

# Un algorithme de clusterisation distribué pour les réseaux ad hoc structurés

Raphaël MASSIN<sup>1</sup>, Christophe J. LE MARTRET<sup>1</sup>, Philippe CIBLAT<sup>2</sup>

<sup>1</sup>Thales Communications & Security, service WFD, 92622 Gennevilliers, France

<sup>2</sup>Institut Mines-Telecom/Telecom ParisTech, département COMELEC, 75013 Paris, France

{raphael-a.massin, christophe.le\_martret}@thalesgroup.com, ciblat@enst.fr

**Résumé** – Cet article traite du problème de la clusterisation distribuée des réseaux ad hoc mobiles lorsque les nœuds appartiennent à une organisation opérationnelle. Dans ce cas, on recherche des algorithmes favorisant l’association de nœuds appartenant au même groupe dans le même cluster. Nous proposons un nouvel algorithme qui se distingue de l’état de l’art par sa conception et ses performances.

**Abstract** – This paper addresses the problem of the distributed clustering of ad hoc networks when nodes belong to an operational organization. In this case, we are looking for algorithms that favor the association of nodes from the same group in the same cluster. We propose here a new algorithm that distinguishes from the state of the art regarding its conception and its performance.

## 1 Introduction

Cet article traite du problème de la clusterisation des réseaux ad hoc mobiles lorsque les nœuds appartiennent à une organisation opérationnelle. C’est le cas des applications professionnelles, par exemple pour les réseaux des forces de sécurité civiles (pompiers, interventions risques naturels / risques technologiques, ...) et militaires. Cette organisation donne lieu à la notion de groupes opérationnels pour lesquels les nœuds à l’intérieur d’un groupe sont amenés à communiquer de manière plus fréquente qu’avec les nœuds des autres groupes. Nous utiliserons simplement *groupes* dans la suite par simplification.

La clusterisation est une technique permettant de regrouper les nœuds en *clusters* formant des sous réseaux connectés autorisant une gestion locale et distribuée des ressources. Elle permet ainsi d’optimiser l’allocation des ressources et de tenir l’effet d’échelle dans le cas de réseaux denses (dans lesquels les nœuds sont en portée radio d’un nombre important d’autres nœuds). De plus, en supposant que le trafic est plus important à l’intérieur d’un groupe qu’entre groupes, il paraît souhaitable de construire des clusters comprenant si possible tous les nœuds d’un même groupe. Cet objectif peut ne pas toujours être atteint par exemple lorsque les groupes ne forment pas un réseau connecté.

La plupart des solutions de clusterisation proposées dans la littérature ne prennent pas en compte la notion de groupe. Ces algorithmes sélectionnent tout d’abord des nœuds appelés *chefs de cluster*, ou cluster head (CH), auxquels s’associent ensuite les autres nœuds constituant ainsi les clusters. Le principe commun à tous ces algorithmes est qu’ils attribuent un poids à chaque nœud, les CH étant élus comme les nœuds de poids le plus élevé dans un voisinage donné. Il existe plusieurs ma-

nières pour calculer le poids d’un nœud : son identifiant, son degré, l’énergie restante de la batterie, les mesures de qualité des liens radios, etc., ou bien une combinaison de ces valeurs [1, 2]. D’autres utilisent la position et la vitesses des nœuds obtenus grâce à un dispositif GPS [3].

Parmi les rares articles prenant en compte la structure de groupe, on peut citer [4] qui définit l’algorithme *type-based algorithm* (TCA). Cet algorithme associe un poids aux nœuds appelé *facteur de stabilité*, qui prend en compte explicitement leur appartenance à une structure de groupe, chaque groupe étant différencié par un identifiant de sous-réseau IP différent. La limitation principale de TCA est que deux CH ne peuvent pas être voisins ce qui a donc pour conséquence la formation de clusters de grandes tailles lorsque les réseaux sont denses, ce qui est réhibitoire. Dans [5], nous avons appliqué l’idée du facteur de stabilité de TCA aux deux algorithmes GDMAC [1] et VOTE [2] pour leur permettre de prendre en compte les groupes.

Nous présentons dans cet article un nouvel algorithme permettant de clusteriser des nœuds possédant une structure opérationnelle et favorisant l’association de nœuds d’un même groupe dans un même cluster. Cet algorithme appelé clusterisation distribuée pour groupes opérationnels (CDGO) se distingue de l’état de l’art car il ne nécessite pas la notion de CH. De plus, ses performances sont nettement meilleures que les algorithmes existants.

L’article est organisé de la manière suivante. Dans la section 2, nous commençons par présenter le modèle du système considéré dans cette étude. Nous présentons ensuite le détail de l’algorithme CDGO dans la section 3, puis comparons ses performances avec les algorithmes développés dans [5].

## 2 Modèle de système

Soit un graphe  $\mathcal{G}$  défini par l'ensemble de ses nœuds  $\mathcal{V}$  et l'ensemble des liens  $\mathcal{E}$ . Le nombre de nœuds de  $\mathcal{G}$  est  $N := |\mathcal{V}|$ . Deux nœuds sont voisins s'il existe un lien entre eux dans  $\mathcal{E}$ . Une clusterisation de  $\mathcal{G}$  définit une partition de  $\mathcal{G}$ , qui contient  $N_c$  clusters. La taille d'un cluster  $k \in \{1, \dots, N_c\}$  est  $n_k^c$ . Dans cet article on impose à un cluster les contraintes suivantes : 1) il doit être connecté, 2) sa taille doit être inférieure ou égale à  $n_{max}$ , et 3) son diamètre doit être inférieur ou égal à  $d_{max}$ . Les nœuds sont organisés en groupes, chaque nœud n'appartenant qu'à un seul groupe. L'ensemble des groupes  $\mathcal{O}$  est défini comme  $\{\mathcal{O}_1, \dots, \mathcal{O}_{N_g}\}$ , avec  $N_g$  le nombre de groupes. La taille du groupe  $\mathcal{O}_k$  est notée  $n_k^g$ . Soit  $\mathcal{I}(\mathcal{C}_k)$  l'ensemble des index des groupes avec au moins un membre dans le cluster  $\mathcal{C}_k$ . Le nombre des membres du groupe  $\mathcal{O}_\ell$  dans le cluster  $\mathcal{C}_k$  est noté  $n_{\ell,k}^{gc}$ .

De manière identique à [6], le trafic utilisateur dépend des groupes : la probabilité qu'un nœud communique avec un autre nœud du même groupe est égale à  $\alpha \in [0, 1]$ , et celle qu'un nœud communique avec un nœud d'un autre groupe est  $1 - \alpha$ . Étant donné qu'un membre d'un groupe  $\mathcal{O}_k$  peut communiquer avec  $n_k^g - 1$  membres de ce groupe, la probabilité de joindre chacun de ces membres est  $\alpha / (n_k^g - 1)$ . De même la probabilité d'un nœud de communiquer avec un nœud en dehors de son groupe est  $(1 - \alpha) / (N - n_k^g)$ . La probabilité  $\pi_{j|i}$  qu'un nœud source  $i \in \mathcal{V}$  du groupe  $\mathcal{O}_k$  communique avec un nœud destination  $j \in \mathcal{V}$  s'écrit donc :

$$\pi_{j|i} := \begin{cases} \frac{\alpha}{n_k^g - 1} & \text{si } j \in \mathcal{O}_k, \\ \frac{1 - \alpha}{N - n_k^g} & \text{sinon.} \end{cases} \quad (1)$$

Dans les réseaux de clusters, les communications inter-cluster peuvent être réalisées avec une technologie MAC différente de celle utilisée pour les communications intra-cluster. En effet, à l'intérieur d'un cluster un nœud appelé allocateur de ressources (AR) peut gérer l'allocation des ressources radio (ARR) pour l'ensemble du cluster et ainsi l'optimiser localement. À contrario entre les clusters les ressources sont allouées d'une manière plus distribuée (par ex. via un dialogue entre les AR de clusters voisins), ce qui rend l'optimisation plus difficile. Par conséquent nous modélisons différemment les performances (délai, débit, etc.) des communication intra- et inter-cluster. Dans ce contexte, afin de mesurer la performance d'une solution de clusterisation, qui appliquée à un graphe  $\mathcal{G}$  donne une partition  $p$ , nous utilisons la métrique de performance  $J(p)$  définie dans [6]. Cette métrique de performance correspond au coût de communication moyen entre toutes les paires de nœuds du réseau, dans le contexte d'une métrique additive. Ses valeurs correspondent par exemple à un délai de bout en bout. Elle est définie de la façon suivante :

$$J(p) := \frac{1}{N} \sum_{(i,j) \in \mathcal{V}^2} \pi_{j|i} \cdot J_{i,j}(p), \quad (2)$$

avec le facteur  $1/N$  qui représente une probabilité égale pour

tous les nœuds  $i$  de transmettre.  $J_{i,j}(p)$  est calculée comme le coût de communication entre le nœud source  $i$  et le nœud destination  $j$  le long d'un plus court chemin. Ce plus court chemin est calculé en prenant en compte les différents coûts  $\tilde{\gamma}$  pour l'inter-cluster et  $\hat{\gamma}$  ( $< \tilde{\gamma}$ ) pour l'intra-cluster.

## 3 Algorithme de clusterisation distribuée

Les clusters formés par CDGO devraient comporter un maximum de membres des mêmes groupes, ainsi qu'une taille la plus proche possible d'une borne supérieure  $n_{max}$ . À cette fin un coût  $c(\mathcal{C}_k)$  est associé à chaque cluster  $\mathcal{C}_k$ . Pour déterminer si des nœuds  $\{u_i\}$  devraient se déplacer de leur cluster  $\mathcal{C}_k$  vers un cluster  $\mathcal{C}_\ell$ , le gain  $g(\{u_i\}, \mathcal{C}_k, \mathcal{C}_\ell)$  associé à ce mouvement doit être strictement positif. Le gain  $g(\{u_i\}, \mathcal{C}_k, \mathcal{C}_\ell)$  résulte de la différence entre 1) le gain lié à l'arrivée des nœuds  $\{u_i\}$  dans  $\mathcal{C}_\ell$ , et 2) la perte liée au départ de ces nœuds de  $\mathcal{C}_k$ . Le gain  $g()$  est défini de la façon suivante :

$$g(\{u_i\}, \mathcal{C}_k, \mathcal{C}_\ell) := [c(\mathcal{C}_\ell) - c(\mathcal{C}_\ell \cup \{u_i\})] - [c(\mathcal{C}_k \setminus \{u_i\}) - c(\mathcal{C}_k)]. \quad (3)$$

L'algorithme CDGO est décrit dans la table 1. Il est découpé en deux étapes après lesquelles les clusters satisfont les contraintes de connectivité, taille et diamètre. Au cours de la première étape (lignes 1-21), chaque cluster détermine si certains de ses membres devraient rejoindre un autre cluster. Dans la seconde étape (lignes 22-33), les nœuds changent de cluster tel que déterminé dans l'étape précédente après avoir vérifié que les conditions autorisant le changement des nœuds sont toujours valides. En effet, l'algorithme CDGO est exécuté indépendamment et de manière désynchronisée à chaque cluster, un choix de mobilité effectué pendant la première étape peut ne plus être valide au moment de le mettre en œuvre si le cluster destinataire a changé entre l'instant où la décision a été prise et où on cherche à l'appliquer.

CDGO associe un coût à chaque cluster, et cherche à construire des clusters dont les coûts sont minimums. Une fonction qui associe un coût à un cluster doit remplir les objectifs suivants :

1. Favoriser les clusters de taille maximale, c.à.d. égale à  $n_{max}$ ,
2. Amener à la construction de clusters qui contiennent un nombre le plus élevé possible de membres des mêmes groupes.

L'équation (4) définit la fonction qui atteint ces deux objectifs et que nous avons utilisée dans cet article. Cette fonction prend ses valeurs dans l'intervalle  $[0, 1]$ , elle est définie de la façon suivante :

$$c(\mathcal{C}_k) := 1 - \left[ f(n_k^c, n_{max}) \cdot \epsilon + \frac{1 - \epsilon}{n_{max}} \sum_{\ell \in \mathcal{I}(\mathcal{C}_k)} n_{\ell,k}^{gc} \cdot f(n_{\ell,k}^{gc}, n_\ell^g) \right], \quad (4)$$

avec

$$f(m, n) := \frac{m(m+1)}{n(n+1)}, \forall (m, n) \in \mathbb{N}^2, m \leq n. \quad (5)$$

La fonction  $f(m, n)$  est telle que :

$$[f(m+1, n) - f(m, n)] < [f(m+2, n) - f(m+1, n)].$$

### Étape 1

- 1 Faire  $\mathcal{M} = \emptyset$ ,  $\mathcal{L} = \emptyset$  et  $g = 0$
- 2 Pour chaque ensemble de nœud  $\{u_i\}$  membres du même groupe dans  $\mathcal{C}_k$  :
- 3 Si  $\mathcal{C}_k \setminus \{u_i\}$  est déconnecté, continuer ligne 2. Fin Si.
- 4 Si  $d(\mathcal{C}_k \setminus \{u_i\}) > d_{max}$ , continuer ligne 2. Fin Si.
- 5 Pour chaque cluster  $\mathcal{C}_\ell$  voisin du cluster  $\mathcal{C}_k$  :
- 6 Si  $|\mathcal{C}_\ell \cup \{u_i\}| > n_{max}$ , continuer ligne 5. Fin Si.
- 7 Si  $\mathcal{C}_\ell \cup \{u_i\}$  est déconnecté, continuer ligne 5. Fin Si.
- 8 Si  $d(\mathcal{C}_\ell \cup \{u_i\}) > d_{max}$ , continuer ligne 5. Fin Si.
- 9 Calculer  $g(\{u_i\}, \mathcal{C}_k, \mathcal{C}_\ell)$  à l'aide de l'équation (3).
- 10 Si  $g(\{u_i\}, \mathcal{C}_k, \mathcal{C}_\ell) > g$  :
- 11 Faire  $g = g(\{u_i\}, \mathcal{C}_k, \mathcal{C}_\ell)$ .
- 12 Faire  $\mathcal{L} = \{\mathcal{C}_\ell\}$ .
- 13 Sinon Si  $g(\{u_i\}, \mathcal{C}_k, \mathcal{C}_\ell) = g$  :
- 14 Faire  $\mathcal{L} = \mathcal{L} \cup \mathcal{C}_\ell$ .
- 15 Fin Si.
- 16 Fin Pour.
- 17 Si  $g = 0$  alors les nœuds  $\{u_i\}$  restent dans le cluster  $\mathcal{C}_k$ .
- 18 Sinon :
- 19 Choisir au hasard  $\mathcal{C}_\ell \in \mathcal{L}$ .
- 20 Faire  $\mathcal{M} = \mathcal{M} \cup (\{u_i\}, \mathcal{C}_\ell, g)$ .
- 21 Fin Si.

### Étape 2

- 22 Pour chaque  $(\{u_i\}, \mathcal{C}_\ell, g) \in \mathcal{M}$  évalués par valeur de  $g$  décroissantes :
- 23 Si  $\mathcal{C}_k \setminus \{u_i\}$  est déconnecté, continuer ligne 22. Fin Si.
- 24 Si  $\mathcal{C}_\ell = \emptyset$ , continuer ligne 22. Fin Si.
- 25 Si  $|\mathcal{C}_\ell \cup \{u_i\}| > n_{max}$ , continuer ligne 22. Fin Si.
- 26 Si  $d(\mathcal{C}_\ell \cup \{u_i\}) > d_{max}$ , continuer ligne 22. Fin Si.
- 27 Calculer  $g(\{u_i\}, \mathcal{C}_k, \mathcal{C}_\ell)$  à l'aide de l'équation (3).
- 28 Si  $g(\{u_i\}, \mathcal{C}_k, \mathcal{C}_\ell) \leq 0$ , continuer ligne 22. Fin Si.
- 29 Si  $\mathcal{C}_\ell$  est prêt :
- 30 Faire  $\mathcal{C}_\ell = \mathcal{C}_\ell \cup \{u_i\}$ .
- 31 Faire  $\mathcal{C}_k = \mathcal{C}_k \setminus \{u_i\}$ .
- 32 Sortir de la boucle Pour.
- 33 Fin Si.
- 34 Fin Pour.

TABLE 1 – Algorithme CDGO sur un cluster  $\mathcal{C}_k$ .

Par conséquent, le terme  $f(n_k^c, n_{max})$  de l'équation (4) signifie qu'il est meilleur d'un point de vue coût de rejoindre un cluster plus gros qu'un cluster plus petit. La forme du second terme  $\sum_{\ell \in \mathcal{I}(\mathcal{C}_k)} n_{\ell,k}^{gc} \cdot f(n_{\ell,k}^{gc}, n_\ell^g)$  permet d'assurer que le meilleur d'un point de vue coût est d'ajouter des membres d'un groupe au cluster incluant le plus grand nombre de membres de ce groupe. Ces deux termes sont pondérés par  $\epsilon$  et  $1 - \epsilon$  pour favoriser l'un des deux objectifs 1 ou 2.

L'adaptation à la mobilité est réalisée par une vérification périodique des contraintes. Un cluster ne satisfaisant plus une ou plusieurs contraintes se scinde de manière à ce que les clusters résultant les respectent à nouveau.

## 4 Résultats numériques

Pour évaluer les performances de l'algorithme proposé, nous l'avons comparé à cinq autres solutions de clusterisation : GDMAC [1], GDMAC-new1 [5], GDMAC-new2 [5] et VOTE-std [4], VOTE-new [5]. Ces protocoles existants ont été choisis car ils permettent d'adapter aisément le nombre et la taille des clusters comme notre algorithme. Seuls les trois protocoles définis dans [5] prennent en compte la structure de groupe.

Le réseau simulé considère  $N$  nœuds déployés sur un carré de 750 unités de distance tandis que les nœuds sont en visibilité jusqu'à 250 unités de distance. Les nœuds sont organisés en groupe de 10 nœuds et le modèle de mobilité de groupe est celui décrit dans [7]. Tous les résultats sont moyennés sur 100 réseaux. Le paramètre de trafic est fixé à  $\alpha = 0.9$ , et les paramètres de coût de communication sont  $\hat{\gamma} = 1$  et  $\tilde{\gamma} = 2$ . Nous avons également  $n_{max} = 20$  et  $\epsilon = 1/(1 + n_{max})^4$ . Cette valeur de  $\epsilon$  favorise la constitution de grands clusters avec beaucoup de membres du même groupe. Pour les paramètres des autres algorithmes, le lecteur consulera [5]. Les algorithmes sont implémentés séquentiellement. Ainsi, pour CDGO, dans une itération l'étape 1 est appliquée à tous les clusters les uns après les autres et l'étape 2 est ensuite appliquée également séquentiellement.

Sur la figure 1, nous traçons le nombre moyen de membres d'un cluster en fonction de  $N$ . Le CDGO conduit à un nombre de membres dans un cluster plus élevé que GDMAC et VOTE. Ceci est une bonne propriété puisque l'on cherche à obtenir les clusters les plus grands possibles dans la limite de  $n_{max}$ .

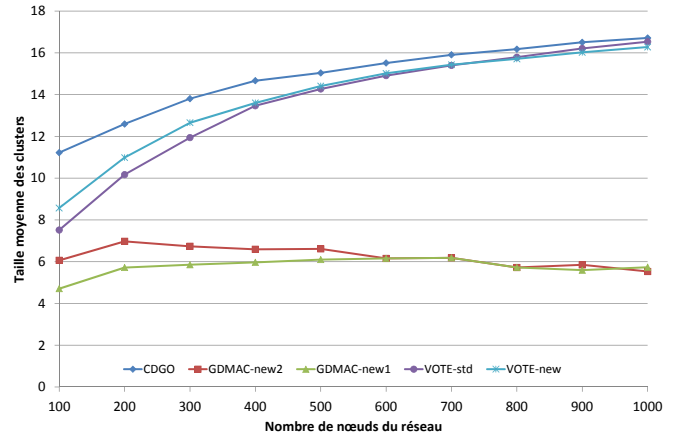


FIGURE 1 – Taille moyenne des clusters en fonction de  $N$ .

Sur la figure 2, nous affichons la diversité de groupes par cluster (CGD), c'est-à-dire, le nombre moyen de groupes présents dans un cluster. Le CDGO offre une faible CGD ce qui signifie que les clusters contiennent des membres de peu de groupes opérationnels et donc que les échanges d'information sont concentrés à l'intérieur des clusters. Par contre, les algorithmes existants ont une CGD plus élevée ce qui est peu satisfaisant.

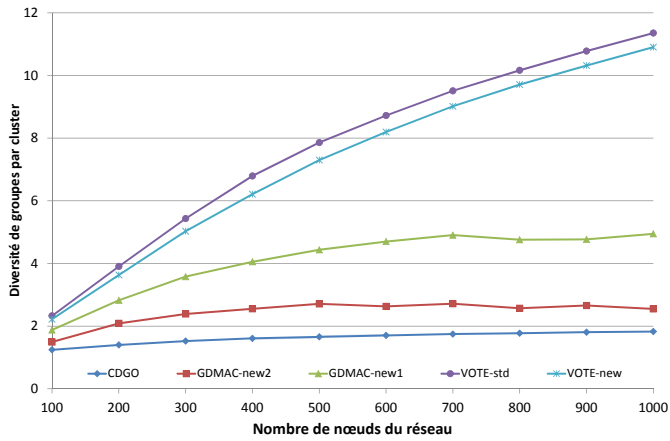


FIGURE 2 – Diversité de groupes par cluster en fonction de  $N$ .

Sur la figure 3, nous traçons  $J$  en fonction de  $N$ . Nous remarquons que le CDGO fournit les meilleures valeurs de  $J$  et donc offre les meilleures performances d'un point de vue applicative, puisque  $J$  typiquement peut représenter le retard moyen d'acheminement de l'information dans le réseau.

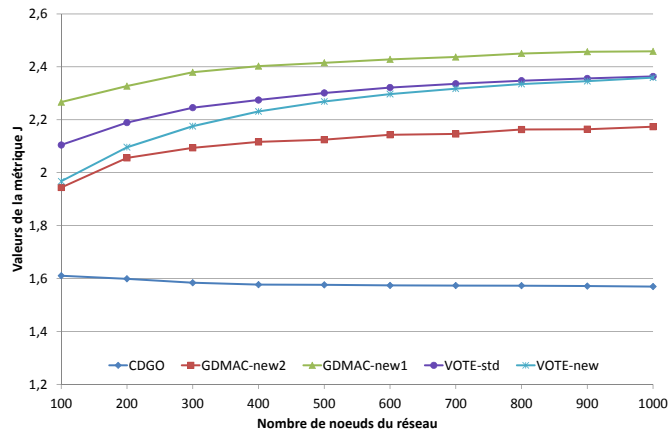


FIGURE 3 –  $J$  en fonction de  $N$ .

Sur la figure 4, nous analysons la stabilité du réseau (calculée comme le nombre d'itérations de simulation pour lesquels la clusterisation n'a pas changé divisé par le nombre total d'itérations) en situation de mobilité quand  $N = 100$ . Nous remarquons que le CDGO est très stable contrairement aux algorithmes existants.

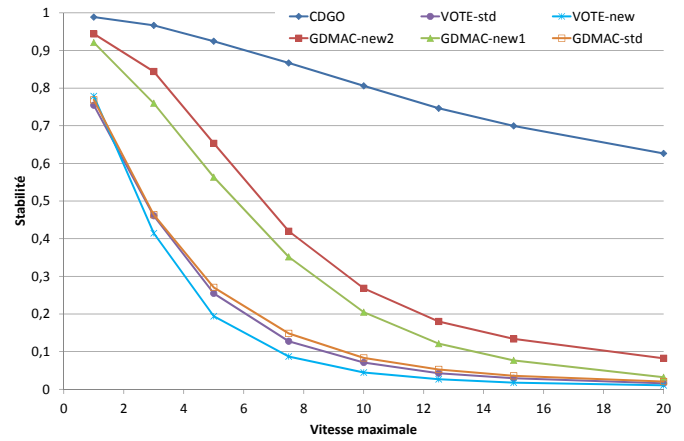


FIGURE 4 – Stabilité de la clusterisation vis-à-vis de la mobilité (vitesse de déplacement des nœuds).

## Références

- [1] R. Ghosh et S. Basagni, "Mitigating the impact of node mobility on ad hoc clustering", dans *Wireless Communications and Mobile Computing*, vol. 8, no. 3, pp. 295-308, Mars 2008.
- [2] F. Li, S. Zhang, X. Wang, X. Xue, et H. Shen, "Vote based clustering algorithm in mobile ad hoc networks," dans *Proc. ICON*, Busan, South Korea, Feb. 2004.
- [3] Y. Zhang et J. Ng, "A Distributed Group Mobility Adaptive Clustering Algorithm for Mobile Ad Hoc Networks", dans *Proc. IEEE ICC*, Beijing, China, May 2008.
- [4] H. Wu, Z. Zhong, et L. Hanzo, "A cluster-head selection and update algorithm for ad hoc networks", dans *Proc. IEEE GLOBECOM*, Dec. 2010.
- [5] R. Massin, C. J. Le Martret, et P. Ciblat, "Distributed Clustering Algorithms in Group-Based Ad Hoc Networks", dans *Proc. EURASIP EUSIPCO* ([http://perso.telecom-paristech.fr/~ciblat/docs\\_recherche/congres/eusipco2015.pdf](http://perso.telecom-paristech.fr/~ciblat/docs_recherche/congres/eusipco2015.pdf)), Nice (France), Sep. 2015.
- [6] R. Massin, C. J. Le Martret, et P. Ciblat, "A network cost function for clustered ad hoc networks : application to group based systems", dans *Proc. IEEE PIMRC*, Washington D.C. (USA), Sep. 2014.
- [7] X. Hong, M. Gerla, G. Pei, et C. Chiang, "A group mobility model for ad hoc wireless network", dans *Proc. ACM Workshop on Modeling and Simulation of Wireless and Mobile Systems (MSWiM)*, Aug. 1999.

## 5 Conclusion

Nous avons proposé un nouvel algorithme de clusterisation distribué adapté à la présence de groupe opérationnels dans le réseau. Il surpasses en performances les algorithmes existants.