

Transmission applicative multipoint pour les réseaux ad-hoc

Cyrine Mrabet, Mohamed Ali Kaafar, Farouk Kamoun

► **To cite this version:**

Cyrine Mrabet, Mohamed Ali Kaafar, Farouk Kamoun. Transmission applicative multipoint pour les réseaux ad-hoc. Colloque Francophone sur l'Ingénierie des Protocoles - CFIP 2006, Eric Fleury and Farouk Kamoun, Oct 2006, Tozeur/Tunisia, Tunisie. 12 p. inria-00110297

HAL Id: inria-00110297

<https://hal.inria.fr/inria-00110297>

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.

Transmission applicative multipoint pour les réseaux ad-hoc

Cyrine Mrabet* — Mohamed Ali Kaafar** — Farouk Kamoun*

*Laboratoire CRISTAL
Ecole Nationale des Sciences de l'Informatique
Campus universitaire - 2010 La Manouba
{cyrine.mrabet, farouk.kamoun}@cristal.rnu.tn

**INRIA Sophia Antipolis
2004 route des Lucioles – B.P. 93
06902 Sophia Antipolis Cedex
mohamed.kaafar@sophia.inria.fr

RÉSUMÉ. AOMP (Ad-hoc Overlay Multicast Protocol) est une nouvelle approche pour la transmission applicative multipoint dans les réseaux ad-hoc mobiles. En exploitant les propriétés de routage IP, nous introduisons, dans une première étape, un algorithme distribué de correspondance des chemins permettant d'extraire des informations sur la topologie physique. Dans une seconde étape, nous construisons un arbre de transmission efficace à moindre coûts et s'adaptant à la mobilité des nœuds ad-hoc. Nous mettons en place un algorithme de raffinement pour améliorer les performances globales de la structure AOMP en cours de transmission. Les résultats de simulations montrent que AOMP, comparé à de récents protocoles multipoints applicatifs, diminue le délai de transmission, évite les redondances de paquets sur un même lien physique avec peu de pertes, en particulier en cas de mobilité des nœuds ad-hoc.

ABSTRACT. AOMP (Ad-hoc Overlay Multicast Protocol) is a novel approach for application-layer multicast in ad-hoc networks. We introduce in this paper a new algorithm that exploits a few properties of IP-routing to extract underlying topology information. The basic idea is to match nodes' path to the source in order to detect near neighbors in the physical topology. Then, in a dynamic and decentralized way, we construct a minimum cost mobility-aware delivery tree, connecting nodes that are close to each other. We design a tree improvement algorithm in order to enhance the global performance of AOMP during data distribution. Our simulations results show that, compared to previously proposed application-layer multicast structures, AOMP yields trees with lower cost and traffic redundancy. In addition, it performs well in terms of packet losses, especially in case of node mobility.

MOTS-CLÉS : réseaux Ad-hoc, réseaux de recouvrement pair à pair, Multicast, Mobilité, DSR, performance.

KEYWORDS: Ad-hoc networks, Peer to peer overlay, Multicast, DSR, performance.

1. Introduction

Les environnements mobiles offrent aujourd'hui une grande flexibilité d'emploi. Les réseaux ad-hoc, en particulier, forment une collection d'entités mobiles interconnectées par une technologie sans fil formant un réseau temporaire sans l'aide de toute administration ou de tout support fixe. Chacun des nœuds joue ainsi le rôle d'un routeur dans le réseau. Par ailleurs, les approches de transmission de flux multimédia multipoint (multicast) applicatives dans les réseaux Internet sont actuellement en pleine expansion, du fait de leur simplicité en termes de déploiement et d'interopérabilité avec les infrastructures et protocoles de bas niveaux. Concevoir et déployer des schémas de transmission efficaces pour des applications multimédia dans les réseaux ad-hoc est de ce fait d'un grand intérêt. La plupart des recherches sur la transmission multipoint pour les réseaux ad-hoc se sont focalisées sur les protocoles de niveau IP. Les solutions (IP) peuvent être classées en deux familles principales, selon qu'elles optent pour la création d'un arbre [WU 99, ROG 99], ou d'une maille [GAR 99, LEE 02]. De tels protocoles requièrent une coopération (implication) de tous les nœuds du réseau, qu'ils soient membres ou non du groupe multipoint.

Les approches applicatives, sont récemment apparues comme alternative au support IP. Elles permettent de construire une structure logique "overlay" masquant ainsi les complications du routage sous jacents. Une topologie virtuelle recouvre la topologie physique où chaque lien de la première est un tunnel mono-directionnel dans la seconde. Elles offrent la possibilité de construire des arbres de transmission multipoints, sans pour cela maintenir des états relatifs à chaque groupe multipoint. Le support du multipoint, est donc relégué aux nœuds participants au niveau applicatif et non à la couche IP. Cette facilité de déploiement est malheureusement compromise par une inefficacité de routage. En effet, la topologie virtuelle et par la suite l'arbre de transmission basée sur cette topologie induisent souvent des transmissions multiples de copies identiques de paquets de données sur un même lien physique. Dans le cas où la construction de l'overlay ne tiendrait pas compte de la topologie du réseau physique, cela engendrerait un coût et un délai de transmission élevés, dégradant les performances perçues par l'utilisateur surtout s'agissant d'applications isochrones. Dans un réseau ad-hoc, où de surcroît les nœuds sont mobiles, cela risque de compromettre tout le fonctionnement de l'overlay.

Dans cet article, nous proposons un nouveau schéma pour la construction d'un overlay multipoint pour les réseaux ad-hoc, efficace et tenant en compte de la topologie, nommé AOMP (Ad-hoc Overlay Multicast Protocol).

Dans une première étape, nous introduisons un nouvel algorithme de connexion des membres à l'overlay. Il consiste à faire correspondre le chemin d'un nouveau venu à partir de la source de l'overlay multipoint avec les chemins des différents membres déjà existants. Cet algorithme partage le même principe que celui du covoiturage. Deux personnes quittant un point donné veulent se rendre à deux destinations différentes. Par souci d'économie, elles partagent la même voiture jusqu'au point le plus loin possible entre les deux chemins. Nous adopterons cette même stratégie pour les paquets émis à partir de la source.

Dans une seconde étape, nous construisons un arbre de transmission multipoint qui desservira tous les membres de l'overlay. Nous introduirons des mécanismes permettant d'améliorer les performances en cours de transmission des nœuds overlay. Ces ajustements sont effectués de manière à optimiser les performances globales de l'overlay et non pas celles de quelques nœuds particuliers (optimisation "non-greedy").

Cet article est structuré comme suit : la section 2 présente les travaux relatifs à l'overlay multipoint dans les réseaux ad-hoc. Nous introduisons, dans la section 3, la

première phase du schéma AOMP, en l'occurrence la phase de connexion à l'overlay. La seconde phase de construction et gestion de l'arbre de transmission est détaillée dans la section 4. Enfin, nous évaluons les performances de AOMP par des simulations dans la section 5. Une conclusion présente les principaux résultats et les travaux futurs de cette étude.

2. État de l'art

Plusieurs travaux se sont penchés sur divers aspects et approches pour le multipoint overlay dans les réseaux filaires en général, et dans les réseaux ad-hoc en particulier. Nous allons dans cette section nous intéresser aux contributions relatives à la transmission multipoint dans les réseaux ad-hoc. Elles peuvent être classifiées en deux catégories principales :

Les protocoles multipoints de niveau IP comme AMRIS [WU 99], MAODV [ROG 99], CAMP [GAR 99] et ODMRP [LEE 02], sont construits d'une manière similaire au multipoint IP natif. Ils exploitent une des propriétés fondamentales des réseaux ad-hoc, en l'occurrence le fait que les nœuds agissent en routeurs. MAODV par exemple, propose une extension du protocole de routage AODV, où les nœuds établissent des routes multipoints à la demande et utilisent ces routes pour la transmission des données. ODMRP crée lui aussi des chemins multipoints spécialement conçus pour le transfert de flux multipoints, et permet aux nœuds de partager et maintenir l'état du groupe multipoint. Ces protocoles permettent une transmission efficace, mais souffrent en revanche de problèmes de déploiement et de maintenance. Tous les nœuds du réseau ad-hoc doivent supporter le protocole de routage multipoint, qu'ils soient membres du groupe ou pas. Le schéma overlay multipoint que nous proposons exploite un protocole de routage réactif de niveau IP, sans pour cela impliquer des nœuds non concernés par la transmission multipoint.

Les protocoles overlay multipoints considèrent le problème au niveau applicatif. Ils construisent une topologie logique, interconnectant les membres du groupe, et recouvrant le réseau physique. Ces différents protocoles peuvent être catégorisés en deux classes principales : (i) les structures hiérarchiques [BLO 04, KIM 05], tentant d'exploiter au mieux la propriété de diffusion du média sans fil et (ii) les structures dérivant un arbre de transmission "plat" sans hiérarchie quelconque [XIE 02, GUI 03].

Parmi la classe de protocoles non hiérarchiques, nous pouvons citer PAST-DM [GUI 03] (Progressively Adapted Sub-Tree in Dynamic Mesh). PAST-DM est aussi un protocole visant à dériver l'arbre de transmission à partir d'une maille adaptée à la mobilité des nœuds ad-hoc. Chaque membre initie une phase de recherche pour découvrir ses voisins logiques, basés sur un mécanisme de diffusion limité par un nombre de sauts maximal. Les nœuds s'échangent périodiquement une table "état des liens" avec leurs voisins. Malgré son adaptation à la mobilité, PAST-DM souffre de problèmes de passage à l'échelle, vu le coût important en termes de messages de contrôle générés par les nœuds lors de la recherche des voisins. Notre schéma exploite la recherche de routes, d'ores et déjà établies par un protocole de routage réactif pour découvrir les voisins logiques les plus proches, sans coûts additionnels.

ALMA [GE 04] (Application Layer Multicast Algorithm) construit un arbre multicast overlay d'une manière incrémentale et décentralisée. À son arrivée dans l'overlay, un nœud émet des requêtes de connexion vers ses voisins "proches" et se connecte au premier nœud qui lui répond. ALMA opte pour des mesures RTT vers ses nœuds pour la détection et la recherche de nouveaux voisins en cas de mobilité. Ainsi, lorsque les mesures RTT d'un nœud vers son parent dépassent un certain seuil, celui-là engage une procédure de reconfiguration de l'arbre de transmission. Durant cette procédure, la

portée de la recherche (nombre de sauts maximum) augmente en fonction de la mesure RTT vers le parent actuel. ALMA se base ainsi sur un mécanisme de diffusion pour cette recherche. Bien que limité par un nombre de sauts, cela risque d'être coûteux, dégradant rapidement les performances lorsque le nombre de nœuds augmente.

NICE-MAN [BLO 04] (NICE for Mobile ad-hoc Networks) est un exemple d'architectures overlay multipoints hiérarchisée. Il s'agit d'une extension pour les réseaux ad-hoc du protocole NICE, un overlay conçu pour la transmission multipoint dans les réseaux filaires. L'idée première de NICE-MAN, et celle des protocoles hiérarchiques en général, est de sélectionner un nœud chef ("clusterhead") pour un groupe de membres overlay et de déléguer la transmission à ce groupe via leur chef ("clusterhead"). Le principal inconvénient de ce type d'approches est qu'elles peuvent induire un délai de bout en bout très élevé en présence d'une densité de nœuds ad-hoc importante. Dans des réseaux ad-hoc de grande taille, ceci est souvent le cas.

Après avoir étudié différents protocoles multipoints au niveau applicatif, nous allons présenter un nouveau protocole appelé AOMP (Ad-hoc Overlay Multicast Protocol) ayant une structure dérivée d'un arbre de transmission tel que ALMA et PAST-DM. AOMP est, en revanche, conçu pour être moins coûteux en termes de messages contrôle et plus performant en terme de mobilité que ces deux protocoles. Dans ce qui suit, nous décrirons notre algorithme de correspondance de chemin de AOMP, exploitant les propriétés des protocoles de routage réactifs pour la connexion des nœuds à l'arbre overlay.

3. Le protocole de connexion de AOMP

Notre objectif est de construire un réseau overlay efficace, sans pour cela engendrer des coûts de communications en terme de mesure. Pour ce faire, nous exploitons les informations de la topologie physique, c'est-à-dire les informations de chemins IP les plus courts que les nœuds ad-hoc maintiennent. La définition de "plus court" dépend en réalité du protocole de routage utilisé, mais dénote souvent le chemin le plus court en termes de nombre de sauts ou de délai, etc. Les protocoles de routage ad-hoc réactifs s'adaptent rapidement à la mobilité (contrairement aux protocoles proactifs), se basent sur une construction dynamique des routes, et offrent la possibilité de découvrir la séquence de sauts entre deux nœuds qui veulent communiquer. Nous allons exploiter ces propriétés pour la construction d'un overlay tenant compte de la topologie du réseau, en particulier des chemins IP entre les nœuds membres du groupe multipoint. Nous choisissons le protocole DSR car il est simple, efficace et réagit rapidement à la mobilité. De plus, le simulation NS implémente DSR, ce qui est appréciable pour l'évaluation des performances.

Dans ce qui suit, nous présentons brièvement le protocole de routage réactif DSR. Nous décrirons ensuite notre algorithme de correspondance de chemin, exploitant DSR pour la connexion des nœuds à l'arbre overlay.

3.1. Le protocole DSR (dynamic Source Routing)

DSR [JOH 03] est un protocole réactif, qui utilise une technique de routage par source dans laquelle les nœuds intermédiaires ne doivent pas nécessairement garder la trace de la route. Chaque paquet contient dans son en-tête la liste complète des adresses des nœuds à traverser vers la destination. Lors de la création des routes, si un destinataire est dans le cache du nœud source, la route connue est utilisée. Sinon, une procédure de découverte de route est *déclenchée* par un message "Route Request". Le chemin vers la destination est créé dans le paquet de recherche de route. Chaque nœud qui reçoit ce paquet ajoute à la route préexistante dans ce paquet sa propre adresse.

En cas de rupture d'un lien sur un trajet, le nœud situé en amont de la rupture envoie à la source une indication concernant la rupture du lien. Les nœuds peuvent, sur une base volontaire, garder en mémoire dans leur cache les routes créées par les procédures de recherche de route. Ces routes permettent de répondre plus rapidement à d'autres requêtes de création de route. En cas de rupture de lien, il est possible à un nœud qui possède une route valide vers la destination de détourner le trafic par ce nouveau chemin, sauvant ainsi l'acheminement des paquets.

3.2. L'algorithme de correspondance des chemins

Notre algorithme de détermination des voisins overlay est lancé par chaque nœud désirant se connecter à l'overlay. Le but est de construire un arbre connectant les nœuds voisins (dans la topologie physique), que nous appelons *arbre principal* (figure 1). La notion de nœuds voisins est définie par le fait que l'un d'entre eux partage avec l'autre le plus long préfixe de chemin vers la source. Deux intuitions principales guident cette approche : (i) Construire un arbre principal qui connecte les nœuds overlay selon leur voisinage réseau permet de réduire le délai de bout en bout de la source jusqu'à chacun des nœuds overlay. En effet, le schéma que nous adoptons fait correspondre les chemins de la source vers les nœuds en se basant sur les tables de routages IP, ce qui permet aux liens overlay de recouvrir les chemins IP les plus courts. (ii) Partager les mêmes préfixes de chemins à partir de la source permet de réduire le nombre de paquets identiques sur le même lien physique, réduisant ainsi le surcoût d'une approche multipoint overlay.

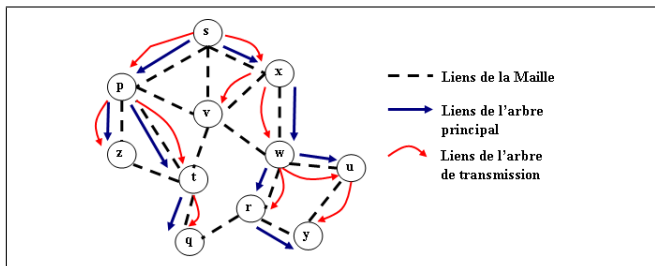


Figure 1. Exemple d'un arbre overlay.

Nous commençons par présenter la terminologie et la notation adoptées dans cet article. Nous notons le chemin IP d'un nœud x vers y par $Ch(x, y)$. Ce chemin, dérivé de la table de routage DSR du nœud x , comporte les sauts qu'un paquet IP traverse pour atteindre la destination y . $|Ch(x, y)|$ est défini comme étant le nombre de sauts que comporte le chemin. Un chemin $Ch(x, y)$ est un **préfixe** de $Ch(x', y')$, si et seulement si, tout le chemin du premier est compris dans le deuxième. Dans la figure 1, $Ch(s, u)$ est un préfixe de $Ch(s, y)$. L'algorithme se base sur un processus de correspondance du plus longue préfixe possible de nœuds (de la source vers le nœud désirant se connecter) :

$$(P(s, y) \text{ préfixe } P(s, x)) \text{ and } \nexists z, P(s, z) \text{ préfixe } P(s, x), tq |P(s, z)| > |P(s, y)| \quad (1)$$

Utilisant un protocole de routage réactif, chaque nœud parent connaît le chemin IP vers ses fils. Lorsque le nœud n veut se connecter à l'arbre principal, il émet une simple requête de connexion vers s et attend une réponse. À la réception de la requête de connexion, s déroule l'algorithme. Si la source estime qu'elle est elle-même le

parent principal, elle envoie une réponse au nouveau venu, sinon la requête de ce dernier est transférée vers le nœud dont le chemin est le plus long préfixe. La procédure est alors ré-exécutée par le nœud réceptionnant la requête de connexion. La Figure 2 montre les trois cas possibles dans le déroulement de cette procédure.

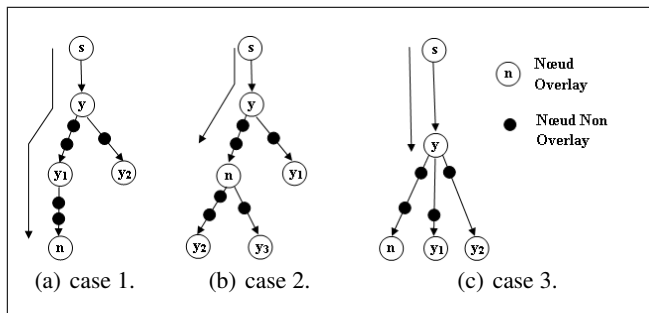


Figure 2. *The three cases for the path matching.*

Chaque nœud y sélectionne, si possible, l'un de ses nœuds fils y_i , tel que $Ch(y, y_i)$ est la plus longue séquence de $Ch(y, n)$. Si un tel fils existe (cas 1), la requête est transférée vers ce fils, et le processus de recherche se propage ainsi au sous-arbre de source y_i , où ce dernier lance de nouveau le processus. Si en revanche, ils existent des nœuds fils de y tel que le chemin de n est une séquence de leurs chemins (cas 2), n est inséré comme fils de y avec les différents y_i comme fils. Si aucun des fils de y ne satisfait les deux premiers cas, n devient fils de y (cas 3). Il est important de noter qu'il ne peut y avoir qu'un seul fils de y satisfaisant le premier cas, puisque nous recherchons la plus longue correspondance des chemins. Le processus de recherche ne se propagera donc que vers un unique sous arbre, dont la source est ce fils unique.

Après avoir établi la connexion entre tous les nœuds participants à la transmission multipoint, c'est à dire avoir construit l'arbre principal, nous allons implanter ce dernier dans une structure maille pour enfin dériver l'arbre de transmission.

4. Opérations de base de l'overlay dans AOMP

La construction de l'overlay se compose de trois étapes distinctes : La construction de l'arbre principal, la construction de la maille en ajoutant des liens supplémentaires pour améliorer les propriétés de délai, et finalement la dérivation de l'arbre de transmission. Ce dernier est obtenu à partir de la maille en utilisant un routage similaire à celui des protocoles de routage reposant sur les vecteurs de distance "path-vector".

L'arbre principal résulte de la phase de connexion. Nous notons ici que lors de cette phase, chaque nouveau venu s'est vu attribuer un parent principal sans se soucier de la capacité individuelle de celui-ci¹. Cela permet de rattacher le fils au nœud le plus proche dans le réseau physique. La manière dont le nœud se connecte à l'arbre de transmission est laissée pour la phase suivante, en l'occurrence la construction de la maille. Dans ce qui suit, nous allons définir la structure maille et par la suite décrire la manière dont le nouveau nœud se joint à la maille de l'overlay.

1. Nous tenons compte ici du fait que chaque nœud possède un nombre maximal de transmission élémentaire (fan-out).

4.1. La structure maillée

L'arbre principal, construit lors de la phase de connexion, est implanté dans une maille et constitue le squelette principal de cette maille (figure 1). Cette structure est dès lors incrémentée de liens supplémentaires vers un ensemble de nœuds spécifiques. L'établissement de ces liens s'effectue comme suit : dès que le nœud se connecte à son parent principal ou à l'un de ses descendants, il est informé de l'adresse de son grand père ainsi que celle de ses oncles. Il établit alors des connexions vers ces derniers constituant les liens de la maille. L'ensemble des voisins d'un nœud, i , est ainsi classifié en trois groupes : V_i^p est l'ensemble des nœuds voisins dans l'arbre principal, il est composé du parent principal et d'éventuels fils. V_i^m est l'ensemble des nœuds avec lesquels i a établi des liens supplémentaires. Et enfin, V_i^t est l'ensemble des nœuds voisins dans l'arbre de transmission.

4.2. Joindre la maille overlay

Une fois la réponse du parent principal reçue, chaque nouveau venu doit essayer de se joindre à la maille. En fait, la réponse du parent principal contient son acceptation ou pas, en tant que fils dans la maille. Dans le cas échéant, c'est-à-dire lorsque le parent principal n'a pas la capacité¹ d'accepter le nouveau venu, le premier lui envoie la liste de ses fils dans l'arbre de transmission. Le nouveau venu émet ainsi un message "JoinRequest" à chacun des descendants récursivement jusqu'à ce qu'il puisse se rattacher à la maille. Ainsi, malgré la phase de recherche du voisin le plus proche, qui a permis au nouveau venu de détecter le nœud le plus proche dans la topologie physique, celui-là peut céder sa position si jamais son parent principal n'est pas apte à le connecter. En effet, tous les fils d'un même parent se valent au vu de la métrique nombre de sauts. Le nouvel arrivant va alors concéder sa position dans l'arbre au profit des autres fils qui existaient déjà. Dans l'exemple de la figure 3(a), le nouvel arrivant ne se connecte pas au père mais au fils².

Un raffinement pourra lui permettre par la suite de remonter dans l'arbre, si jamais il acquiert un poids dans l'arbre de transmission supérieur à ses parents. Nous discuterons dans les paragraphes suivants comment un nœud pourrait remonter dans l'arbre au fur et à mesure que des nœuds se connectent à lui, ou que ses performances l'imposent.

4.3. Raffinement de l'overlay

Les nœuds sont rattachés dans l'arbre overlay à leur parent le plus proche physiquement ou à l'un des descendants de ce dernier s'il ne peut pas l'accepter. Dans ce dernier cas, le père va donc reléguer la transmission de trafic à l'un de ses descendants. Cette relégation peut entraîner un rallongement dans l'acheminement du flux que nous pourrions éviter. Reprenons l'exemple de la figure 3(a). Pour atteindre le nouvel arrivant, le trafic doit cheminer par le père p et le fils x_2 . Nous aurions amélioré les performances de l'overlay, en transitant le trafic du nœud grand-père, vers le nouvel arrivant, et en terminant par le père. Un raffinement du rattachement à l'overlay est donc nécessaire. Il consiste en une inversion des rôles entre un fils et son père (comme dans notre exemple) ou un fils et l'un de ses oncles.

4.3.1. État au niveau des nœuds overlay

En cours de transmission, chaque nœud i est capable de maintenir son temps de latence (noté L_i) induit par l'overlay à partir de la source s . Ce temps de latence est calculé comme étant la différence entre les moments d'émission et de réception des paquets de données. En se basant sur ce temps de latence, le nœud i peut estimer son

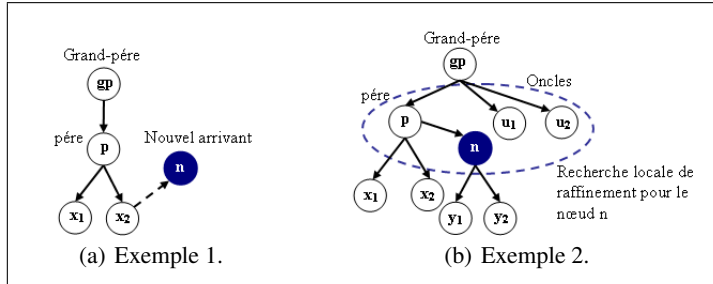


Figure 3. Exemple de relégation d'un nœud fils.

pois p_i dans l'arbre de transmission. Ce poids est la somme des délais de tous les nœuds appartenant au sous-arbre rattaché à i . Le calcul de ce poids est important car, dans le cas où un nœud i est mal desservi dans la maille overlay, tout le sous-arbre de i va pâtir d'une augmentation significative de délai de transmission. Le poids est calculé comme suit :

$$p_i = L_i + \sum_{j \in \text{Fils}(i)} p_j \quad (2)$$

où le poids p_f d'un nœud feuille f (n'ayant pas de fils) est égal à son temps de latence L_f . Ce poids est la métrique sur laquelle se base un père (ou un oncle) pour céder, ou non, sa place à l'un de ses fils (ou neveux). Notons ici que l'information poids est partagée entre chaque parent et ses fils à travers les messages d'entretien de l'overlay. Dans cet article, nous ne nous étendrons pas sur les mécanismes d'entretien classiquement utilisés dans les réseaux overlay.

4.3.2. Sélection du candidat de raffinement

La sélection d'un candidat pour le raffinement se fait au cas où un nœud a été relégué par son père lors de son rattachement à l'arbre de transmission. Elle peut aussi se faire pour un nœud choisi au hasard périodiquement, ou encore lors de la détection d'un mouvement de nœud. Le nœud choisi va essayer d'améliorer les performances de l'overlay en se substituant à son père ou à l'un de ses oncles dans l'arbre principal (figure 3(b)). Il doit donc maintenir dans une liste toutes leurs adresses.

4.3.3. Choix d'un voisin potentiel

Une fois qu'un candidat i est sélectionné, il calcule son poids potentiel. Celui-ci représente le poids du nœud s'il se rattachait directement à son grand-père. Il envoie ensuite cette information au nœud j auquel il veut se substituer (son père ou l'un de ses oncles).

$$p_i^{\text{potentiel}} = p_i + (N_i + 1)[d(gp, i) - d(i, j) - d(j, gp)] \quad (3)$$

Où N_i est le nombre des nœuds dans le sous-arbre de source i .

Le nœud j , recevant une requête de raffinement de la part de son fils, considère désormais son poids en éliminant le sous arbre du fils i , $p'_j = p_j - p_i$. Il calcule ensuite son poids potentiel si jamais il cédait sa position dans l'arbre de transmission, au nœud i :

$$p_j^{\text{potentiel}} = p'_j + (N_j - N_i)[d(gp, i) + d(i, j) - d(gp, j)] \quad (4)$$

La substitution ne se fera que lorsque le poids potentiel du nœud i est supérieur à celui du nœud j : $p_i^{\text{potentiel}} > p_j^{\text{potentiel}}$.

4.4. Adaptation à la mobilité

L'arbre de transmission doit s'adapter à la mobilité des nœuds ad-hoc dans le réseau, ainsi qu'aux changements des conditions réseaux. Une adaptation basique résulterait en une vérification périodique des routes (à travers des "route Request" déclenchés) afin de déterminer si les chemins ont changé. Si des changements de route sont détectés, un processus pour rejoindre de nouveau l'overlay (en particulier l'arbre principal) est imposé aux nœuds concernés. Toutefois, un tel mécanisme induit des problèmes de passage à l'échelle, ainsi qu'une surcharge éventuelle de la source.

Nous proposons un mécanisme permettant aux nœuds intermédiaires jusqu'à la source de participer au processus de détection et de reconnexion à l'arbre principal. Le mécanisme se base sur l'entretien des routes réalisé automatiquement par le protocole de routage réactif utilisé. L'idée est de procéder localement à une connexion à l'arbre principal dans un premier temps, puis de reconfigurer l'arbre de transmission lorsque la connexion au parent principal est établie. Reprenons l'exemple de la figure 1 et supposons que le nœud w , a effectué un mouvement dans le réseau ad-hoc vers le nœud v , ayant entraîné une modification dans le chemin $Ch(S, w)$. Ce changement de route, grâce au protocole de routage réactif, sera détecté par x , son parent dans l'arbre de transmission (puisque'il maintient l'adresse IP ainsi que le chemin vers son fils dans sa table de routage). x doit donc vérifier l'impact d'un tel mouvement. Il vérifie dans la nouvelle séquence générée par le mouvement de son fils w si elle contient un ou plusieurs nœuds qu'il connaît dans son voisinage (père, oncles, frères, fils...). Si aucun nœud overlay n'est détecté dans $Ch(x, w)$, le parent x reconfigure uniquement son chemin IP vers son fils sans imposer aucun changement dans la structure overlay. Sinon, des reconnexions à l'arbre principal sont effectuées par d'une part le fils w et d'autre part les fils de w , en l'occurrence u et r . Le fils w initie ainsi une procédure locale de connexion à l'arbre principal vers le premier nœud overlay rencontré dans la nouvelle séquence, v . Cette procédure, identique à une procédure de correspondance des chemins locale ayant pour source v , s'effectue d'une manière récursive jusqu'à une connexion à un parent principal (comme discuté dans 3.2). Le chemin vers le sous arbre de w a aussi été modifié. Les fils de w initient, à leur tour, une reconnexion locale à leur grand parent x dans l'arbre principal, identique à celle de w . Il est important de noter ici que l'arbre de transmission n'est modifié que lorsque la connexion de i et de ses fils est établie dans l'arbre principal. Dès lors qu'il s'est reconnecté à un nouveau parent principal, w doit attendre des acquittements de tous ses fils, pour entamer la dérivation de son nouvel arbre de transmission.

5. Évaluation de AOMP

Dans cette section, nous allons évaluer AOMP en le comparant à deux protocoles multipoints overlay récents : ALMA et PAST-DM. Nous discutons dans un premier temps l'environnement de simulation ainsi que des métriques utilisées. Ensuite, nous présentons les résultats de ces simulations.

5.1. Description des simulations

Nous avons utilisé le simulateur NS-2.28 [CAN 04]. La durée de simulation est de 900s. Le réseau est de taille 1000x1000m. Il comporte 100 nœuds se déplaçant de manière aléatoire suivant le modèle "Random WayPoint". La vitesse moyenne des nœuds est de 2 m/s. Le temps de pause est au maximum de 30s. La bande passante

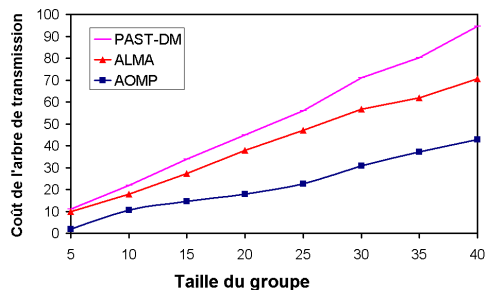


Figure 4. Coût de l'arbre de transmission.

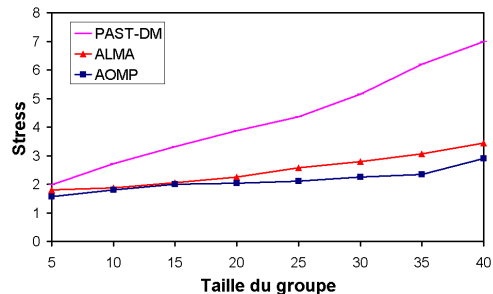


Figure 5. Stress sur les liens physiques.

des canaux est de 2 MHz. Le trafic généré par la source est de 64Kbits/s ce qui représente une excellente qualité pour la transmission de la voix. La capacité individuelle de chaque nœud (en terme de fan-out) est uniformément distribuée dans l'intervalle [2..20]. Nous avons utilisé les métriques suivantes :

- Coût de l'arbre de transmission : le nombre total de liens physiques que traversent les paquets dans un arbre de transmission. Cette métrique représente la qualité de la structure de l'arbre overlay.
- Stress : le nombre de copies identiques de paquets sur un même lien physique. Cette métrique quantifie l'efficacité du schéma multipoint.
- Ratio des paquets reçus : le ratio du nombre de paquets reçus par le destinataire par rapport au nombre total de paquets envoyés par la source. Cette métrique est utilisée pour quantifier la fiabilité du protocole multipoint.
- Surcoût en message de contrôle : le ratio des paquets de contrôle transmis par rapport au nombre total des paquets transmis. Cette métrique quantifie le coût du protocole.

5.2. Coût de l'arbre de transmission

La figure 4 présente le coût de l'arbre de transmission par rapport à la taille du groupe overlay. Nous observons que la structure AOMP est considérablement plus efficace que PAST-DM et ALMA, avec un coût d'environ 1,5 à 2,5 fois moins que ALMA et PAST-DM. AOMP exploite les propriétés des protocoles de routage réactifs pour connecter les nœuds à leurs voisins les plus proches dans la topologie physique. De plus, il met en place des opérations de raffinements permettant d'améliorer les performances globales de l'arbre de transmission en substituant un nœud avec son parent. PAST-DM, en revanche, construit l'arbre de transmission selon des décisions prises par la source. Ces décisions non locales nuisent à l'adaptation à l'effet de mobilité. En se basant sur des mesures RTT de bout en bout vers la source, ALMA est moins performant que AOMP du fait qu'il n'exploite pas les propriétés du plus court chemin du niveau routage.

5.3. Nombre de copies identiques de paquets sur un même lien physique

Nous cherchons à quantifier l'efficacité du schéma multipoint overlay AOMP. En calculant le stress sur les liens physiques en fonction de la taille du groupe overlay. Nous observons dans la figure 5 que PAST-DM génère un stress plus important (2 fois

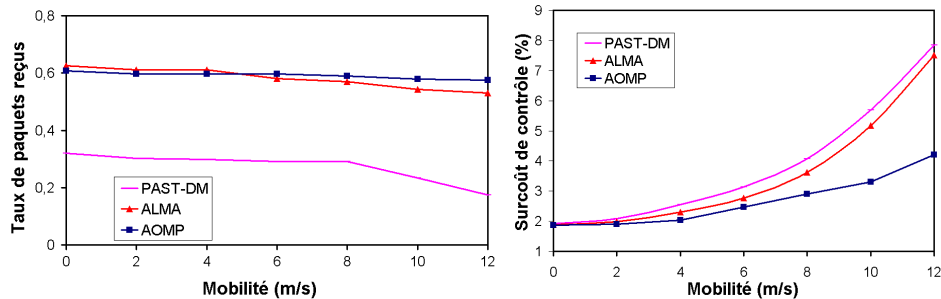


Figure 6. Ratio des paquets reçus vs mobilité. Figure 7. Surcoût de contrôle vs mobilité.

plus pour une taille de groupe de 40) que ALMA et AOMP. La technique de recherche des voisins de PAST-DM, basée sur la métrique nombre de sauts, n'est pas adaptée à une structure overlay applicative. En se rattachant dans l'arbre de transmission au nœud y le plus proche selon le nombre de sauts, un nœud x pourrait être plus proche de la source que le parent qu'il a choisi, en l'occurrence y . Ceci engendrerait une duplication inutile du paquet sur un même lien physique. ALMA utilise une métrique dynamique, le délai RTT, pour connecter deux voisins. La figure 5 montre que l'utilisation d'une telle métrique donne de meilleurs résultats que PAST-DM. Enfin, AOMP établit la connexion des nœuds overlay en utilisant un algorithme de correspondance de chemins. Grâce à cette technique, un paquet a de grandes chances pour ne pas repasser plusieurs fois par un même chemin physique.

5.4. Ratio des paquets reçus

Dans les simulations qui suivent, nous faisons varier la vitesse des nœuds de 0 à 12 m/s. La taille de l'overlay est de 40 nœuds. La figure 5 présente le ratio des paquets reçus par rapport à la mobilité des nœuds. Notons que ce ratio diminue en fonction de la mobilité et particulièrement pour PAST-DM (il passe de 0.35% vers 0.18%). PAST-DM perd des paquets lorsqu'un nœud recherche un nouveau parent et se connecte à lui. Les courbes de ALMA et AOMP sont très proches avec un léger avantage pour ALMA lorsque les nœuds sont peu mobiles (inférieur à 4m/s) et inversement pour AOMP lorsque la vitesse dépasse 4m/s. Ceci s'explique par le fait que ALMA utilise la métrique RTT de bout en bout pour déterminer un changement de topologie. Lorsque les nœuds sont peu mobiles, cette métrique est suffisante mais lorsque la vitesse augmente, AOMP qui détecte les changements grâce à DSR, réagit mieux. En outre, en cas de mouvement d'un nœud, s'il ne peut plus se connecter à son parent principal, l'arbre de transmission n'est modifié que lorsque ce nœud rejoint un nouveau parent principal.

5.5. Surcoût des messages de contrôle

La figure 7 présente le surcoût de contrôle par rapport à la mobilité. Notons que le surcoût de message augmente avec la mobilité des nœuds. En particulier, PAST-DM génère un coût important de messages de contrôle lors de la recherche de nouveaux voisins. ALMA a un surcoût moins important que PAST-DM mais reste très audessus de AOMP lorsque la vitesse des nœuds augmente (2 fois plus de surcoût de contrôle). Dans ALMA, tous les nœuds du groupe génèrent des messages de contrôle dans deux cas : vers leur parent périodiquement pour connaître son RTT, vers leurs voisins lors-

qu'ils détectent un RTT dépassant un seuil. AOMP produit moins de trafics de contrôle à cause de son mécanisme de reconnaissance des chemins (routage réactif). Il n'y a échange de paquets de contrôle que lorsque le mécanisme de raffinement est initié et lorsqu'un nœud intermédiaire détecte un changement de route.

6. Conclusion

Dans cet article, nous proposons un nouveau protocole AOMP (Ad-hoc Overlay Multicast Protocol) de transmission applicative multipoint dans les réseaux ad-hoc. AOMP est un protocole décentralisé, à moindre coût et s'adaptant rapidement à la mobilité. En exploitant les propriétés des protocoles de routage réactifs, nous introduisons un algorithme de correspondance de chemins permettant de connecter les nœuds les plus proches dans la topologie physique à un arbre principal. Une structure maillée est construite au dessus de l'arbre principal afin d'améliorer d'une part les performances en terme de surcoût de contrôle et d'assurer d'autre part la robustesse de la structure. Nous dérivons ainsi un arbre de transmission tenant en compte de la topologie physique et possédant un mécanisme de raffinement optimisant les performances globales. Nous avons effectué des simulations pour évaluer AOMP en le comparant à deux protocoles multipoints overlay ad-hoc : PAST-DM et ALMA. Ces simulations montrent que la qualité de l'arbre de transmission de AOMP est meilleure que ALMA du fait qu'il exploite les propriétés des chemins au niveau IP. Il engendre aussi peu de trafic de contrôle ce qui améliore les performances des réseaux ad-hoc. Dans les travaux futurs, nous explorerons des mécanismes de découverte de chemins plus généraux que DSR (niveau applicatif) afin d'assurer une indépendance vis à vis du protocole de routage. Il faudrait alors trouver un compromis entre les coûts des messages applicatifs générés pour cette découverte et l'indépendance recherchée. Nous allons, par ailleurs, approfondir les simulations et plus particulièrement, les conséquences de la mobilité.

7. Bibliographie

- [BLO 04] BLODT S, *Efficient End System Multicast for Mobile ad-hoc Networks*, Proceedings of Pervasive Computing and Communications Workshops, 2004.
- [CAN 04] MCCANNE S., ET AL., *NS network simulator*, <http://www.isi.edu/nsnam/ns/>, 2004.
- [GAR 99] GARCIA-LUNA-ACEVES J.J., ET AL., *The Core-Assisted Mesh Protocol*, IEEE Journal on Selected Areas in Communications, 1999.
- [GE 04] GE M., ET AL., *Overlay multicasting for ad hoc networks*, In The Third Annual Mediterranean Ad Hoc Networking Workshop, 2004.
- [GUI 03] GUI C., MOHAPATRA P., *Efficient Overlay Multicast for Mobile ad-hoc Networks*, Proceedings of IEEE WCNC, 2003.
- [JOH 03] JOHNSON D.B., ET AL., *The Dynamic Source Routing Protocol for Mobile ad-hoc Networks (DSR)*, draft IETF MQNET 2003.
- [KIM 05] KIM K.I, KIM S.H, *DESIRE : Density Aware Heterogenous Overlay Multicast Forwarding Scheme in Mobile ad-hoc Networks*, Proceedings of IEICE, 2005.
- [LEE 02] LEE S.J., ET AL., *On-Demand Multicast Routing Protocol in Multihop Wireless Mobile Networks*, Proceedings of Communications in Wireless Mobile Networks, 2002.
- [ROG 99] ROYER E., PERKINS C.E., *Multicast Operations of the Ad-hoc On-Demand Distance Vector Routing Protocol*, Proceedings of ACM/IEEE MOBICOM'99, 1999.
- [WU 99] WU C.W, TAY Y.C., *AMRIS : A Multicast Protocol for Ad-hoc Wireless Networks*, Proceedings of IEEE MILCOM'99, 1999.
- [XIE 02] XIE J., ET AL., *AMRoute : ad-hoc Multicast Routing Protocol*, Mobile Networks and Applications, 2002.