Utilisant l'inondation contrôlée pour la découverte de routes dans les réseaux ad hoc

Luis H. M. K. Costa* — **Marcelo D. de Amorim**** — **Serge Fdida****

* *GTA/COPPE – Universidade Federal do Rio de Janeiro*
P.O. Box 68504 – 21945-970 – Rio de Janeiro – RJ – Brasil
luish@gta.ufrj.br [1]

** *LIP6 – Université Pierre et Marie Curie*
8, rue du Capitaine Scott – 75015 – Paris – France
{Marcelo.Amorim, Serge.Fdida}@lip6.fr

ABSTRACT. *Ad hoc routing protocols often use broadcast for route discovery. This can be inefficient if the path between any source-destination pair brokes frequently. We propose and evaluate a simple mechanism that allows fast route repair in on demand ad hoc routing protocols. We apply our proposal in the context of the Ad hoc On-demand Distance Vector (AODV) routing protocol. The proposed Controlled Flooding (CF) framework is used to build alternative routes around the main original path. With alternative routing, data packets are forwarded through a secondary path without requiring the source to re-flood the whole network, as often the case of AODV. We show that our proposal reduces the connection disruption probability as well as the frequency of broadcasts.*

RÉSUMÉ. *Les protocoles de routage ad hoc utilisent très souvent la transmission en broadcast pour la découverte de routes. Dans beaucoup de cas, cette technique peut s'avérer très inefficace. Dans cet article, nous proposons l'inondation contrôlée (CF – Controlled Flooding), qui permet d'accélérer la réparation des routes dans les protocoles de routage ad hoc réactifs. Nous appliquons la technique proposée au protocole de routage AODV. En utilisant CF, des routes alternatives sont construites autour de la route principale. Ces routes sont utilisées à la fois comme chemin de backup et comme des sentiers pour la reconstruction des routes, évitant ainsi l'inondation du réseau avec des requêtes de routes. Nous démontrons au travers d'un simulateur réseau que notre mécanisme peut réduire les temps de déconnexion ainsi que le nombre de paquets de contrôle nécessaires au protocole de routage.*

KEYWORDS: *Controlled flooding, alternative routing,connectivity maintenance, broadcast.*

MOTS-CLÉS : *Inondation contrôlée, routes alternatives, maintien de la connectivité, broadcast.*

1. Nous remercions la CAPES/PRODOC pour le financement de Luís Costa.

1. Introduction

Très souvent, les protocoles de routage ad hoc à la demande émettent des messages en broadcast pour la découverte de routes [Perkins et al., 2002, Johnson et al., 2002]. Le protocole DSR (*Dynamic Source Routing protocol*) [Johnson et al., 2002] utilise le routage à la source, où le nœud d'origine déclare dans l'entête des paquets toute la séquence des nœuds à traverser jusqu'à la destination. Pour découvrir un chemin vers la destination, la source inonde le réseau avec des messages RREQ (Route REQuest) jusqu'à ce que la destination soit atteinte. Au fil de la traversée du réseau, les paquets RREQ enregistrent l'identifiant des nœuds traversés. Par la suite, la destination peut envoyer un message RREP (Route REPly) en utilisant le chemin inverse. Le protocole AODV utilise le même mécanisme de requête-réponse, mais réalise la transmission des données nœud à nœud, avec des tables de routage qui gardent le prochain saut pour la destination. Par ailleurs, quand un nœud AODV reçoit un message RREQ, il garde le chemin inverse vers le nœud source. Quand la destination reçoit un message RREQ, elle envoie un RREP en utilisant la route inverse laissée par la requête.

Malgré la simplicité du broadcast pour la découverte de routes, ce mécanisme présente des inconvénients, comme la redondance, la contention et les collisions. Différentes techniques ont été proposées afin d'augmenter l'efficacité des protocoles basés sur l'inondation de requêtes [Ko et al., 2000, Castaneda et al., 1999, Park et al., 1997, Royer et al., 1999]. Néanmoins, peu d'attention a été prêtée au mécanisme de broadcast lui-même. Le broadcast est en effet difficile à éviter si la source n'a aucune idée sur la localisation de la destination. Néanmoins, l'inondation du réseau entier pour chaque découverte de route est inefficace. Dans cet article, nous argumentons que même si le broadcast est nécessaire pour la première recherche de route, il pourrait être évité pour les recherches suivantes [Costa et al., 2002]. Pour ce faire, nous proposons la technique d'inondation contrôlée (CF – *Controlled Flooding*) qui a pour but la redécouverte d'une route sans avoir à inonder le réseau entier. CF peut être utilisé pour la découverte et réparation de routes. Pour la découverte des routes, le message RREQ est "dirigé" vers la localisation probable de la destination (supposons que la destination a bougé, ses nœuds voisins gardent une piste sur sa nouvelle localisation).

Dans ce travail, nous appliquons la technique CF au mécanisme de réparation de routes du protocole à la demande AODV, mécanisme directement lié au maintien de la connectivité. Avec la mobilité, la communication entre deux nœuds peut être souvent coupée, ce qui nuit aux applications sensibles au délai une fois que les paquets doivent être stockés, voir jetés, le temps qu'un autre chemin soit établi. Le maintien des chemins alternatifs serait l'idéal pour minimiser la période de déconnections. Dans des protocoles proactifs, la construction de chemins alternatifs est plus simple, car des tables de routage avec toutes les destinations du réseau. Pour les protocoles réactifs, des mécanismes supplémentaires sont nécessaires car il n'existe, a priori, qu'une route par destination active. Dans notre approche, le réseau établit des routes alternatives "autour" de la route principale.

Nous évaluons notre approche dans des différents scénarios, où le nombre de nœuds, la densité et le facteur de mobilité sont variés. Le protocole AODV-CF (qui incorpore notre mécanisme) est comparé avec AODV d'origine en termes du temps de déconnexion et de la fréquence d'inondations de tout le réseau. Les résultats montrent les avantages de l'implantation du mécanisme proposé dans un protocole de routage ad hoc à la demande tel que AODV.

Cet article est ainsi organisé. Dans la Section 2 nous décrivons l'approche proposée, CF, et la comparons avec la technique ERS (Expanding Ring Search). Les détails de l'implémentation des chemins alternatifs sont présentés dans la Section 3. La Section 4 présente la comparaison entre AODV-CF et AODV et la Section 5 présente les conclusions et directions possibles dans cet axe de recherche.

2. L'inondation contrôlée

La technique d'inondation contrôlée (CF) a pour but de réduire le trafic de contrôle normalement utilisé par des algorithmes de routage ad hoc pour la réparation de routes. CF augmente la probabilité qu'un chemin jusqu'à la destination soit retrouvé après une éventuelle faille de la route principale. CF peut être également utilisé pour la construction de routes alternatives (ou de *backup*) au cas de rupture de la route principale. Cette approche est particulièrement intéressante si le chemin source-destination est long et le mobiles bougent régulièrement. Dans ces deux cas, la probabilité qu'une route soit perdue est plus importante. Si le mécanisme de redécouverte de routes passe mal à l'échelle (comme par exemple l'inondation), le système global peut aboutir à des performances très pauvres. Avec l'inondation contrôlée, le nœuds gardent quelques états supplémentaires mais la charge de signalisation tende à être plus faible.

L'idée principale de l'algorithme CF est basée sur le multicast orienté (OM - *Oriented Multicast*), proposé par Magoni et Pansiot dans [Magoni et al., 2002]. Dans OM, les arbres multicast sont utilisés pour guider la recherche de services. Même si le principe est le même, comme nous allons le voir tout au long de cet article, l'objectif de l'algorithme CF est différent. OM suppose l'existence d'une entité (dont la route à partir de la source est connue) qui sert de cible pour une recherche. Ce système exige donc un protocole de routage sous-jacent. Contrairement, CF est utilisé pour la découverte et pour la maintenance des routes – il n'exige la présence d'aucune infrastructure de routage. Une fois que la première route a été découverte, les chemins subséquents sont obtenus sans provoquer l'inondation tous les liens du réseau.

L'algorithme CF suppose l'existence d'un premier chemin entre la source et la destination. Une fois que ce chemin est établi, l'algorithme CF détermine des routes alternatives pour le cas où la route principale soit perdue. Il peut être appliqué dans deux situations différentes : la découverte et la réparation de routes.

Pendant la phase de découverte des routes, la source doit déclencher une nouvelle procédure de recherche de façon à obtenir un autre chemin valable. Si le protocole utilise ERS (comme par exemple le protocole AODV), la recherche est "omnidirectionnelle", c'est-à-dire tous les nœuds distants de la source d'au maximum TTL sauts reçoivent le message RREQ. Le protocole ne tient pas en compte la position précédente de la destination. Si le protocole utilise CF, certains nœuds sont capables d'indiquer la direction de la destination. Une nouvelle procédure de recherche est lancée uniquement dans cette direction.

Notre approche peut utiliser différentes techniques pour déterminer de routes alternatives. Ces stratégies diffèrent en ce qui concerne l'étendue de l'inondation contrôlée autour du chemin principal entre la source s et la destination d, $\mathcal{S}_{s,d}$. Basiquement, lorsqu'un nœud dans $\mathcal{S}_{s,d}$ n'est plus fonctionnel, s'il existe des chemins alternatifs, une nouvelle route peut être immédiatement tracée.

L'algorithme CF donne l'impression de générer plus de charge dans le réseau puisque, en plus des RREQs de la première recherche, les nœuds dans la route principale échangent des messages de contrôle pour établir les routes alternatives. Ceci dit, il n'y aurait aucun avantage à utiliser CF si autant le chemin principal que les routes alternatives étaient perdus. Néanmoins, si la mobilité des nœuds est moins rapide que le temps de récupération après la perte d'une route, l'algorithme CF donne des résultats plus performants.

3. L'opération d'AODV-CF

Cette section présente les principaux détails de l'opération d'AODV-CF. L'algorithme qui suit est responsable pour la construction et le maintien des routes alternatives utilisés par AODV-CF. Le rafraîchissement des routes alternatives est fait au travers des mécanismes utilisés par AODV pour la détection des voisins actifs.

Le problème clé pour la maintenance des routes alternatives est que dans l'opération de base d'AODV, les routes sont rafraîchies par le trafic de données. Or, dans AODV-CF, une route alternative ne transporte pas de données jusqu'à ce qu'une faille arrive. L'idée est donc qu'un nœud gardera une route alternative aussi longtemps que son voisin qui est utilisé comme prochain saut soit actif. Ceci peut être réalisé en utilisant des messages AODV `Hello` ou la détection de lien actif d'AODV. Dans tous les cas, AODV-CF ne rajoute pas de trafic de contrôle pour le maintien des routes alternatives. Le mécanisme est simple car les nœuds des routes alternatives sont à un seul saut de la route principale. Une de nos directions de recherche traitera des routes plus "éloignées" de la route principale.

Nous présentons maintenant la construction des routes alternatives. Le protocole AODV-CF ajoute deux variables à chaque entrée de routage standard d'AODV. Premièrement, l'entrée inclut un vecteur qui garde l'ensemble de prochains sauts des routes alternatives. Deuxièmement, les entrées de routage AODV-CF possèdent un drapeau "primaire". Ce drapeau indique si la route pour la destination d est une route principale, de backup, ou neutre. Par défaut, une route est neutre (c'est le cas des routes créées par AODV standard). Une route principale est celle utilisée pour le trafic des données, et une route de backup (ou alternative) constitue des chemins alternatifs construits par AODV-CF.

AODV-CF utilise deux messages de signalisation dans la construction des routes alternatives : `Controlled` et `Controlled-Ack`. Les deux messages sont envoyés avec TTL égal à 1. Les nœuds dans la route principale sont les seuls à produire des messages `Controlled`, tandis que les nœuds dans les routes des backup produisent les messages `Controlled-Ack`. Chaque message est uniquement identifié, cette identifiant étant temporairement valide, comme pour les messages RREQ d'AODV. Un nœud est donc capable de savoir les messages qui ont déjà été traités et éventuellement retransmis.

La Fig. 1 décrit de façon simplifiée le règles de traitement des messages du type `Controlled`. Nous présentons ici les points de décision critiques. Supposons que le nœud n reçoit un message `Controlled` généré par le nœud m (qui appartient forcément à la route principale). Le message `Controlled` signale que m possède une route pour d, avec $l_{m,d}$ sauts, et que le nœud h est le prochain saut utilisé par m pour aller à d. En plus, le message `Controlled` porte le numéro de séquence (l'âge) de la route annoncée. La construction des routes alternatives est lancée par le nœud qui a demandé la route pour d en premier (la "source"). Quand la source reçoit le message AODV Reply, elle génère le premier message `Controlled`. Chaque nœud dans la route principale, y compris le nœud destination, d, retransmet ce message `Controlled` une seule fois ((1) et (2) dans la Fig. 1). Un nœud dans la route principale qui reçoit le message `Controlled` avec un numéro de séquence plus grand (plus "frais") que son propre numéro de séquence mettra à jour la route pour d si le message `Controlled` a été envoyé par le prochain saut, ou le saut précèdent, dans la route pour d ((3) et (4) dans la Fig. 1). Quand un nœud procède à une réparation de route, il remplace son prochain saut par une des routes alternatives existantes, augmente le numéro de séquence de la route, et envoie un message `Controlled`. Avec les règles (3) et (4), les nœuds qui sont toujours dans la route principale mettent à jour leurs numéros de séquence, mais pas les nœuds qui ne sont plus dans la route principale pour d. Ces règles servent à éviter la formation de boucles.

Supposons maintenant que le nœud n n'est pas dans la route principale pour d quand il reçoit le message `Controlled` ((5) dans la Fig. 1). Si n ne possède pas de route pour n, alors n crée une route de "bootstrap" qui pointe vers m (le nœud qui a envoyé le `Controlled`). Cette route de bootstrap est marquée DOWN (dans le drapeau standard d'AODV) et BACKUP (dans le drapeau "primaire" d'AODV-CF). Supposons à présent que le nœud n possède une route, non-primaire, pour d ((6) dans la Fig. 1). Si le numéro de séquence du message `Controlled` est le même que celui de la route dans n, alors n vérifie si m fournit une route plus courte pour d. En cas affirmatif, n met à jour sa route de backup pour d. Sinon, n insère m dans sa liste de précurseurs AODV (une fois que dans ce cas, m utilise potentiellement n comme prochain saut vers d) ((7) dans la Fig. 1). Si le message `Controlled` est plus récent que la route dans n, alors n vérifie si lui-même est le prochain saut utilisé par m (le nœud dans la route principale). En cas affirmatif, n efface la route de backup actuelle et recommence l'opération de construction de routes en créant une route de bootstrap ((8) dans la Fig. 1).

Les règles de traitement des messages `Controlled-Ack` sont plus simples (Fig. 2). Seulement les nœuds qui sont actuellement dans la route principale pour la destination d traitent les messages `Controlled-Ack` (Fig. 2 (1)). (La destination d ignore elle aussi ces messages car d n'a pas de route pour d.) Si le nœud n est dans la route principale avec le même numéro de séquence que le message `Controlled-Ack` (i.e., la route annoncée a le même âge que la route dans n – Fig. 2(2)), il y a deux possibilités. Si n est le prochain saut utilisé par m pour atteindre d, n doit insérer m dans sa liste de précurseurs ((3) dans la Fig. 2). Sinon, n pourra éventuellement ajouter une route alternative pour d passant par m ((4) dans la Fig. 2). Ceci ne sera possible que si la route pour d passant par m est plus courte que, égale à, ou exactement un saut plus longue que la route principale utilisée par n pour aller à d (une fois qu'une route alternative est éloignée d'un saut de la route principale dans cette version d'AODV-CF). Si n possède une route pour d qui est plus âgée que la route annoncée dans le message `Controlled-Ack` ((5) dans la Fig. 2), alors n efface ses routes alternatives pour d une fois que dans ce cas, une réparation de route a probablement eu lieu. Dans ce cas, le nœud n ira éventuellement mettre à jour sa route pour d après avoir reçu un message `Controlled` avec un numéro de séquence plus récent (si n est toujours dans la route pour d), et reconstruire les routes alternatives. Dans tous les cas, la liste de routes alternatives n'est plus valide dans cette situation.

4. Résultats de simulation

Pour évaluer l'efficacité de l'approche de l'inondation contrôlée dans AODV-CF, nous avons utilisé le simulateur *ns* (Network Simulator) [Fall et al., 2001] qui possède une implémentation d'AODV réalisé par le groupe Monarch/CMU [Broch, 1998]. Nous avons développé un nouveau module qui implémente AODV-CF, en héritant du module AODV original.

Dans cette section, nous analysons le trafic de contrôle généré par AODV et AODV-CF, ainsi que les temps de déconnexion (après les failles de routes). Nous démontrons

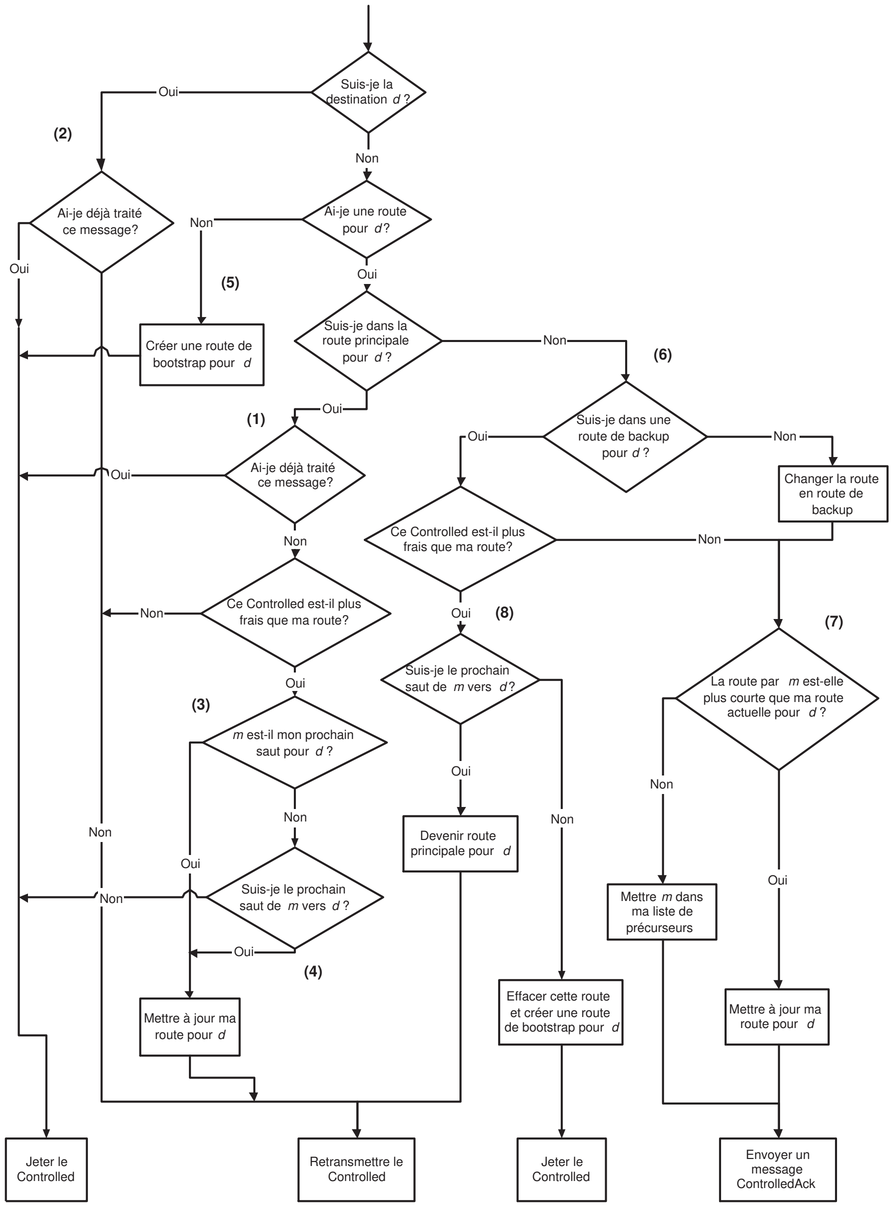


Figure 1. *Le nœud n reçoit un message* `Controlled` *du nœud m (route pour d avec h comme prochain saut).*

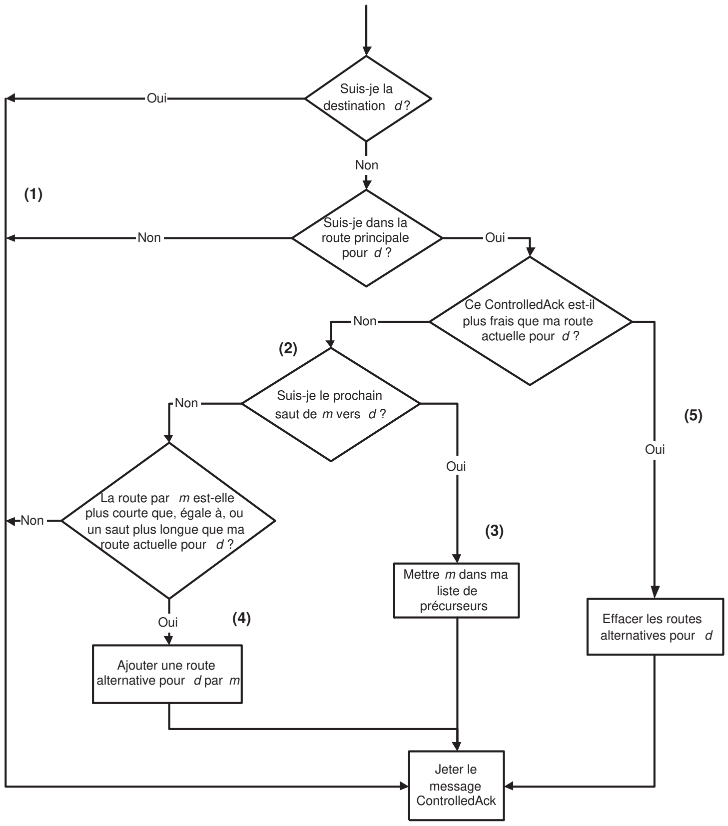


Figure 2. *Le nœud* n *reçoit un message* `Controlled-Ack` *du nœud* m *(route pour* d *avec* h *comme prochain saut).*

que le protocole AODV-CF possède une surcharge de contrôle plus petite qu'AODV avec un maintien de la connectivité plus efficace.

4.1. *Scénario de simulation*

Dans les simulations, nous avons utilisé la couche MAC IEEE 802.11 avec le modèle de propagation radio "free space" [Fall et al., 2001], ainsi que le générateur de scénario de mobilité implémenté par CMU dans la distribution de *ns*. Les nœuds se déplacent dans un plan de 1000×1000 m, à une vitesse maximale de 5 m/s avec des pauses de 50 s. Tous les nœuds sont mobiles.

Nous varions la taille du réseau entre 50 et 150 nœuds (en ajoutant 25 nœuds à chaque fois). Pour le réseau de 50 nœuds, la portée radio était de 225 m. La portée était ensuite variée de façon à garder le nombre moyen de voisins d'un nœud constant. Ainsi, en augmentant la taille du réseau la longueur de routes augmente elle aussi. Une deuxième possibilité serait de garder la portée radio constante et augmenter les dimensions du plan simulé, mais ce choix conduit à des temps de simulation plus longs à cause du générateur de mobilité. Pour chaque taille de réseau, nous avons réalisé 50 essais de 200 secondes simulées.

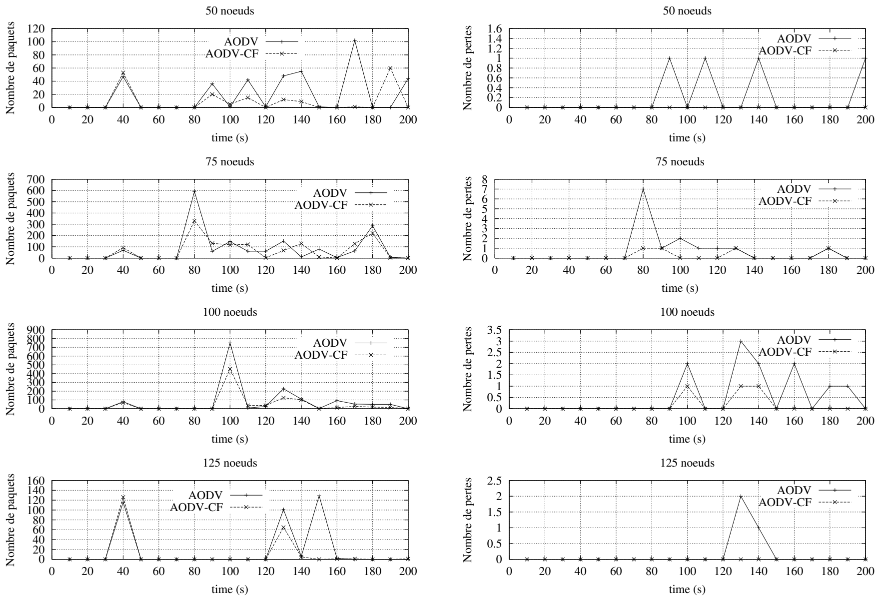
Dans les simulations, il y a une paire source-destination et la source génère un paquet de 512 octets par seconde. Nous n'utilisons qu'une source une fois que le but des simulations est d'évaluer l'efficacité d'AODV-CF dans la réduction du temps de déconnexion (grâce à l'utilisation des routes alternatives) et dans la reconstruction des routes. Dans le cas de plusieurs paires source-destination, les deux protocoles (AODV et AODV-CF) bénéficieront de l'existence potentielle de routes construites au préalable quand une même destination est demandée pas des différentes sources.

4.2. *Résultats*

La Fig. 3(a) montre le nombre total de paquets de contrôle générés par AODV et AODV-CF dans quelques uns des scénarios générés comme décrit précédemment. Dans tous les cas observés, la source CBR envoie une requête de route à l'instant $t = 30$ s. La communication dure tout le temps restant de la simulation. Il faut aussi noter que les résultats sont intégrés sur une période de 10 secondes, c'est à dire que le nombre de paquets montré à l'instant t dans le graphe est le nombre total de paquets générés dans l'intervalle $](t-10),t]$.

La Fig. 3(a) révèle quelques comportements intéressants. À $t = 40$ s, le mécanisme implémenté dans AODV-CF fait en sorte qu'AODV-CF produise plus de messages de contrôle qu'AODV. Cette différence était attendue en conséquence des messages `Controlled` et `Controlled-Ack` générés par l'algorithme CF dans la construction des routes alternatives (pour la première fois après la construction de la route principale). Néanmoins, cette surcharge de contrôle sera compensée par les gains obtenus dans les réparations de routes. Par exemple, avec le réseau de 75 nœuds à l'instant $t = 80$ s. Tandis que le protocole AODV redémarre une procédure de découverte de route en inondant le réseau avec des requêtes, AODV-CF est capable de réparer la route perdue et envoyer le trafic des données par une route alternative. Dans ce cas, le trafic de contrôle généré par AODV-CF correspond à l'inondation contrôlée utilisée pour reconstruire les route alternatives après la faille de la route principale.

Dans ces simulations, AODV-CF fonctionne en mode *promiscuous*. Tout nœud qui reçoit deux, ou plus, messages `Controlled` émet un message `Controlled-Ack`. Dans le cas où le réseau est dense, le nombre de messages `Controlled-Ack` peut être important. Ceci explique pourquoi dans certains cas la quantité de trafic de contrôle est grande même après une réparation de route avec succès (la plupart des messages correspond à des `Controlled-Ack`). Dans nos travaux futurs, des règles pour limiter le nombre de routes alternatives qu'un nœud peut implémenter seront étudiées pour réduire d'avantage le trafic de contrôle dans ces scénarios.

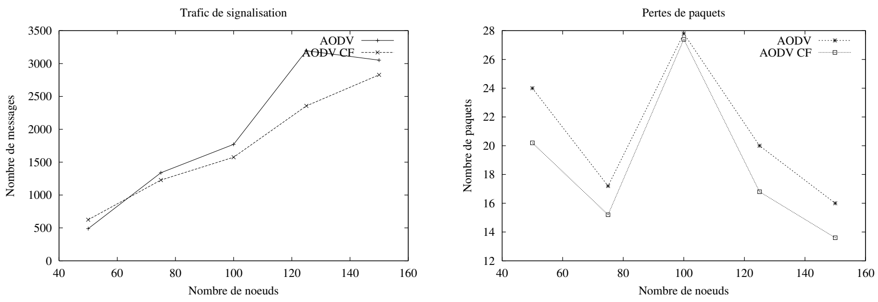


(a) Trafic de contrôle. (b) Pertes de paquets de données.

Figure 3. *Résultats.*

La Fig. 3(b) montre le nombre de pertes de paquets pour les mêmes scénarios de la Fig. 3(a). Les résultats reflètent le temps de déconnexion imposé quand une autre route doit être construite ou quand une route alternative devient route principale. Dans toutes les scénarios, AODV-CF présente des meilleures performances qu'AODV. Une observation intéressante : un nœud peut être amené à envoyer des paquets des données dans une route alternative qui n'est plus valide. Ceci cause des pertes de données jusqu'à ce que le nœud trouve un chemin alternatif valide. Ce phénomène arrive par exemple à t = 100 s pour le réseau de 100 nœuds.

La Fig. 4 montre la moyenne des résultats obtenus avec 50 essais pour chaque scénario, en variant la taille du réseau. Il est possible d'observer que les différences entre AODV et AODV-CF en terme de trafic de contrôle tendent à être plus importantes pour les plus grands réseaux (les routes sont potentiellement plus longues). Dans l'autre côté, les gains en terme de temps de déconnexion sont quasiment constants. Ces gains sont plutôt liés à la quantité de chemins alternatifs stockés dans chaque nœud et donc aussi à la densité du réseau (le nombre de voisins qu'un nœud possède en moyenne).



(a) Trafic de contrôle.

(b) Pertes de paquets de données.

Figure 4. *Comparaison de la moyenne des résultats d'AODV et d'AODV-CF.*

5. Conclusions

Dans cet article, nous avons proposé la technique de l'inondation contrôlée (CF) pour améliorer le maintien de la connectivité et l'efficacité de la découverte de routes dans les protocoles de routage ad hoc à la demande, qui utilisent souvent l'inondation comme mécanisme de recherche d'un nœud. Avec CF, des routes alternatives sont construites une fois que la route principale entre la source et la destination a été établie. Le mécanisme ne remplace pas le protocole de routage original. Celui-ci étant toujours utilisé dans l'établissement de la route principale, CF ne joue son rôle que lors des découvertes des routes ultérieures.

Dans AODV, une nouvelle procédure de découverte de route est réalisée à chaque fois que le chemin entre la source et la destination est rompu. Dans les réseaux mobiles ad hoc, les failles de routes peuvent être causées principalement par la mobilité des nœuds ou par la faille d'un nœud. Quand la topologie du réseau est éparse, la nouvelle route peut être complètement différente de la route d'origine. Dans ce cas, l'inondation des requêtes de routes dans le réseau entier peut être une solution acceptable. Néanmoins, si le réseau est dense, il est probable qu'une partie de la nouvelle route comprenne des liens de la route précédente. D'un autre point de vue, nous pouvons considérer que les nœuds voisins de la destination ont une piste sur sa nouvelle localisation et dans ce cas le broadcast dans le réseau entier peut être évité.

Dans le protocole AODV-CF, version modifiée d'AODV qui utilise le mécanisme que nous avons proposé, les nœuds dans la route principale gardent des pointeurs pour le prochain saut vers une destination, comme dans AODV, mais aussi des pointeurs vers un ensemble des voisins qui pourvoient des routes alternatives pour cette destination. La construction de ces routes alternatives est réalisée par un algorithme simple. Les nœuds dans la route principale n'échangent de messages qu'avec des nœuds dans leur voisinage. Les simulations réalisées jusqu'à présent ont démontrée que l'approche est prometteuse. AODV-CF a montré une réduction considérable des messages de contrôle par rapport à AODV, ainsi que des temps de déconnexion plus courts.

6. References

[Broch, 1998] Broch, J. A performance comparison of multihop wireless ad hoc network routing protocols. In *ACM/IEEE Mobicom*, 1998.

[Castaneda et al., 1999] Castaneda, R. et Das, S. R. Query localization techniques for on-demand routing protocols in ad hoc networks. In *ACM/IEEE Mobicom*, 1999.

[Costa et al., 2002] Costa, L. H. M. K., de Amorim, M. D., et Fdida, S. Avoiding network-wide broadcasting with controlled flooding for on-demand ad hoc routing protocols. In *IFIP Med-Hoc-Net*, 2002.

[Fall et al., 2001] Fall, K. et Varadhan, K. *The ns Manual*. UC Berkeley, LBL, USC/ISI, and Xerox PARC, 2001, Available at http://www.isi.edu/nsnam/ns/ns-documentation.html.

[Johnson et al., 2002] Johnson, D. B., Maltz, D. A., Hu, Y.-C., et Jetcheva, J. G. *The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)*, 2002, Work in progress, <draft-ietf-manet-dsr-07.txt>.

[Ko et al., 2000] Ko, Y.-B. et Vaidya, N. H. Location-aided routing (LAR) in mobile ad hoc networks. *ACM/Baltzer Wireless Networks Journal (WINET)*, 6(4):307–321s, 2000.

[Magoni et al., 2002] Magoni, D. et Pansiot, J.-J. Network layer search service using oriented multicasting. In *IEEE Infocom'02*, 2002.

[Park et al., 1997] Park, V. et Corson, M. A highly adaptive distributed routing algorithm for mobile wireless networks. In *IEEE Infocom*, 1997.

[Perkins et al., 2002] Perkins, C. E., Belding-Royer, E. M., et Das, S. R. *Ad hoc On-Demand Distance Vector (AODV) Routing*, 2002, Work in progress, <draft-ietf-manet-aodv-11.txt>.

[Royer et al., 1999] Royer, E. et Toh, C.-K. A review of current protocols for ad hoc wireless networks. *IEEE Personal Communications*, pages 45–55, 1999.