

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mohammed Seddik Ben Yahia de Jijel  
Faculté des Sciences Exactes et Informatique  
Département d'Informatique



*Mémoire de fin d'étude  
pour l'obtention du diplôme de  
master en informatique*

*Option : Réseaux et sécurité*

Thème

**Election de k-meilleurs leaders dans  
les réseaux mobiles ad hoc**

**Présenté par :**

Benmicia Lamia.

Lebbada wiam.

**Encadré par :**

Melit leila

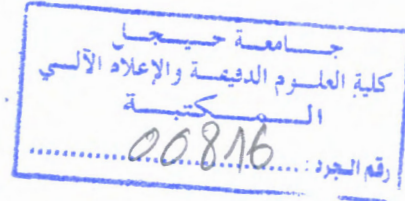
Promotion : 2019.

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mohammed Seddik Ben Yahia de Jijel  
Faculté des Sciences Exactes et Informatique  
Département d'Informatique

١  
٤

inf. RS, 3/19



*Mémoire de fin d'étude  
pour l'obtention du diplôme de  
master en informatique*

Option : Réseaux et sécurité

Thème

Election de k-meilleurs leaders dans  
les réseaux mobiles ad hoc

Présenté par :

Benmicia Lamia.

Lebbada wiam.

Encadré par :

Melit leila



Promotion : 2019.

## \* *Remerciements* \*

Nous tenons à remercier en premier lieu le Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce modeste travail.

Nos remerciements s'adressent aussi à notre promotrice madame **Melit Leila** qui nous a fait le grand honneur d'accepter de diriger notre travail. Aussi pour sa précieuse présence assistance, sa disponibilité et l'intérêt qu'elle a manifesté pour ce modeste travail, nous la remercions pour ses orientations et son enthousiasme envers notre travail. Les judicieux conseils et rigueur qu'elle nous a prodigué tout au long de ce travail. Nous avons pris un grand plaisir en travaillant avec elle.

Nous adressons également, nos remerciements chaleureux à Mademoiselle **Laouici Zineb** et aussi à tous les enseignants du département d'informatique et surtout nos enseignants monsieur **Rouibah Said** et **Khelifi Manel** pour leurs aides.

Nous tenons à remercier très sincèrement l'ensemble des membres du jury qui nous font le grand honneur d'accepter de juger notre travail.

Un grand merci à nos familles, la famille **Benmicia** et **Lebbada** qui nous ont toujours soutenu et encouragé sans oublier tous ceux qui ont participé à la réalisation de notre travail.

En fin, nous adressons nos plus sincères remerciements à tous nos amies, qui nous ont toujours soutenu et encouragé au cours de la réalisation de ce mémoire.

MERCI À TOUS

## *\* Dédicaces\**

*Je dédie ce modeste travail à :*

A mes chers parents **Fatima et Ismail** symbole de sacrifice, de tendresse et d'amour ; sont les moindres sentiments que je puisse vous témoigner.

Quoi que je fasse, je ne pourrais jamais vous récompenser pour les grands sacrifices que vous avez fait et continuez de faire pour moi .. merci

A mes frères **Ibrahim et Moussa**

A mes soeurs **Khaoula, Halla** et ma petite soeur **Amina Serine** ;

Je le dédie aussi à tous mes oncles **Hoçine** et tantes **Hassina, Hanane** et **Teldja**, cousins et cousines. Surtout **Soumia, Zineb** ;

A Ma binôme **Wiam** qui a partagé avec moi la préparation et la réalisation de ce travail .  
Tous mes amies **Farida, Khadija, Rabia, Radia, Badiaa** et tout mes amies et collègues sans exception .

A mon ami **Rami** pour leur aide, conseils et son encouragement, merci énormément

Ainsi qu'à Tous Ceux qui me sont Chers ....

BENMICIALAMIA

*\* Dédicaces\**

*Je dédie ce modeste travail à :*

*A mes chers parents **Fatiha et Massoud** symbole de sacrifice, de tendresse et d'amour ; sont les moindres sentiments que je puisse vous témoigner.*

*Quoi que je fasse, je ne pourrais jamais vous récompenser pour les grands sacrifices que vous avez fait et continuez de faire pour moi .. merci*

*A mes frères **Djiad, Mohammed taher et Abd aldjalil** ;*

*A mes soeurs **Ines, Mona** et ma petite soeur **Amira** ;*

*Je le dédie aussi à tous mes tantes **Naima, Nassira**  
A ma binôme **Lamia** qui a partagé avec moi la préparation et la réalisation de ce travail*

*Tous mes amies **Sara , Hoda , Wafa, Firouz , Naziha**, et tout mes amies et collègues sans exception .*

*Ainsi qu'à Tous Ceux qui me sont Chers ....*

**LEBBADA WIAM**

# Table des matières

<b>Table des matières</b>	<b>i</b>
<b>Liste des tableaux</b>	<b>iii</b>
<b>Table des figures</b>	<b>v</b>
<b>Liste des abréviations</b>	<b>vi</b>
<b>Introduction générale</b>	<b>1</b>
<b>1 Introduction aux réseaux mobiles Ad hoc</b>	<b>4</b>
1.1 Introduction . . . . .	4
1.2 Réseau mobile et réseau sans fil . . . . .	4
1.2.1 Les réseaux mobiles . . . . .	4
1.2.2 Les réseaux sans fils . . . . .	5
1.2.2.1 Définition . . . . .	5
1.2.2.2 Les catégories des réseaux sans fils . . . . .	5
1.3 Les environnements mobiles . . . . .	9
1.3.1 Les réseaux avec infrastructure . . . . .	10
1.3.2 Les réseaux sans infrastructure . . . . .	11
1.4 Présentation des réseaux mobiles ad hoc . . . . .	11
1.4.1 Historique des réseaux Ad Hoc . . . . .	11
1.4.2 Définition des réseaux mobiles Ad Hoc . . . . .	11
1.4.3 Caractéristiques des réseaux ad hoc . . . . .	12
1.4.4 Le routage dans les réseaux Ad Hoc . . . . .	14
1.4.4.1 Problématiques de routage dans les réseaux ad hoc . . . . .	15
1.4.4.2 Les protocoles de routage des MANETs . . . . .	15
1.4.5 Les applications des MANETs . . . . .	16
1.4.6 Les avantages des MANETs . . . . .	17
1.4.7 Les inconvénients des MANETs . . . . .	18
1.5 Conclusion . . . . .	19

<b>2</b>	<b>Le problème d'élection de K-meilleurs leaders dans les réseaux Ad hoc</b>	<b>20</b>
2.1	Introduction . . . . .	20
2.2	Élection de leader . . . . .	20
2.2.1	Définition . . . . .	20
2.2.2	Élection de leader dans les réseaux ad hoc . . . . .	21
2.3	Les algorithmes d'élection de leader . . . . .	22
2.3.1	Algorithmes basés sur le protocole TORA . . . . .	22
2.3.2	Les algorithmes d'élection selon la priorité ( Extrema-finding Algorithms)	24
2.3.2.1	L'algorithme LEAA( Leader Election Algorithm for Ad hoc Networks) . . . . .	24
2.3.2.2	MELFA : Message Efficient Leader Finding Algorithm for Mobile ad- hoc networks . . . . .	29
2.4	Algorithmes d'élection de k-meilleurs leaders . . . . .	35
2.4.1	L'algorithme CBLEAA (Candidate Based Leader Election Algorithm)	35
2.4.2	ELFA :Elite Leader Finding Algorithm for MANETs . . . . .	39
2.4.3	L'algorithme DELFA : DEMocratic Leader Finding Algorithm . . . . .	45
2.5	Conclusion . . . . .	55
<b>3</b>	<b>Simulation et résultats</b>	<b>56</b>
3.1	Introduction . . . . .	56
3.2	La simulation des réseaux ad hoc . . . . .	56
3.2.1	JSim . . . . .	57
3.2.2	NS . . . . .	57
3.2.3	OMNET++ . . . . .	58
3.2.4	GlomoSim . . . . .	58
3.3	Le choix de l'environnement de simulation . . . . .	58
3.4	Les paramètres de simulation . . . . .	59
3.5	Les métriques de performance . . . . .	59
3.6	Évaluation de performance . . . . .	60
3.6.1	Évaluation de performance dans un environnement statique . . . . .	60
3.6.2	Évaluation de performance dans un environnement dynamique . . . . .	64
3.6.2.1	Cas de 10 noeuds . . . . .	64
3.6.2.2	Cas de 20 noeuds . . . . .	66
3.6.2.3	Cas de 30 noeuds . . . . .	69
3.7	Conclusion . . . . .	71
	<b>Conclusion générale</b>	<b>72</b>
	<b>Bibliographie</b>	<b>74</b>
	<b>Annexe</b>	<b>1</b>

# Liste des tableaux

2.1	Les types de messages de l'algorithme LEAA . . . . .	25
2.2	Les variables maintenues par un noeud $i$ pendant le processus d'élection . . .	26
2.3	Types de message pour ELFA . . . . .	40
2.4	Les structures de données utilisées dans l'algorithme DELFA . . . . .	48



# Table des figures

1.1	Les catégories des réseaux sans fils . . . . .	5
1.2	La technologie Bluetooth . . . . .	6
1.3	La technologie Infrarouge . . . . .	7
1.4	La technologie WIFI . . . . .	8
1.5	Schéma de décomposition des réseaux mobiles . . . . .	10
1.6	Le modèle des réseaux mobiles avec infrastructure . . . . .	10
1.7	Le modèle des réseaux mobiles sans infrastructure . . . . .	11
1.8	Exemple d'un réseau ad hoc simple . . . . .	12
1.9	Le changement de topologie des réseaux ad hoc . . . . .	13
1.10	Le routage par relais . . . . .	14
1.11	Les opérations de secours . . . . .	17
2.1	Exemple d'exécution de l'algorithme CBLEAA. . . . .	37
2.2	Exemple en cas de partitionnement du réseau . . . . .	38
2.3	Exemple de fusion des partitions . . . . .	39
2.4	Modèle du système de ELFA . . . . .	42
2.5	Exemple de l'exécution de l'algorithme ELFA. . . . .	45
2.6	Vue abstraite de DELFA . . . . .	46
3.1	La topologies des scénarios statiques . . . . .	60
3.2	La topologies de partitionnement . . . . .	61
3.3	La topologie initiale du fusionnement . . . . .	61
3.4	La topologie après fusionnement . . . . .	61
3.5	Nombre de message avec 5 noeuds . . . . .	62
3.6	Temps d'élection avec 5 noeuds . . . . .	62
3.7	Temps de convergence avec 5 noeuds . . . . .	63
3.8	Taux de stabilisation avec 5 noeuds . . . . .	63
3.9	Nombre de messages avec 10 noeuds . . . . .	64
3.10	Temps d'élection avec 20 noeuds . . . . .	65
3.11	Temps de convergence avec 10 noeuds . . . . .	65
3.12	Taux de stabilisation avec 10 noeuds . . . . .	66
3.13	Nombre de messages avec 20 noeuds . . . . .	67

3.14 Temps d'élection avec 20 noeuds . . . . .	68
3.15 Temps de convergence avec 20 noeuds . . . . .	68
3.16 Taux de stabilisation avec 20 noeuds . . . . .	69
3.17 Nombre de messages avec 30 noeuds . . . . .	69
3.18 Temps d'élection avec 30 noeuds . . . . .	70
3.19 Temps de convergence avec 30 noeuds . . . . .	70
3.20 Taux de stabilisation avec 30 noeuds . . . . .	71

# Liste des abréviations

<b>WPAN</b>	Wireless Personal Area Network
<b>PDA</b>	Personal Digital Assistant
<b>irDA</b>	(infrared data association)
<b>WLAN</b>	Wireless Local Area Network
<b>WIFI</b>	Wireless Fidelity
<b>WECA</b>	(Wireless Ethernet Compatibility Alliance)
<b>ETSI</b>	European Telecommunications Standards Institut
<b>WMAN</b>	Wireless Metropolitan Area Network
<b>BLR</b>	Boucle Locale Radio
<b>WIMAX</b>	Worldwide Interoperability for Microwave Access
<b>BTS</b>	Base Transceiver Station
<b>WWAN</b>	Wireless Wide Area Network
<b>GSM</b>	Global System for Mobile Communication
<b>GPRS</b>	General Packet Radio Services
<b>UMTS</b>	Universal Mobile Telecommunications System
<b>DARPA</b>	Defence Advanced Research Project Agency
<b>ONR</b>	Office Naval Research
<b>MANET</b>	Mobile Ad hoc NETWORK
<b>TORA</b>	Temporally-Ordered Routing Algorithm
<b>DAG</b>	Directed Acyclic Graph
<b>LEAA</b>	Leader Election Algorithm for Ad hoc Networks
<b>FIFO</b>	First In First Out
<b>MELFA</b>	Message Efficient Leader Finding Algorithm for Mobile ad-hoc networks
<b>CBLEAA</b>	Candidate Based Leader Election Algorithm
<b>ELFA</b>	Elite Leader Finding Algorithm for MANETs
<b>DLFA</b>	DEMocratic Leader Finding Algorithm

# Introduction générale

Urant ces dernières années, nous avons été marqués par un développement très rapide et récent des technologies de communication sans fil et l'émergence de terminaux mobiles (portables, smartphones, etc). Avec ce succès, il devient possible d'accéder au réseau partout et à tout moment sans avoir besoin de connecter physiquement les appareils communicants à une infrastructure. Un avantage indéniable de ces technologies sans fil est la possibilité "d'être mobile tout en restant connecté". Ainsi, le paradigme sans fil a vu naître, au cours de son évolution, diverses architectures dérivées, telles que les réseaux cellulaires, les réseaux locaux sans fil, etc. Les réseaux ad hoc représentent une composante clé de cette évolution qui répondent à ce besoin en supprimant la nécessité d'infrastructure fixe pour communiquer.

Un réseau mobile ad hoc qui s'appelle aussi MANET (Mobile Ad hoc NETwork) est un réseau composé d'un ensemble des équipements ou des noeuds mobiles hétérogènes qui se déplacent de façon libre et arbitraire dans un territoire quelconque et dont le seul moyen de communication est l'utilisation des interfaces sans fil, sans l'aide d'une infrastructure préexistante ou administration centralisée. En outre, les réseaux ad hoc sont caractérisés par des changements de topologie rapides et aléatoires, d'une manière où l'interconnexion entre les noeuds peut changer à tout moment. Ce qui nous oblige à changer la vision classique des problèmes liés aux systèmes distribués.

Dans un système réparti, beaucoup d'applications exigent qu'un seul processus soit désigné pour la coordination et l'ordonnancement d'une tâche effectuée par d'autres processus. Généralement, beaucoup de processus dans le système sont potentiellement capables d'offrir une telle coordination. Cependant, seulement un processus est en mesure d'offrir vraiment cette fonction. Par conséquent, un processus appelé le leader, doit être choisi pour assurer cette fonction. Un leader dans un système réparti est un processus coordinateur qui exécute certaines fonctions pour le compte des autres processus du système. En plus, si le processus élu tombe en panne, un nouveau leader doit être élu.

La notion d'élection un leader est d'élire finalement un leader unique parmi un ensemble fini de noeuds, les algorithmes d'élection ont plusieurs domaines d'application. Un algorithme d'exclusion mutuelle, par exemple, peut être modifié pour être plus efficace avec un algorithme d'élection incorporé dans les noeuds. L'élection d'un leader est aussi utile

dans les protocoles de communication de groupes afin de choisir un nouveau coordinateur lorsque la composition du groupe change. Cependant, dans un environnement mobile ad hoc qui se caractérise essentiellement par des changements fréquents de topologie induits par la mobilité des sites, des connexions/déconnexions et des défaillances de communication, l'élection d'un leader devient plus compliquée. Il y aura une période où une ou plusieurs composantes n'auront pas de leader. Un algorithme d'élection doit garantir que chaque nouvelle composante (séparée de l'ancien leader) doit élire exactement un leader unique. Il peut aussi y avoir une période pendant laquelle il y aura co-existence de deux ou plusieurs leaders suite à la fusion de deux ou plusieurs composantes. Cependant, l'algorithme doit garantir que seulement un seul de ces leaders survivra. Il est à noter qu'il est impossible de garantir un leader tout le temps. Par exemple, lors de la fusion de deux composantes, il faut un peu de temps pour que le nouveau leader soit élu. Ainsi, le problème d'élection est défini dans le contexte des réseaux mobiles ad hoc comme suit :

*Chaque composante connexe du réseau mobile ad hoc finira par avoir exactement un seul leader.*

Cependant, dans plusieurs situations, il pourrait être souhaitable d'élire un leader selon une priorité qui représente une caractéristique inhérente au système tel que l'énergie résiduelle de la batterie ou bien la capacité de calcul des noeuds, etc. Par conséquent, le noeud élu doit être le noeud qui a la plus grande priorité parmi tous les noeuds dans sa composante.

En outre, dans de nombreuses applications des systèmes distribués, la présence du leader est nécessaire pour la réalisation de certaines tâches. Lors de départ de ce leader, le système reste en panne jusqu'à l'élection d'un nouveau leader qui nécessite un temps de ré-élection qui peut être important. Ainsi, un nouveau concept est introduit pour réduire le temps d'absence de leader. La solution consiste à élire un ensemble de leaders et remplacer le leader actuel, en cas de son départ, par le vice leader et ainsi de suite jusqu'à la terminaison de l'ensemble de leaders. Dans ce cas, seulement on doit refaire le processus d'élection à nouveau.

L'objectif principal de ce projet est d'étudier les algorithmes d'élection de K meilleurs leaders dans les réseaux ad hoc, afin d'évaluer leurs performances par rapport à ceux qui font d'élection d'un seul leader.

Notre mémoire s'articule en trois chapitres comme suit :

Le premier chapitre intitulé « Introduction aux réseaux mobiles Ad hoc » est consacré à la présentation de certains concepts fondamentaux relatifs aux réseaux sans fils ainsi que des généralités sur les réseaux mobiles ad hoc ; une définition, leurs caractéristiques principaux, leurs domaines d'application, ainsi que les problèmes de routages liés à ce type de réseaux.

Le second chapitre nommé « Le problème d'élection de K-meilleurs leader dans les réseaux Ad hoc » présente un état de l'art des algorithmes d'élections de K meilleurs leader. Nous commençons par la définition de l'élection de leader d'une manière générale puis en particulier dans les MANETs. Ensuite, nous présentons les principaux algorithmes résolvant le problème d'élection existants dans la littérature en mettant l'accent sur les algorithmes qui font l'élection de K meilleurs leaders.

Le troisième chapitre intitulé « Simulation et Résultats » est consacré à l'analyse des performances des algorithmes de CBLEAA, DELFA qui font l'élection d'un nombre K de leaders en les comparant avec l'algorithme de LEAA qui élit un leader unique. La simulation est faite en utilisant le simulateur réseau JSim.

Nous terminerons ce mémoire par une conclusion générale et des perspectives.

# Introduction aux réseaux mobiles Ad hoc

## 1.1 Introduction

Les communications sans fil ont un rôle crucial à jouer au sein des réseaux informatiques. Elles offrent des solutions ouvertes pour fournir de la mobilité ainsi que des services essentiels là où l'installation d'infrastructures n'est pas possible. Ces réseaux sont en plein développement du fait de leur flexibilité de leur interface, qui offre à un utilisateur la mobilité. Les environnements mobiles permettent une grande flexibilité d'emploi.

Les réseaux mobiles ad hoc constituent un nouveau type de réseaux basés sur cette technologie. L'évolution récente de cette dernière et l'apparition des unités de calculs portables (les laptops par exemple), poussent aujourd'hui les chercheurs à faire des efforts afin de réaliser le but des réseaux : " l'accès à l'information n'importe où et n'importe quand ".

Dans ce chapitre nous allons présenter les réseaux sans fil et plus précisément les réseaux mobiles ad hoc, leurs caractéristiques, leurs domaines d'applications ainsi que le problème de routage dans ces environnements instables. En outre, nous énumérons quelques avantages et inconvénients des réseaux ad hoc.

## 1.2 Réseau mobile et réseau sans fil

### 1.2.1 Les réseaux mobiles

Un réseau est dit mobile, s'il permet à ses utilisateurs d'accéder à l'information, indépendamment de leurs positions géographiques. Pour communiquer entre eux, les noeuds du réseau mobile, utilisent une interface de communication sans fil ( médium radio ou infra-rouge), qui permet de propager les signaux sur une certaine distance. Les réseaux mobiles offrent une plus grande flexibilité d'emplois et un plus grand confort par rapport aux réseaux statiques [1].

## 1.2.2 Les réseaux sans fils

### 1.2.2.1 Définition

Un réseau sans fil (wireless network en anglais) est un réseau informatique qui connecte différents hôtes ou noeuds par des ondes radios. Les réseaux sans fil constituent avant tout une alternative aux réseaux câblés. Leur compatibilité avec les réseaux câblés permet également de les ajouter comme extension. C'est une technique qui permet aux particuliers, aux réseaux de télécommunications et aux entreprises de limiter l'utilisation de câbles entre diverses localisations [2].

### 1.2.2.2 Les catégories des réseaux sans fils

Les réseaux sans fil sont basés sur une liaison utilisant des ondes radioélectriques (radio et infrarouges) au lieu et à la place des câbles habituels. Il existe plusieurs technologies se distinguant d'une part par la fréquence d'émission utilisée ainsi que le débit et la portée des transmissions. Les réseaux sans fil permettent de relier très facilement des équipements distants d'une dizaine de mètres à quelques kilomètres.

On distingue habituellement plusieurs catégories de réseaux sans fil, selon le périmètre géographique offrant une connectivité (appelé zone de couverture) (voir la figure suivante) :

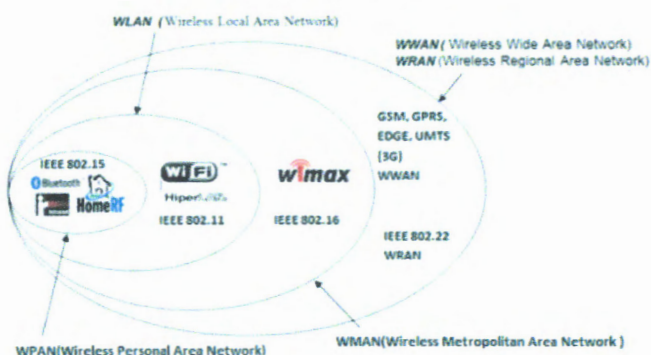


FIGURE 1.1 – Les catégories des réseaux sans fils [3]

**Réseaux personnels sans fil (WPAN) :** Les réseaux personnels sans fil (appelés également réseaux individuels sans fil ou réseaux domestiques sans fil et notés WPANs pour Wireless Personal Area Networks), sont des réseaux sans fil à très faible portée, de l'ordre d'une dizaine de mètres. Ils sont le plus souvent utilisés à faire communiquer entre eux des matériels présents sur une personne (par exemple une oreillette et un téléphone portable). Ils sont également utilisés pour relier des équipements informatiques entre eux sans liaison filaire : par exemple pour relier une imprimante ou un PDA (Personal Digital Assistant) à bureau ou faire communiquer deux machines très peu distantes [4].



Il existe plusieurs technologies utilisées pour les WPANs :

- **Bluetooth**

La norme Bluetooth (prise en charge par IEEE 802.15.1) est une technologie de moyen débit, elle permet d'atteindre un débit maximal théorique de 1Mbps (environ 720Kbps effectif) à basse consommation énergétique. Bluetooth utilise la bande de fréquence 2.4GHz.

Il permet notamment une communication d'une pièce à une autre, sur de petits espaces. Chaque appareil doit disposer d'une puce électronique contenant le protocole Bluetooth.

L'objectif de Bluetooth est de permettre la transmission des données ou de la voix entre des équipements possédant un circuit radio de faible coût, sur un rayon de l'ordre d'une dizaine de mètres à un peu moins d'une centaine de mètres et avec une faible consommation électrique [5] [6].

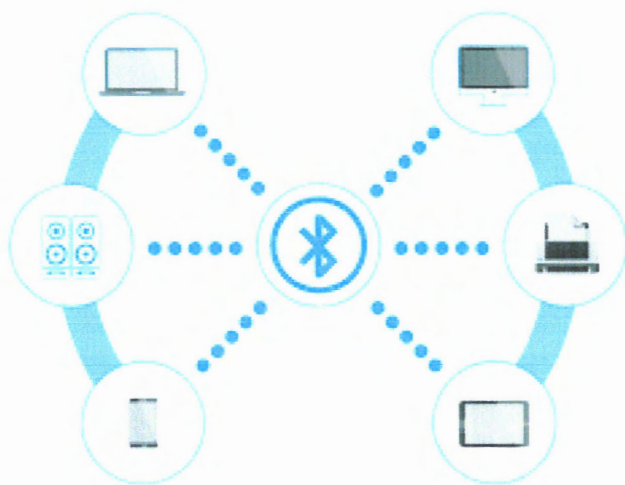


FIGURE 1.2 – La technologie Bluetooth

- **Liaisons infrarouges (IrDA : Infra-red Data Association)**

Enfin les liaisons infrarouges permettent de créer des liaisons sans fil de quelques mètres avec des débits pouvant monter à quelques mégabits par seconde. Cette technologie est largement utilisée pour la domotique (télécommandes) mais souffre toutefois des perturbations dues aux interférences lumineuses. L'association IrDA formée en 1995 regroupe plus de 150 membres [5].



FIGURE 1.3 – La technologie Infrarouge

**Réseaux locaux sans fil (WLAN)** Le réseau local sans fil correspond au périmètre d'un réseau local installé dans une entreprise, dans un foyer ou encore dans un espace public. Tous les terminaux situés dans la zone de couverture du WLAN peuvent s'y connecter. Plusieurs WLAN peuvent être synchronisés et configurés de telle manière que le fait de traverser plusieurs zones de couverture est pratiquement indécélable pour un utilisateur [3] :

- **WIFI (Wireless Fidelity) :**

La norme IEEE 802.11 (ISO/IEC 8802-11) est un standard international décrivant les caractéristiques d'un réseau local sans fil (WLAN). Le nom Wi-Fi (contraction de Wireless Fidelity, parfois notée à tort WiFi) correspond initialement au nom donné à la certification délivrée par la Wi-Fi Alliance, anciennement WECA (Wireless Ethernet Compatibility Alliance), l'organisme chargé de maintenir l'interopérabilité entre les matériels répondant à la norme 802.11.

Grâce au Wi-Fi il est possible de créer des réseaux locaux sans fils à haut débit pour que la station à connecter ne soit pas trop distante par rapport au point d'accès. Dans la pratique, le Wi-Fi permet de relier des ordinateurs portables, des machines de bureau, des assistants personnels (PDA) ou tout type de périphérique à une liaison haut débit (11 Mbps ou supérieur) sur un rayon de plusieurs dizaines de mètres en intérieur (généralement entre une vingtaine et une cinquantaine de mètres) à plusieurs centaines de mètres en environnement ouvert. Ainsi des opérateurs commencent à irriguer des zones à fortes concentration d'utilisateurs (gares, aéroports, hôtels, trains, ...) avec des réseaux sans fils. Ces zones d'accès sont appelées "hot spots" [5].



FIGURE 1.4 – La technologie WIFI

- **Hiperlan :**

Élaboré par l'ETSI (European Telecommunications Standards Institut), Hiperlan est exclusivement une norme européenne. La technologie de Hiperlan exploite la bande de fréquence de 5Ghz et les débits changent selon la version, ainsi : Hiperlan1 apporte un débit de 20 Mbit/s et Hiperlan2 offre un débit de 54 Mbit/s sur une portée d'action semblable dans celui de la Wi-Fi (100 mètres) [4].

**Les réseaux métropolitains sans fil (WMAN)** Les réseaux métropolitains sans fil ou Wireless Metropolitan Area Network (WMAN) sont aussi connus sous l'appellation de boucle locale radio (BLR). Les réseaux basés sur la technologie IEEE 802.16 ont une portée de l'ordre de quelques dizaines de kilomètres (50km de portée théorique annoncée) et un taux de transmission radio théorique pouvant atteindre 74 Mbit/s pour IEEE 802.16, plus connu sous le nom commercial de WiMAX(acronyme pour Worldwide Interoperability for Microwave Access) [2].

L'objectif du WiMAX est de fournir une connexion internet à haut débit sur une zone de couverture de plusieurs kilomètres de rayon. Ainsi, dans la théorie, le WiMAX permet d'obtenir des débits montants et descendants de 70 Mbit/s avec une portée de 50 kilomètres. Le standard WiMAX possède l'avantage de permettre une connexion sans fil entre une station de base (en anglais Base Transceiver Station, notée BTS) et des milliers d'abonnés sans nécessiter de ligne visuelle directe (en anglais Line Of Sight, parfois abrégés LOS) ou NLOS

pour Non Line Of Sight) [5]. Elle nécessite que les antennes émettrices et réceptrices soient situées l'une en face de l'autre pour que les transmissions passent, cette technologie est très utile pour éviter les coûteuses liaisons câblées qui étaient jusque là nécessaires pour apporter l'internet à haut débit dans les régions moins peuplées. WiMAX peut être utilisé en complément de Wifi pour relier deux réseaux trop éloignés l'un de l'autre, par exemple deux bâtiments d'une même entreprise [7].

**Les réseaux sans fil étendus (WWAN) :** Le réseau étendu sans fil (WWAN pour Wireless Wide Area Network) est également connu sous le nom de réseau cellulaire mobile. Il s'agit des réseaux sans fil les plus répandus puisque tous les téléphones mobiles sont connectés à un réseau étendu sans fil. Les principales technologies sont les suivantes :

- **GSM :**

Constitue au début du 21<sup>ème</sup> siècle le standard de téléphonie mobile le plus utilisé en Europe. Il s'agit d'un standard de téléphonie dit de seconde génération (2G), car contrairement à la première génération de téléphones portables, les communications fonctionnent selon un mode entièrement numérique.

- **GPRS :**

Permet d'étendre l'architecture du standard GSM, afin d'autoriser le transfert de données par paquets, avec des débits théoriques maximums de l'ordre de 171,2 Kbits/s (en pratique jusqu'à 114 Kbits/s). Grâce au mode de transfert par paquets, les transmissions de données n'utilisent le réseau que lorsque c'est nécessaire. Le standard GPRS permet donc de facturer l'utilisateur au volume échangé plutôt qu'à la durée de connexion, ce qui signifie notamment qu'il peut rester connecté sans surcout [8].

- **UMTS :**

UMTS L'abréviation de « Universal Mobile Telecommunications System », l'UMTS désigne une nouvelle norme de téléphonie mobile. Le principe de l'UMTS consiste à exploiter une bande de fréquences plus grande pour faire transmettre plus des données et donc obtenir un débit plus important. En théorie, il peut atteindre 2 Mb/s.

La norme d'UMTS exploite de nouvelles bandes de fréquences situées entre 1900 et 2200 MHz. Cette technologie permet de faire passer des données simultanément et offre alors des débits nettement supérieurs à ceux atteints par le GSM et le GPRS [4].

### 1.3 Les environnements mobiles

Un environnement mobile est un système composé de sites mobiles qui permet à ses utilisateurs d'accéder à l'information indépendamment de leurs positions géographiques. Les réseaux mobiles ou sans fil, peuvent être classés en deux catégories : les réseaux avec infrastructure et les réseaux sans infrastructure [9].

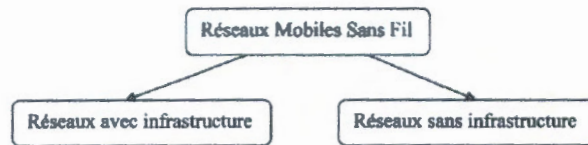


FIGURE 1.5 – Schéma de décomposition des réseaux mobiles

### 1.3.1 Les réseaux avec infrastructure

Le modèle de réseau mobile avec infrastructure intègre deux ensembles d'entités distinctes : les sites fixes d'un réseau de communication filaire classique (wired network), et les sites mobiles (wireless network). Certains sites fixes, appelés stations support mobile (Mobile Support Station) ou station de base (SB) sont munis d'une interface de communication sans fil pour la communication directe avec les sites ou unités mobiles (UM), localisés dans une zone géographique limitée, appelée cellule.

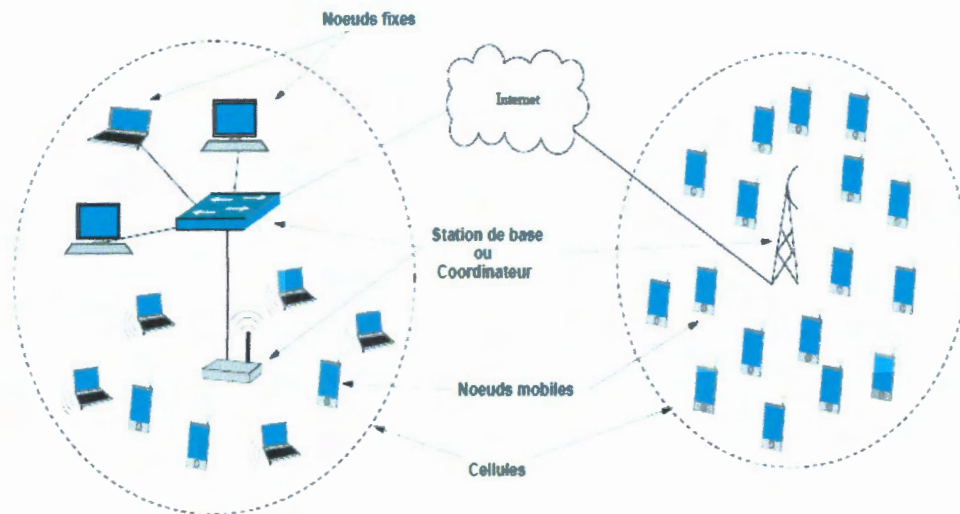


FIGURE 1.6 – Le modèle des réseaux mobiles avec infrastructure [10]

A chaque station de base correspond une cellule à partir de laquelle des unités mobiles peuvent émettre et recevoir des messages. Les sites fixes sont interconnectés entre eux à travers un réseau de communication filaire, généralement fiable et d'un débit élevé. Les liaisons sans fil ont une bande passante limitée qui réduit sévèrement le volume des informations échangées [11]. Dans ce modèle, une unité mobile ne peut être, à un instant donné, directement connectée qu'à une seule station de base. Elle peut communiquer avec les autres sites à travers la station à laquelle elle est directement rattachée [9].

### 1.3.2 Les réseaux sans infrastructure

Le modèle de réseau mobile sans infrastructure préexistante ne comporte pas de site fixe. Tous les sites du réseau sont mobiles et communiquent d'une manière directe en utilisant leurs interfaces de communication sans fil (voir figure 1.7). L'absence d'infrastructure ou de réseau filaire composé de stations de base, oblige les unités mobiles à se comporter comme des routeurs qui participent à la découverte et la maintenance des chemins pour les autres hôtes du réseau.

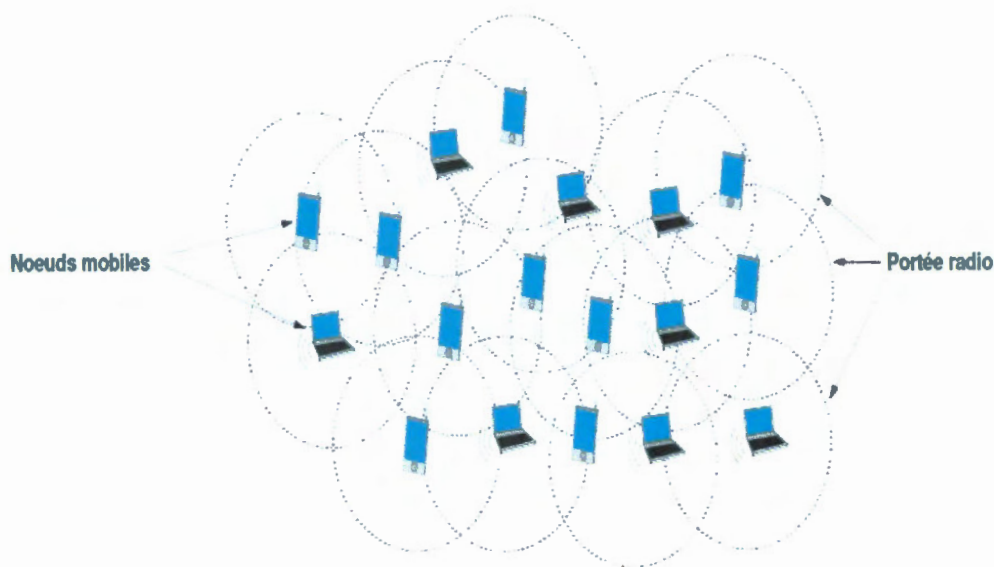


FIGURE 1.7 – Le modèle des réseaux mobiles sans infrastructure [10]

## 1.4 Présentation des réseaux mobiles ad hoc

### 1.4.1 Historique des réseaux Ad Hoc

A l'origine les réseaux ad hoc mobiles ont été introduits pour améliorer les communications dans le domaine militaire, vu la nature dynamique de leurs opérations et champs d'action. Les recherches sur ces réseaux ont été financées par le gouvernement américain et supervisées principalement par le DARPA (Defence Advanced Research Project Agency) et l'ONR (the Office Naval Research) en 1994. Aujourd'hui, avec l'émergence des standards tels que Bluetooth, HiperLan et IEEE 802.11, les projets de recherches civiles dans ce domaine ont afflué de partout dans le monde [12].

### 1.4.2 Définition des réseaux mobiles Ad Hoc

Un réseau mobile ad hoc, appelé généralement Mobile Ad hoc NETWORK (MANET), est un ensemble de nœuds mobiles qui se déplacent dans un territoire quelconque d'une manière

autonome et coopérative, sans l'utilisation d'une infrastructure préexistante ou d'une administration centralisée. Les ondes radio qui se propagent entre les différents noeuds mobiles sont le seul moyen de communication. Dès qu'un ensemble de noeuds mobiles se trouve à portée radio les uns des autres, alors le réseau se forme spontanément mais de manière provisoire [13].

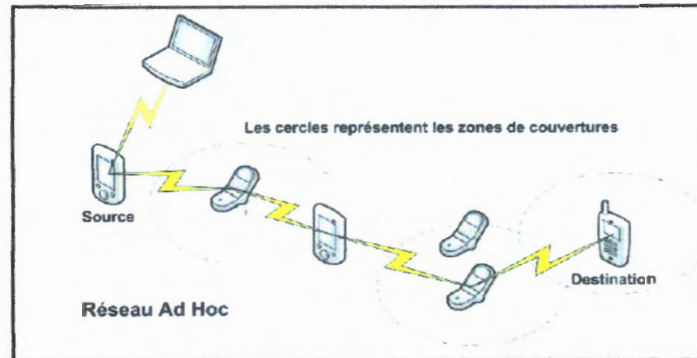


FIGURE 1.8 – Exemple d'un réseau ad hoc simple

### 1.4.3 Caractéristiques des réseaux ad hoc

Les MANETs ont plusieurs caractéristiques particulières [9] [14] [15] [16] :

#### ✓ Absence d'infrastructure

Les réseaux ad hoc se distinguent des autres réseaux mobiles par la propriété d'absence d'infrastructure préexistante et de tout genre d'administration centralisée. Les hôtes mobiles sont responsables d'établir et de maintenir la connectivité du réseau d'une manière continue.

#### ✓ Topologies dynamiques

Les noeuds sont libres de se déplacer arbitrairement, ce qui fait que la topologie du réseau typiquement multi-saut, peut changer aléatoirement et rapidement n'importe quand, et peut être constituée à la fois de liaisons unidirectionnelles et bidirectionnelles. Ainsi, un noeud peut quitter ou rejoindre le réseau à tout instant. Par conséquent, la topologie du réseau peut changer à des instants imprévisibles, d'une manière rapide et aléatoire. Cela affecte fortement la disponibilité des chemins de routage, et les protocoles de routage doivent s'adapter à la mobilité des noeuds.

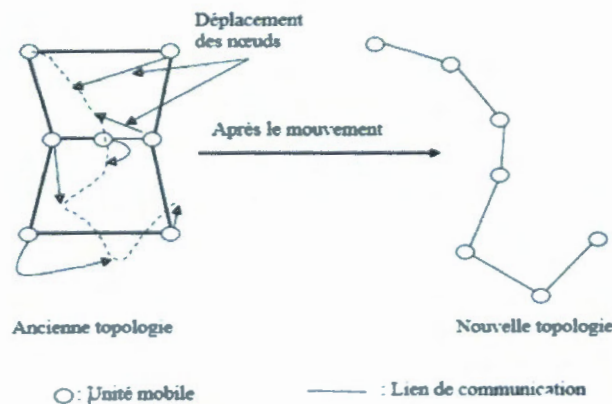


FIGURE 1.9 – Le changement de topologie des réseaux ad hoc [17]

### ✓ Liaisons à débits variables et à bande passante limitée

Les liaisons sans fil auront toujours une capacité inférieure à leurs homologues câblés. En plus, le débit réel des communications sans fil après avoir déduit les effets des accès multiples, du bruit, des interférences, etc. est souvent inférieur aux taux de transfert maximum de la radio.

Un des effets de ces débits de liaison relativement faibles est que la congestion sera généralement la norme plus que l'exception, i.e. la demande sur les applications distribuées approchera ou dépassera souvent la capacité du réseau. Comme le réseau mobile est souvent une simple extension d'un réseau fixe, les utilisateurs mobiles ad hoc demanderont les mêmes services. Cette demande ne cessera de croître avec l'augmentation des traitements multimédia et des applications basées sur les réseaux.

### ✓ Source d'énergie limitée

Une partie des noeuds d'un MANET, voire l'ensemble des noeuds, peut reposer sur des batteries ou un autre moyen limité pour puiser leur énergie. De plus, suivant la topologie du réseaux, certains mobiles peuvent se trouvés dans des positions clefs et sont appelés pour assurer le routage pour un grand nombre de flux (entre plusieurs sous-parties du réseau indépendantes par exemple). Ces noeuds peuvent être amenés à consommer très vite leurs ressources énergétiques. Pour ces noeuds, le plus important est sans doute de mettre en place des critères d'optimisation pour la conservation de l'énergie.

### ✓ L'hétérogénéité des noeuds

Les noeuds ad hoc peuvent correspondre à une multitude d'équipements, par exemple des ordinateurs portables, des PDA, téléphones mobiles, etc. Ainsi, ces noeuds peuvent avoir des différences en termes de capacité de traitement (CPU, mémoire), de mobilité



(lent, rapide) et de logiciel, mais ils doivent inter-opérer pour maintenir le réseau.

#### ✓ Équivalence des noeuds

Dans les réseaux ad hoc il n'existe pas de différence entre noeuds tels que les autres réseaux (hôte et station) car tous les noeuds peuvent être amenés et assurer les fonctions de routage.

#### ✓ Routage par relais

Dans un réseau ad hoc, un noeud ne peut communiquer qu'avec des noeuds qui se trouvent dans sa zone de propagation par exemple sur la figure 1.10, le noeud A ne peut communiquer qu'avec le noeud B et le noeud C ne peut communiquer qu'avec le noeud B à cause de leur limitation du rayon de propagation. Par contre B peut communiquer avec les deux, ce qui lui attribuera la fonction de routeur qui achemine le trafic entre les deux hôtes (A et C).

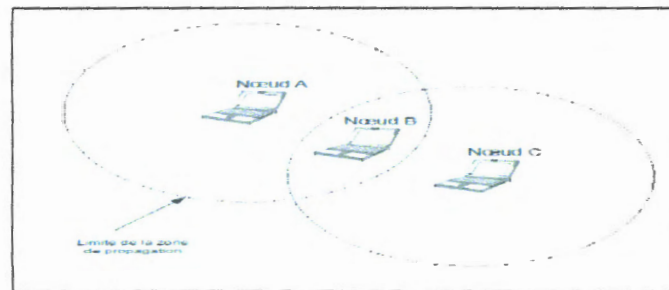


FIGURE 1.10 – Le routage par relais

#### ✓ Sécurité physique limitée

Les réseaux sans fil mobiles sont généralement plus sensibles aux menaces physiques que ne le sont les réseaux câblés fixes. Les possibilités accrues d'attaques par écoute passive, par usurpation d'identité et par déni de service doivent être étudiées avec attention. Les techniques existantes pour la sécurité des liaisons sont souvent appliquées au sein des réseaux sans fil pour réduire les risques d'attaques.

### 1.4.4 Le routage dans les réseaux Ad Hoc

Les protocoles de routage dans les réseaux Ad hoc sont fondés sur les principes fondamentaux du routage, qui sont : l'inondation, le vecteur de distance, le routage à la source et l'état de lien. Deux grandes catégories de protocoles se sont formées à partir de la normalisation de MANET : Les protocoles de routage proactifs qui établissent les routes à l'avance et les protocoles de routage réactifs qui cherchent les routes à la demande. D'autres classes sont à citer à savoir : les protocoles de routage hybrides (mélange de réactifs et proactifs), géographiques, hiérarchique, à qualité de service et multicast [18].

#### 1.4.4.1 Problématiques de routage dans les réseaux ad hoc

Dans le but d'assurer la connectivité du réseau, malgré l'absence d'infrastructure fixe et la mobilité des stations, chaque noeud est susceptible d'être mis à contribution pour participer au routage et pour retransmettre les paquets d'un noeud qui n'est pas en mesure d'atteindre sa destination; tout noeud joue ainsi le rôle de station et de routeur. Le fait que la taille d'un réseau ad hoc peut être énorme, souligne que la gestion de routage de l'environnement doit être complètement différente des approches utilisées dans le routage classique. Le problème qui se pose dans le contexte des réseaux ad hoc est l'adaptation de la méthode d'acheminement utilisée avec le grand nombre d'unités existant dans un environnement caractérisé par de modestes capacités de calcul et de sauvegarde. Dans la pratique, il est impossible qu'un hôte puisse garder les informations de routage concernant tous les autres noeuds, dans le cas où le réseau serait volumineux.

Le problème ne se pose pas dans le cas de réseaux de petites tailles, car l'inondation (la diffusion pure) faite dans ces réseaux n'est pas coûteuse; par contre dans un réseau volumineux, le manque de données de routage concernant les destinations peut impliquer une diffusion énorme dans le réseau, et cela si on considère seulement la phase de découverte de routes. Le trafic causé par la diffusion, dans ce cas, est rajouté au trafic déjà existant dans le réseau ce qui peut dégrader considérablement les performances de transmission du système caractérisé principalement par une faible bande passante.

Dans le cas où le noeud destination se trouve dans la portée de communication du noeud source, le routage devient évident et aucun protocole de routage n'est initié. Malheureusement, ce cas est généralement rare dans les réseaux ad hoc. Une station source peut avoir besoin de transférer des données à une autre station (noeud intermédiaire) qui ne se trouve pas dans sa portée de communication ce qui nécessite un protocole de routage approprié [19].

#### 1.4.4.2 Les protocoles de routage des MANETs

Il existe plusieurs critères pour la conception et la classification des protocoles de routage dans les réseaux ad-hoc : la manière dont les informations de routage sont échangées, quand et comment les routes sont calculées, etc. Ainsi, comme nous avons vu précédemment il est possible de distinguer deux grandes catégories de routage [9] [14] [18] :

**protocole de routage proactif** Dans cette catégorie dite à diffusion de table, les protocoles maintiennent à jour une table de routage dans chaque noeud. A chaque changement du réseau des messages de mise à jour sont communiqués aux noeuds. Les protocoles de routage proactifs essaient de maintenir les meilleurs chemins existants vers toutes les destinations possibles au niveau de chaque noeud du réseau. Pour le faire, ils utilisent l'échange régulier de messages de contrôle pour mettre à jour les tables de routage vers toute destination atteignable depuis celui-ci. Cette approche permet de disposer d'une route vers chaque

destination immédiatement au moment où un paquet doit être envoyé. Les tables de routage sont modifiées à chaque changement de la topologie du réseau.

**Le protocole de routage réactif** Ces protocoles se basent sur la découverte et le maintien des routes. Suite à un besoin, une procédure de découverte globale de routes est lancée. Ce processus s'arrête une fois la route trouvée ou toutes les possibilités sont examinées. Dès que la communication est établie, cette route est maintenue jusqu'à ce que la destination devienne inaccessible ou jusqu'à ce que la route ne soit plus désirée.

**Le protocole de routage hybride** Les protocoles hybrides combinent les deux idées : celle des protocoles proactifs et celle des protocoles réactifs. Ils utilisent un protocole proactif pour avoir des informations sur les voisins les plus proches (au maximum les voisins à deux sauts). Au-delà de cette zone prédéfinie, le protocole hybride fait appel aux techniques des protocoles réactifs pour chercher des routes. Ce type de protocoles s'adapte bien aux grands réseaux, cependant, il cumule aussi les inconvénients des protocoles réactifs et proactifs en même temps (messages de contrôle périodique, le coût d'ouverture d'une nouvelle route). Plusieurs protocoles hybrides existent dont le CBRP (Cluster Based Routing Protocol) et le ZRP (Zone Routing Protocol).

#### 1.4.5 Les applications des MANETs

La particularité du réseau ad hoc est qu'il n'a besoin d'aucune installation fixe, ceci lui permettant d'être rapide et facile à déployer. Les opérations tactiques comme les opérations de secours, militaires ou d'explorations trouvent en ad hoc, le réseau idéal. La technologie ad hoc intéresse également la recherche des applications civiles. On distingue entre autre [9] :

✓ **Le travail collaboratif et les communications dans des entreprises ou bâtiments**

Dans le cadre d'une réunion ou d'une conférence par exemple.

✓ **Domaine militaire**

Le matériel militaire contient maintenant couramment une sorte de matériel informatique. Le réseaux Ad hoc permet aux forces armées de tirer parti de la technologie réseau courante pour maintenir un réseau entre les soldats, les véhicules et le quartier général de l'information militaire. Les techniques de base du réseau ad hoc est venu de ce domaine.

✓ **Les services d'urgence**

Le réseau Ad hoc peut être utilisé dans les opérations de sauvetage d'urgence pour les opérations de secours en cas de catastrophe, par exemple. En feu, inondation ou tremblement de terre. Les opérations de secours d'urgence doivent avoir lieu lorsque

des communications inexistantes ou endommagées l'infrastructure et le déploiement rapide d'un réseau de communication sont nécessaires. L'information est relayée par un sauvetage membre de l'équipe à un autre sur un petit ordinateur de poche. [20].

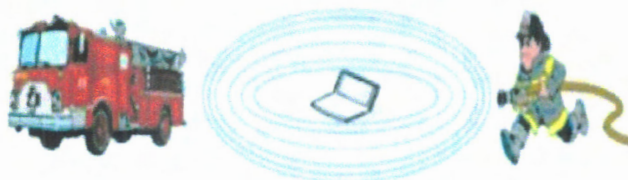


FIGURE 1.11 – Les opérations de secours

#### ✓ Applications commerciales

Pour un paiement électronique distant (taxi) ou pour l'accès mobile à l'Internet, ou service de guide en fonction de la position de l'utilisateur [9].

#### ✓ Réseaux domestique

MANET à courte portée peut simplifier l'intercommunication entre différents périphériques mobiles (tels qu'un PDA, un ordinateur portable et un téléphone cellulaire). Les câbles fastidieux sont remplacés par des connexions sans fil. Un tel réseau ad hoc peut également étendre l'accès à Internet ou à d'autres réseaux par des mécanismes, par exemple, LAN sans fil(WLAN), GPRS et UMTS [20].

En effet, les applications potentielles des réseaux ad hoc sont nombreuses. Par exemple, on peut penser qu'un groupe de personnes avec des ordinateurs portables dans une conférence qui souhaite échanger des fichiers peut rapidement mettre en place un réseau ad hoc sans avoir recours à une infrastructure supplémentaire. Les réseaux ad hoc sont idéals dans des zones où un tremblement de terre ou d'autres catastrophes naturelles ont détruit les infrastructures de communication.

D'une façon générale, les réseaux ad hoc sont utilisés dans toute application où le déploiement d'une infrastructure réseau filaire est trop contraignant, soit parce que difficile à mettre en place, soit parce que la durée d'installation du réseau ne justifie pas de câblage à demeure [9].

### 1.4.6 Les avantages des MANETs

Le mode de fonctionnement des réseaux ad hoc présente de nombreux avantages [21] [22] :

- ✓ **Déploiement simple et rapide** : Par contre aux réseaux cellulaires qui nécessitent un grand effort de planification pour leur déploiement, les réseaux ad hoc ne nécessite aucun pré requis puisqu'il suffit de disposer d'un certain nombre de terminaux dans un espace pour créer un réseau ad hoc, et rapide puisqu'il est immédiatement fonctionnel dès lors que les terminaux sont présents.
- ✓ **La mobilité** : L'absence de câblages autorise les noeuds à se déplacer librement l'un par rapport aux autres, au cours du temps. A condition de ne pas s'éloigner trop les uns des autres pour garder la connectivité du réseau.
- ✓ **Coût** : Il n'y a aucun coût d'installation des stations de bases ou de câblage. La simple présence des hôtes possédants des interfaces de communication radio forme le réseau.
- ✓ **Robustesse** : Les réseaux Ad Hoc ont en théorie une très grande robustesse. En effet si un des éléments du réseau devient indisponible, des nouvelles routes vont être créés puis empruntées pour acheminer les données comme si l'élément ne fonctionnant plus n'avait jamais existé.
- ✓ **Extensible** : Un réseau mobile ad hoc peut s'étendre facilement dans sa taille. En effet, le fonctionnement d'un nouveau noeud au sein du réseau nécessite quelques configurations.

#### 1.4.7 Les inconvénients des MANETs

Le mode de fonctionnement des réseaux ad hoc présente de nombreux inconvénients :

- ✓ **Capacités limitées (puissance de calcul, mémoire, énergie)** : Tant que les sources d'énergie sont limitées et les noeuds sont hétérogènes, un équilibre entre la connexité du réseau et la consommation énergétique est nécessaire. En effet, il faut que la portée d'émission des noeuds soit suffisante pour assurer la connexité du réseau. Mais plus on accroît la portée des mobiles, plus les communications demandent de l'énergie. En outre, l'utilisation des batteries en tant que source d'énergie par les unités mobiles, affecte la durée de vie du réseau ad hoc.
- ✓ **Taux d'erreur important** : Ce taux d'erreur dépend des collisions qui augmentent avec le nombre de noeuds partageant le médium radio.

- ✓ **Sécurité** : Les réseaux sans fil sont par nature plus sensibles aux problèmes de sécurité (diffusion totale).
  
- ✓ **Connectivité limité** La connectivité limite les possibilités de communication, ainsi deux station ne sont joignables s'il existe un ensemble de stations pouvant router afin de faire suivre les paquets de données échangés entre les deux stations.

## 1.5 Conclusion

Dans ce chapitre, nous avons présenté les concepts des environnements mobiles et en particulier les réseaux mobiles ad hoc qui sont considérés comme une collection des stations ou noeuds mobiles connectés via des liaison sans fil, qui forment un réseau de topologie arbitraire sans utiliser une infrastructure préexistante ou administration centralisée. Les réseaux ad hoc sont Auto-organisé et ont objet d'assurer les échanges des informations et les services fournis aux utilisateurs.

Bien que ces réseaux présentent des avantages énormes, malheureusement, les limitations imposées par l'environnement mobile ad hoc, nous obligent à changer la vision classique de résolution des problèmes liés aux systèmes distribués tel que celui de l'élection d'un leader qui devient plus compliqué à résoudre dans ces environnements dynamiques et instables.

# Le problème d'élection de K-meilleurs leaders dans les réseaux Ad hoc

## 2.1 Introduction

Ces dernières années, les systèmes répartis se développent rapidement. Par conséquent, gérer et contrôler ces systèmes devient un problème difficile. En fait, il existe beaucoup d'applications réparties qui exigent qu'un seul processus soit désigné pour coordonner une tâche effectuée par un ensemble des processus. Généralement, beaucoup de processus dans le système sont potentiellement capables d'offrir une telle coordination. Cependant, les processus doivent « s'entendre » pour décider lequel d'entre eux prend désormais le rôle de coordonnateur. Le processus choisi est appelé « leader ».

Considéré comme un cas particulier du problème de consensus, l'élection de leader constitue un composant fondamental dans les systèmes distribués statiques et dynamiques particulièrement ceux sujets à des défaillances. Dans le contexte des réseaux mobiles ad hoc où les défaillances de liens et des processus ainsi que les changements fréquents de topologie sont considérés comme la norme et non l'exception, la résolution du problème d'élection devient plus difficile.

Dans ce chapitre nous nous intéressons aux algorithmes d'élection de leader dans un environnement mobile ad hoc. Nous présentons d'abord le problème d'élection dans un système réparti conventionnel puis dans un réseau mobile ad hoc, ensuite quelques protocoles d'élection. L'accent dans ce chapitre sera mis par la suite sur les algorithmes d'élection de K-meilleurs leaders dans les réseaux mobiles ad hoc.

## 2.2 Élection de leader

### 2.2.1 Définition

Un leader dans un système réparti est un processus coordonnateur du système qui exécute certaines tâches pour le compte des autres processus du système. En plus, en cas

de défaillance du leader, un nouveau leader doit être élu et prend en charge le rôle de coordonnateur.

En informatique distribué, l'élection de leader est une primitive importante. Elle est utile pour de nombreuses applications qui nécessitent la sélection d'un processeur unique parmi plusieurs processeurs, en particulier lorsque les défaillances sont fréquentes. A titre d'exemple, si la défaillance d'un processus provoque la perte d'un jeton dans un algorithme d'exclusion mutuelle, les autres processus doivent élire un nouveau 'leader' qui sera chargé de régénérer le jeton. L'élection d'un leader est aussi utile dans les protocoles de communication de groupe afin de choisir un nouveau coordinateur lorsque la composition du groupe change.

Un algorithme d'élection de leader valide doit remplir les conditions suivantes [23] :

- **Terminaison**

L'algorithme devrait se terminer dans un temps fini une fois le leader est sélectionné. Dans les approches randomisées cette condition est parfois affaibli (par exemple, ce qui nécessite une terminaison avec probabilité).

- **Unicité**

Il y a exactement un processeur qui se considère comme leader.

- **Accord**

Tous les autres processeurs savent qui est le leader.

## 2.2.2 Élection de leader dans les réseaux ad hoc

L'élection de leader consiste à choisir finalement un leader unique d'un ensemble de noeuds. L'algorithme d'élection part d'une configuration dans laquelle tous les processus sont dans le même état (non élu) et termine par une configuration dans laquelle un seul processus est dans l'état élu et tous les autres dans l'état battu. En plus, dans les réseaux mobiles ad hoc caractérisés essentiellement par des changements fréquents de topologie induits par la mobilité des noeuds, de nouveaux problèmes se posent. Par exemple, la mobilité des noeuds peut causer le partitionnement du réseau ou bien la fusion de composantes du réseau. Par conséquence, il y aura une période où une ou plusieurs composantes n'auront pas de leader. Un algorithme d'élection doit garantir que chaque nouvelle composante (séparée de l'ancien leader) doit élire exactement un leader unique. Il peut aussi y avoir une période pendant laquelle il y aura co-existence de deux ou plusieurs leaders suite à la fusion de deux ou plusieurs composantes. Cependant, l'algorithme doit garantir que seulement un seul de ces leaders survivra. Il est à noter qu'il est impossible de garantir un leader tout le temps. Par exemple, lors de la fusion de deux composantes, il faut un peu de temps pour que le nouveau leader soit élu.

Ainsi, le problème d'élection est défini dans le contexte des réseaux mobiles ad hoc comme suit : "Chaque composante connexe du réseau mobile ad hoc finira par avoir exactement un



leader”.

## 2.3 Les algorithmes d'élection de leader

Comme nous l'avons vu précédemment, L'élection du leader est un problème fondamental en informatique distribuée qui a été largement étudié sur une grande variété de modèles et de topologies. En fait, les solutions proposées diffèrent par le mécanisme de communication utilisé (asynchrone ou synchrone), l'identité des processus (identités uniques ou anonymes), la topologie du réseau (anneau, graphe complet, topologie arbitraire, arbre, etc.), et même par la manière de résoudre le problème. Il a fait l'objet de recherches intensives depuis que son importance a été exprimée pour la première fois par Gerard Le Lann en 1977 [24]. L'importance pratique des élections dans l'informatique distribuée a été encore soulignée par l'algorithme Bully de Garcia Molina [25] en 1982. De plus, certains articles pertinents ont été fournis en considérant une topologie en anneau tel que ceux de Change-Roberts [26], Peterson [27], Franklin [28], etc. Il existe également des algorithmes pour la topologie arbitraire [29] et la topologie en arbre [30]. Cependant, ces solutions proposées pour les réseaux statiques (topologie fixe) ne peuvent pas être facilement implémentées dans les réseaux dynamiques comme les MANETs à cause de la mobilité des noeuds, des ressources limitées, d'une faible bande passante des canaux sans fil, etc. Par conséquent, plusieurs nouveaux algorithmes d'élection de leader dans les MANET ont été proposés pour s'adapter aux changements de topologie. Nous pouvons classer ces algorithmes en deux grandes classes : les algorithmes basés sur le protocole de routage TORA (Temporally-Ordered Routing Algorithm) et les algorithmes d'élection selon une priorité.

### 2.3.1 Algorithmes basés sur le protocole TORA

TORA [31] est un protocole de routage distribué qui utilise une technique de routage appelée 'Inversement de Liens' (Link Reversal). Il a été conçu principalement pour minimiser l'effet des changements de la topologie (mobilité) qui sont fréquents dans les réseaux Ad hoc. Pour le faire, plusieurs chemins vers une même destination sont stockés, ce qui fait que beaucoup de changements de topologie n'auront pas d'effets sur le routage des données, ceci en utilisant quatre fonctions de base : création de route, maintenance de route, suppression de routes et optimisation de route. La création de routes correspond à la création d'un graphe acyclique orienté DAG (Directed Acyclic Graph) vers la destination, et la maintenance signifie l'adaptation de la structure de routage suite à un changement dans la topologie du réseau.

Le protocole TORA est basé sur le principe des graphes acycliques orientés DAGs pour la création d'une route vers la destination. Chaque noeud maintient une valeur, appelée sa taille qu'il l'échange avec l'ensemble de ses voisins directs. Cette notion de taille est utilisée

pour l'orientation des liens du réseau. Un lien est toujours orienté du noeud qui a la plus grande taille, vers le noeud qui a la plus petite taille. Un graphe acyclique orienté est orienté destination s'il y a toujours un chemin possible vers une destination spécifiée. Le graphe devient non orienté destination, si un lien (ou plus) devient défaillant. Dans ce cas, TORA utilise le concept d'inversement de liens. Ce concept assure la transformation du graphe précédent, en un graphe orienté destination durant un temps fini.

Plusieurs algorithmes d'élection sont inspirés du protocole de routage TORA en lui apportant les modifications de base suivantes :

- Au lieu d'avoir un seul DAG orienté destination, chaque composante du réseau doit former un DAG orienté leader.
- Lors de la détection d'une partition (en utilisant le mécanisme de TORA), un nouveau leader est élu et son identificateur est propagé à travers toute la nouvelle composante.
- Suite à la fusion de deux composantes, une compétition aura lieu entre les deux leaders de telle sorte que l'identificateur du gagnant soit propagé et écrase l'identificateur du perdant (seulement un seul de ces deux leaders survivra).
- Quand plusieurs changements de topologie se produisent, des complications supplémentaires surgissent. C'est dû au fait que pendant la propagation de l'identificateur d'un nouveau leader, des changements de topologie pourraient se produire dans la composante et le processus d'élection peut être répété.

Deux algorithmes d'élection basés sur TORA ont été proposés par Malpani, Welch et Vaidya (MWV) [32] dont le premier supporte un seul changement topologique à la fois, alors que le deuxième algorithme marche même s'il y a plusieurs changements topologiques simultanés. Ces algorithmes assurent que chaque composante soit un graphe acyclique orienté leader, c-à-d, que chaque composante connexe aura exactement un seul leader. D'autres travaux ont analysé et amélioré les idées de MWV. Derhab et Badache ont proposé un algorithme auto-stabilisant pour les réseaux dynamique asynchrone [33] basé sur la création de DAG. L'algorithme de Derhab utilise le mécanisme de protocole TORA pour la détection automatique des partitions. Il a le même principe de l'algorithme de Malpani, mais avec la modification du critère utilisé pour choisir le niveau de référence afin de diminuer le nombre des messages échangés et améliorer le temps de stabilisation. En outre, l'algorithme d'Ingram, Shields, Walter, et Welch [34] qui fonctionne dans un système asynchrone avec des changements de topologie arbitraires, est basé sur l'algorithme d'élection de leader de Malpani. Une nouvelle caractéristique de cet algorithme est l'ajout d'une septième valeur à la taille, un horodatage associé à l'identifiant de leader qui enregistre le temps où le leader a été élu. Il ajoute également une nouvelle règle par laquelle les noeuds peuvent choisir de nouveaux leaders.

## 2.3.2 Les algorithmes d'élection selon la priorité ( Extrema-finding Algorithms)

Bien que, les protocoles basés sur TORA gèrent bien les cas de partitionnements et des fusions de réseaux, ils font une élection aléatoire et ignorent les contraintes sur les ressources telles que l'énergie résiduelle, la puissance de calcul ou l'espace de stockage, alors que dans plusieurs situations, il pourrait être souhaitable d'élire un leader selon une priorité qui représente une caractéristique inhérente au système. Par conséquent, le noeud élu doit être le noeud qui a la plus grande priorité parmi tous les noeuds dans sa composante. Ainsi, Ce que doit garantir l'algorithme d'élection dans ce cas est : "Après un nombre fini de changements topologiques, toute composante connexe finira par avoir choisi un leader unique, qui est le noeud le plus prioritaire dans sa composante ". Nous allons dans les sous sections suivantes présenter quelques algorithmes d'élection de leader selon une priorité.

### 2.3.2.1 L'algorithme LEAA( Leader Election Algorithm for Ad hoc Networks)

Sudarshan Vasudevan, Jim Kurose et Don Towsley ont proposé un algorithme d'élection de leader qui appartient à la classe 'extrema-finding' et qui consiste à essayer de trouver un extrema en utilisant des calculs diffusants. Leur algorithme nommé LEAA [35] fait l'élection selon une caractéristique du noeud comme le niveau de l'énergie résiduelle ou bien la capacité de calcul, etc.

L'idée générale de l'algorithme est de 'grandir' et 'rétrécir' un arbre couvrant enraciné par le noeud qui commence l'élection (la source). Le noeud source après le rétrécissement de l'arbre couvrant obtient l'information adéquate pour déterminer le noeud ayant la plus grande valeur, et diffuse son identifiant aux autre noeuds du réseau.

#### a) Objectifs, contraintes et hypothèses

Le réseau ad hoc est modélisé par un graphe non orienté et dynamique à cause de la mobilité des noeuds. Les sommets du graphe correspondent à des noeuds mobiles et les arêtes à des liens de communication entre les noeuds. Une arête entre deux sommets du graphe signifie que les deux noeuds sont dans la portée de transmission l'un de l'autre et peuvent donc directement communiquer entre eux. Le graphe peut comporter plusieurs composantes à cause des déplacements des noeuds. Les hypothèses suivantes sont faites sur les noeuds et l'architecture du système :

- **Valeur du noeud** : Chaque noeud a une valeur qui lui est associée. La valeur d'un noeud peut être tout attribut lié à la performance tel que l'énergie restante de la batterie, nombre de voisins directs, les capacités de calculs, etc.

- **Identifiants de noeuds uniques et ordonnés** : Chaque noeud a un identifiant unique. Les identifiants sont utilisés pour identifier les participants pendant le processus d'élection.
- **Liens** : Les liens sont bidirectionnels et FIFO, c'est-à-dire que les liens ne réordonnent pas les paquets.
- **Comportement des noeuds** : Les noeuds peuvent tomber en panne à n'importe quel moment et revenir dans le réseau n'importe quand. Les pannes des noeuds sont implicitement modélisées dans l'algorithme d'élection par des déconnexions des noeuds du réseau. En outre, l'ajout d'un nouveau noeud est implicitement modélisé par de nouvelles formations de liens avec ses voisins.
- **Communications de noeud à noeud** : Une livraison de message est garantie seulement quand l'émetteur et le récepteur restent reliés durant le transfert de message.
- **Taille de la mémoire tampon** : Chaque noeud dispose d'une mémoire tampon de réception suffisamment grande pour éviter débordement de la mémoire tampon.

L'objectif de cet algorithme d'élection est d'assurer qu'après un nombre fini de changements de topologie, éventuellement chaque noeud  $i$  possède un leader ayant doté de la valeur la plus grande parmi tous les noeuds connectés.

### b) Variables et types de message

L'algorithme utilise 5 types de message :

Message	Objectif
Élection	Agrandir l'arbre couvrant
Ack	Accuser de réception d'un message d'élection qui contient la valeur de leader .
Leader	Annoncer le nouveau leader
Probe	Déterminer si le noeud est toujours connecté
Replay	Envoie d'une réponse à un message Probe

TABLE 2.1 – Les types de messages de l'algorithme LEAA

Chaque noeud maintient les structures de données suivantes :

Variable	Explication
$\delta_i$	Une variable booléenne indiquant si le noeud $i$ est dans une élection ou non
$\Delta_i$	Un booléen qui indique si un noeud $i$ a envoyé l'Ack à son père.
$p_i$	Le noeud parent de $i$ dans l'arbre couvrant
$lid_i$	Le leader du noeud $i$
$N_i$	Une liste qui contient les noeuds voisins de $i$
$S_i$	L'ensemble des noeuds que $i$ attend leur Ack.
$src_i$	L'index d'élection
$max_i$	Une variable utilisée pour contenir l'identité du maximum noeud de $i$ dans la construction de l'arbre couvrant, il est inclus dans le message Ack envoyé à son parent $p_i$ .
$flag_i$	Une variable booléenne qui égale à 1 si le noeud $i$ possède un leader, $flag_i = 0$ sinon.
$Num_i$	Une variable entière qui définit sur le champ num le plus grand de l'indice de calcul
$L_i$	La liste des noeuds auquel le noeud $i$ a envoyé le message Ack.

TABLE 2.2 – Les variables maintenues par un noeud  $i$  pendant le processus d'élection

### c) L'algorithme

---

```

1 if  $(\delta_i = 0 \wedge d_{i,lid_i} = \infty) \vee (\exists j : j \in S_i : d_{i,j} = \infty \wedge \Delta_i = 1) \vee (d_{i,p_i} = \infty \wedge \Delta_i = 1) \vee (\Delta_i = 0 \wedge \delta_i = 1 \wedge d_{i,src_i.id} = \infty) \vee (rcv_{i,j}(L) \wedge \delta_i = 1 \wedge src_i = L.src \wedge d_{i,L.lid} = \infty)$  then
2    $src_i, E.src := \langle Num_i, i \rangle, \langle Num_i, i \rangle$  ;
3    $L_i, S_i, \Delta_i, \delta_i, p_i, flag_i := N_i, N_i, 1, 1, i, 0$  ;
4    $(\forall j : j \in N_i) Send_{i,j}(E)$  ;
5    $Num_i := Num_i + 1$  ; Remove(L) if present ;
6 if  $rcv_{i,j}(NL) \wedge j \in N_i \wedge j \notin L_i \wedge src_i \neq NL.src \wedge (\delta_i = 1 \vee NL.\delta = 1)$  then
7    $Num_i := \max(src_i.num, NL.src.num) + 1$  ;
8   all commandes as in previous action ;
9   Remove(NL) ;
10 if  $rcv_{i,j}(E) \wedge (\delta_i = 0 \vee (\delta_i = 1 \wedge E.src > src_i))$  then
11    $src_i := E.src$  ;
12    $L_i, S_i, \Delta_i, \delta_i, p_i, flag_i := N_i \setminus \{j\}, N_i \setminus \{j\}, 1, 1, j, 0$  ;
13    $(\forall k \in N_i \setminus \{j\}) Send_{i,k}(E)$  ;
14   if  $Num_i \leq E.src.num$  then
15      $Num_i := E.src.num + 1$  ;
16   Remove(E) ;
17 if  $rcv_{i,j}(E) \wedge \Delta_i = 1 \wedge src_i = E.src \wedge d_{i,j} \neq \infty$  then
18    $A.src, A.flag := src_i, 0$  ;  $send_{i,j}(A)$  ; Remove(E) ;
19 if  $rcv_{i,j}(A) \wedge \Delta_i = 1 \wedge src_i = A.src \wedge d_{i,j} \neq \infty$  then
20    $S_i := S_i \setminus \{j\}$  ;
21   if  $(A.flag = 1 \text{ and } A.id > max_i)$  then
22      $max_i := A.id$  ;
23 if  $S_i = \{\}$   $\wedge src_i.id \neq i \wedge \Delta_i = 1 \wedge d_{i,p_i} \neq \infty$  then
24    $A.src, A.flag, A.id, \Delta_i := src_i, 1, max_i, 0$  ;
25    $send_{i,p_i}(A)$  ; Remove(E and A msgs w src <  $src_i$ ) ;
26 if  $S_i = \{\}$   $\wedge src_i.id = i \wedge \Delta_i = 1$  then
27    $lid_i := max_i$  ;
28    $L.src ; L.lid, \delta_i := src_i, lid_i, 0, 0$  ;
29    $(\forall k \in N_i \setminus \{j\}) send_{i,k}(L)$  ;
30   Remove(E and A msgs w src <  $src_i$ ) ;
31 if  $rcv_{i,j}(L) \wedge d_{i,L.lid} \neq \infty ((\delta_i = 1 \wedge \Delta_i = 0 \wedge L.src = src_i) \vee (\delta_i = 0 \wedge lid_i > L.lid))$  then
32    $lid_i, \delta_i, flag_i := L.lid, 0, 1$  ;  $(\forall k \in N_i \setminus \{j\}) send_{i,k}(L)$  ; Remove(L) ;
33 if  $NewLinkFormed(i, j) \wedge j \in N_i$  then
34    $NL.src, NL.\delta, NL.flag, NL.lid := src_i, \delta_i, flag_i, lid_i$  ;
35    $send_{i,j}(NL)$  ;  $NewLinkFormed(i, j) := false$  ;
36 if  $rcv_{i,j}(NL) \wedge NL.flag = 1 \wedge NL.\delta = 0 \wedge ((\delta_i = 1 \wedge src_i = NL.src) \vee (\delta_i = 0 \wedge d_{i,lid_i} \neq \infty \wedge d_{i,NL.lid} \neq \infty \wedge lid_i < NL.lid))$  then
37    $lid_i, \delta_i, flag_i := NL.lid, 0, 1$  ;
38    $L.src, L.lid := NL.src, lid_i$  ;
39    $(\forall k \in N_i \setminus \{j\}) send_{i,k}(L)$  ; Remove(NL) ;

```

---

- **Initialisation du processus d'élection**

Chaque noeud qui détecte le départ de son leader commence une élection par l'agrandissement d'un arbre couvrant en envoyant des messages d'élection à ses voisins. Chaque noeud  $i$  qui se rend compte de l'existence d'une fraîche élection met sa variable  $\delta[i]$  à 1. Il n'annonce un leader qu'après la réception des Ack de tous les noeud aux quels il a envoyé un message d'élection. La liste  $S[i]$  est donc initialisée à  $N[i]$ .

- **La construction de l'arbre couvrant**

Lors de la réception d'un message d'élection d'un noeud source  $i$ , le noeud  $j$  joint l'arbre couvrant en plaçant sa variable  $P[j]=i$ , et envoie un message d'élection à ses voisins  $N[j]$ . Ce dernier est propagé vers tous les noeuds et finalement un arbre couvrant est construit.

Quand un noeud reçoit un message d'élection d'un voisin autre que son père, il répond par un Ack après la réception des Ack de toutes ses fils.

Une fois que l'arbre couvrant grandis, il va se rétrécir en arrière vers la source. Le rétrécissement commence à partir des feuilles qui ne sont pas des pères à d'autres noeuds. Ces feuilles envoient, finalement, leurs Ack différés à leurs pères respectifs, qui envoient à leur tour leurs Ack différés à leurs propres pères, et ainsi de suite, jusqu'à ce que le noeud source reçoive tous ses Ack différés. L'Ack envoyé au père contient l'identificateur du noeud qui a la plus grande valeur parmi les noeuds descendants. Finalement, le noeud source aura l'information pour déterminer le noeud qui a la plus grande valeur dans le réseau.

- **Gestion multiple, calculs simultanés**

Le processus d'élection est déclenché au niveau du noeud  $i$  quand il détecte le départ de son leader. Il est évident que plus d'un noeud puisse détecter simultanément le départ du leader et que chacun d'eux puisse initier le calcul diffusant conduisant indépendamment à des calculs diffusant simultanés.

Les calculs diffusants simultanés sont traités en exigeant que chaque noeud participe à un seul calcul diffusant à la fois. Afin de réaliser ceci, chaque calcul diffusant est identifié par un index. Cet index contient l'identificateur du noeud et un numéro qui représente sa désidérabilité d'être un leader.

$(num1, id1) > (num2, id2) \Leftrightarrow ((num1 > num2) \text{ ou } ((num1 = num2) \text{ et } (id1 > id2)))$ .

### 2.3.2.2 MELFA : Message Efficient Leader Finding Algorithm for Mobile ad-hoc networks

Awadhesh Kumar Singh et Shantanu Sharma ont proposé l'algorithme MELFA [36] qui utilise le multicast et l'unicast pour l'envoi des messages afin d'élire un seul leader pour le réseau ad hoc entier.

#### a) Modèle du système

Les principales hypothèses de cet algorithme sont les suivantes :

- **Identité du Noeud(ID)** : Chaque noeud a un identifiant unique. Les identifiants sont utilisés pour identifier les participants pendant le processus d'élection.
- **Lien de communication** : Les liens de communications sont bidirectionnel, fiable non nécessairement FIFO.
- **Informations sur les voisins** Chaque noeud a des informations sur ses voisins seulement. Aucun noeud n'a des informations sur le réseau entier.
- **Communication des noeuds** : les noeuds communiquent en passant les messages sur le lien sans fil. Les noeuds voisins peuvent communiquer directement les uns avec les autres.
- **Contrainte de la mobilité des noeuds** : Une forte mobilité des noeuds peut causer des changements de topologie arbitraires, y compris des partitionnements et des fusions.
- **Livraison de message** : une livraison de message est garantie uniquement lorsque l'émetteur et le récepteur restent connectés pendant toute la durée du transfert de message.
- **Transmission des message d'élection (EM) et message de coordonnateur(CM)** : les deux messages sont considérés comme prioritaires sur les messages d'application.
- **Poids d'un noeud (NW)** : Chaque noeud a une priorité pour devenir coordonnateur. La priorité d'un noeud peut être basé sur son id, son niveau de batterie, sa puissance de calcul, etc.
- **Un seul MANET** : MELFA suppose qu'il n'y a qu'un seul MANET (réseau non partitionné).
- **Tampon de noeud** : Chaque noeud a un tampon suffisamment grand qui sert à stocker temporairement les messages de l'application.



## b) Message et structure de données

### • Types de messages

Cet algorithme utilise trois types de messages :

- **Election(EM)** pour le processus d'élection.
- **Accusé de réception(AM)** Utilisé en réponse à EM qui comprend NW de l'expéditeur et toujours destiné vers l'initiateur.
- **Message de coordonnateur(CM)** Pour l'annonce de coordonnateur(leader).

### • Structures de données

Il existe quatre catégories de structures de données.

#### i. Au noeud ori

Un noeud appelé noeud ori, déclenche le protocole d'élection à cause de la perte de contact avec son coordonnateur ou la panne de ce dernier.

- ✓ **receive\_ack [n]** : Un tableau de tuples représentant les messages AM reçus. Quand ori reçoit AM de j, il met  $\text{receive\_ack}[j]=j$ . Initialement  $\text{receive\_ack}[j] = -1$ .
- ✓ **Receive\_weight[n]** : Un tableau de tuples représentant NW des noeuds, il est modifié par le noeud ori lorsque il reçoit un AM de j, comme suite :  $\text{receive\_weight}[j] = NW_j$ . Initialement,  $\text{receive\_weight}[j] = -1$ .
- ✓ **Time Out Value(*te*)** l'intervalle de temps pendant lequel ori attend des AM de tous les noeuds possibles.

#### ii. À tous les noeuds

Tous les noeuds y compris ori maintiennent les structures de données suivantes :

- ✓ **ID** identité du noeud.
- ✓ **Neighbor\_name<sub>i</sub>[j]** un tableau de tuples représentant les voisins du noeud i.
- ✓ **NW<sub>i</sub>** représente la priorité du noeud i pour devenir coordonnateur.
- ✓ **CODR<sub>i</sub>** représente le coordonnateur actuel du noeud i. Initialement,  $CODR_i = \emptyset$ .
- ✓ **CODRelect<sub>i</sub>** une variable booléenne et égale TRUE si le node i a élu son coordonnateur. Initialement,  $CODRelect_i = FALSE$ .
- ✓ **STATE<sub>i</sub>** représente l'état actuel du noeud i. Les noeuds mobiles peuvent se trouver dans l'un des trois états : NORMAL, CANDIDAT, Coordonateur, Initialement  $STATE_i = NORMAL$ .

- ✓ **election\_send<sub>i</sub>** : une variable booléenne égale à TRUE si le noeud *i* transmet EM à ses voisins. Initialement,  $election\_send_i = FALSE$ .
- ✓ **send\_for<sub>i</sub>** : une variable qui contient l'ID de l'expéditeur de l'EM reçu par le noeud *i*. Initialement,  $send\_for_i = \emptyset$ .
- ✓ **Coordinator\_send<sub>i</sub>** : une variable booléenne qui vaut TRUE si le noeud *i* transmet CM à ses voisins. Initialement,  $Coordinator\_send_i = FALSE$ .

### iii. Transporté dans EM

EM collabore avec les structures de données suivantes qui sont utilisées pour transmettre les EM.

- ✓ **Receive\_election[n]** : un tableau de tuples représentant les ID des noeuds qui ont déjà reçu EM.
- ✓ **Ori** représente l'identité de l'initiateur actuel du protocole.
- ✓ **Sender\_election** : représente l'ID du noeud qui a transmis l'EM le plus récent.

### iv. Retourné avec CM

Comme EM, CM collabore également avec les structures de données suivantes qui sont utilisées pour transférer CM.

- ✓ **Receive\_coordinator[n]** : un tableau de tuples représentant les identifiants de noeuds qui ont déjà reçu CM. Lorsque le noeud *i* envoie CM à tous ses voisins *k*, il met à jour  $receive\_coordinator[k] = 1$ . Initialement,  $receive\_coordinator[j] = -1$ .
- ✓ **CID<sub>new</sub>** représente l'ID du nouveau coordinateur élu.

## c) Principe de MELFA

### Phase 1 - *Initiation*

Le coordinateur envoie périodiquement un message heart-beat à tous les autres noeuds. Un noeud NORMAL déclenche une élection s'il ne reçoit pas le message heart-beat ou s'il se déplace à un emplacement qui est au-delà de la portée du coordinateur et il devient un candidat et, par conséquent, appelé *ori* (CANDIDAT). Maintenant, le noeud *ori* diffuse l'EM et démarre le temporisateur *te*. Avant de diffuser le message EM, le noeud *ori* met à jour  $receive\_election[]$ ,  $sender\_election$ , et définit  $election\_send$  comme TRUE.

---

```

1 if  $STATE_i = NORMAL$  and  $CODR_i = \emptyset$  then
2    $ori := i$  ;
3    $STATE_i := CANDIDACY$  ;
4    $sender\_election := ori$  ;
5    $election\_send_{ori} := TRUE$  ;
6    $coordinator\_send_{ori} := FALSE$  ;
7    $codr\_elect_{ori}[i] := FALSE$  ;
8    $\forall j : j \in neighbor\_name_{ori}$  ;
9    $receive\_election[j] := 1$  ;
10  Set timer  $t_e$  ;
11   $\forall j : j \in neighbor\_name_{ori}$  ;
12  Broadcast EM to  $j$  ;

```

---

### Phase 2 - réception de message EM simultanée

Dans cette phase, le noeud  $j$  recevrait le message EM à partir d' $ori$  ou d'un autre noeud  $i$ . En recevant EM, le noeud  $j$  passe à l'état CANDIDAT. Par la suite, le noeud  $j$  prend l'une des actions suivantes selon la situation :

- **Cas 1**

Si le noeud  $j$  a déjà renvoyé un EM (c'est-à-dire  $election\_send_j = TRUE$ ), il supprime tous les EM suivants reçus des noeuds  $ori$  ayant un ID supérieur à l'identifiant d'origine de l'EM remis le plus récemment en le comparant à  $send - for_j$  ; Sinon, le noeud  $j$  diffuse EM à tous ses voisins et envoie AM.

- **Cas 2**

Si  $sender\_election_j = FALSE$  et le noeud  $j$  reçoit plus d'un EM simultanément, il transmet l'EM qui a le plus petit  $ori$  aux noeuds qui n'ont pas encore reçu EM. En outre, il rejette les autres EM.

---

```

1 Le noeud j reçoit un message EM de i et x ;
2 STATEj := CANDIDACY ;
3 coordinator_sendj := FALSE ;
4 codr_electori := FALSE ;
5 if (election_sendj == TRUE and EM.ori[i] ≥ send_forj and EM.orii >
   EM.orix) then
6   Discard EM.orix /* EM.orix represents ori of the EM*/ ;
7   ∀ k : k ∈ neighbor_namej ;
8   Broadcast EM.orii to all k ;
9 else if (election_sendj == FALSE and EM.orii > EM.orix) then
10  Discard EM.orix ;
11  if (∀ k : k ∈ neighbor_namej, receive_election[k] == 1) then
12    Send AM ;
13  else if (∃ k : k ∈ neighbor_namej, receive_election[k] == -1) then
14    Send AM ;
15    election_sendj := TRUE ;
16    send_forj := ori ;
17    receive_election[k] := 1 ;
18    sender_election := j ;
19    ∃ k : k ∈ neighbor_namej, receive_election[k] == -1 ;
20    Forward EM to k ;

```

---

#### Procédure - Élection de Coordinateur à ori

Cette procédure se déroule à ori avec le début de la phase 1. Elle remet en sortie le coordinateur après expiration du temporisateur. Si temporisateur te a expiré et pour tout j, receive\_election [j] est toujours -1, alors la phase 1 est à nouveau déclenchée. Maintenant, le node ori déclare le noeud ayant le NW le plus élevé en tant que coordinateur et passe à l'état NORMAL ou COORDONNATEUR, selon le cas, et diffuse le message CM et met à jour les structures de données pertinentes.

---

```

1 if (receive_ack[j] == -1 and te == TRUE) then
2   reinitiate Phase 1(4 times);
3 else
4   ori receive AM from j ;
5   receive_weight[] = receive_weight[] ∪ NW[j] ;
6   receive_ack[] = receive_ack[j] ∪ j ;
7   if (te == TRUE) then
8     CID := max_weight_id(receive_weight[]) ;
9     CODRori := CID ;
10    CODRelectori := TRUE ;
11    if (CID != oriid) then
12      STATEori := NORMAL ;
13    else
14      STATEori := COORDINATOR ;
15    ∃ k : k ∈ neighbor_nameori ;
16    receive_coordinator[k] := 1 ;
17    coordinator_sendori := TRUE ;
18    election_sendori := FALSE ;
19    ∃ k : k ∈ neighbor_nameori ;
20    Broadcast CM to k ;

```

---

### Phase 3 : - Distribution de CM

C'est la dernière phase de MELFA. Si le noeud j n'a pas encore réussi à élire le coordinateur et reçoit CM, il accepte le noeud nouvellement élu en tant que leader et son état devient *NORMAL* ou *COORDONNATEUR* selon le cas.

---

```

1 Noeud j reçoit CM de i ;
2 if (CODRelectj == FALSE) then
3   CODRj := CIDnew ;
4   CODRelectj := TRUE ;
5   election_sendj := FALSE ;
6   if (CIDnew != j) then
7     STATEj := NORMAL ;
8   else
9     STATEj := COORDINATOR ;
10  if (∃ k : k ∈ neighbor_namej, receive_coordinator[k] == 1) then
11    coordinator_sendj := TRUE ;
12  else if (∃ k : k ∈ neighbor_namej, receive_coordinator[k] == -1) then
13    coordinator_sendj := TRUE ;
14    receive_coordinator[k] := 1 ;
15    ∃ k : k ∈ neighbor_namej, receive_coordinator[k] == -1 ;
16    Forward CM to k ;

```

---

## 2.4 Algorithmes d'élection de k-meilleurs leaders

Le problème d'élection de leader a été largement étudié dans les systèmes distribués, est présenté dans le contexte des réseaux ad hoc mobiles (MANET). Plusieurs algorithmes d'élection de leader ont donc été proposés dans la littérature. Bien que la plupart des algorithmes se concentrent sur la réduction du nombre de messages de contrôle (messages dont la priorité est la plus haute à transmettre), il y avait peu d'attention à assurer la haute disponibilité d'un leader malgré divers types d'échecs, en particulier dans des scénarios critiques où l'absence du chef, même de courte durée, peut causer des ravages. En outre, dans les grands MANET, un grand nombre d'applications ne parviennent pas à exécuter en l'absence d'un leader. Élire plusieurs meilleurs leaders est une approche intéressante afin de minimiser l'échange de messages et préserver l'énergie d'un côté, et d'assurer la haute disponibilité du leader de l'autre côté. Contrairement aux solutions précédentes, les algorithmes de cette approche proposent que chaque noeud maintienne une liste de candidats afin de minimiser le nombre total d'élections de leader, où le premier noeud est considéré comme le leader actif du réseau. Si le premier est absent pendant une période spécifiée, le second devient le leader actif, etc. Nous allons dans ce qui suit présenter les algorithmes d'élection de top K leaders dans les MANET. Ces algorithmes appartiennent à la classe Extrema-finding expliquée précédemment.

### 2.4.1 L'algorithme CBLEAA (Candidate Based Leader Election Algorithm)

Cet algorithme est une amélioration de l'algorithme LEAA. Muhammad Mizanur Rahman, M.Abdullah-Al-Wadud et OksamChae ont proposé l'algorithme CBLEAA [37] afin de minimiser le nombre de messages échangés et le nombre des processus d'élection. Dans cet algorithme, chaque noeud possède une liste leader de cinq noeuds (dans l'ordre décroissant), où le premier noeud est considéré en tant que leader actif. Si le premier est absent pendant un certain temps le second sera considéré comme leader actif et ainsi de suite.

#### a) Modèle du système

Les hypothèses de cet algorithme sont :

- Chaque noeud a un identifiant unique.
- le noeud ayant l'identifiant le plus élevé dans le réseau sera considéré comme leader.
- Les liens sont bidirectionnels et FIFO.
- La mobilité des noeuds peut changer la topologie arbitrairement.
- Chaque noeud a un grand tampon(buffer).



## b) Principe de l'algorithme CBLEAA

Dans cet algorithme le noeud  $i$  initialise un processus d'élection en envoyant un message Election à ses voisins, et dans ce cas les noeuds voisins sont considérés comme des fils du noeud  $i$ .

- A la réception des messages Election : les noeuds voisins envoient à leurs tours un message Election à leurs voisins et ainsi de suite jusqu'à la construction de l'arbre.
- Les noeuds feuilles ajoutent leurs identifiants dans la liste, et envoient ces derniers à leurs parents dans des messages Ack.
- Dans cet algorithme l'identifiant vide est dénotée par -1 dans  $L$  (la liste leader).

Contrairement à LEAA, dans la section suivante, nous allons voir comment cet algorithme gère le partitionnement du réseau et fusion des composantes.

(a) vérifier l'existence du leader, après un intervalle de 20 secondes (message heart-beat);

- Si le leader n'existe pas dans le réseau après une période de  $20 \times 6$  secondes alors :

(a) Choisir un noeud initiateur.

(b) Le noeud initiateur envoie un message Election à tous ses enfants. Ce processus est continué jusqu'à ce que tous les noeuds feuilles reçoivent ces messages.

(c) Pour chaque noeud feuille,  $l$ , dans le réseau :

I. Ajouter  $l$  à sa propre liste leader  $L$ .

II. Envoyer  $L$  à son père dans un message ACK.

III. Le noeud parent rajoute son identifiant à la liste  $L$  tout en conservant un ordre décroissant puis propage  $L$  à son parent.

IV. Répéter ce processus jusqu'à ce que le noeud initiateur obtienne toutes les listes leader de chaque branche.

V. Dans cette phase de construction de l'arbre chaque noeud envoie et reçoit des messages Probe et Reply périodiquement pour maintenir la connectivité du réseau.

(d) l'initiateur construit une liste qui contient les identifiants des cinq noeuds ayant les plus grandes valeurs (priorités) puis la propage dans le réseau à travers le message Leader.

- Si au moins un leader existe dans la position  $k$  au niveau de la liste leader alors :

(a) Le leader annonce un message heart-beat.

- (b) La liste leader est mise à jour en plaçant le k-ième noeud de la liste L (le chef actif) à la première position et en décalant les leaders inadmissibles de L à la fin de la liste.
- S'il existe plusieurs leaders actifs de différentes composantes au sein du même réseau alors :
  - (a) Combiner les différentes listes leader des composantes précédentes et construire une nouvelle liste qui contient les cinq noeuds ayant les plus grandes valeurs.

L'exemple de la figure suivante illustre l'exécution de l'algorithme de CBLEAA.

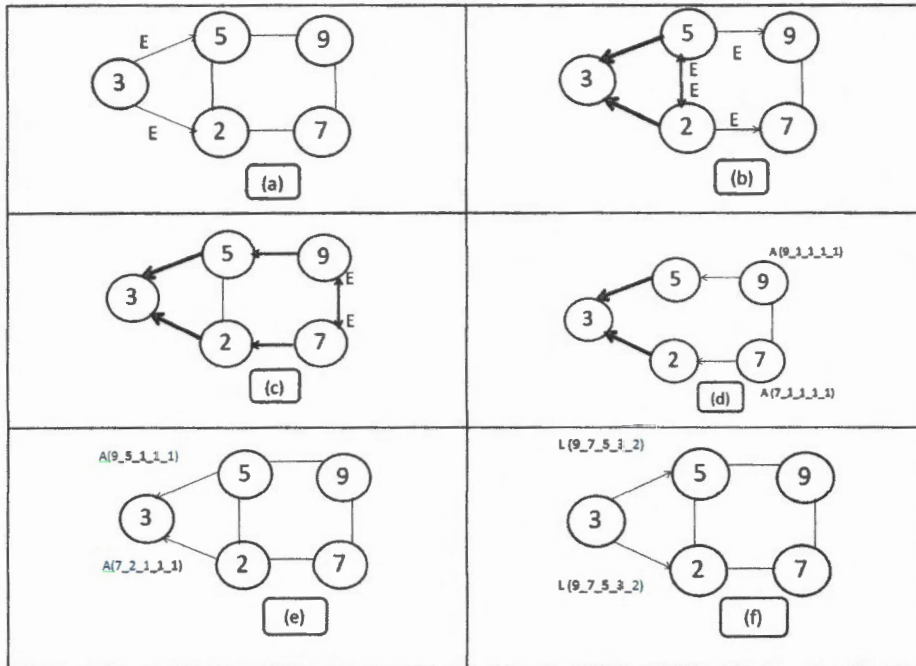


FIGURE 2.1 – Exemple d'exécution de l'algorithme CBLEAA.

Les figures 2.1 (a) à (c) montrent les phases de croissance de l'arbre couvrant. Chaque noeud maintient une liste de leaders. Dans cet exemple, la taille de la liste des leaders est égale à cinq. La figure 2.1 (d) montre que les noeuds feuilles 7 et 9 ajoutent leurs identifiants à la liste et les envoient à leurs parents dans les messages Ack. La figure 2.1 (e) montre que l'initiateur reçoit deux listes de ses deux branches qui sont A (7, 2, -1, -1, -1) et A (9, 5, -1, -1, -1). De ces deux listes, le noeud initiateur sélectionne la liste Leader L (9, 7, 5, 3, 2) et la diffuse vers tous les noeuds du réseau. Comme LEAA, ici tous les noeuds envoient des messages Probe périodiques et attendent la réponse Replay des voisins pour maintenir la connectivité.

c) **CBLEAA dans un environnement mobile**

— **Traitement des partitions réseau**

Dans la figure 2.2, nous voyons l'avantage d'avoir plusieurs candidats. Dans la



figure 2.2 (a), tous les noeuds du réseau maintiennent la même liste de leaders, où le leader actif est 50. Mais comme le noeud 3 disparaît, deux réseaux sont créés. Dans la figure 2.2 (b), sur le réseau à droite, les noeuds 50, 20, 17 existent. Donc, ce réseau ne modifie pas la liste des chefs. Mais sur le réseau de gauche, les identifiants des trois premiers candidats sont absents de la liste des leaders. Donc après un certain temps nécessaire pour faire les trois niveaux de vérification des noeuds principaux, le noeud 10 envoie le message heart-beat à tous les noeuds du réseau.

Alors contrairement à LEAA, une nouvelle élection de chef n'est pas nécessaire ici. Pour réduire le temps d'attente, tous les noeuds mettent à jour leurs listes de leaders en déplaçant les identifiants invalides à la fin de la liste de leader. Mais il ne faut pas supprimer ces nœuds de la liste pendant un certain temps, pour une future rejoinure.

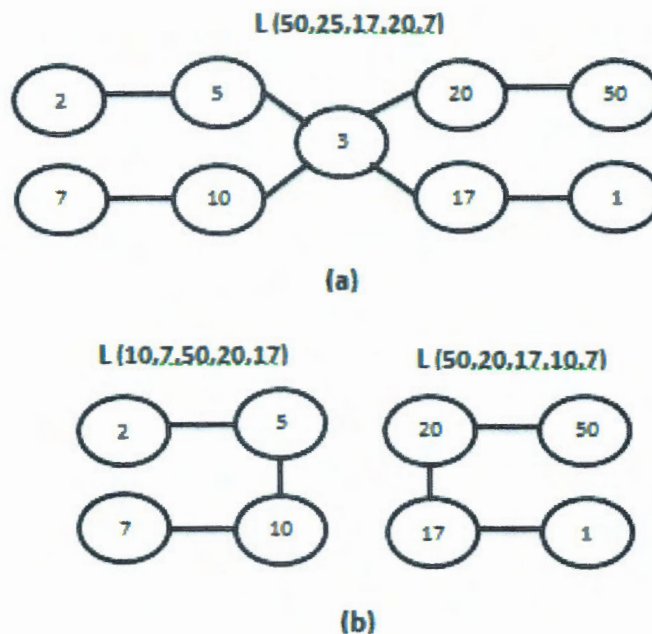


FIGURE 2.2 – Exemple en cas de partitionnement du réseau

### — Gestion des fusions de partitions

La fusion de réseaux peut être gérée efficacement par cette méthode. Dans la figure 2.3 (a), il y a trois réseaux qui maintiennent les listes de leaders  $L(10,7,5,2,1)$ ,  $L(25,-1,-1,-1,-1)$  et  $L(50,20,17,1,-1)$ . Selon cet algorithme, du réseau fusionné a finalement  $L(50,25,10,20,17)$ , où les noeuds 50,25 et 10 étaient les leaders actifs. Après la fusion, le noeud 50 devient le leader actif. Comme montre la figure 2.3.

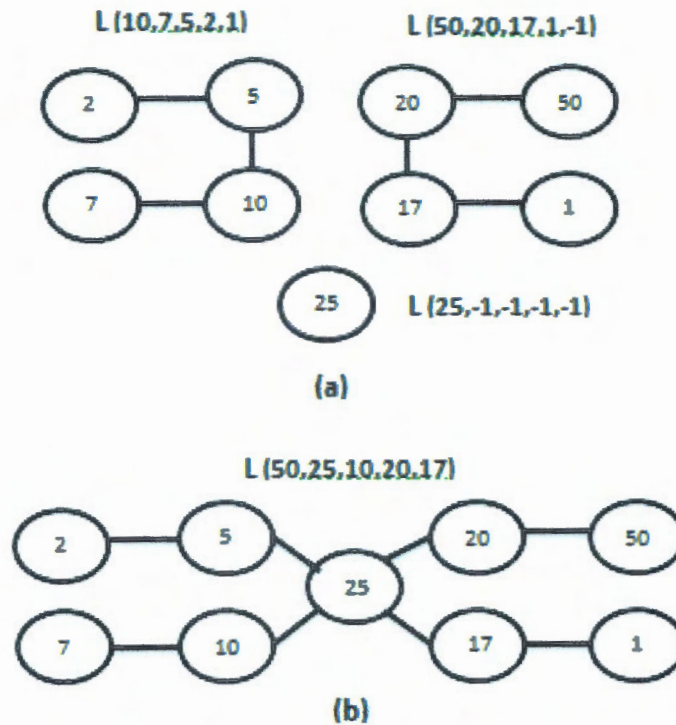


FIGURE 2.3 – Exemple de fusion des partitions

### 2.4.2 ELFA :Elite Leader Finding Algorithm for MANETs

Awadhesh Kumar Singh and Shantanu Sharma ont proposé un algorithme d'élection baptisé ELFA [38] basé sur l'algorithme MELFA. Cet algorithme déclare un nouveau coordinateur pour le système dès le départ du leader actuel en maintenant un vice-coordinateur et un ensemble dit cabinet de  $n$  nœuds d'élite, appelés CAGs (Coordinator Advisory Group members). Les nœuds ayant les meilleures capacités sont appelés nœuds "d'élite". En cas d'échec, le vice-coordinateur devient le leader du système et un nouveau vice-coordinateur sera élu parmi les nœuds d'élite. Ainsi, le protocole réduit le nombre de messages et le temps d'attente pour choisir le nouveau leader.

#### a) Types de messages et structures de données

- **Types de messages**

Cet algorithme utilise trois types de messages :

- **Message d'élection(EM)** : pour le processus électoral.
- **Message d'accusé de réception(AM)** : utilisé en réponse à EM qui comprend NW de l'expéditeur et toujours destiné à l'initiateur ori.

- **Message de coordonnateur(CM)** : pour l'annonce de Coordinateur, en MELFA. En outre,, ELFA utilise sept types de messages.

Message		Description
Information Message	IM	Envoyé par le CAG au coordinateur actuel
Acknowledge Information message	AIM	Envoyé par le coordinateur en réponse à IM afin de confirmer sa présence dans le système.
Notify Message	NM	Envoyé par le noeud i pour informer le nouveau n ?ud j joignant le MANET sur l'ID du coordinateur actuel.
Update Message	UM	Envoyé par le noeud i pour fournir au coordinateur actuel l'ID du nouveau noeud j joignant le MANET
Leader Message	LM	Transfère la responsabilité du leader crashé au vice-coordonnateur par n'importe quel noeud du CAG
CABinet message	CAB	Envoyé par le vice-coordonnateur pour lancer le MELFA au sein du CAG afin d'élire le prochain vice-coordonnateur.
Wait Message	WM	Envoyé par le noeud en attente i à un autre noeud en attente j pour éviter les flux EM inutiles si noeud j n'a pas l'id du leader actuel

TABLE 2.3 – Types de message pour ELFA

- **Structures de données :**

Nous présentons les structures de données maintenues à chaque n ?ud plus les structures de données dans EM et CM.

- i. **Au noeud ori**

Un noeud, appelé ori node, qui déclenche le protocole d'élection à cause de la perte de contact avec son coordinateur ou la panne de ce dernier. Ori possède les structures de données suivantes :

- $Receive\_ack[n]$  : Un tableau de tuples représentant les messages AM reçus. Quand ori reçoit AM de j, il est met  $receive\_ack[j]=j$ . Initialement  $receive\_ack[j] = -1$ .
- $Receive\_weight[n]$  : Un tableau de tuples représentant les poids (NW) transportés par les messages AM et mis à jour par ori. Quand il reçoit un message AM de j, ori met  $receseive\_weight[j] = NWj$ . Initialement,  $receive\_weight[j] = -1$ .
- $Time\ Out\ Value(te)$  : L'intervalle de temps pendent lequel ori attend des AM de tous les noeuds possibles.

- ii. **À tous les noeuds** Tous les noeuds du système maintiennent les structures de données suivantes :

- ✓ **ID** Identité du noeud.

- ✓ **Neighbor\_name<sub>i</sub> [j]** Voisins du noeud i.
  - ✓ **NW[j]** Priorité du noeud i pour devenir coordinateur.
  - ✓ **STATE<sub>i</sub>** Représente l'état actuel du noeud i. Les noeuds mobiles peuvent être dans l'un des cinq états : NORMAL, CANDIDACY, LEADER, VC (vice-coordinateur), ELITE. Initialement,  $STATE_i = \text{NORMAL}$ .
  - ✓ **CAG\_status<sub>i</sub>** Une variable booléenne qui vaut TRUE si le noeud i est CAG. Initialement,  $CAG\_STATE_i = \text{FALSE}$
  - ✓ **Wait\_set<sub>i</sub> [W]** Un tableau de tuples qui stocke les IDs des noeuds du noeud auquel le message d'attente a été envoyé. Initialement,  $\forall w, wait\_set_i[w] = \emptyset$ ;
- iii. **Au CAG** Chaque membre du groupe consultatif du Cabinet Conservez les informations sur tous les autres CAG en utilisant la structure de données suivante :
- ✓ **CAG\_set<sub>i</sub> [n]** un tableau de tuples représentant les IDs de tous les CAGs. Initialement,  $\forall n, CAG\_set_i[n] = \emptyset$ .
  - ✓ **VID<sub>i</sub>** représente l'identité du vice coordinateur actuel du noeud i. Initialement,  $VID[i] = \emptyset, \forall i$ .
- iv. **Enchaîné avec EM** EM collabore par les structures de données suivantes qui sont utilisées pour transférer EM.
- ✓ **Receive\_election[n]**
  - ✓ **Ori** ID de l'initiateur actuel du protocole.
  - ✓ **Sender\_election**
- v. **Retourné avec CM** Comme EM, CM collabore également par les structures de données suivantes qui sont utilisées pour transférer CM.
- ✓ **Receive\_coordinator[n]**
  - ✓ **CID<sub>new</sub>** ID du nouveau coordinateur élu.
  - ✓ **VCID** ID du nouveau vice-coordinateur élu.
  - ✓ **CAG[K]** un tableau de tuples représentant les IDs des noeuds membres consultatifs du coordinateur.

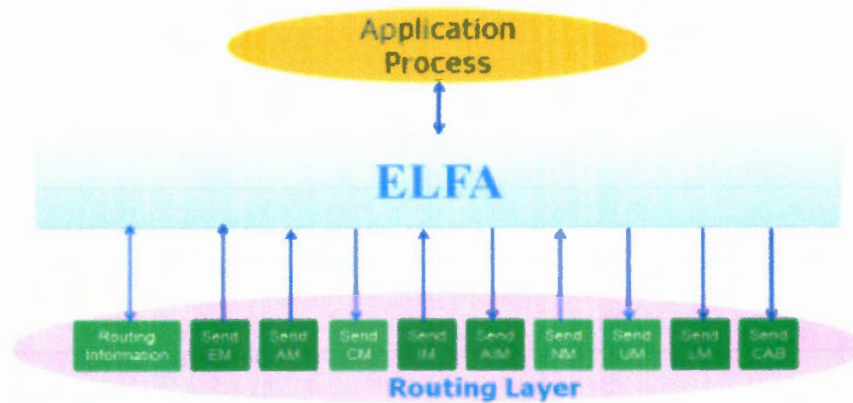


FIGURE 2.4 – Modèle du système de ELFA

### b) Principe de l'algorithme

Bien que ELFA utilise un tout nouveau concept de cabinet, il exécute MELFA au sein des CAGs pour décider du nouveau vice-coordonateur parmi les CAGs. L'algorithme ELFA utilise la PHASE 1 et la PHASE 2 de MELFA pour la distribution du message Election EM. Cependant, ELFA diffère dans le fonctionnement de la PHASE 3 et la PROCEDURE A de MELFA, afin de créer le cabinet et élire le vice-coordonateur.

#### **PROCÉDURE B** *Élection du Coordinateur et Formation de Cabinet à ori*

Cette procédure est presque similaire à la PROCEDURE A de MELFA, et s'exécute à ori avec le début de la PHASE1 et annonce le noeud qui possède NW le plus prioritaire comme coordinateur et les K noeuds NW ayant les plus grandes valeurs comme CAG, après l'expiration du temporisateur  $t_e$ , et le noeud ori passe à l'état NORMAL ou COORDINATOR ou ELITE ou VC, selon le cas, et diffuse un CM. Un peu après, l'ori passe à n'importe quel état sauf NORMAL et maintient également l'information complète sur les autres CAGs qui ont reçus CM.

#### **Phase adjointe 3** *Distribution CM*

La phase actuelle distribue le CM dans l'ensemble du réseau comme discuté plus tôt. En outre, il construit également les cabinets dans le système en présence du vice coordinateur et du coordinateur. En outre, le noeud  $j$  devient NORMAL, COORDINATOR ou ELITE, selon le cas, si le noeud  $j$  n'a pas encore réussi à élire le coordinateur. Ensuite, la fin cette phase marque l'achèvement des deux tâches, c'est-à-dire la construction du cabinet et l'élection du coordinateur et du vice-coordonateur. A ce stage, chaque noeud, CAG et vice-leader connaissent le coordonnateur et de chaque CAG connaît également les autres CAGs. Cette phase est suivie par quelques autres événements, qui sont les suivants :

**ÉVÉNEMENT 1** *Nouveau Noeud Envoie EM Pour rejoindre le réseau ayant déjà un leader*

L'événement 1 se produira lorsqu'un ou plusieurs noeuds arrivent dans la portée de transmission d'un MANET ayant déjà un leader, où chaque noeud connaît le courant coordinateur et le nouveau noeud joignant le réseau souhaite participer dans ce MANET. Dans cette situation, les nouveaux noeuds diffusent EM à leurs voisins seulement (phase 1 modifiée) et attendent l'expiration du temporisateur  $t_e$ , sinon réinitialisent EVENT1.

**ÉVÉNEMENT 2** *Réception De L'EM En Présence De LEADER.*

L'événement 2 se produit quand un noeud du MANET (noeud ou CAG ou leader), en présence d'un coordinateur existant, reçoit EM de n'importe quel nouveau noeud pour rejoindre le système. Par la suite, le destinataire, qui peut être un noeud, un CAG ou un leader, renverra NM ; Cependant, il envoie UM aussi si le destinataire n'est pas leader. Néanmoins, si le destinataire n'a pas d'information sur le coordinateur, il envoie un message d'attente(WM).

**ÉVÉNEMENT 3** *Réception D'un message EM Dans L'absence de leader.*

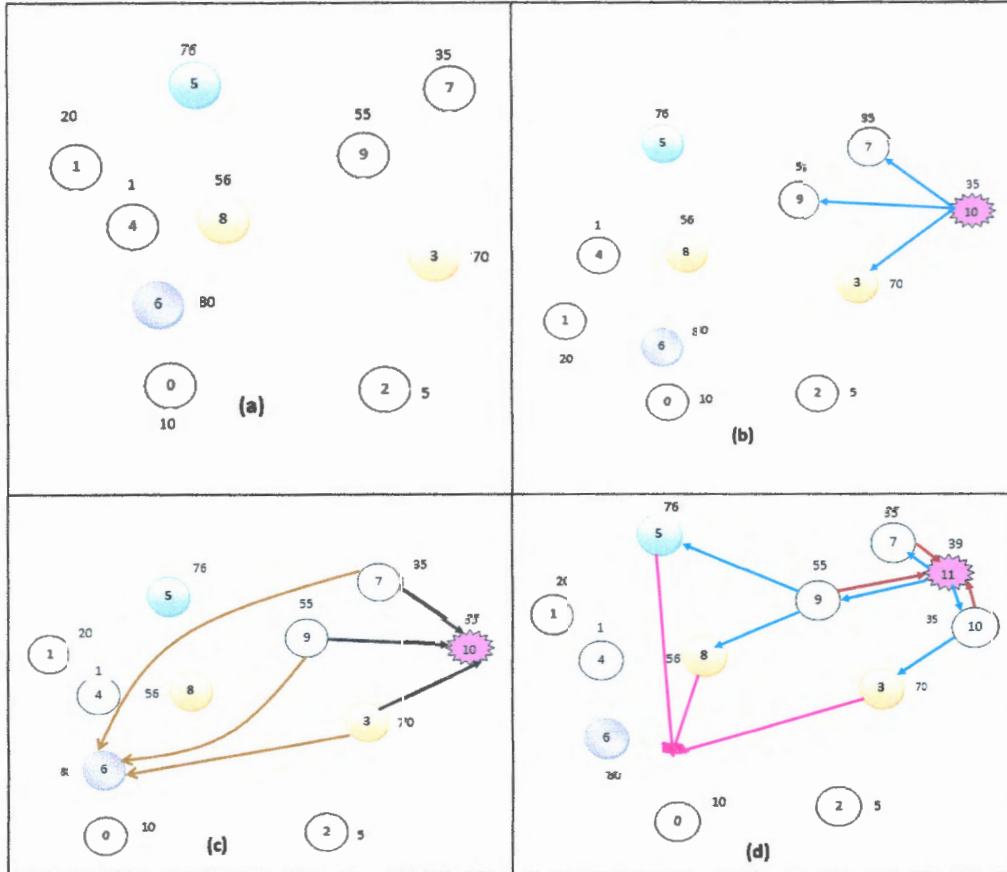
MELFA s'exécute au sein des CAGs sur l'occurrence de l'événement 3 qui rend cet algorithme tolérant aux fautes. Pour des raisons comme le départ du leader actuel ou l'arrivée d'un nouveau noeud dans le système, si les CAGs ou le vice-leader reçoivent l'EM de n'importe quel noeud et ils ne reçoivent pas le message heart-beat du leader, ils envoient IM seulement deux fois. Si le CAG reçoit l'AIM dans  $T_{max}$  propagate, il renvoie le NM ; Sinon, il transfère la charge et la responsabilité du chef vers le vice-coordinateur en utilisant LM. En outre, le vice-coordinateur en charge envoie un message heart-beat à tous les noeuds et envoie également le message CAB au plus petit CAG, qui se termine par l'élection de nouveau vice-coordinateur au sein des CAGs.

**ÉVÉNEMENT 4** *Pas De Message HeartBeat De Vice-Coordinateur En Charge.*

En réponse à LM, le nouveau leader envoie un message HeartBeat. Ainsi, l'absence de ce message indique que soit le vice-coordonnateur en charge a tomber en panne ou il peut y avoir partition réseau. Pour surmonter cette situation, les CAGs initient le MELFA similaire, expliqué dans l'événement 3. Après un délai raisonnable, si aucun nouveau leader ne pouvait être choisi parmi les CAGs, ces derniers initient encore MELFA, récursivement, au sein de composantes individuelles pour trouver de nouveau leader.

**ÉVÉNEMENT 5** *Fusion de Réseaux*

Ce problème est traité comme dans TORA. L'ÉVÉNEMENT 5 s'exécutera lorsque deux composantes partitionnées se rencontrent en raison de la formation d'un nouveau lien. Le coordinateur de l'un des composants ayant le NW le plus élevé devient le coordinateur de toute la composante. Cependant, et contrairement à TORA, dans cette approche l'autre leader devient le CAG. L'ÉVÉNEMENT 5 élit deux vice-coordonateurs. Ces vice-leaders deviennent les nouveaux leaders un par un chaque fois que le leader actuel part.



## Conclusion générale

Le problème d'élection de leader constitue une brique de base utile dans les systèmes distribués statiques et dynamiques particulièrement ceux sujets à des défaillances. Dans le contexte des réseaux mobiles ad hoc où les défaillances et les changements fréquents de topologie sont considérés comme la norme et non l'exception, la résolution du problème d'élection devient plus difficile.

Dans ce projet de fin d'étude, nous nous sommes intéressés au problème d'élection de leader pour les réseaux mobiles ad hoc, en particulier l'élection de  $K$  meilleurs leaders. Pourquoi  $K$  leaders ? C'est pour réaliser un système cohérent et fiable tout en réduisant le nombre de processus d'élection grâce au remplacements direct de leader lorsqu'il tombe en panne, où le vice leader prend en charge les responsabilités et les tâches du leader sans être besoin de refaire le processus d'élection. Les algorithmes font l'élection selon une valeur dite ' priorité ' qui peut correspondre à l'énergie résiduelle des noeuds, ses capacités de calcul, ses espaces de stockage, etc. Les noeuds élu doivent être les noeuds qui ont la plus grande priorité dans leurs composante.

Afin d'étudier les performances de ses algorithmes, nous avons utilisé le simulateur J-Sim. L'étude comparative entre les algorithmes d'élection que nous avons simulé a été en fonction du nombre de messages échangés, le temps d'élection, le temps de convergence de l'algorithme ainsi que le taux de stabilisation. Les résultats de simulations obtenus ont montré que les algorithmes DELFA et CBLEAA sont meilleurs que LEAA en termes de nombre de messages échangés, temps de convergence et taux de stabilisation aussi. Ceci montre l'efficacité des algorithmes d'élection de  $k$  meilleurs leaders par rapport à ceux qui font l'élection d'un seul leader.

### Perspectives

Comme perspectives de notre travail, nous envisageons de :

- Calculer le taux de stabilisation de l'algorithme CBLEAA avec un grand nombre de noeud.
- Calculer le temps de convergence de l'algorithme DELFA dans un environnement dy-



namique, et ainsi que le taux de stabilisation.

- Implémenter une amélioration de l'algorithme CBLEAA.

# Bibliographie

- [1] N.Boukhechem. *Routage dans les réseaux mobiles ad hoc par une approche à base d'agents* . Mémoire de magister, Université de constantine, 2008.
- [2] A.BOUDRIES. *Maintien de la Connectivité dans les Réseaux Ad hoc sans fil*. PhD thesis, Université Ferhat Abbas de Sétif 1, 2014.
- [3] B.Benmammam et A.Amraoui. *Réseaux de radio cognitive*. Mémoire de magister, LTT Laboratoire de Télécommunications Tlemcen, 2012.
- [4] A.Boudjaadar. *Plateforme basée Agents pour l'aide à la conception et la simulation des réseaux de capteurs sans fil*. PhD thesis, Université de Skikda, 2010.
- [5] Les réseaux sans fil. [http://www.icriq.com/fr/productique\\_tfp.html/-/asset\\_publisher/MeX1/content/les-reseaux-sans-fil/maximized](http://www.icriq.com/fr/productique_tfp.html/-/asset_publisher/MeX1/content/les-reseaux-sans-fil/maximized).
- [6] Kh.Belkheir et A.Haned. *Réseaux WiFi ad hoc*. PhD thesis, Institut de télécommunication d'Oran, 2008.
- [7] F. LEMAINQUE. *Tous sur les Réseaux sans fils*. DUNOD, Avril 2009.
- [8] G. PUJOLLE. *Les Réseaux Edition 2008*. EYROLLES, Septembre 2007.
- [9] B.Salim HAGGAR. *Auto-organisation et routage dans les réseaux mobiles ad hoc*. PhD thesis, Université de Reims Champagne-Ardenne, 2011.
- [10] BA.Mandicou. *Vers une structuration auto-stabilisante des réseaux ad hoc : cas des réseaux de capteurs sans fil*. PhD thesis, Université de Reims Champagne-Ardenne, 2014.
- [11] *Measured performance of wereless lan* . Technical report, Computer Science Department Colimbia University, September 1992.
- [12] NA Nouri. *Évaluation des performances du protocole 802.11 mode adhoc*. Mémoire de Magistère, Département d'Informatique, Université de Bejaïa, 2006.
- [13] S.NAIMI. *Gestion de la mobilité dans les réseaux Ad Hoc par anticipation des métriques de routage*. PhD thesis, Université Paris-Sud, 2015.
- [14] FRIKHA Mounir. *Réseaux ad hoc : routage, qualité de service et optimisation*. Lavoisier, (2010).

- [31] S. Corson V. Park. *Temporally-Ordered Routing Algorithm (TORA) Version 1. Internet Draft*, 26 Nov 1997.
- [32] N. Vaidya N. Malpani, J. Welch. *Leader Election Algorithms for Mobile Ad Hoc Networks. 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, PP 96-103, 2000.
- [33] Nadjib Badache Abdelouahid Derhab. *A Self-Stabilizing Leader Election Algorithm in Highly Dynamic Ad Hoc Mobile Networks . IEEE transactions on parallel and distributed Systems transactions on parallel and distributed systems*, VOL.19, NO.7, JULY 2008.
- [34] Jennifer E. Walter Jennifer L. Welch Rebecca Ingram, Patrick Shields. *An Asynchronous Leader Election Algorithm for Dynamic Networks . In 23rd IEEE International Symposium on Parallel and Distributed Processing*, PP 1-12, 2010.
- [35] Jim Kurose Don Towsley Sudarshan Vasudevan, Neil Immerman. *A Leader Election Algorithm for Mobile Ad Hoc Networks . University of Massachusetts*, VOL ,PP 96-103,2004.
- [36] Shantanu Sharma Awadhesh Kumar Singh. *Message Efficient Leader Finding Algorithm for Mobile Ad Hoc Networks. International Journal of Computer Networks and Communications Security National Institute of Technology, Kurukshetra, Haryana, India., January 2011.*
- [37] M.Mizanur Rahman, M.Abdullah-Al-Wadud, and O.Chae. *Performance analysis of Leader Election Algorithms in Mobile Ad hoc Networks . IJCSNS International Journaj of Computer Science and Network Security*, VOL.8 No.2, February 2008.
- [38] Shantanu Sharma Awadhesh Kumar Singh. *Elite Leader Finding Algorithm for MANETs. National Institute of Technology Kurukshetra, Haryana, India*, PP 125-132, January 2011.
- [39] Shantanu Sharma and Awadhesh Kumar Singh. *An Election Algorithm to Ensure the High Availability of Leader in Large Mobile Ad Hoc Networks . Parallel, Emergent and Distributed Systems, IJPEDS*, PP 172-196, 2016.
- [40] Les réseaux sans fil. <https://sites.google.com/site/jsimofficial/j-sim/>.
- [41] Les réseaux sans fil. <http://www.isi.edu/nsnam/ns/>.
- [42] Kamil Chebira. *Etude et analyse de la stabilité des protocoles de routages dans les réseaux ad hoc*. PhD thesis, 2006-2007.
- [43] Deepesh Rawat Mrs. Poonam Chhimwal, Dhajvir Singh Rai. *Comparison between Different Wireless Sensor Simulation Tools . Journal of Electronics and Communication Engineering (IOSR-JECE)*, VOL.5 No.2, PP 54-60, 2013.
- [44] Les réseaux sans fil. <http://pcl.cs.ucla.edu/projects/glomosim/>.
- [45] Gezahegn Gelta. *Performance evaluation of wireless sensor network routing protocoles for critical condition monitoring applications*, Octobre 2007.



## RÉSUMÉ

L'élection de leader est un composant fondamental dans les systèmes distribués. La définition classique de ce problème est d'élire finalement un leader unique parmi un ensemble fini de noeuds. Cependant, la résolution de ce problème devient plus difficile dans les réseaux mobiles ad hoc où les coupures des liens et la défaillance des noeuds sont considérées comme norme et pas comme exception. Bien que la plupart des algorithmes d'élection se concentrent sur la réduction du nombre de messages, il y avait peu d'attention à assurer la haute disponibilité d'un leader malgré divers types d'échecs. Élire plusieurs meilleurs leaders est une approche intéressante afin de minimiser l'échange de messages et préserver l'énergie d'un côté, et d'assurer la haute disponibilité du leader de l'autre côté. Contrairement aux solutions précédentes, les algorithmes de cette approche proposent que chaque noeud maintienne une liste de candidats afin de minimiser le nombre total d'élections de leader.

Le but de ce travail est d'étudier, en premier lieu, les algorithmes d'élection de K meilleurs leaders, puis faire une étude comparative, après simulation, entre un algorithme d'élection d'un seul leader et des algorithmes d'élection de k meilleurs leaders dans un environnement mobile ad hoc, ceci afin de voir les avantages de cette nouvelle approche. La comparaison est faite en fonction du nombre de messages échangés, le temps nécessaire pour faire l'élection, le temps de convergence de l'algorithme ainsi que le taux de stabilisation. Nous avons utilisé le simulateur J-Sim pour implémenter les trois algorithmes : LEAA, CBLEAA et DELFA. Les résultats de simulations obtenus ont montré que les algorithmes DELFA et CBLEAA génèrent moins de messages que l'algorithme LEAA. En outre, ils convergent rapidement à une situation légitime et montrent une meilleure stabilisation que LEAA.

**Mots clés :** élection de leader ; réseaux mobiles ad hoc ; priorité ; K meilleurs leaders ; . .

## ABSTRACT

Leader election is a fundamental control problem in both wired and wireless systems. The classical statement of leader election problem is to eventually elect a unique leader from a fixed set of nodes. However, in the context of mobile ad hoc networks, complications may arise due to frequent and unpredictable topological changes. Several leader election algorithms have been proposed in the literature. Although, most of them focus on reducing the total number of messages exchanged, they have been almost no attention on ensuring high availability of a leader despite various types of failures. Electing several top leaders is an interesting approach to minimize message exchange, preserve energy and ensure high availability of the leader. Unlike previous solutions, the algorithms of this approach propose that each node maintains a list of candidates in order to minimize the total number of leader elections.

The aim of this work is to study K top leaders election algorithms and make a comparative study between a single leader election algorithm and top k-leaders election algorithms in mobile ad hoc networks. The comparison is based on message overhead, election time, convergence time and the Fraction of stabilization time. We used J-Sim simulator to implement the three algorithms : LEAA, CBLEAA and DELFA. The results of simulations obtained have shown that DELFA and CBLEAA generate fewer messages than LEAA. In addition, they quickly converge to a legitimate state.

**Key words :** leader election ; mobile ad hoc networks ; priority ; K top leaders