



**HAL**  
open science

## Un algorithme de gestion des adjacences pour les réseaux ad-hoc mobile

Husnain Ali, Amina Naimi, Anthony Busson, Veronique Veque

► **To cite this version:**

Husnain Ali, Amina Naimi, Anthony Busson, Veronique Veque. Un algorithme de gestion des adjacences pour les réseaux ad-hoc mobile. 10ème Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel'08), 2008, Saint-Malo, France. pp.49-52. inria-00374454

**HAL Id: inria-00374454**

**<https://inria.hal.science/inria-00374454v1>**

Submitted on 8 Apr 2009

**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 algorithme de gestion des adjacences pour les réseaux ad-hoc mobile.*

Husnain Ali, Amina Naimi, Anthony Busson et Véronique Vèque

*Institut d'Electronique Fondamentale  
Bat 220, Université Paris-Sud, Centre Scientifique d'Orsay  
91405 cedex Email : prenom.nom@ief.u-psud.fr*

---

Dans cet article nous présentons un algorithme de gestion des adjacences dans les réseaux ad hoc. Cet algorithme permet à un nœud de lister les voisins. Notre algorithme améliore les mécanismes utilisés aujourd'hui, entre autres dans OLSR, car il prend en compte la puissance du signal des messages reçus par les voisins. Ces informations permettent d'anticiper les ruptures de liens et de garantir des liens de bonne qualité. Une comparaison par simulation de l'algorithme proposé et de l'algorithme défini dans le RFC de OLSR montre que, d'une part on améliore nettement le taux de délivrance des paquets, et que d'autre part notre algorithme aide à stabiliser le réseau.

**Keywords:** réseau Ad Hoc, OLSR, Gestion de la mobilité et réseaux sans fil

---

## 1 Introduction

Un réseau ad hoc est un réseau où aucune infrastructure est requise pour que les nœuds puissent communiquer. Les nœuds sont équipés d'interfaces sans fil configurées en mode ad-hoc leur permettant de communiquer directement entre eux. Les nœuds n'étant pas à portée radio sont alors atteints via le relaiage des autres nœuds. Tous les nœuds sont alors des routeurs acheminant les données pour les autres nœuds. La mobilité des nœuds est alors gérée au niveau réseau. Lorsqu'un nœud se déplace, les protocoles de routage proactif sont censés prendre en compte la nouvelle topologie et mettre à jour les tables de routage de l'ensemble des nœuds. Comme pour quasiment tous les protocoles de routage, les informations de routage sont basées sur la connaissance du voisinage. Le voisinage étant défini comme l'ensemble des nœuds à portée radio. Par exemple, avec le protocole OLSR [1], une partie des nœuds annonce la liste de leurs voisins dans tout le réseau. Ces informations permettent alors à chaque nœud de connaître la topologie du réseau (au moins partiellement) et de calculer les routes vers les différents nœuds.

Dans ce contexte, les informations sur le voisinage sont des données sensibles et doivent être les plus précises possibles. Malheureusement, les protocoles de routage sont relativement indépendants de la technologie sous-jacente. Un nœud établit alors son voisinage uniquement en fonction du fait qu'il reçoit ou non des messages de ses voisins. Ces messages sont appelés des HELLOs ; ils sont envoyés périodiquement en broadcast avec une portée d'un seul saut. Une adjacence est supposée exister avec un voisin donné si on reçoit des HELLOs de ce voisin. Dûes aux collisions qui peuvent se produire dans les réseaux sans fil, on tolérera l'absence de quelques HELLOs (généralement 2 ou 3) avant de considérer qu'un voisin a disparu du voisinage. Cette approche relativement efficace dans un contexte statique est assez catastrophique dans un milieu mobile. Lorsqu'un nœud sort de la portée radio d'un autre nœud, il faudra attendre deux ou trois périodes de HELLO avant de considérer que le nœud ne se trouve plus dans le voisinage. Durant tout ce temps le lien est supposé exister ; tous les paquets envoyés sur ce lien sont alors perdus.

Il est donc important d'améliorer la gestion de ces adjacences de manière à ce que les modifications dans le voisinage soient prises en compte le plus rapidement possible. Il n'existe pas une solution universelle à ce problème. Les solutions dépendent du type de mobilité (rapide ou non), de l'environnement (outdoor/indoor où les liens n'apparaissent et disparaissent pas du tout de la même manière) et des technologies radio. Nous présentons dans cet article un algorithme efficace de gestion des adjacences pour les réseaux mesh [2] a priori en environnement outdoor et utilisant la technologie 802.11. Un réseau mesh est un réseau ad-hoc

où certains nœuds sont déployés à seule fin de relayer les informations pour les autres nœuds. Les nœuds "mesh" sont alors supposés relativement statiques et les autres assez mobiles. Ce contexte particulier est dû au fait que ce travail est effectué dans le cadre du projet ANR RNRT R2M.

Dans la section suivante (section 2) nous présentons l'algorithme de gestion des liens. Nous l'évaluons au travers de simulations dans la section 3 et concluons section 4.

## 2 Présentation de l'algorithme

L'algorithme a déjà été présenté dans un précédent article [4]. Plutôt que de donner les détails de l'algorithme assez peu lisible, nous avons choisi de décrire ici l'algorithme de manière plus pédagogique. Certains détails sont donc omis. L'idée est la suivante. Nous utilisons, comme dans OLSR, une métrique sans unité décrivant la qualité du lien. Lorsque cette métrique passe un seuil haut cela valide le lien, et lorsque cette métrique descend en deçà d'un seuil bas, cela invalide le lien. On a donc un hystérésis.

Mais contrairement aux protocoles de routage classiques et à OLSR en particulier, cette métrique ne sera pas évaluée uniquement à partir de la réception ou non réception d'un HELLO, mais prendra en compte la puissance du signal radio avec laquelle ces HELLOs ont été reçus. C'est donc une technique se basant sur des informations de la couche 2, donc dite "cross-layer" qui est utilisée ici. Cette puissance du signal permet à la fois d'anticiper le fait qu'un lien va se valider (du fait que la puissance augmente lors de la réception des HELLOs successifs) mais aussi et surtout d'anticiper les ruptures de lien. En effet, lorsque l'on observera une dégradation du signal, on diminuera la métrique relative à la qualité du lien de manière à invalider le lien avant que le nœud/ex-voisin ne soit plus à portée radio.

On introduit alors deux seuils relatifs à la puissance du signal  $SHaut$  et  $SBas$ .  $SHaut$  est le seuil de bonne réception. Lorsque les HELLOs sont reçus avec une puissance supérieure à ce seuil, l'émetteur est considéré suffisamment proche et la métrique de qualité du lien est augmentée. Lorsque la puissance du signal du HELLO reçue est inférieure au seuil  $SBas$ , le HELLO est ignoré (c'est comme s'il avait subi une collision ou que l'émetteur était au delà de la portée radio) et la métrique de qualité est diminuée. Lorsque la puissance du signal du HELLO est entre ces deux seuils, on compare cette puissance avec les précédents HELLO, s'il y a une nette amélioration, on augmente la métrique de qualité (le nœud est a priori en train de se rapprocher). Plus précisément, une variable *somme* est utilisée pour cumuler les améliorations/détériorations successives des HELLOs. Lorsque cette variable dépasse en valeur absolue un seuil  $\Delta$ , c'est qu'il y a eu une nette amélioration/détérioration et on augmente/diminue la métrique du lien en conséquence.

On note  $MQ$  la métrique de qualité du lien pour un voisin donné. L'algorithme est le suivant. Lorsque l'on reçoit un HELLO provenant d'un voisin, on considère la puissance du signal  $S$  associée à la réception de ce HELLO :

- Si  $S > SHaut$ ,  $MQ$  est augmenté
- Si  $S < SBas$ ,  $MQ$  est diminué
- Si on est entre les deux seuils :
  - Et que le lien n'est pas encore valide. On rajoute à la variable *somme* l'écart avec le précédent HELLO. Si la somme des améliorations est plus grande que  $\Delta$  ( $somme > \Delta$ ), on augmente  $MQ$  et on réinitialise *somme* ( $somme = 0$ ).
  - Et que le lien est déjà valide. On rajoute à la variable *somme* l'écart avec le précédent HELLO. Si la somme des détériorations est plus grande que  $\Delta$  ( $somme > \Delta$ ) on diminue  $MQ$  et on réinitialise *somme* ( $somme = 0$ ).
- Dans les autres cas, on ne change pas la valeur de la métrique du lien  $MQ$ .

## 3 Résultats de simulation

Pour nos simulations nous avons utilisé le protocole OLSR comme protocole de routage et le simulateur NS2 [3] comme simulateur de réseau. Nous avons intégré au code d'OLSR dans NS2 le nouvel algorithme. Nous considérons deux scénarios. Un premier scénario où les nœuds "mesh" statiques sont placés comme une chaîne de nœud (alignés dans le plan et régulièrement espacés). Un unique nœud mobile se déplace alors

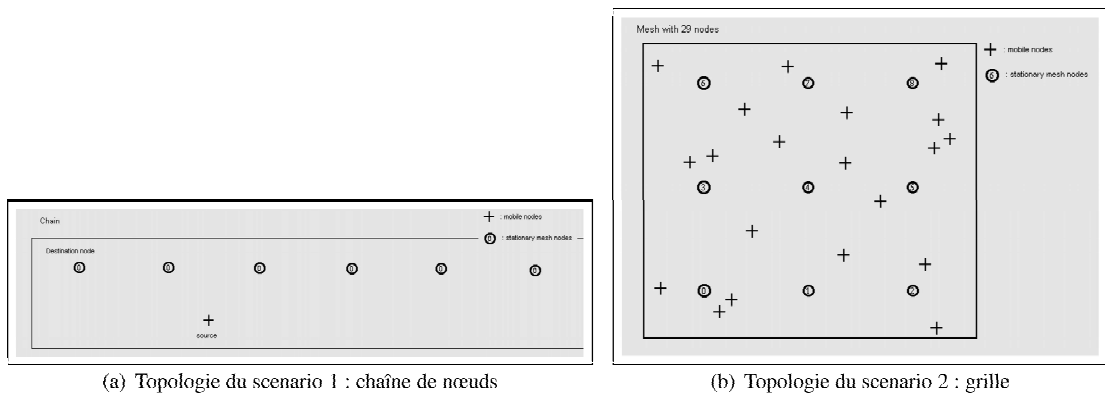


FIGURE 1: Topologies des scenarios 1 et 2.

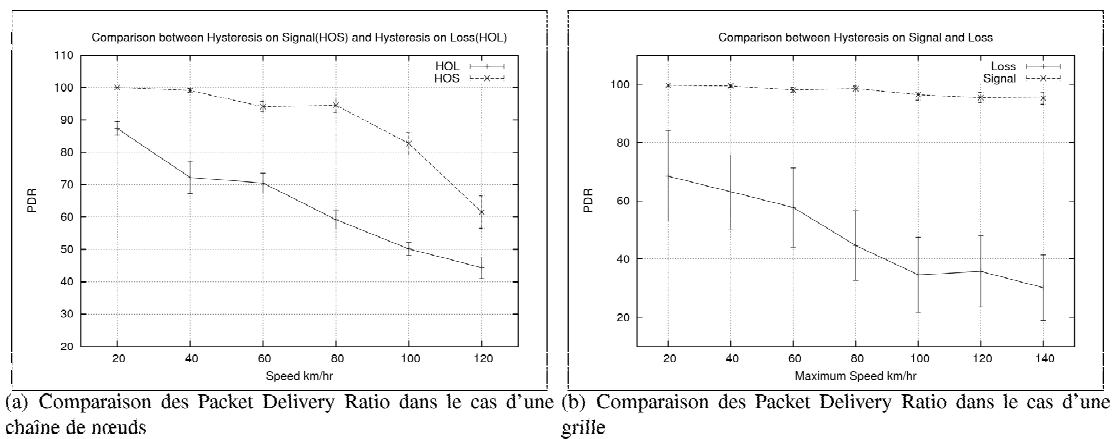


FIGURE 2: Comparaison des PDR pour les deux scenarios.

devant cette chaîne de nœuds. Le nœud mobile envoie alors un flux UDP à débit constant sur le premier nœud de la chaîne. Ce flux UDP nous permet de mesurer le taux de délivrance (PDR : Packet Delivery Ratio), autrement dit l'opposé du taux de perte. La topologie pour ce premier scenario est représentée figure 1(a). On compare alors le PDR obtenu avec le mécanisme (hystérésis basés uniquement sur les pertes de HELLO) de OLSR et notre algorithme. Cette comparaison représentée sur la figure 2(a) est effectuée en faisant varier la vitesse du nœud mobile. Il apparaît clairement que notre algorithme présente de meilleurs résultats. Il est important de noter que la distance entre les nœuds fixes et les différents paramètres de l'algorithme ont été fixés pour que celui-ci fonctionne jusqu'à une vitesse de 60 km/h. C'est la raison pour laquelle le PDR commence à se dégrader à cette vitesse.

Le deuxième scenario correspond à une topologie de réseau où les nœuds fixes forment une grille. La topologie est représentée figure 1(b). La grille est formée de 9 nœuds fixes. Une vingtaine de nœuds mobiles se déplacent dans cette grille suivant le modèle de mobilité **Random WayPoint Model (RWP)**. En deux mots, pour chaque nœud, on choisit au hasard une destination dans la fenêtre d'observation et une vitesse uniformément réparti entre 0 et une vitesse maximale. Le nœud se déplace alors à vitesse constante vers son point destination. Quand il y est, on recommence.

On fait varier la vitesse maximale des nœuds de 20 à 140 km/h. On choisit au hasard une source et une destination parmi les 20 nœuds mobiles. On applique un flux UDP entre ces deux nœuds et on calcule le PDR pour ce flux. Comme pour le scenario précédent on compare le mécanisme d'OLSR et notre propre mécanisme. On observe sur la figure 2(b) que notre algorithme améliore nettement le PDR.

Pour une vitesse maximale des nœuds constante, nous augmentons alors le nombre de sources de 1 à

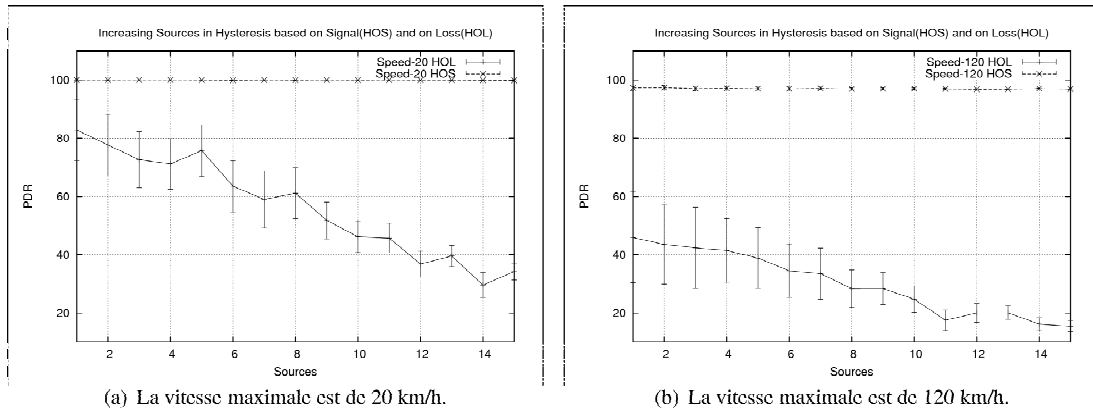


FIGURE 3: Comparaison du PDR lorsque le nombre de sources augmente. La topologie est la grille.

15. Sur les figures 3(a) et 3(b), nous avons représenté le PDR en fonction du nombre de sources pour deux vitesses maximales de 20 km/h et 120 km/h. Il apparaît alors que le PDR dans notre cas est stable avec le nombre de sources. Au contraire, avec l’algorithme d’OLSR, le PDR chute avec le nombre de sources. Ceci montre que le mécanisme initial de gestion des adjacances est très sensible à la charge. Une étude approfondie des traces a montré que la gestion des adjacances lorsqu’elle est peu précise amène des erreurs de routage et même des boucles. En effet les adjacances étant obsolètes, les informations sur la topologie le sont aussi. A noter que ce phénomène avait déjà été souligné dans [5]. Dans nos simulations, ce sont ces boucles qui créent et qui expliquent l’effondrement du PDR.

## 4 Conclusion

Dans cet article nous avons présenté un nouvel algorithme de gestion des adjacances dans les réseaux ad hoc. Cet algorithme prend en compte la puissance du signal des messages reçus par les voisins pour assurer une certaine qualité aux liens et pour anticiper les ruptures de ces liens. Au travers de simulations, on a montré que cet algorithme améliorait significativement le taux de réception dans le réseau. De plus, il s’avère que notre algorithme améliore nettement la capacité du réseau. En effet, une mauvaise gestion des adjacances conduit à la la création de boucles qui effondre les capacités du réseau.

## Références

- [1] Thomas Clausen (ed) and Philippe Jacquet (ed), *Optimized Link State Routing protocol (OLSR)*, October 2003, RFC 3626, Experimental.
- [2] I. F.Akyildiz and X. Wang., *A survey on wireless mesh networks.*, September 2005, IEEE Communications magazine, 43(9) :S23–S30.
- [3] K. Fall and K. V. Editor., *The ns manual.*, January 2005, UC Berkeley and LBL and USC/ISI and Xerox PARC, <http://www.isi.edu/nsnam/ns/ns-documentation.html>.
- [4] A. Busson V. Vèque. H. Ali, A. Meraihi Naimi, *Signal strength based link sensing for mobile ad hoc networks.*, October 2007, The 5-th ACM International Workshop on Mobility Management and Wireless Access, Chania, Crete Island Greece.
- [5] F. Filali. J. Haerri, C. Bonnet, *Olsr and mpr, mutual dependences and performances.*, June 2005, 4th Annual Mediterranean Ad Hoc Networking Workshop, MedHocNet’2005, Porquerolles, France.