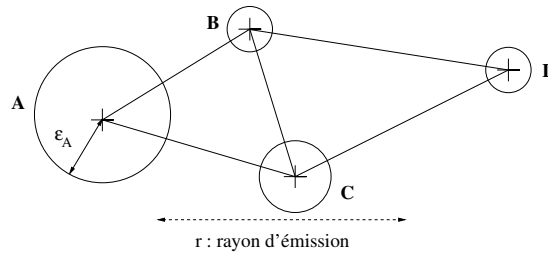


FIG. 1 – Représentation d'un réseau de capteurs sans fil, avec erreurs de positionnement.

### 3 Estimation du coût d'un lien

Dans cette partie, nous allons proposer une métrique afin d'attribuer un coût à chaque arc du graphe. Soit deux capteurs  $A$  et  $B$ . Le coût d'un arc  $(A, B)$  est défini en fonction des critères suivants : la probabilité que la communication soit possible entre les capteurs  $A$  et  $B$ , la dépense d'énergie et le progrès réalisé si le capteur  $A$  transmet le message au capteur  $B$ . Le progrès est la différence entre la distance euclidienne du nœud émetteur au puits, et la distance du nœud récepteur au puits. Dans un premier temps nous allons voir comment estimer la probabilité  $P_{AB}$  d'existence d'un lien de communication entre  $A$  et  $B$ , puis le rapport  $R_{AB}$  entre la dépense d'énergie et le progrès réalisé lors de la transmission d'un message de  $A$  vers  $B$ .

#### 3.1 Estimation de la probabilité d'existence d'un lien

Connaissant la position de chaque capteur à une erreur de position près, et leur rayon d'émission  $r$ , nous proposons dans cette section une fonction permettant d'estimer la probabilité d'existence d'un lien de communication entre deux capteurs  $A$  et  $B$ . Notons  $P_{AB}$  cette probabilité associée à l'arc  $(A, B)$ .

Il est possible de différencier trois cas pour calculer la probabilité  $P_{AB}$  :

- Le cas où les deux capteurs sont placés avec précision (i.e.  $\epsilon_A = \epsilon_B = 0$ );
- Le cas où un seul des deux capteurs est positionné avec une incertitude (par exemple  $\epsilon_B = 0$  et  $\epsilon_A \neq 0$ );
- Le cas où les deux capteurs ont une erreur de positionnement.

Le premier cas est le plus simple. Si les  $\epsilon$  sont nuls cela signifie que les positions de  $A$  et de  $B$  sont exactes. Dans ce cas  $P_{AB} = 1$  si la distance entre  $A$  et  $B$  est inférieure ou égale à  $r$ . Autrement,  $P_{AB} = 0$ .

Le second cas se résoud en effectuant un simple rapport de surface : soit  $S$  la surface de la zone définie par l'intersection du disque centré en  $B$  de rayon  $r$  et celui de  $A$  de rayon  $\epsilon_A$ . La probabilité  $P_{AB}$  est égale à  $\frac{S}{\pi \times \epsilon_A^2}$ .

Le dernier cas est le plus difficile. Il est illustré en figure 2 :

- La surface  $S_1$ , la plus à gauche, correspond aux positions que peut prendre le capteur  $A$  sans pouvoir recevoir les messages émis par  $B$  ;
- La surface  $S_3$ , la plus à droite, correspond, elle, aux positions que peut prendre le capteur  $A$  en garantissant la réception les messages émis par  $B$  ;
- La surface  $S_2$ , comprise entre les deux zones précédentes, correspond aux positions où l'existence, ou non, d'un lien ne peut être garantie. La probabilité d'existence de lien dans cette zone devra être calculée ou estimée.

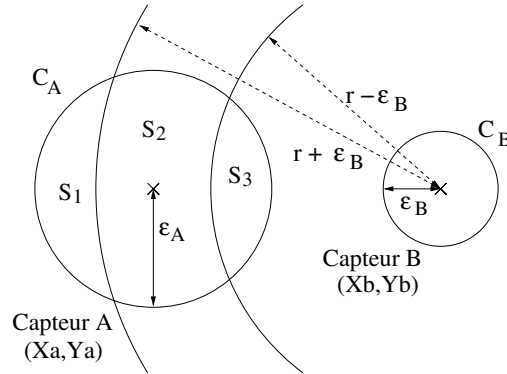


FIG. 2 – Emission de  $B$  vers  $A$  avec  $\epsilon_A$  et  $\epsilon_B > 0$

La probabilité d'existence d'un lien entre les capteurs  $A$  et  $B$  dépend des surfaces  $S_1, S_2, S_3$  ainsi que des probabilités  $P_1, P_2$  et  $P_3$  d'existence du lien dans chacune de ces zones. Le calcul permettant d'obtenir la probabilité que la communication soit possible entre les deux capteurs est le suivant :

$$P_{AB} = \frac{S_1 * P_1 + S_2 * P_2 + S_3 * P_3}{\pi * \epsilon_A^2} \quad (1)$$

Lorsque le capteur  $A$  est dans la surface  $S_3$ , les capteurs  $A$  et  $B$  sont sûrs de pouvoir communiquer,  $P_3 = 1$ . Par contre, la communication sera impossible si  $A$  est positionné dans la surface  $S_1$ , donc  $P_1 = 0$ . Le calcul de la probabilité  $P_{AB}$  se simplifie en :

$$P_{AB} = \frac{S_2 * P_2 + S_3}{\pi * \epsilon_A^2} \quad (2)$$

Le calcul de la probabilité  $P_2$  est plus délicat. Pour le détail des calculs, le lecteur est invité à se reporter à [Champ 07]. Intéressons nous maintenant au calcul du rapport entre la dépense d'énergie et le progrès réalisé lorsqu'on traverse un arc.

### 3.2 Calcul du rapport entre la dépense d'énergie et le progrès réalisé en traversant un arc

Dans cette section, nous utilisons la modélisation énergétique définie dans [Rodoplu 98]. Considérons deux capteurs  $A$  et  $B$  et la distance  $d_{AB}$  qui les sépare. Le coût énergétique  $J_{AB}$  d'un envoi sur le lien  $(A, B)$  est donné par la formule :

$$J_{AB} = d_{AB}^a + c \quad (3)$$

avec  $a$  et  $c$  des constantes dépendantes de l'environnement du réseau. Nous prenons ici  $a = 4$  et  $c = 2 \times 10^8$  comme dans [Rodoplu 98].

Le progrès  $prog_{AB}$  lorsqu'on traverse le lien  $(A, B)$  est donné par la formule :

$$prog_{AB} = d_{A,puits} - d_{B,puits} \quad (4)$$

Considérons le rapport suivant :  $\frac{J_{AB}}{prog_{AB}}$ . Afin de normaliser cette valeur entre 0 et 1 on utilise le rapport  $\frac{J_{opt}}{prog_{opt}}$  qui correspond au meilleur compromis possible entre la dépense d'énergie et le progrès pour un modèle énergétique donné.

Le rapport entre la dépense d'énergie et le progrès réalisé, en traversant un arc, est donné par la formule :

$$R_{AB} = \frac{J_{opt}}{prog_{opt}} \times \frac{prog_{AB}}{J_{AB}} \quad (5)$$

Cependant, le progrès pouvant être négatif,  $R_{AB}$  appartiendrait à l'intervalle  $[-1, 1]$ . Afin que  $R_{AB}$  soit compris dans l'intervalle  $[0, 1]$ , on utilise la formule suivante :

$$R_{AB} = \frac{\left(\frac{J_{opt}}{prog_{opt}} \times \frac{prog_{AB}}{J_{AB}}\right) + 1}{2} \quad (6)$$

Etant donné que  $J_{AB} = J_{BA}$  mais que  $prog_{AB} \neq prog_{BA}$ , alors  $R_{AB} \neq R_{BA}$  pour les arcs  $(A, B)$  et  $(B, A)$ .

### 3.3 Calcul du coût d'un arc

Maintenant que nous savons comment calculer  $P_{AB}$  et  $R_{AB}$ , il nous reste à combiner les deux pour définir le coût de l'arc  $(A, B)$ . Nous cherchons maintenant à pénaliser un arc, en lui attribuant un coût élevé, lorsqu'il ne présente pas un bon compromis entre la probabilité et le rapport dépense d'énergie progrès réalisé en le traversant. Quel que soit  $(A, B) \in E$ , le coût de l'arc  $(A, B)$ , noté  $C_{AB}$ , est donné par la formule suivante :

$$C_{AB} = 1 - [\alpha P_{AB} + (1 - \alpha) R_{AB}] \quad (7)$$

La constante  $\alpha$  comprise entre 0 et 1 permet d'avantager soit la probabilité soit le rapport «dépense d'énergie» / «progrès réalisé» en traversant un arc.

Le coût d'un chemin se calcule en faisant la somme des coûts des arcs appartenant à ce chemin. L'algorithme de routage cherchera à prendre un chemin ayant un coût minimal.

## 4 Principe de l'algorithme

Avant le déploiement, chaque capteur a une vision globale et identique du réseau, il connaît les coûts associés à chacun des arcs. Suite au déploiement, par exemple au moyen d'un hélicoptère, chaque capteur émet un message *HELLO* afin de prendre connaissance de son voisinage et mettre à jour les coûts des arcs avec ses voisins. L'idée générale de l'algorithme est la suivante : lorsqu'un capteur veut envoyer un message il calcule, grâce à sa connaissance du réseau, le meilleur chemin (i.e. celui possédant le plus faible coût) entre lui et la destination et il transmet le message au nœud suivant dans le chemin.

Dans un premier temps, un capteur calcule pour chaque voisin  $v$  le coût du meilleur chemin pour atteindre la destination en passant par  $v$ . Ces coûts sont stockés dans la table *Tab\_Couts* qui associe un voisin à son coût. Lorsqu'un capteur désire transmettre un message au point de collecte, il sélectionne le voisin par lequel passe le chemin de plus faible coût. Il lui transmet un message contenant les champs suivants<sup>2</sup> :

<sup>2</sup>La destination peut être ajoutée au message dans le cas où il y a plusieurs points de collecte différents.

- la position du capteur qui a détecté l'événement,
- le coût de son chemin de secours (un chemin de secours est calculé comme étant le deuxième meilleur chemin),
- des informations sur l'événement détecté.

Lors de la réception par  $u$  d'un message envoyé par  $v$ ,  $u$  déduit qu'il est le meilleur choix de  $v$ . Il met à jour sa table  $Tab\_Couts$ , en changeant le coût associé à l'expéditeur  $v$  par le coût du chemin de secours inclu dans le message auquel il additionne le coût  $C_{uv}$ . À son tour,  $u$  va sélectionner le voisin correspondant à son meilleur chemin, auquel il transmet le message, en y ajoutant la valeur de son chemin de secours.

On constate que le chemin choisi au départ n'est pas forcément respecté par les nœuds chargés de faire suivre le message car chacun prendra une décision en fonction de ses propres connaissances. C'est ce genre de comportement qui peut entraîner l'apparition de boucles.

**Gestion des boucles :** Afin d'empêcher qu'un message tourne en boucle, nous prenons la décision suivante : si le capteur  $A$  a envoyé le message au capteur  $B$ , et que le message revient à  $A$  par un autre capteur que  $B$ , cela signifie qu'il y a une boucle. Le capteur  $A$ , en s'interdisant de renvoyer le message à  $B$  empêche que cette boucle se reproduise.

**Détection de pannes :** En supposant qu'un capteur envoie périodiquement à son voisinage un message  $I\_AM\_ALIVE$ , il est possible de détecter les pannes de capteurs. Lorsqu'un capteur tombe en panne, ses voisins détectent cet événement au bout d'un certain temps (i.e. pas de réception de message  $I\_AM\_ALIVE$  en provenance de ce capteur) et mettent à jour leurs données relatives à ce capteur. Cette détection se fait localement et n'est pas propagée dans le réseau, mais permet de prendre en compte la panne d'un capteur.

---

**Algorithme 1 :** Lorsque le capteur  $u$  reçoit le message  $Msg$  envoyé par  $v$

---

```

/* Msg = < Expéditeur, CoutExp,D > */
/* Expéditeur l'identifiant de l'expéditeur et CoutExp,D le coût de son chemin de secours */
/* MettreAJourCout(voisin,cout) Met à jour la table Tab_Couts. */
/* Interdire(voisin) Interdit au capteur de choisir voisin comme destinataire. */
/* Stocker(Msg) Mémoire le message Msg. */
/* PasRecu(Msg) Retourne Vrai si Msg est reçu pour la première fois, faux sinon. */
/* CalculMeilleurChemin() Retourne parmi tous les voisins non interdits celui qui a le meilleur chemin, -1 s'il y en a aucun. */
/* CalculCoutCheminSecours() Retourne parmi tous les voisins non interdits, le coût du 2ème meilleur chemin, -1 s'il y en a aucun. */

if Coutv,D != -1 then
  MettreAJourCout(v, CoutExp,D + Coutv,D)
else
  Interdire(v) // Celui qui nous a envoyé Msg n'a plus de chemin à proposer
if (PasRecu(Msg)) then
  Stocker(Msg)
else
  if (v ≠ MeilleurVoisin) then
    Interdire(MeilleurVoisin) // On interdit MeilleurVoisin, à l'origine de la boucle
MeilleurVoisin = CalculMeilleurChemin()
Msg.Coutu,D = CalculCoutCheminSecours()
if (MeilleurVoisin ≠ -1) then
  Envoyer(MeilleurVoisin, Msg)
else
  Abandon

```

---

## 5 Résultats de simulation

Afin de mesurer l'intérêt de la méthode que nous proposons, nous avons implémenté cet algorithme dans un logiciel développé par nos soins. La couche MAC est considérée comme étant idéale, et le canal radio comme étant parfait : tout message émis arrive correctement à la destination (pas de perte de paquets).

Nous avons généré des topologies de réseaux contenant 100 capteurs à l'intérieur d'un carré de taille 1200 par 1200. Afin d'avoir une densité qui varie entre 6 et 20 nous faisons varier le rayon d'émission maximum  $r$  du capteur. Pour chaque nœud  $u \in V$ , avec  $V$  l'ensemble des nœuds, on tire aléatoirement et en suivant une distribution uniforme :

- la position estimée  $(X_u, Y_u)$  de  $u$  dans le carré de taille  $1200 \times 1200$ ,
- son erreur de positionnement  $\epsilon_u$  compris entre 0 et  $r$ ,
- sa position réelle à l'intérieur du cercle centré en  $(X_u, Y_u)$  et de rayon  $\epsilon_u$ .

La position du point de collecte est elle aussi tirée aléatoirement, mais sans erreur de positionnement. Pour chacune des densités nous générons 100 topologies, dans nos scénarios tous les capteurs émettent un message en direction de la station de base. Pour comparer notre algorithme, chaque routage est comparé au routage dans la même topologie en ayant une connaissance exacte de la présence ou non des liens (i.e. on calcule le plus court chemin en terme d'énergie par exemple grâce à l'algorithme de Dijkstra).

Nous nous intéressons à plusieurs points dans ces simulations. Le tableau 3, nous indique la valeur à choisir pour le paramètre  $\alpha$ , de manière à avoir 99%, si possible, des messages qui arrivent à destination, tout en ayant une dépense d'énergie raisonnable.

Densité	6	8	10	12	14	16	18	20
$\alpha$	0.1	0.3	0.6	0.7	0.7	0.7	0.8	0.8
% de messages à destination	94	99	99	99	99	99	100	100

FIG. 3 – Choix du paramètre  $\alpha$  en fonction de la densité.

Les figures 4 et 5 nous permettent de voir l'influence du paramètre  $\alpha$  dans les réseaux de densité 6. La première figure analyse l'impact sur le taux d'acheminement des messages et la deuxième, l'impact sur la dépense d'énergie. Il est important de noter que dans un réseau de densité 6 plus le paramètre  $\alpha$  est bas, plus le taux d'acheminement est haut. Inversement, plus le paramètre  $\alpha$  est haut, plus la consommation d'énergie est basse.

Sur la figure 6, nous faisons varier la densité du réseau et nous étudions le pourcentage d'énergie nécessaire en plus de la dépense d'énergie optimale. Pour chaque densité nous choisissons le meilleur paramètre  $\alpha$  d'après le tableau 3.

Nous pouvons remarquer que dans notre méthode la dépense d'énergie décroît lorsque la densité du réseau augmente. Même dans des réseaux de faible densité (i.e. densité égale à 6) notre algorithme consomme moins de 30% d'énergie en plus de la dépense d'énergie optimale. Dans des réseaux à forte densité (i.e. dont la densité est supérieure ou égale à 14), la dépense d'énergie supplémentaire par rapport à la dépense d'énergie optimale est de moins de 20%.

Pour conclure, nous pouvons noter, grâce à ces simulations, que pour avoir un réseau où la majorité des messages arrivent à destination (i.e. plus de 99% des messages) tout en ayant une dépense d'énergie raisonnable, il est possible d'utiliser des réseaux de densité supérieure ou égale à 8. L'utilisation de réseaux de densité supérieure permet alors de réduire la dépense totale d'énergie.



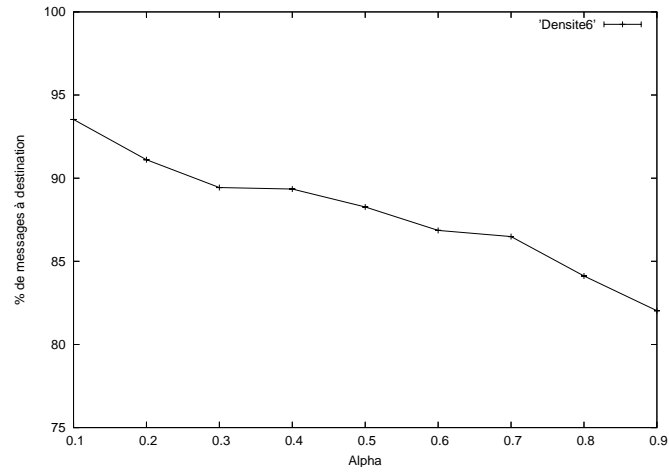


FIG. 4 – Influence de  $\alpha$  sur le taux d'acheminement dans des réseaux de densité 6.

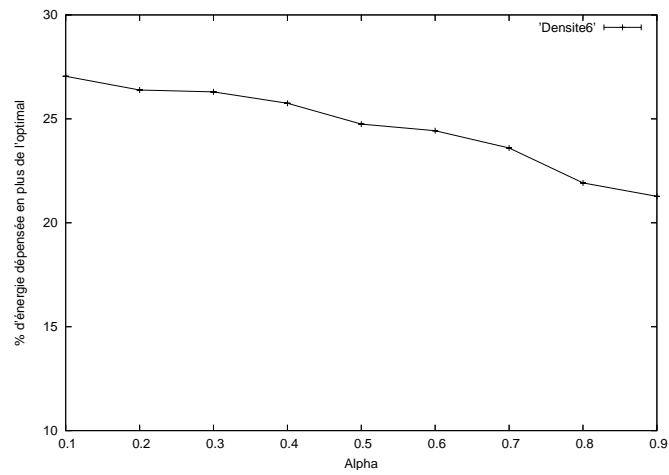


FIG. 5 – Influence de  $\alpha$  sur la consommation d'énergie dans des réseaux de densité 6.

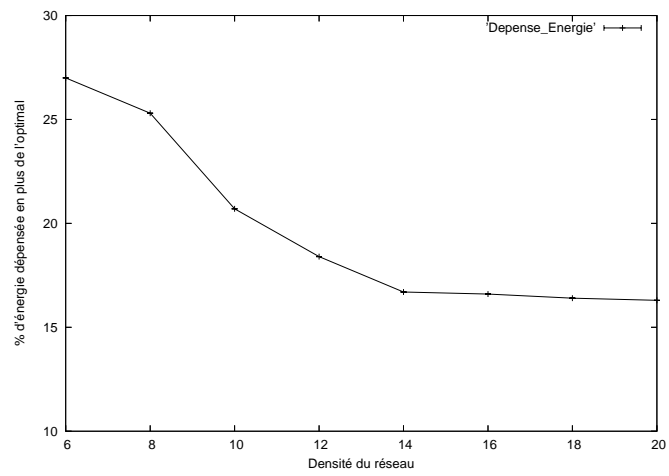


FIG. 6 – Comportement de la dépense d'énergie lorsque la densité varie.

## 6 Conclusion

Ce papier présente un algorithme de routage géographique prenant en compte les erreurs de positionnement des capteurs, ce qui engendre des incertitudes sur l'existence de liens entre les nœuds. Cet algorithme achemine les messages le long des chemins ayant le meilleur compromis entre les probabilités d'existence des liens et la consommation d'énergie. Il a notamment la propriété d'être résistant aux pannes.

Certaines améliorations peuvent être apportées à cette méthode. Dans le but d'augmenter la durée de vie du réseau, si un capteur connaît le pourcentage de batterie restant de ses voisins, il peut pénaliser celui ou ceux qui ont une batterie faible, de manière à ne pas toujours passer par les mêmes capteurs. La gestion des boucles peut elle aussi être améliorée. En interdisant un lien à l'origine d'une boucle, on ne peut pas garantir que les messages arrivent à destination s'il existe un chemin entre la source et la destination. Il serait intéressant de gérer plus finement les boucles. Enfin, la connaissance intégrale du réseau par un capteur n'est pas nécessaire. Un capteur peut définir un sous graphe du réseau lui permettant d'atteindre la destination. L'étude de cette restriction est en cours.

**Remerciements :** Les auteurs tiennent à remercier Jean-Claude König et Jérôme Palaysi pour leurs conseils, et leurs remarques constructives.

## Références

- [Akyildiz 02] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam & E. Cayirci. *Wireless sensor networks : a survey*. Computer Networks, vol. 38, no. 4, pages 393–422, 2002.
- [Bose 01] Prosenjit Bose, Pat Morin, Ivan Stojmenovic & Jorge Urrutia. *Routing with Guaranteed Delivery in Ad Hoc Wireless Networks*. Wireless Networks, vol. 7, no. 6, pages 609–616, 2001.
- [Champ 07] J. Champ. *Routage dans les réseaux Ad hoc*. Mémoire de Master Recherche, LIRMM, Juin 2007.
- [Intanagonwiwat 00] Chalermek Intanagonwiwat, Ramesh Govindan & Deborah Estrin. *Directed diffusion : a scalable and robust communication paradigm for sensor networks*. In Mobile Computing and Networking, pages 56–67, 2000.
- [Jacquet 01] P. Jacquet, P. Mühlthaler, T. Clausen, A. Laouiti, A. Qayyum & L. Viennot. *Optimized link state routing protocol for ad hoc networks*. In Proceedings of the 5th IEEE Multi Topic Conference (INMIC 2001), 2001.
- [Kuruwila 06] Kuruwila, Nayak & Stojmenovic. *Progress based localized power and cost aware routing algorithms for ad hoc and sensor wireless networks*. International Journal of Distributed Sensor Networks, vol. 2, no. 2, pages 147–159, 2006.
- [MIT 03] MIT. *10 Emerging Technologies That Will Change the World*. MIT Technology Review, Feb. 2003.
- [Niculescu 01] D. Niculescu & B. Nath. *Ad Hoc Positioning System (APS)*. In Proc. IEEE GlobeCom, San Antonio, AZ, November 2001.
- [Rodoplu 98] V. Rodoplu & T.H. Meng. *Minimum energy mobile wireless networks*. In Proc. of IEEE International Conference on Communications (ICC), '98, volume 3, pages 1633 – 1639, 1998.
- [Saad 07a] Clément Saad, Abderrahim Benslimane, Jean-Claude König & Jacques Turbert. *AT-Free : A Preliminary Method for Localization Techniques in Sensor Networks*. 7th International Conference on New Technologies of Distributed Systems, 2007.
- [Saad 07b] Clément Saad, Abderrahim Benslimane & Jean-Claude König. *AT-Dist : A Distributed Method For Localization With High Accuracy in Sensor Networks*. Special Issue on "Wireless Ad Hoc and Sensor Networks" in the international journal Studia Informatica Universalis (To Appear), 2007.
- [Stojmenovic 01] Ivan Stojmenovic & Xu Lin. *Power-Aware Localized Routing in Wireless Networks*. IEEE Transactions on Parallel and Distributed Systems, vol. 12, no. 11, pages 1122–1133, 2001.