



**HAL**  
open science

## **JUMBO : protocole de routage unicast dans les réseaux ad-hoc**

Johanne Cohen, Eric Fleury, Jens Gustedt

► **To cite this version:**

Johanne Cohen, Eric Fleury, Jens Gustedt. JUMBO : protocole de routage unicast dans les réseaux ad-hoc. 2ièmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications - ALGOTEL 2000, 2000, La Rochelle, France. pp.31-34. inria-00099028

**HAL Id: inria-00099028**

**<https://inria.hal.science/inria-00099028>**

Submitted on 22 Dec 2010

**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.

# *JUMBO : protocole de routage unicast dans les réseaux ad-hoc.*

Johanne Cohen and Eric Fleury and Jens Gustedt<sup>†</sup>

LORIA, Université Henry Poincaré & INRIA, campus scientifique, BP 239, F-54506 Vandœuvre lès Nancy, France

---

Un réseau ad-hoc est 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. Du fait que le rayon de propagation des transmissions des entités de chacune soit limité, il se peut qu'un sommet se trouve dans l'obligation de demander de l'aide à un autre sommet pour pouvoir communiquer avec son correspondant. Nous présentons un protocole pour réaliser des communications dans les réseaux ad-hoc sans fils de type HIPERLAN. Après une description des contraintes propres à ces réseaux, nous décrivons une fonction de routage pour ces réseaux construite à partir d'une décomposition en sous graphes complets du réseau. Enfin, nous présentons succinctement un algorithme distribué permettant de construire une telle décomposition du graphe.

**Keywords:** Réseaux ad-hoc, fonction de routage, protocole de communications, algorithme distribué.

---

## 1 Modélisation d'un réseau sans fils

L'architecture d'un réseau sans fil ad-hoc peut se caractériser par une absence d'infrastructure fixe préexistante, à l'inverse des réseaux filaires ou des réseaux sans fil tels que nous les connaissons dans les télécommunications classiques (GSM par exemple). Les nœuds d'un réseau ad-hoc sont des hôtes mobiles ayant des capacités et une puissance de transmission identique, et possédant une certaine puissance de calcul. Un tel réseau doit s'organiser automatiquement de façon à être déployable rapidement et pouvoir s'adapter aux conditions de propagation, aux trafics et aux différents mouvements pouvant intervenir au sein des nœuds mobiles.

Du fait de l'absence d'une architecture fixe et de la mobilité des nœuds, la topologie d'un réseau ad-hoc peut évoluer à tout instant. Afin que le réseau reste connecté au sens classique du terme (tout sommet peut atteindre tout autre), chaque sommet est susceptible d'être mis à contribution pour participer au routage et pour retransmettre les paquets d'un nœud qui n'est pas en mesure d'atteindre directement sa destination. Tout nœud joue ainsi le rôle d'hôte et de routeur. Le groupe de travail de l'IETF<sup>‡</sup> étudie ce genre de problème.

Notre étude a été réalisée en prenant comme type de réseau ad-hoc la norme d'HIPERLAN<sup>§</sup>. Nous les principes des protocoles de routage que nous présentons s'applique aussi à d'autres réseaux ad-hoc.

Une des caractéristiques primordiales des réseaux sans fils est l'utilisation d'un médium de communication partagé qui est le canal de communication hertzien dans notre cas. Une transmission sélective se révèle donc impossible : lorsqu'un nœud émet, tous ces voisins (*i.e.*, tous les nœuds se trouvant dans la zone de couverture du sommet émetteur) vont recevoir le message et une collision se produit si tout autre sommet voisin tente de communiquer en même temps. Pour plus de détail sur les résolutions de collision et plus généralement sur HIPERLAN on se référera à [1, 5].

---

<sup>†</sup>Ce travail a été en partie financé par l'action coopérative COMPAS de l'INRIA <http://menetou.inria.fr/>

<sup>‡</sup> <http://www.ietf.org>

<sup>§</sup> <http://www.etsi.org>

La topologie d'un r eseaux ad-hoc  evoluant  a chaque instant, nous allons la mod eliser par un graphe  $G = (V, E)$  o u  $V$  repr esentent l'ensemble des n oeuds et  $E$  mod elise l'ensemble des connections existantes entre les n oeuds du r eseau. Il existe une ar ete  $e = (u, v) \in E$  si les n oeuds  $u$  et  $v$  sont en mesure de communiquer directement  a l'instant  $t$ .

 Etant donn e que tous les sommets utilisent une m eme fr equence pour communiquer, lorsqu'un n oeud  emet, il emp eche ces voisins d' emettre au m eme moment. Cette caract eristique fondamentale se traduit en la propri ete de base suivante valable  a chaque instant  $t$  au sein de chaque sous graphe complet (clique) du graphe  $G_t$  :

**Propri ete 1** Pour toute clique  $C$  de  $G_t$  :

- Au plus un n oeud  emet un message  a l'instant  $t$  ;
- Lorsqu'un des n oeuds de la clique  emet un message, tous les autres n oeuds de la clique peuvent entendre ce m eme message.

Pour g erer les  evolutions de la topologie, les n oeuds d'un r eseau HIPERLAN effectuent r eguli erement des communications pour signaler leur pr esence  a leurs voisins. De cette fa con, les n oeuds ont  a tout moment connaissance des n oeuds avec lesquels ils peuvent directement communiquer : la r eception d'un message « HELLO » signale la pr esence d'un n oeud avec lequel il est possible de communiquer directement ; la non r eception depuis un certain temps de messages de la part d'un n oeud signale la disparition de ce n oeud du voisinage, soit parce que le n oeud est sortie de la zone de couverture du fait de sa mobilit e, soit que le lien de communication soit devenu trop faible (trop de bruit, changement des conditions de propagation...). Chaque n oeud g ere localement ces  ev enements pour maintenir sa connaissance locale du r eseau.

## 2 Fonction de routage

Les hypoth eses de travail que nous avons faite rejoignent celles  enonc ees dans [3]. Le protocole de routage JUMBO est adapt e aux r eseaux assez denses dans lequel le trafic est al eatoire et sporadique entre plusieurs n oeuds du r eseaux. En effet, JUMBO est un protocole de type « pro-actif » dans le sens o u il va chercher  a maintenir une vue globale du r eseau dans chaque n oeud afin que ceux ci puissent router les messages contrairement  a une approche « r eactive » dans laquelle un n oeud va chercher  a d' ecouvrir une route avant de pouvoir communiquer [4].

Nous supposons qu' a tout instant il existe une **d ecomposition du r eseau en cliques** et que chaque n oeud la conna t. Une telle d ecomposition en cliques est repr esent ee par une famille  $C = \{C_1, \dots, C_l$  de cliques du graphe.

La fonction de routage est alors d' ecrite de la fa con suivante :

$$\mathcal{R}(v_{current}, dest) = (C, n_{next}) \quad (1)$$

o u

- $v_{current}$  est le n oeud courant.
- $dest$  est le n oeud destination du message.
- $C$  est la clique du graphe qui transmettra l'information  a la prochaine  etape.
- $n_{next} \in C$  est un n oeud de cette clique qui sera choisit de retransmettre l'information.

Cette d efinition ne d' ecrit pas un chemin entre deux sommets du graphe compos e d'ar etes comme il est d'usage dans les r eseaux filaires, mais le d' ecrit en terme de clique et de sommet responsable de la retransmission au sein de chacune des cliques. Cette progression d'un message de clique en clique pour aller d'un sommet  $u$   a un sommet  $v$  mod elise le fait que lorsqu'un sommet  emet, tous les autres sommets de la clique non seulement ne peuvent pas  emettre mais en plus re oivent le message. On peut alors appliquer le m eme genre de raisonnement pour la construction d'un arbre de multicast. Ce qui nous int eresse ce n'est plus un arbre recouvrant tous les membres du groupe mais bien un arbre recouvrant l'ensemble des cliques o u se trouve des membres du groupe. Cette notion de d ecomposition en clique est en fait n ee de notre volont e de proposer un protocole de routage multicast et de la constatation que la notion de multi

---

**Algorithme 1:** Gestion des événements liés à la modification de la topologie au niveau du nœud  $u$ .

---

```

1 D'écouter un événement lié à une modification du réseau ;
  suivant événement faire
2   cas où "  $v$  est un nouveau nœud du graphe "
3     Insérer si possible le nœud  $v$  dans les cliques dont  $u$  est responsable ;
4     Répondre à  $v$  en envoyant un message contenant les voisins de  $u$  plus les arêtes couvertes par
      insertion éventuelle de  $v$  dans les cliques dont  $u$  est responsable ;
5   cas où "  $v$  est un nouveau voisin de  $u$  "
6     Préparer l'émission d'un message qui avertit  $v$  de la gestion de cet événement ;
7     si réception du message avertissant  $u$  que  $v$  gère cette événement avant émission de ce message
      alors
9       Répondre à  $v$  en renvoyant un message contenant les voisins de  $u$  ;
8     sinon
9       Envoi du message ;
10      si réception de la réponse de  $v$  alors
11        insérer  $v$  dans les cliques dont  $u$  est responsable ;
12        si besoin alors créer une nouvelle clique réduite à l'arête  $uv$  ;
13   cas où " disparition avertie du nœud  $v$  "
14     Supprimer le nœud  $v$  dans la décomposition en cliques ;
15   cas où " manque de HELLO du nœud  $v$  "
16     Supprimer le nœud  $v$  dans la décomposition en cliques ;
17     Avertir les voisins de  $u$  de la suppression temporaire de  $u$  dans le réseau ;
18     Déclencher l'événement " insertion d'un nouveau nœud " pour  $u$  ;

```

---

points relais telle qu'elle est définie dans HIPERLAN (les multi points relais d'un sommet  $u$  forment un sous ensemble de ses voisins permettant d'atteindre tous les sommets à distance 2 de  $u$ ) est mal adaptée à la construction d'arbres de multicast.

Nous imposons que la décomposition en cliques du réseau respecte certaines contraintes :

**Propriété 2** *L'union des cliques de la décomposition est égale au graphe  $G_t$ .*

C'est-à-dire, toute arête est couverte par au moins un élément de la décomposition, et inversement, un élément de la décomposition ne doit pas couvrir des arêtes n'étant pas dans  $G$

**Propriété 3** *La décomposition  $C = C_1, \dots, C_\ell$  du graphe  $G_t$  doit vérifier  $\ell \leq m$  où  $m = |E_t|$  correspond aux nombres d'arêtes de  $G_t$ .*

Cette dernière propriété permet d'éviter une gestion trop coûteuse du routage, en raison d'un nombre trop élevé de cliques. Observons que la solution qui consisterait à construire une décomposition du graphe minimale en nombre d'éléments semble elle aussi très coûteuse (le problème MINIMUM CLIQUE COVER est NP-complet [2]). La propriété 3 permet de garantir un bon compromis entre les deux extrêmes.

### 3 Algorithme de décomposition en cliques

Les nœuds ont une vision locale du réseau : ils connaissent leurs voisins par l'intermédiaire de la réception ou de la non réception de messages HELLO. Pour que l'algorithme de routage s'effectue correctement, il faut toutefois qu'ils puissent en plus connaître à tout moment la décomposition globale du réseau en cliques afin de pouvoir calculer et mettre à jour des routes entre les cliques. Pour résoudre ce problème, nous proposons qu'à chaque clique soit associée un responsable. Ce nœud se charge régulièrement de diffuser les cliques dont il est le responsable à l'ensemble des sommets du réseau (par inondation). Ces diffusions permettent de

garantir que chacun des nœuds du réseau connaît à intervalles de temps régulier la décomposition globale du réseau en cliques. L'inondation est un moyen simple de diffusion mais il n'est pas à écarter l'utilisation de techniques plus efficace et moins coûteuse en terme de ressources utilisées pouvant se baser sur la décomposition en clique sous-jacente même si cette dernière est erronée ou incomplète.

Nous proposons maintenant un algorithme distribué pour construire une décomposition du graphe en sous graphe complets qui vérifie les propriétés 2 et 3.

Continuellement, les nœuds reçoivent des messages de contrôles (incluant les messages HELLO). Ces messages permettent de gérer les événements liés à la modification de la topologie du graphe (apparition d'un nœud, détection d'un nouveau voisin, disparition d'une arête). Les nœuds gèrent ces événements dans l'ordre de leurs arrivées selon l'algorithme 1 (écrit de façon informelle). Le coût de chaque modification du réseau pour réactualiser localement la décomposition du graphe vaut en nombre de messages  $O(d)$  et en opérations élémentaires  $O(d)$  où  $d$  correspond au nombre maximum de voisins dans le graphe.

Lors d'une insertion d'un nouveau nœud  $u$  dans le réseau,  $u$  détermine dans un premier temps son voisinage en écoutant de façon passive ce qui se passe. Ensuite, il informe tous ces voisins de l'événement " $u$  est un nouveau nœud du graphe" tout en leur transmettant la connaissance qu'il a de son voisinage. Il attend alors les réponses (cf. instruction 2 de l'algorithme) de ses voisins tout en gérant, si besoin les modifications locales du réseau (disparition d'un voisin, apparition d'un nouveau voisin, apparition d'un nouveau nœud du réseau dans le voisinage de  $u$ ). Lors ce qu'il a reçu toutes les réponses de ces voisins courants, il crée de nouvelles cliques pour couvrir toutes les arêtes non encore couvertes, et il devient responsable de toutes ces cliques. Si par exemple, pendant cette attente, le nœud  $u$  détecte une disparition d'un de ces voisins, le nœud réactualise les ensembles des nœuds dont il attend une réponse.

On démontre assez facilement que :

**Lemme 1** *La décomposition en cliques du réseau par l'algorithme distribué précédent est une décomposition valide qui vérifie les propriétés 2 et 3.*

## 4 Conclusion

Dans ce papier, nous avons proposé un algorithme de routage dans les réseaux sans fils ad-hoc de type HIPERLAN basé sur une décomposition en cliques du réseau. Cette approche se base sur le maintien d'une vue du réseaux afin de pouvoir tenir des tables de routage à jour et ce malgré la mobilité des hôtes au sein du réseau ad-hoc. L'approche de décomposition en clique, permettant de construire des routes est à l'origine destinée à la construction d'arbres de multicast. Nous allons donc poursuivre dans cette voie mais nous allons aussi tenter de déterminer les performances de notre algorithme de routage via des simulations sous MS mais via la mise en œuvre d'un prototype au dessus d'IP.

## Références

- [1] S. K. Barton, editor. *Wireless Personal Communication: Special issue on the High Performance Radio Local Area Network (HIPERLAN)*, volume 4. Kluwer Academic Publishers, January 1997.
- [2] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the theory of NP-Completeness*. a Serie of books in mathematical sciences. Freeman edition, 1979.
- [3] P. Jacquet, P. Muhlethaler, and A. Qayyum. Optimized link state routing protocol. Internet Draft draft-ietf-manet-olsr-00.txt, IETF MANET Working Group, November 1998.
- [4] D. B. Johnson. Routing in ad hoc networks of mobile hosts. In *IEEE Workshop on Mobile Computing Systems and Applications*, December 1994.
- [5] ETSI TC-RES. Radio Equipment and System (res); High Performance Radio Local Area Network (HIPERLAN), Type 1, Functional specification. Technical Report ETS 300 652, European Telecommunication Standards Institute, December 1995. Draft.