

*République Algérienne Démocratique et Populaire*  
*Ministère de l'Enseignement Supérieur et de La Recherche Scientifique*  
***Université Ammar Telidji Laghouat***



*Faculté des Sciences*  
*Département d'Informatique*  
*et de Mathématiques*

***Domaine*** : *Mathématiques et Informatique (MI)*

*Filière* : *Informatique*

***Option*** : *Réseaux, Systèmes et Applications Répartis (ReSar)*

***Mémoire de Master***

*Présenté par :*

***Benmebarek Brahim***

***Benlahbib Abdessatar***

*Thème*

*L'influence de mobilité sur les algorithmes de partage de ressource  
dans les Manets*

## *Remerciement*

*Dedicase*

## **Résumé**

*La mobilité des nœuds dans les réseaux mobiles Ad hoc provoque des partitionnements et des fusionnements dans le réseau, dans ce mémoire nous présentons une solution de fusionnement après un partitionnement dans le réseau, nous allons d'abord introduire le concept des systèmes distribués leur caractéristique et leur but ainsi que les réseaux sans fil, et les réseaux mobiles AD HOC.*

*Nous expliquerons également la notion de partage de ressource et les types des ressources partagées et le système distribué de gestion des ressources, puis nous allons expliquer le principe de verrouillage des ressources (exclusion mutuelle), puis nous allons présenter l'un des problèmes fondamentaux dans les réseaux MANET qui est le partitionnement, et nous présenterons une solution en cas de fusionnement.*

*Nous présentons un algorithme qui permet de détecter le fusionnement qui est basé sur les critères d'énergie et le minimum identificateur.*

*Nous avons utilisé l'outil de simulation NS-2 afin d'étudier la performance de l'algorithme proposé, les résultats montrent que le comportement de notre algorithme est logique, ce qui valide la performance de la solution proposée.*

***Mots-clés : Réseaux mobiles AD HOC, partage de ressources, système Distribué, exclusion mutuelle, AODV, simulation, NS-2.***

## ***Abstract***

*The mobility of nodes in mobile networks Ad hoc causes partitioning and mergers in the network, in this paper we present a merger after a partitioning solution in the network, we will first introduce the concept of distributed systems and their characteristic their goal and wireless networks, and mobile networks AD HOC.*

*We also explain the concept of resource sharing and types of shared resources and systems distributed resource management, then we will explain the resource lock principle (mutual exclusion), then we will present one of the fundamental problems in MANET networks is partitioning, and present a solution merger cases. We present an algorithm that detects the merger which is based on energy criteria and the minimum identifier.*

*We used the NS-2 simulation tool to study the performance of the proposed algorithm, the results show that the behavior of our algorithm is logical, which validates the performance of the proposed solution.*

***Keywords: AD HOC Mobile networks, resource sharing, Distributed system, mutual exclusion, AODV, simulation, NS-2.***

# Table Des Matières

<u>1. Notions Générales.....</u>	<u>11</u>
<u>1.1. Introduction.....</u>	<u>12</u>
<u>1.2. Les systèmes distribués.....</u>	<u>12</u>
<u>1.2.1. Définition.....</u>	<u>12</u>
<u>1.2.2. Exemples.....</u>	<u>12</u>
<u>1.2.3. Les caractéristiques d'un S.D.....</u>	<u>13</u>
<u>1.2.4. Les Buts d'un S.D.....</u>	<u>13</u>
<u>1.3. Les réseaux mobiles.....</u>	<u>14</u>
<u>1.3.1. Définition.....</u>	<u>14</u>
<u>1.3.2. Les types des réseaux mobiles.....</u>	<u>14</u>
<u>1.3.2.1. Les réseaux mobiles avec infrastructure.....</u>	<u>14</u>
<u>1.3.2.2. Les réseaux mobiles sans infrastructure.....</u>	<u>15</u>
<u>1.3.3. Les avantages des réseaux mobiles ad hoc.....</u>	<u>15</u>
<u>1.3.4. Les inconvénients des réseaux mobiles ad hoc.....</u>	<u>15</u>
<u>1.4. Partage des ressources.....</u>	<u>16</u>
<u>1.4.1. Définition.....</u>	<u>16</u>
<u>Conclusion.....</u>	<u>17</u>
<u>2. L'Influence de Mobilité Sur Les Algorithmes de Partage des Ressources Dans Les Réseaux Ad hoc.....</u>	<u>18</u>
<u>2.1. Introduction.....</u>	<u>20</u>
<u>2.2. Partage des ressources.....</u>	<u>20</u>
<u>2.2.1. Définition.....</u>	<u>20</u>
<u>2.2.2. Gestion des ressources.....</u>	<u>20</u>
<u>2.3. L'exclusion mutuelle.....</u>	<u>22</u>
<u>2.3.1. Définition.....</u>	<u>22</u>
<u>2.3.2. Synchronisation des processus.....</u>	<u>22</u>
<u>2.3.3. Les mécanismes d'exclusion mutuelle.....</u>	<u>23</u>
<u>2.4. Problème de partitionnement.....</u>	<u>23</u>
<u>2.4.1. Définition.....</u>	<u>23</u>
<u>2.4.2. Les causes d'un partitionnement.....</u>	<u>23</u>
<u>2.4.3. Les inconvénients du partitionnement.....</u>	<u>23</u>

<u>2.4.4. Techniques de détection du partitionnement.....</u>	<u>24</u>
<u>2.5. Clustering.....</u>	<u>24</u>
<u>2.5.1. Définition.....</u>	<u>24</u>
<u>2.5.2. Cluster.....</u>	<u>24</u>
<u>2.5.2.1. Définition.....</u>	<u>24</u>
<u>2.5.3. Avantages du clustering.....</u>	<u>25</u>
<u>2.5.4. Inconvénients du clustering.....</u>	<u>26</u>
<u>2.5.5. Les facteurs d'élection du cluster head.....</u>	<u>26</u>
<u>2.5.6. Les fameux algorithmes d'élection du cluster head.....</u>	<u>26</u>
<u>2.5.6.1. Identifier based clustering.....</u>	<u>27</u>
<u>2.5.6.2. Power (Energy) based clustering.....</u>	<u>27</u>
<u>2.6. Détection du fusionnement.....</u>	<u>28</u>
<u>2.7. L'algorithme proposé.....</u>	<u>28</u>
<u>2.7.1. L'algorithme.....</u>	<u>29</u>
<u>Conclusion.....</u>	<u>30</u>
<u>3. Simulation et Analyse des Résultats.....</u>	<u>31</u>
<u>3.1. Introduction.....</u>	<u>34</u>
<u>3.2. Le simulateur ns-2.....</u>	<u>34</u>
<u>3.2.1. Définition.....</u>	<u>34</u>
<u>3.2.2. Présentation du simulateur ns-2.....</u>	<u>34</u>
<u>3.2.3. Les phases de simulation.....</u>	<u>34</u>
<u>3.3. Les paramètres de simulation.....</u>	<u>35</u>
<u>3.3.1. Les paramètres fixes.....</u>	<u>35</u>
<u>3.3.2. Les paramètres variables.....</u>	<u>35</u>
<u>3.4. Code de l'algorithme.....</u>	<u>36</u>
<u>3.4.1. Les variables.....</u>	<u>36</u>
<u>3.4.2. Les messages utilisés.....</u>	<u>36</u>
<u>3.5. Le protocole de routage utilisé.....</u>	<u>36</u>
<u>3.6. Les procédures d'algorithme.....</u>	<u>36</u>
<u>3.6.1. Procédures d'initialisation.....</u>	<u>36</u>
<u>3.6.2. Procédure du découverte.....</u>	<u>38</u>

<u>3.6.3.Procédure de choix.....</u>	<u>38</u>
<u>3.6.4.Procédure de Mise à Jour.....</u>	<u>39</u>
<u>3.7.Scénario de simulation.....</u>	<u>40</u>
<u>3.8.Résultats de simulation.....</u>	<u>40</u>
<u>Conclusion.....</u>	<u>41</u>
<u>Conclusion Générale.....</u>	<u>42</u>
<u>Bibliographie.....</u>	<u>43</u>
<u>Annexe.....</u>	<u>44</u>
<u>Acronyme.....</u>	<u>51</u>

# Liste des Figures

1.1 Architecture du Système Distribué[2].....	15
1.2 Architecture du réseau mobile avec infrastructure.....	17
1.3 Architecture du réseau mobile sans infrastructure.....	18
2.1 Architecture du système de gestion des ressources distribuée[5].....	24
2.2 Structure du cluster.....	28
2.3 Classification des techniques du clustering[13].....	30
3.1 Scénario de simulation.....	41
3.2 Scénario de simulation.....	41
3.3 Influence du nombre de clusters.....	42
3.4 Influence du nombre de nœuds.....	42

## **Introduction Générale**

Ces dernières années le développement de la technologie sans fil a ouvert de nouvelles perspectives dans le domaine des télécommunications, les réseaux mobiles basés sur la technologie sans fil connaissent aujourd'hui une forte expansion. Les réseaux mobiles offrent une grande flexibilité d'emploi, ils permettent aux utilisateurs de se déplacer librement tout en continuant normalement leurs communications. Il existe deux types de réseaux mobiles, les réseaux mobiles avec infrastructure et les réseaux mobiles ad hoc. Le concept des réseaux mobiles ad hoc essaie d'étendre les notions de la mobilité à toutes les composantes de l'environnement, contrairement aux réseaux basés sur la communication avec infrastructure, aucune administration centralisée n'est disponible, ce sont les hôtes mobiles eux mêmes qui forment une infrastructure du réseau[1].

Aucune supposition ou limitation n'est faite sur la taille du réseau mobile ad hoc. Parmi les conséquences de la mobilité la création d'un réseau mobile partitionnée sous forme des clusters et ainsi la mobilité peut provoquer un réseau mobile fusionnée (la fusionnement de deux clusters ou plus), un cluster peut être vue comme un regroupement des nœuds, chaque cluster a un seul cluster head (leader) responsable de la communication entre les nœuds intra cluster ou bien inter cluster, et un nœud de coordination (Gateway) localisé dans les bordures du cluster, qui joue le rôle d'un passerelle entre les clusters, le fusionnement de deux clusters ou plus fait apparaître le problème du choix d'un nouveau cluster head au nouveau réseau fusionnée. Surpasser cette lacune revient à la réponse des questions suivantes:

- Comment détecter un fusionnement dans le réseau?
- Sur quelle critère est basé le choix d'un nouveau cluster head?

Dans notre travail nous allons étudier l'influence du mobilité sur les algorithmes du partage des ressources dans les réseaux MANET. Nous avons présenté une solution permettant de détecter le fusionnement dans les réseaux et permet également de choisir un nouveau leader dans le nouveau réseau fusionné. Afin de valider notre algorithme une simulation a été réalisée par le simulateur du réseau NS2 (Network Simulator).

Le reste de ce mémoire est organisée comme suit:

le premier chapitre présente un survol sur le concept d'un système distribué, leurs caractéristiques et leurs buts, puis un survol sur les notions des réseaux sans fils et des réseaux ad hoc.

Dans le deuxième chapitre nous expliquons le problème de partage de ressource et l'exclusion mutuelle, et nous étudions le problème de partitionnement, ses causes et ses inconvénients, nous expliquons ensuite notre solution proposée.

Dans le troisième chapitre, le code de la solution proposée, et les résultats obtenus ont été présentés.

*Nous terminons par une conclusion générale qui conclut notre travail et les résultats obtenus et annonce quelques perspectives.*

# CHAPITRE 1

## 1. Notions Générales

### Sommaire

<u>1.1.Introduction.....</u>	<u>14</u>
<u>1.2.Les systèmes distribuées.....</u>	<u>14</u>
<u>1.2.1.Définition.....</u>	<u>14</u>
<u>1.2.2.Exemples.....</u>	<u>14</u>
<u>1.2.3.Les caractéristiques d'un S.D.....</u>	<u>15</u>
<u>1.2.4.Les Buts d'un S.D.....</u>	<u>15</u>
<u>1.3.Les réseaux mobiles.....</u>	<u>16</u>
<u>1.3.1.Définition.....</u>	<u>16</u>
<u>1.3.2.Les types des réseaux mobiles.....</u>	<u>16</u>
<u>1.3.2.1.Les réseaux mobiles avec infrastructure.....</u>	<u>16</u>
<u>1.3.2.2.Les réseaux mobiles sans infrastructure.....</u>	<u>17</u>
<u>1.3.3.Les avantages des réseaux mobiles ad hoc.....</u>	<u>17</u>
<u>1.3.4.Les Inconvénients des Réseaux Mobiles Ad hoc.....</u>	<u>17</u>
<u>1.4.Partage des ressources.....</u>	<u>18</u>
<u>1.4.1.Définition.....</u>	<u>18</u>
<u>Conclusion.....</u>	<u>19</u>

## 1.1. Introduction

L'essor des technologies sans fil offre aujourd'hui de nouvelles perspectives dans le domaine des télécommunications. L'évolution récente des moyens de la communication sans fil a permis la manipulation de l'information à travers des unités de calculs portable qui ont des caractéristiques particulières (une faible capacité de stockage, une source d'énergie autonome...) et accèdent au réseau à travers une interface de communication sans fil, comparant avec un environnement statique, le nouvel environnement résultant appelé l'environnement mobile, permet aux unités de calcul une libre mobilité, et il ne pose aucune restriction sur la localisation des usages.

Dans ce chapitre, nous allons d'abord présenter une vue générale sur les systèmes distribués, puis nous allons présenter les réseaux mobiles, les réseaux Ad hoc en particulier avec quelques problèmes liés à ces réseaux tel que le problème de partage de ressources.

## 1.2. Les systèmes distribués

### 1.2.1. Définition

Un S.D (Système Distribué) est un ensemble des ordinateurs autonomes vus à l'utilisateur comme un seul système cohérent où la coordination entre les différents sites (ordinateurs) est faite par l'échange des messages dans un réseau câblé ou sans fils, un S.D implémente la même couche particulière (middleware) son rôle principale est l'accès uniforme à la même interface à partir de n'importe quel site et de faciliter la communication entre les sites[2].

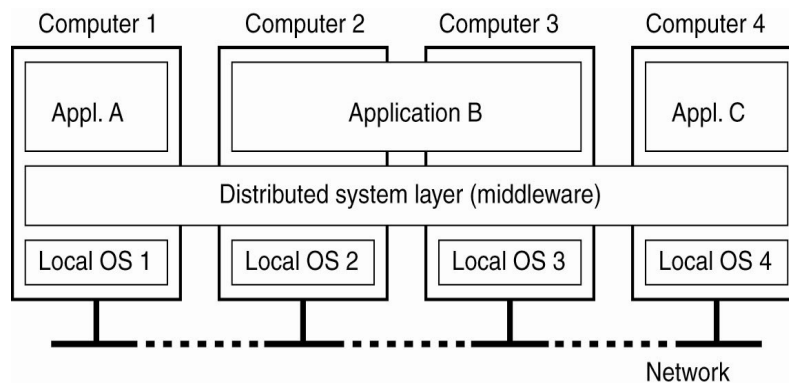


Figure 1.1 Architecture du Système Distribué[2].

### 1.2.2. Exemples

- Internet.
- Clusters (des groupes des unités mobiles dispersées dans des localisations géographique différents).
- Une ligne d'assemblage automatique.

- *Multi-player online games.*
- *Un système de transaction financière.*

### 1.2.3. *Les caractéristiques d'un S.D*

- **La Concurrence** : la propriété de l'indépendance entre les différents sites fait de la gestion de la concurrence pour protéger les ressources partagées entre les différents sites un point critique pour le bon fonctionnement du SD[3].
- **Asynchrones** : Une horloge globale n'existe pas entre les différents sites du SD, alors la seule méthode qui assure la coordination entre les sites est l'échange des messages[3].
- **Tolérance aux pannes** : quand un site tombe en panne, le SD va continuer à fonctionner sans aucun problème[3].

### 1.2.4. *Les Buts d'un S.D*

- **Hétérogénéité** : examiner la capacité du S.D de faire tourner une large gamme des différents types des sites[4].
- **Robustesse** : est ce que la conception du S.D est bien protégé contre les crashes du système[4] ?
- **La Disponibilité** : est ce que les données et les ressources partagées toujours disponible[4] ?
- **L'Accessibilité** : fournir toujours un accès aux différentes ressources partagées (fichiers, imprimantes, etc.) et le partage égaux entre les différents sites[2].
- **Transparence** : le but de ce point est de masquer au maximum le fonctionnement interne du S.D dans les points suivants :
  - **L'Accès** : l'accès uniforme aux ressources distantes et locales[2].
  - **La Location** : l'accès à n'importe quelle ressource sans avoir besoin de connaître sa location[2].
  - **La Migration** : aucun effet sur l'accès à une ressource en cours d'être déplacé[2].
  - **Redondance** : augmenter la disponibilité des ressources par autorisation de plusieurs instances du même ressource[2].
  - **La Concurrence** : autoriser plusieurs accès à la même ressource sans aucun dérangement[2].
  - **Tolérance aux Pannes** : masquer les erreurs qui peuvent avoir lieu durant la location d'une ressource[4].
  - **La Mobilité** : masquer la réassociation (déplacement) d'une ressource aux nouveaux utilisateurs à chaque demande[4].
  - **Efficacité** : est ce que le S.D fonctionne à 100% et est ce que le service offert est de qualité ou bien non?.
- **Ouvert**: an open S.D a les caractéristiques suivants[2] :
  - **Interopérabilité** : deux systèmes différents peuvent se communiquer entre eux sans

problème.

- **Portabilité** : une application exécutée sur un système d'exploitation peut s'exécuter sur un autre.
- **Extensibilité** : ajout des nouveaux services.
- **La scalabilité** : est ce que le S.D peut supporter l'ajout des nouveaux nœuds sans une dégradation remarquée dans le performance[2] ?.
- **La Sécurité** : le S.D doit protéger les informations partagées, la sécurité d'un S.D considérée comme un grand défi aux concepteurs du S.D[4].

### 1.3. Les réseaux mobiles

#### 1.3.1. Définition

Un réseau mobile est un système composé de sites mobiles, qui donne à l'utilisateur la capacité de se connecter indépendamment de sa position géographique, on distingue deux types des réseaux mobiles, les réseaux mobiles avec infrastructure et sans infrastructure.

#### 1.3.2. Les types des réseaux mobiles

##### 1.3.2.1. Les réseaux mobiles avec infrastructure

Un ensemble des nœuds mobiles connectés entre eux à travers une station de base (point d'accès) reliée avec les autres stations de base par un réseau filaire (statique).

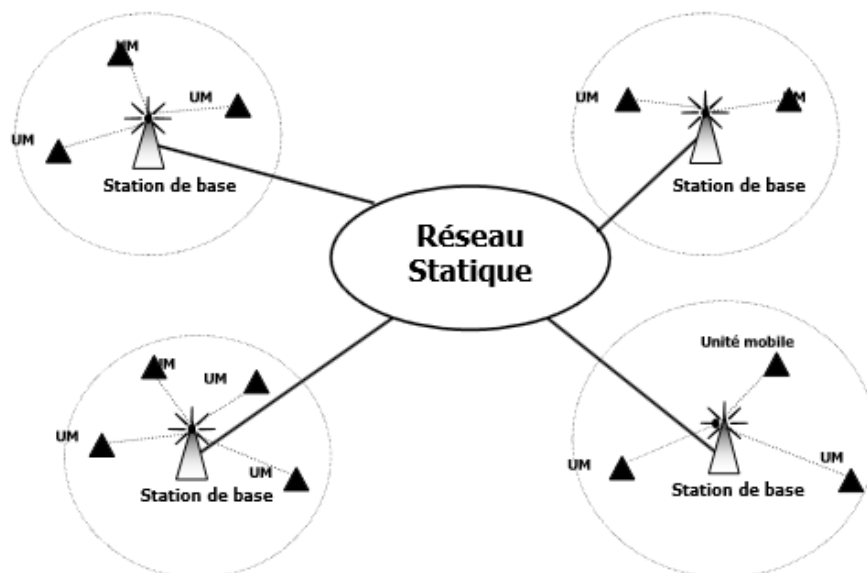
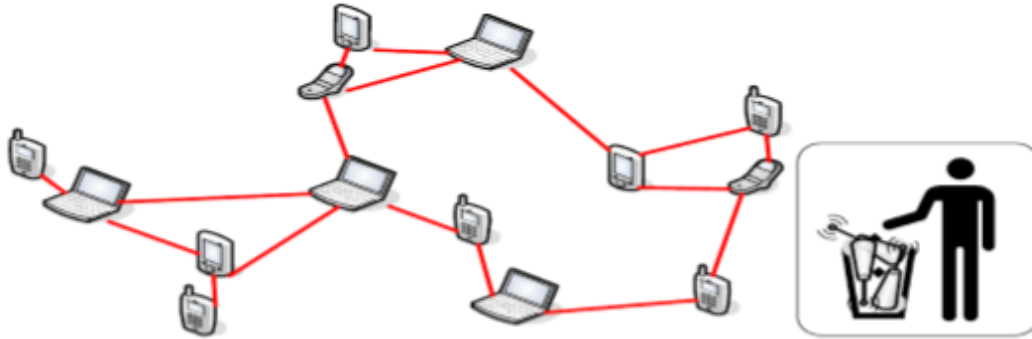


Figure 1.2 Architecture du réseau mobile avec infrastructure.

### **1.3.2.2. Les réseaux mobiles sans infrastructure**

*Est un ensemble des nœuds mobiles qui peuvent communiquer entre eux sans avoir besoin d'une administration centralisée (point d'accès), généralement appelée Ad hoc ou MAN(ET).*



*Figure 1.3 Architecture du réseau mobile sans infrastructure.*

### **1.3.3. Les avantages des réseaux mobiles ad hoc**

1. *La Facilité de déploiement : il suffit de mettre en place plusieurs machines pour créer le réseau. Ceci rend la construction d'un réseau AD HOC rapide et peu onéreuse.*
2. *Moins chère dans l'équipement par rapport au réseau avec infrastructure.*
3. *Evolutifs : pour ajouter un nœud à un réseau AD HOC préexistant, il suffit d'approcher le nouveau venu d'au moins l'un des membres du réseau. De même il suffit de l'en éloigner pour le retirer du réseau.*

### **1.3.4. Les inconvénients des réseaux mobiles ad hoc**

1. *Les réseaux Ad hoc présente un délai de communication supplémentaire par rapport au réseau avec infrastructure.*
2. *N'importe quelle nœud peut acheminer les données qui se posent un problème de confidentialité.*
3. *Consommation rapide dans les ressources d'un nœud à cause d'acheminement des données.*
4. *Le changement très fréquent dans les routes pose un problème de mise à jour fréquente ce qui représente un gaspillage dans les ressources du nœud.*

## ***1.4. Partage des ressources***

### ***1.4.1. Définition***

*Le partage de ressource est parmi les buts principaux d'un système distribué, différents types des ressources matérielles et logicielles peuvent être partagées, la communication dans ce système est une communication peer-to-peer où il n'y a aucune distinction entre les différents processus, pour l'allocation d'une ressource, un site fait la demande pour détenir la ressource en question, la synchronisation entre les différentes demandes est gérée par un système de gestion des ressources distribuées à l'aide des mécanismes de l'exclusion mutuelle[5].*

## ***Conclusion***

*Dans ce chapitre on a mis les points sur les réseaux mobiles, parmi ses avantages qu'elle donne aux utilisateur la capacité de connecter entre eux indépendamment de leurs positions géographiques, il existe deux types de réseaux mobile, réseau mobile avec infrastructure et réseaux mobile sans infrastructure, la seule différence entre eux est l'existence ou l'absence d'une administration centralisée, puis on a mis le point sur les avantages et les inconvénients de chaque type.*

*Dans le chapitre suivant on va exposer en détail le partage de ressource, l'exclusion mutuelle (les technique de verrouillage des ressource), le problème de partitionnement et les réseaux cluster avec l'algorithme d'élection d'un cluster head .*

# CHAPITRE 2

## 2. L'Influence de Mobilité Sur Les Algorithmes de Partage des Ressources Dans Les Réseaux Ad hoc

### Sommaire

<u>2.1.Introduction.....</u>	<u>21</u>
<u>2.2.Partage des ressources.....</u>	<u>21</u>
<u>2.2.1.Définition.....</u>	<u>21</u>
<u>2.2.2.Gestion des ressources.....</u>	<u>22</u>
<u>2.3.L'exclusion mutuelle.....</u>	<u>23</u>
<u>2.3.1.Définition.....</u>	<u>23</u>
<u>2.3.2.Synchronisation des processus.....</u>	<u>24</u>
<u>2.3.3.Les mécanismes d'exclusion mutuelle.....</u>	<u>24</u>
<u>2.4.Problème de partitionnement.....</u>	<u>25</u>
<u>2.4.1.Définition.....</u>	<u>25</u>
<u>2.4.2.Les causes d'un partitionnement.....</u>	<u>25</u>
<u>2.4.3.Les inconvénients du partitionnement.....</u>	<u>25</u>
<u>2.4.4.Techniques de détection du partitionnement.....</u>	<u>25</u>
<u>2.5.Clustering.....</u>	<u>26</u>
<u>2.5.1.Définition.....</u>	<u>26</u>
<u>2.5.2.Cluster.....</u>	<u>26</u>
<u>2.5.2.1.Définition.....</u>	<u>26</u>
<u>2.5.3.Avantages du clustering.....</u>	<u>27</u>
<u>2.5.4.Inconvénients du clustering.....</u>	<u>28</u>
<u>2.5.5.Les facteurs d'élection du cluster head.....</u>	<u>28</u>
<u>2.5.6.Les fameux algorithmes d'élection du cluster head.....</u>	<u>28</u>
<u>2.5.6.1.Identifier based clustering.....</u>	<u>29</u>

<u>2.5.6.2.Power (Energy) based clustering.....</u>	<u>29</u>
<u>2.6.Détection du fusionnement.....</u>	<u>30</u>
<u>2.7.L'algorithme proposé.....</u>	<u>30</u>
<u>2.7.1.L'algorithme.....</u>	<u>31</u>
<u>Conclusion.....</u>	<u>32</u>

## **2.1. Introduction**

*Parmi les conséquences de la mobilité des nœuds dans les réseaux MANET le partitionnement et aussi le fusionnement, donc il y a une possibilité de former des clusters à cause d'une partitionnement, ce qui pose un problème dans la gestion des ressources et ces clusters peuvent se fusionne ce qui rendre le réseau comme un seule cluster, il est donc nécessaire de trouver des moyens permettant de détecter une fusionnement et aussi faire le choix entre les clusters heads et mettre à jour l'information globale dans le réseau.*

## **2.2. Partage des ressources**

### **2.2.1. Définition**

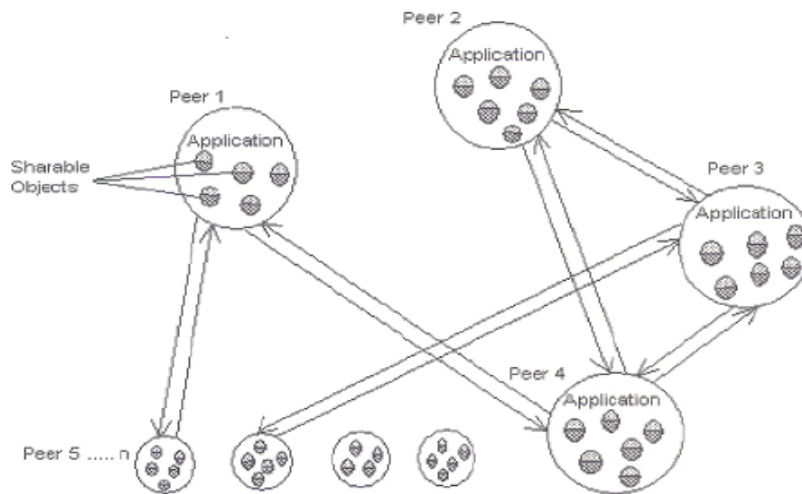
*La nature des ressources partagées dans un S.D se diffère, la ressource partagée peut être hardware comme des imprimantes ou des ressources logicielles comme des fichiers, L'avantage apporté est de grande importance comme pour instance d'économiser le coût dans le cas de partager une imprimante trop chère, pour exploiter une ressource il y a évidemment un service implémenter dans le S.D, un service est la partie du système responsable de gérer un ensemble des ressources et offert aux utilisateurs les fonctionnalité de ces derniers. Comme exemple l'accès aux fichiers partagés est gérer par un service des fichiers qui offre des opérations de manipulations des fichiers comme le read, le write ou bien le delete[3].*

*Un serveur est un processus du système chargé de répondre aux demandes des utilisateurs (client) pour lui fournir un service particulier, un tel approche est connue sous l'appellation client-serveur, les demandes initialisées par des client (users) et envoyées aux serveur sous forme des messages, les réponses envoyées du serveur aux clients, un serveur peut jouer le rôle d'un client si il demande un service des autres serveurs, les clients sont actives (faire des demandes) et les serveur sont passives attendues pour les demandes des clients[3].*

### **2.2.2. Gestion des ressources**

*Puisque les ressources doivent être gérées dans un système distribué alors il faut un mécanisme pour gérer les ressources entre les différents sites et coordonner l'accès aux ressources partagées. Le système de gestion des ressources dans un environnement distribué est un système avec un but principal: faciliter l'accès des utilisateurs aux ressources distants comme aux ressources locales, les ressources dans un S.D se trouvant au niveau des sites où chaque site peut demander l'accès à une ressource dans un autre site par l'envoi d'une requête dans le réseau, cette architecture de communication appelée*

peer-to-peer, l'avantage de cette architecture par rapport à l'architecture centralisée client-serveur est qu'elle est plus rapide et plus flexible où il n'y a aucune distinction entre les processus clients et serveurs dans leur interactions et ils coopèrent entre eux pour l'accomplissement d'une tâche, le grand défi pour un système de gestion des ressources est l'organisation de l'interaction entre les sites en communication, les messages validés doivent être délivrés aux destinations et les messages invalides doivent être rejetés ou bien tous le S.D va tomber en panne. Dans un S.D on trouve plusieurs copies d'une même ressource dans plusieurs sites dont le but est de balancer la charge entre les sites du S.D ce qui augmente la disponibilité des ressources et par conséquent la qualité du service[5].



**Figure 2.1 Architecture du système de gestion des ressources distribuée[5].**

Lorsqu'un site demande une ressource particulière, elle doit être mise en disposition au plus vite possible et si un autre site demande la même ressource, le même mécanisme doit être exécuté, alors il faut un mécanisme de timing (scheduling) pour organiser la possession de la ressource par les sites.

On peut distinguer deux types de ressources[5]:

**1. Ressource réutilisable**

Ce type de ressource ne se consomme pas à cause de son utilisation et peut être utilisée plusieurs fois.

**2. Ressource consommable**

Ce type de ressources termine à cause de son utilisation, quand un processus alloue une ressource on dit que la ressource a été consommée. Il n'y a pas un nombre fixe pour une telle ressource dans un S.D tant que ces unités peuvent être créées et détruites.

## 2.3. L'exclusion mutuelle

### 2.3.1. Définition

Considérée comme le brique de base de la programmation distribuée et de conception des systèmes d'exploitation, l'exclusion mutuelle considérée comme un problème réel dans un environnement mobile où l'accès à une ressource partagée doit être assurée entre les différents sites, il y a deux types des ressources :

1. **Ressources partageables** : le type de ressource qui peuvent être utilisée par plusieurs processus à la fois sans le risque que les processus peuvent entrer en conflit ou que le ressource peut être en état incohérent, pour des raisons comme exemple la lecture d'un fichier[6].
2. **Ressource non partageables** : ce type des ressources ne peut être utilisé que par un seul processus à la fois, comme pour instance l'accès à un fichier contient des données critiques et susceptibles d'être modifiées où si un processus entrer pour lire alors qu' un autre en cours de faire une modification alors le résultat sera incohérent[6].

### 2.3.2. Synchronisation des processus

Quand on partage des ressources critiques lors de son utilisation comme exemple lors d'imprimer des fichiers, l'imprimante est une ressource qui ne peut être utilisée que par un seul processus à la fois, les processus qui veulent imprimer partagent deux variables **in** et **out** où chacune représente le prochain numéro associée au fichier enregistrée pour être imprimée et le prochain numéro du fichier qui sort de l'imprimante respectivement, on doit être certaine que aucune interblocage aura lieu entre les processus et que les processus doit être servi en tout équité (problème de la famine), alors les variables partagées représente les ressources critique et la partie du code pour modéfier ces variables appelée la section critique, les mécanismes de contrôler la section critique (de l'exclusion mutuelle) doit satisfaire les propriétés suivants[6] :

- Au plus un processus par la section critique.
- Les processus en dehors de la section critique ne doit pas être bloquées.
- Les processus en attente ne doit pas mis en fil d'attente trop long pour être servi (problème de la famine).

Un processus passe par 3 états pendant son exécution[7] :

1. **Demandeur** : demande à utiliser la ressource et à entrer en section critique.
2. **Dedans** : dans la section critique et utilise la ressource.
3. **Dehors** : n' utilise pas la ressource et dehors la section critique.

L'interblocage (deadlock) peut se produire lorsqu'on a une attente circulaire entre les sites demandeurs, un site demande une ressource détenue par un autre site qui à son tour demande une ressource détenue par le premier site demandeur[6].

La famine (starvation) aura lieu quand un processus attende indéfiniment dans la file d'attente pour être servi et détenue un ressource[6].

### **2.3.3. Les mécanismes d'exclusion mutuelle**

- Approche centralisée : gérée par un serveur qui centralise l'accès à la ressource partagée entre les processus[7].
- Approche par jeton : un jeton circule entre les processus et le processus qui demande l'accès au ressource doit avoir le jeton (la coordination sur l'accès au ressource est gérée entre les processus eux mêmes)[7].
- Approche par permission : les processus doit s'autoriser entre eux afin de désigner quel processus doit utiliser la ressource partagée[7].

## **2.4. Problème de partitionnement**

### **2.4.1. Définition**

Le partitionnement est la division (breakdown) de la topologie connectée d'un réseau en plusieurs topologies déconnectées les une des autres. Un des grands problèmes dans les réseaux en général est le partitionnement et comment le prédire pour l'éviter[8].

### **2.4.2. Les causes d'un partitionnement**

Un partitionnement peut avoir lieu dans un réseau à base d'infrastructure à cause des pannes dans la plupart des parties de l'infrastructure, mais dans les MANET plusieurs raisons peuvent le causer[9]:

- Le mouvement des nœuds mobiles.
- L'arrêt (Shutdown) d'un nœud.
- Une destruction physique d'un nœud.
- Déchargement de la batterie du nœud.

### **2.4.3. Les inconvénients du partitionnement**

Parmi les inconvénients d'un partitionnement dans un réseau[9] :

- Les nœuds dans les différentes partitions ne peuvent plus communiquer entre eux.

- Minimiser la qualité de services offerts par le réseau.
- L'apparition des partitions dans un réseau, fait du réseau au pire des cas un ensemble des partitions déconnectées avec un seul nœud par partition.

#### **2.4.4. Techniques de détection du partitionnement**

Plusieurs techniques ont été proposées pour détecter le partitionnement, on peut citer la technique de **PING/ACK**; un nœud envoie un message du ping aux autres nœuds, après une durée du temps, et s'il ne reçoit pas un **ACK** d'un certain nœud, il l'ajoute au liste suspecte, certains chercheurs ont proposé une autre technique de prédiction qui consiste à utiliser trois équipements : un prédicteur d'échec, un planificateur du mouvement et un évaluateur d'environnement, ces équipements dépendent sur une quantité énorme des informations qui doit être échangées et des équipements de détection, aussi il y a des techniques qui utilisent des messages de contrôle (heartbeat messages) où l'absence d'un heartbeat message attendu conduit à un suspect du partitionnement, cette technique n'offre pas la distinction entre une panne du nœud et une véritable événement de partitionnement.

Parmi les solutions proposée dans les réseaux à base du cluster pour détecter un partitionnement est l'exploitation de certains nœuds situées à la bordure du cluster appelé gateway (passerelle) nœuds qui ont un rôle vitale dans la connexion entre les clusters, si un nœud gateway ne reçoit plus des messages d'autres nœuds gateway alors il suspecte qu'un partitionnement a lieu[8].

### **2.5. Clustering**

#### **2.5.1. Définition**

Le clustering est une méthode (une technique) de reformer le réseau **MANET** ou **WSN** (Wireless Sensor Network) de telle sorte que le nouveau réseau reformé contient des petits groupes des nœuds chacun appelé cluster dispersées géographiquement, alors ce qui va améliorer le routage dans la couche réseau par la minimisation de la taille de la table de routage et par conséquence minimiser la taille de l'overhead circulée dans le réseau[10].

#### **2.5.2. Cluster**

##### **2.5.2.1. Définition**

Un cluster est un regroupement des nœuds selon un certain critère (mobility, energy, trust, identité, etc..), les deux étapes essentielles pour tout algorithme de clustering sont :

### 1. Formation du Cluster

Cette étape est l'étape essentielle dans la construction du cluster, car dans cette étape un nœud particulier appelé **cluster head (CH)** doit être élu parmi tous les nœuds du cluster, le rôle du CH est de coordonner le fonctionnement du cluster comme la mise à jour des table de routage, le CH doit être capable de communiquer avec le reste du cluster. Lors d'une communication intra cluster, les nœuds ordinaires du cluster vont propager le paquet données au CH qui va prendre la responsabilité de le distribuer, cela est fait par une recherche dans sa table de routage d'un des nœuds particulier appelé le gateway, un nœud gateway est un nœud qui connecte deux ou plusieurs clusters (située dans la portée de transmission de deux ou plusieurs CH), dans un tel cas le gateway va envoyer le paquet au cluster head de l'autre cluster[11].

### 2. Maintenance du cluster

Elle aura lieu quand un mouvement (déconnexion) d'un nœud sera présentée, dans cette phase la réformation du cluster (reclustering) doit être évitée au maximum et la structure courante du cluster doit être préservée au maximum pour éviter le taux de calcul résultant à cause de la phase de formation du cluster.

La communication dans un cluster est divisée en trois étapes[12]:

1. Le cluster head reçoit les données arrivées de ces membres (voisins).
2. Le cluster head compresse les données reçues.
3. Finalement le cluster head envoie les données compressées aux autres cluster heads

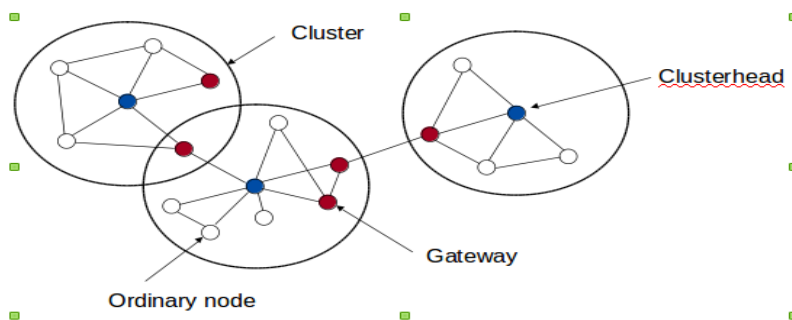


Figure 2.2 Structure du cluster[13].

### 2.5.3. Avantages du clustering

Parmi les avantages du clustering on peut citer[11] :

- Une gestion efficace de la mobilité.

- Un meilleur routage à cause de la minimisation de la taille des tables de routage de chaque nœud (chaque nœud garde une portion du réseau globale) et par conséquence minimisation de l'overhead dans les paquets des données.
- Une meilleure exploitation de la bande passante du médium d'accès et une préservation importante dans l'énergie du chaque nœud (a longer life time System).
- la Faciliter dans l'agrégation des informations de la topologie grâce au nombre réduit des nœuds dans le cluster.
- Minimisation de l'interférence dans le réseau car seul le cluster head est autorisé à propager les paquets aux autres destinations.

#### **2.5.4. Inconvénients du clustering**

- Pour un grand réseau dynamique (mobilité très fréquente) le nombre des messages échangées dans le réseau durant la formation et de maintenance du cluster sera très important et l'utilisation de la bande passante arrive à un point critique, l'énergie de chaque nœud sera consommée très rapidement[11].
- quand un nouveau leader du cluster est choisit, le processus du reclustering doit être lancé dans tout le réseau[11].
- L'un des inconvénients du clustering c'est la différence de la consommation dans l'énergie des nœuds du cluster, les nœuds cluster heads et les nœuds gateway envoient un nombre important des messages par rapport aux nœuds membres du cluster[11].

#### **2.5.5. Les facteurs d'élection du cluster head**

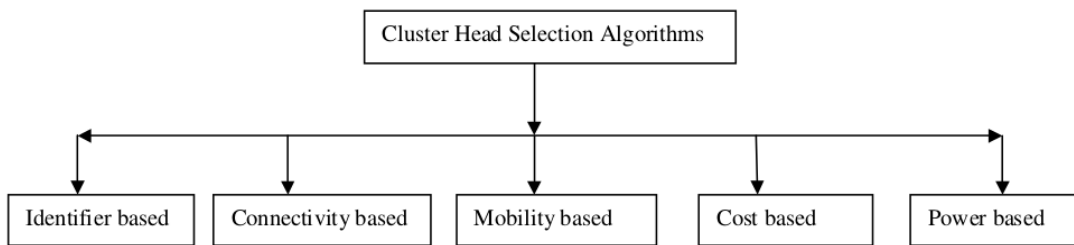
Les algorithmes d'élection du cluster head se concentrent sur la sociabilité du réseau, l'égalité à choisir un certain nœud pour devenir le cluster head, la stabilité du nœud qui est calculée selon plusieurs choix comme la mobilité du nœud et l'énergie du nœud, métriques combinée, etc., on peut citer les facteurs suivants dans le choix d'un cluster head[12] :

- Location géographique du nœud par rapport aux autres nœuds.
- Mobilité.
- Energie.
- Trust.
- Throughput.

#### **2.5.6. Les fameux algorithmes d'élection du cluster head**

Il existe plusieurs algorithmes d'élection du CH dans les réseaux MANET, parmi ces algorithmes on peut citer: lowset id clustering technic, highest degree clustering algorithm, weighted clustering

technic. Chacune des ces techniques a un objectif, en générale les algorithmes du clustering peuvent être classifiés en 5 catégories différentes[14].



*Figure 2.3 Classification des techniques du clustering[14].*

### **2.5.6.1. Identifier based clustering**

Le processus d'élection du CH se base sur l'identificateur du nœud, classifiée en deux fameux algorithmes:

#### **1. Loweset id clustering algorithm**

Dans cette algorithme un ID (identifiant) unique est associée à chaque nœud du cluster et le nœud avec minimum id est élu comme cluster head, les autres nœuds du cluster ont toujours un ID supérieur à celui du cluster head[14].

#### **2. Max-Min d-cluster formation algorithm**

Cette algorithme basée sur la métrique de la taille du cluster, un cluster selon cette approche est formée de tous les nœuds à distance au maximum  $d$  hops du cluster head, comment déterminera la valeur de  $d$  n'est pas spécifiée[15].

### **2.5.6.2. Power (Energy) based clustering**

Le critère de l'énergie est considéré comme le facteur de décision de base dans ces classifications, parmi les algorithmes basés sur cette approche on citer :

#### **1. Power aware connected dominant set**

Le nœud avec maximum énergie level (el) est élu comme cluster head[14].

#### **2. Clustering for energy conservation**

Cette approche suppose l'existence de deux types des nœuds : a master nœud and slave nœud, un nœud esclave a au maximum un seule nœud master, les connections entre les nœuds esclaves est interdit, la surface du cluster est la distance le plus loin entre un nœud master et un nœud esclave[14].

#### **3. Weighted clustering algorithm**

Cette algorithme basée sur des facteurs différents pour éluer un cluster head, il se base sur le nombre des nœuds qu'un cluster head peut gérer et la mobilité de nœud et la force de transmission et l'énergie résiduelle dans la batterie du nœud[14].

## ***2.6. Détection du fusionnement***

- *Quand un nœud appartenant à un cluster rentre dans un nouveau cluster, il va recevoir un message de découverte par le cluster head du nouveau cluster.*
- *Dans ce cas, ce nœud doit envoyer deux messages, l'un vers son cluster head contenant l'identité du nouveau cluster head, et l'autre vers le cluster head du nouveau cluster contenant l'identité de l'autre cluster head.*
- *La réception de ces messages par les cluster head lui permettent de se connaître pour prendre une décision.*

## ***2.7. L'algorithme proposé***

*La question principale qui se pose est la suivante: quel est le nœud qui va jouer le rôle d'un leader quand plusieurs clusters seront regroupés?*

*On a proposé pour cela un mécanisme basé sur le critère de l'énergie, le cluster head ayant le maximum d'énergie parmi les anciens clusters head sera choisi. Si deux ou plusieurs cluster head ont le même niveau d'énergie alors on va choisir le leader ayant le minimum identificateur.*

*Les étapes de l'algorithme pour compléter le processus de formation du cluster sont comme suivantes :*

- *Initialement identifier les clusters head dans la table de routage.*
- *Puis, les anciens clusters head peuvent se connaître, donc ils doivent communiquer entre eux pour désigner le nouveau leader.*
- *Enfin le nouveau leader diffuse cette information à tous les nœuds.*

### 2.7.1. L'algorithme

*Début*

*max* = 0 ;

*pour tout nœud n dans le réseau*

*si((n.energie > max) && (n.lead == 1)) {*

*max = n.energie;*

*leaderID = n.ID;*

*}*

*finpour*

*pour tout nœud n dans le réseau*

*if(n.ID != leaderID) n.leade = 0 ;*

*finpour*

*Fin*

## ***Conclusion***

*Dans ce chapitre nous avons présenté des notions de partage des ressource dans un environnement distribuée en particulier ,puis nous avons présenté l'exclusion mutuelle leur définition et leur mécanisme , puis on a mis les points sur l'un des conséquences de mobilité le partitionnement ,puis nous avons présenté les cluster et les facteurs et les algorithmes d'élection d'un cluster head ,et nous avons également présenté l'un des conséquence de mobilité le fusionnement avec une proposition d'un mécanisme pour lui détecter, puis nous avons présenté un algorithme d'élection d'un cluster head .*

*Dans le chapitre suivant, on va expliquer le code de l'algorithme et la simulation, ainsi la discussion des résultats obtenus.*

# CHAPITRE 3

## 3. Simulation et Analyse des Résultats

### Sommaire

<u>3.1.Introduction.....</u>	<u>34</u>
<u>3.2.Le simulateur ns-2.....</u>	<u>34</u>
<u>3.2.1.Définition.....</u>	<u>34</u>
<u>3.2.2.Présentation du simulateur ns-2.....</u>	<u>34</u>
<u>3.2.3.Les phases de simulation.....</u>	<u>34</u>
<u>3.3.Les paramètres de simulation.....</u>	<u>35</u>
<u>3.3.1.Les paramètres fixes.....</u>	<u>35</u>
<u>3.3.2.Les paramètres variables.....</u>	<u>35</u>
<u>3.4.Code de l'algorithme.....</u>	<u>36</u>
<u>3.4.1.Les variables.....</u>	<u>36</u>
<u>3.4.2.Les messages utilisés.....</u>	<u>36</u>
<u>3.5.Le protocole de routage utilisé.....</u>	<u>36</u>
<u>3.6.Les procédures d'algorithme.....</u>	<u>36</u>
<u>3.6.1.Procédures d'initialisation.....</u>	<u>36</u>
<u>3.6.2.Procédure du découverte.....</u>	<u>38</u>
<u>3.6.3.Procédure de choix.....</u>	<u>38</u>
<u>3.6.4.Procédure de Mise à Jour.....</u>	<u>39</u>
<u>3.7.Scénario de simulation.....</u>	<u>40</u>
<u>3.8.Résultats de simulation.....</u>	<u>40</u>
<u>Conclusion.....</u>	<u>41</u>

### **3.1. Introduction**

Pour tester l'algorithme proposé et pour satisfaire leur insuffisance nous allons l'implémenter, et cela à l'aide d'un outil de simulation, Nous présentons dans ce chapitre le code de l'algorithme et les résultats des simulations effectuées en générant des scénarios proposés.

### **3.2. Le simulateur ns-2**

#### **3.2.1. Définition**

NS est un outil logiciel de simulation de réseaux informatique, il est parmi les simulateurs les plus utilisés dans les laboratoires de recherche, Afin de simplifier et étudier les performances des protocoles réseau.il offre une plateforme de développement de nouveaux protocoles et permet de les tester.

#### **3.2.2. Présentation du simulateur ns-2**

Il est développé lors d'un projet de la DARPA. Il est principalement bâti avec l'idée de la conception par objet, de réutilisabilité du code et de modularité. Il est devenu aujourd'hui un standard de référence en ce domaine. C'est un logiciel dans le domaine public disponible sur l'internet. Son utilisation est gratuite. Le simulateur se compose d'une interface de programmation en Tcl et d'un noyau écrit en C++ dans lequel la plupart des protocoles réseaux ont été implémentés :

- Application : Web, ftp, Telnet, générateur de trafic (CBR, ..).
- Transport : TCP, UDP, RTP, SRM.
- Routage unicast : Statique, dynamique (vecteur distance).
- Routage multicast : DVMRP, PIM.
- Gestion de fil d'attente : RED, DropTail, Token bucket.

#### **3.2.3. Les phases de simulation**

La simulation en NS-2 passe par quatre phases :

##### **1. L'écriture d'un script en tcl :**

Durant cette phase, nous allons créer le script principal en TCL, ce script contient le code de génération du trafic sur le réseau avec le mouvement des nœuds dans le réseau. Ce fichier constitue un scénario de simulation.

## **2. La création des fichiers (.h et .cc) :**

Durant cette phase nous allons créer deux fichiers l'un de l'extension .h et l'autre de l'extension .cc, le premier est un fichier d'en-tête qui contient la structure des messages échangées entre les nœuds, le deuxième contient les fonctions nécessaires de l'algorithme (envoi et réception des messages, ...).

## **3. La visualisation des résultats :**

Dans cette phase Nous allons récupérer le fichier de trace NS2 et on extrait les résultats que nous voulons visualiser ou interpréter.

## **4. Exploitation des résultats :**

L'exploitation des résultats consiste à utiliser ces résultats pour tracer des courbes ou bien effectuer d'autre calculs en faisant appel aux outils qui l'accompagnent (NS-2) ou bien avec d'autres programmes.

### **3.3. Les paramètres de simulation**

#### **3.3.1. Les paramètres fixes**

- **Surface de Réseau:** Nous avons choisi une surface de 1500m\*1500m.
- **Protocole de Routage :** Le protocole utilisé est le Protocole réactif AODV car il limite l'entête des paquets (overhead).
- **Modèle de Propagation :** Nous avons utilisé le modèle de propagation Two-ray-ground dans notre simulation, ce modèle est devenu un standard dans la recherche sur les réseaux mobiles.
- **Modèle de Mobilité :** Nous avons choisi le modèle Random-WayPoint, dans ce modèle, les nœuds sont distribués uniformément dans l'espace de simulation, leurs positions initiales sont aléatoires ainsi que leurs déplacements.
- **La Portée de Communication :** ce paramètre prend la valeur 200 mètre.

#### **3.3.2. Les paramètres variables**

- **L'Energie:** variation du niveau d'énergie de nœud.
- **La Vitesse :** variation dans la vitesse entre 2m/s et 5m/s.

### **3.4. Code de l'algorithme**

#### **3.4.1. Les variables**

- **Leader** : une variable locale booléenne initialisée à **Faux**, indiquant si le nœud est un cluster head ou non.
- **L'énergie** : une variable locale indique le niveau d'énergie initialisée à 100 joules dans tous les nœuds.

#### **3.4.2. Les messages utilisés**

1. **Le message de découverte**: est un message envoyé par le nœud qui a détecté un fusionnement vers les cluster head.
2. **Le message de décision**: est un message envoyé par chacun des cluster head vers l'autre, ce message contient le niveau d'énergie afin de décider après une comparaison local avec son propre niveau d'énergie quelle sera le cluster head de nouveau réseau fusionnée.
3. **Le message de mise à jour** : est un message envoyé par le cluster head sélectionné vers tous les nœuds pour les informer qui est le cluster head sélectionnée.

### **3.5. Le protocole de routage utilisé**

Le protocole utilisée est le protocole **AODV** (Ad hoc On-demand Distance Vector) est un protocole réactif appartenant à la famille des protocoles On-Demand, il est basé sur deux mécanismes, la découverte de route et la maintenance de route.

### **3.6. Les procédures d'algorithme**

#### **3.6.1. Procédures d'initialisation**

Ces procédures indiquent les initialisations réalisées aux niveaux des différents sites : les détenteurs des jetons, les détenteurs des ressources, les leaders et les sites simples.

*Initialisation: des variables de sites simple*

*Début*

$\mathcal{AODV}[i, jpres] \leftarrow 0;$

$\mathcal{AODV}[i, rpres] \leftarrow 0;$

$\mathcal{AODV}[i, sitlead] \leftarrow 0;$

*Fin*

\*\*\*\*\*

*Initialisation: site détient un jeton*

*Début*

$\mathcal{AODV}[i, jpres] \leftarrow 1;$

*Fin*

\*\*\*\*\*

*Initialisation: site détient une ressource*

*Début*

$\mathcal{AODV}[i, rpres] \leftarrow 1;$

*Fin*

\*\*\*\*\*

*Initialisation: de site Leader*

*Début*

$Leader_i \leftarrow \text{vrai};$

$AODV[i, sitlead] \leftarrow 1;$

*Fin*

### **3.6.2. Procédure du découverte**

*Est une procédure lancée par le nœud intermédiaire quand il reçoit un message de hello (Hello Msg) par un nœud cluster head d'un cluster différent pour que les nœuds cluster head peuvent se connaître et peuvent échanger des messages entre eux.*

**Procédure N°1 : Découvert ()**

**Début**

*Lors de réception hello Msg par le nœud intermédiaire  $k$  par un nœud cluster head  $i$*

*Si ( $AODV[i, sitlead] = 1$ ) alors*

*Envoyer  $Msg(ID_j)$  à  $i$*

*Envoyer  $Msg(ID_i)$  à  $j$*

*Fin si*

*Fin*

### **3.6.3. Procédure de choix**

*Est une procédure lancée par les nœuds cluster head lorsque chacun de ces cluster head reçoit la valeur d'énergie de l'autre, afin de faire le choix entre les cluster head, ensuite le choix sera par la comparaison entre l'énergie lequel entre eux a le plus d'énergie, donc il sera choisi comme un cluster head, si il y a une égalité la comparaison sera par rapport aux identificateurs( $ID$ ) le nœud qui a le minimum identificateur( $ID$ ) sera choisi comme un cluster head.*

*Procédure N°2 : choix ()*

*Début*

*Si ( $\mathcal{AODV}[i, \text{energy}] > \mathcal{AODV}[j, \text{energy}]$ ) alors*

*Leader<sub>j</sub> ← Faux;*

*Sinon*

*Si ( $\mathcal{AODV}[i, \text{energy}] < \mathcal{AODV}[j, \text{energy}]$ ) alors*

*Leader<sub>i</sub> ← Faux;*

*Sinon*

*Si ( $\mathcal{AODV}[i, ID] < \mathcal{AODV}[j, ID]$ ) alors*

*Leader<sub>j</sub> ← Faux;*

*Sinon*

*Leader<sub>i</sub> ← FAUX;*

*Fin si*

*Fin*

### **3.6.4. Procédure de Mise à Jour**

*Est un procédure de mise à jour pour informer tous les nœuds et procéder des modifications sur la table de routage, ces modifications sont faites au niveau du champ sitlead.*

*Procédure mise\_a\_jour ()*

*Début*

*Lors de réception un message par le cluster head i 'je suis le cluster head'*

*Pour tous les nœuds dans le réseau faire :*

*AODV [i, sitlead] =1;*

*Fin*

### **3.7. Scénario de simulation**

*Pour étudier la performance de l'algorithme proposé, nous, et pour spécifier les paramètres qui influent sur la performance de notre algorithme, avons établi des scénarios.*

➤ *Scénario 1 : variation du nombre de cluster*

*Dans ce scénario nous avons varié le nombre de cluster entre (2 à 6).*

<i>Paramètre</i>	<i>Nombre de clusters</i>
<i>Valeur</i>	<i>2 à 6</i>

*Figure 3.1 Scénario de simulation.*

➤ *Scénario 2 : variation du nombre de nœud*

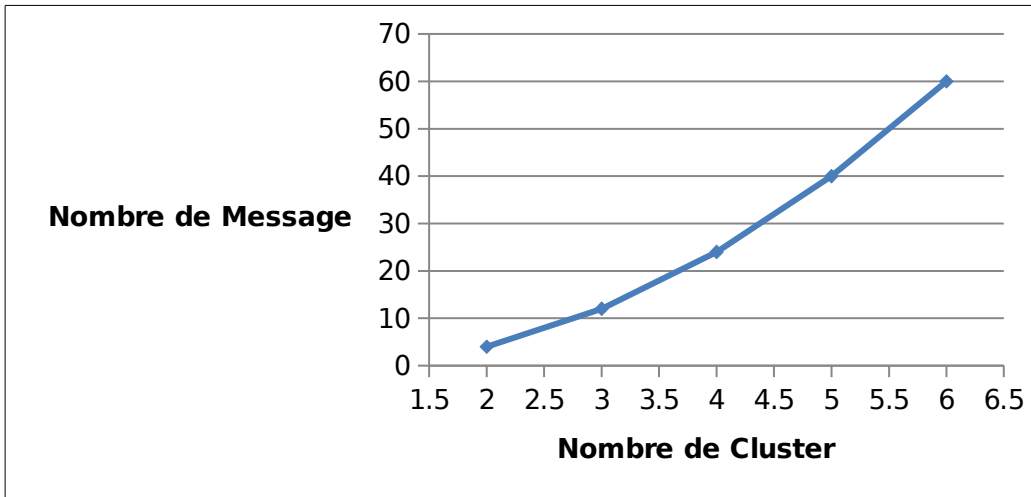
*Dans ce scénario nous avons varié le nombre de nœud.*

<i>Paramètre</i>	<i>Nombre de nœuds</i>
<i>Valeur</i>	<i>10 à 50</i>

*Figure 3.2 Scénario de simulation.*

### 3.8. Résultats de simulation

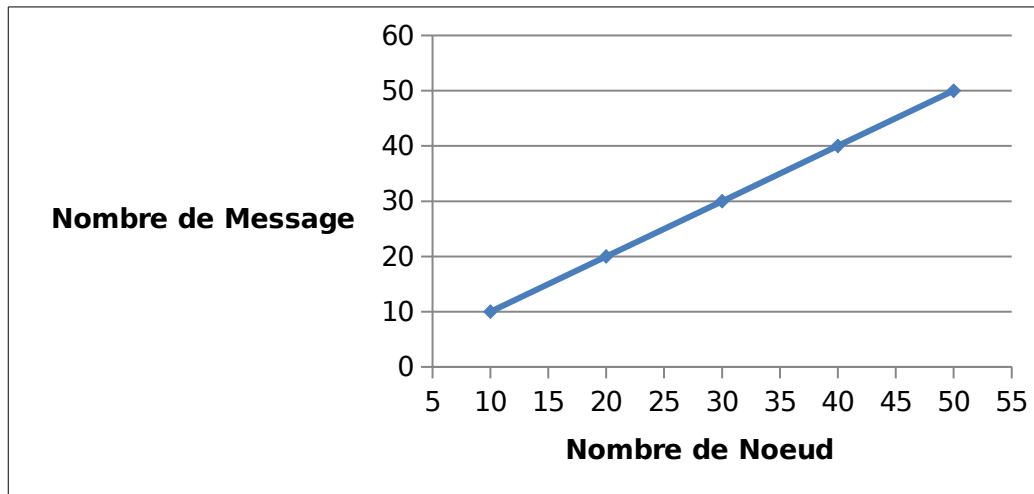
- *Variation de nombre de cluster :*



*Figure 3.3 Influence du nombre de clusters.*

Dans la figure ci-dessus on remarque une augmentation constante du nombre de message avec l'augmentation du nombre de cluster, cela peut être justifié par l'utilisation de messages de découvert qui doit être envoyé par le nœud intermédiaire vers les nœuds cluster head et l'utilisation de message de décision qui envoier par chaque nœud cluster head vers l' autre.

- *Variation du nombre de nœuds :*



*Figure 3.4 Influence du nombre de nœuds.*

*Dans cette figure on remarque une augmentation constante du nombre de message avec l'augmentation de nombre de nœuds. Cela peut être justifié par l'utilisation de message de mise à jour qui doit être envoyer par le nœud cluster head sélectionner à tous les nœuds de réseau fusionner.*

## ***Conclusion***

*dans ce chapitre nous avons présenté le code de l'algorithme proposé et les résultats de simulation de notre algorithme.*

*Les courbes obtenues montrent un comportement logique de notre algorithme qui dépend essentiellement du nombre de nœuds et nombre de cluster,*

## ***Conclusion Générale***

*Dans ce mémoire nous avons étudié le problème de l'influence de la mobilité sur les algorithmes de partage des ressources dans les réseaux ad hoc.*

*Nous avons commencé par la présentation de la généralité sur les systèmes distribués, leurs caractéristiques et leurs buts. Puis nous avons présenté les réseaux mobiles ad hoc, leurs avantages et leurs inconvénients. Puis nous avons abordé l'objet de notre étude : le phénomène de fusionnement par la description d'un algorithme proposée et cet algorithme a été validé par une simulation via l'outil de simulation NS-2 et nous avons discuté ces résultats qui montrent le comportement logique de la solution.*

*Ce travail nous a permis de.*

*Étudier et comprendre le problème de partage des ressources.*

*Avoir une idée sur l'état de fusionnement.*

*Se familiariser avec l'outil NS-2.*

## **Bibliographie**

### **Bibliography**

- [1]: Mr.Boukhechem Nadhir, *Routage Dans Les Réseaux Mobiles Ad-Hoc par Une Approche a Base d'Agent*, 2008
- [2]: Tanenbaum, ,
- [3]: George Coulouris , Jean Dollimore , Tim Kindberg , Gordon Blair, , 2012
- [4]: Diganta Goswami, *CS542: Topics in Distributed Systems*, ,
- [5]: Shalini Kapoor , Poonam Kashyap, *Resource Management In Distributed System*,
- [6]: , *Les communication inter-processus*, , <http://www.courstechinfo.be/OS/>
- [7]: Eric Cariou, *Algorithmique distribuée Exclusion mutuelle*, ,
- [8]: Hartmut Ritter, Rolf Winter, Jochen Schiller, *A Partition Detection System for Mobile Ad-Hoc Networks*,
- [9]: Bratislav Milic, Nikola Milanovic, Mirosław Malek, *Prediction of Partitioning in Location-aware Mobile Ad Hoc Networks* ,
- [10]: Arathy Nair , Mr.S.Kannan , Dr.S.Karthik, *A Methodology for Cluster Head Selection to Improve Throughput and Channel Space Utilization in Power Heterogeneous MANET*, 2014
- [11]: Wojciech Bednarczyk , Piotr Gajewski, *An Enhanced Algorithm for MANET Clustering Based on Weighted Parameters*, 2013
- [12]: Mandeep Singh , Mr.Gagangeet Singh, *A Secure and Efficient Cluster Head Selection Algorithm for MANET*, 2015
- [13]: V.Preetha, Dr.K.Chitra, *Clustering a Cluster Head Selection Techniques in Mobile Adhoc Networks*, 2014
- [14]: Soumyabrata Talapatra , Alak Roy, *Mobility Based Cluster Head Selection Algorithm for Mobile Ad-Hoc Network*, 2014

## Annexe

# Script de Simulation

*# Définition des Options*

```
#-----  
set val(chan) Channel/WirelessChannel ;# channel type  
set val(prop) Propagation/TwoRayGround ;# radio-propagation model  
set val(ant) Antenna/OmniAntenna ;# Antenna type  
set val(ll) LL ;# Link layer type  
set val(ifq) Queue/DropTail/PriQueue ;# Interface queue type  
set val(ifqlen) 50 ;# max packet in ifq  
set val(netif) Phy/WirelessPhy ;# network interface type for range of 200m  
set val(mac) Mac/802_11 ;# MAC type  
set val(rp) AODV ;# ad-hoc routing protocol  
set val(nn) 20 ;# number of mobilenodes  
set val( $\chi$ ) 1500  
set val(y) 1500  
set val(energy) EnergyModel ;#Energy set up  
set initialEnergy 10 ;# Energy Initiale
```

*# la configuration des antennes*

```
#-----  
$val(ant) set X_ 0  
$val(ant) set Y_ 0  
$val(ant) set Z_ 1.5  
$val(ant) set Gt_ 1.0  
$val(ant) set Gr_ 1.0
```

*# la configuration dans la portée de signal en 200 metres*

```
#-----  
# Phy/WirelessPhy set CPTthresh_ 10.0  
# Phy/WirelessPhy set Rb_ 2*1e6  
# Phy/WirelessPhy set Pt_ 0.2818  
# Phy/WirelessPhy set freq_ 914e+6  
# Phy/WirelessPhy set L_ 1.0  
# Phy/WirelessPhy set CSTthresh_ [expr pow ($ $\chi$ , -8)] # sensing range of 200
```

```

Phy/WirelessPhy set RXThresh_ 7.69113e-08
# -----
# creation d'instance de simulateur
# -----
set ns [new Simulator]

#$ns color 1 Blue
#$ns color 2 Red

# création des fichiers de trace
# -----
set tracefd [open wireless.tr w]
$ns trace-all $tracefd
set namtrace [open wireless.nam w]
$ns namtrace-all-wireless $namtrace $val(x) $val(y)
# -----
# la définition du procedure 'finish'
# -----
proc finish {} {
global ns tracefd namtrace
$ns flush-trace
close $tracefd
close $namtrace
exec nam wireless.nam &
exit 0
}
# Définition de la Topographie
# -----
set topo [new Topography]
$topo load_flatgrid 1500 1500
# -----
# creation d'objet GOD
# -----
create-god $val(nn)
# -----
# Configuration globale des noeudes
# -----
$ns node-config -adhocRouting $val(rp) \
-llType $val(ll) \
-macType $val(mac) \

```

```

-ifqType $val(ifq) \
-ifqLen $val(ifqlen) \
-antType $val(ant) \
-propType $val(prop) \
-phyType $val(netif) \
-topoInstance $topo \
-energyModele $val(energy) \
                                -rxPower 0.5 \
                                -txPower 1.0 \

-idlePower 0.5 \
-sleepPower 0.001 \
-transitionPower 0.2 \
-channelType $val(chan) \
-agentTrace ON \
-routerTrace ON \
-macTrace OFF \
-movementTrace OFF \

# Création des noeuds
#-----
for {set i 0} {$i < $val(nn)} {incr i} {
set energy($i) [expr rand() * 100]
$ns node-config -initialEnergy $energy($i)
set node($i) [$ns node ]
$node($i) random-motion 0
}
#-----
# Positions initiales des noeuds
#-----
$node(0) set X_ 200.0
$node(0) set Y_ 380.0
$node(0) set Z_ 0.0
$node(0) color Blue
$ns at 0.0 "$node(0) color Blue"

$node(1) set X_ 220.0
$node(1) set Y_ 300.0
$node(1) set Z_ 0.0
$node(1) color Blue
$ns at 0.0 "$node(1) color Blue"

```

*\$node(2) set X\_ 40.0*  
*\$node(2) set Y\_ 350.0*  
*\$node(2) set Z\_ 0.0*  
*\$node(2) color Blue*  
*\$ns at 0.0 "\$node(2) color Blue"*

*\$node(3) set X\_ 100.0*  
*\$node(3) set Y\_ 200.0*  
*\$node(3) set Z\_ 0.0*  
*\$node(3) color Blue*  
*\$ns at 0.0 "\$node(3) color Blue"*

*\$node(4) set X\_ 200.0*  
*\$node(4) set Y\_ 100.0*  
*\$node(4) set Z\_ 0.0*  
*\$node(4) color Blue*  
*\$ns at 0.0 "\$node(4) color Blue"*

*\$node(5) set X\_ 250.0*  
*\$node(5) set Y\_ 200.0*  
*\$node(5) set Z\_ 0.0*  
*\$ns at 0.00 "\$node(5) label \"Cluster Head\""*  
*\$node(5) color Blue*  
*\$ns at 0.0 "\$node(5) color Blue"*

*\$node(6) set X\_ 15.0*  
*\$node(6) set Y\_ 110.0*  
*\$node(6) set Z\_ 0.0*  
*\$node(6) color Blue*  
*\$ns at 0.0 "\$node(6) color Blue"*

*\$node(7) set X\_ 100.0*  
*\$node(7) set Y\_ 300.0*  
*\$node(7) set Z\_ 0.0*  
*\$node(7) color Blue*  
*\$ns at 0.0 "\$node(7) color Blue"*

*\$node(8) set X\_ 180.0*

*\$node(8) set  $\mathcal{Y}_-$  180.0*  
*\$node(8) set  $\mathcal{Z}_-$  0.0*  
*\$node(8) color Blue*  
*\$ns at 0.0 "\$node(8) color Blue"*

*\$node(9) set  $X_-$  15.0*  
*\$node(9) set  $\mathcal{Y}_-$  280.0*  
*\$node(9) set  $\mathcal{Z}_-$  0.0*  
*\$node(9) color Blue*  
*\$ns at 0.0 "\$node(9) color Blue"*

*\$node(10) set  $X_-$  11.0*  
*\$node(10) set  $\mathcal{Y}_-$  200.0*  
*\$node(10) set  $\mathcal{Z}_-$  0.0*  
*\$node(10) color Blue*  
*\$ns at 0.0 "\$node(10) color Blue"*

*\$node(11) set  $X_-$  80.0*  
*\$node(11) set  $\mathcal{Y}_-$  80.0*  
*\$node(11) set  $\mathcal{Z}_-$  0.0*  
*\$node(11) color Blue*  
*\$ns at 0.0 "\$node(11) color Blue"*

*\$node(12) set  $X_-$  150.0*  
*\$node(12) set  $\mathcal{Y}_-$  250.0*  
*\$node(12) set  $\mathcal{Z}_-$  0.0*  
*\$node(12) color Blue*  
*\$ns at 0.0 "\$node(12) color Blue"*

*\$node(13) set  $X_-$  1050.0*  
*\$node(13) set  $\mathcal{Y}_-$  180.0*  
*\$node(13) set  $\mathcal{Z}_-$  0.0*  
*\$node(13) color Red*  
*\$ns at 0.0 "\$node(13) color Red"*

*\$node(14) set  $X_-$  1100.0*  
*\$node(14) set  $\mathcal{Y}_-$  350.0*  
*\$node(14) set  $\mathcal{Z}_-$  0.0*  
*\$node(14) color Red*  
*\$ns at 0.0 "\$node(14) color Red"*

```
$node(15) set X_ 1100.0  
$node(15) set Y_ 150.0  
$node(15) set Z_ 0.0  
$node(15) color Red  
$ns at 0.0 "$node(15) color Red"
```

```
$node(16) set X_ 1100.0  
$node(16) set Y_ 220.0  
$node(16) set Z_ 0.0  
$node(16) color Red  
$ns at 0.0 "$node(16) color Red"
```

```
$node(17) set X_ 1150.0  
$node(17) set Y_ 280.0  
$node(17) set Z_ 0.0  
$node(17) color Red  
$ns at 0.00 "$node(17) label \"Cluster Head\""  
$ns at 0.0 "$node(17) color Red"
```

```
$node(18) set X_ 1050.0  
$node(18) set Y_ 320.0  
$node(18) set Z_ 0.0  
$node(18) color Red  
$ns at 0.0 "$node(18) color Red"
```

```
$node(19) set X_ 1000.0  
$node(19) set Y_ 250.0  
$node(19) set Z_ 0.0  
$node(19) color Red  
$ns at 0.0 "$node(19) color Red"
```

```
# initialiser les noeuds avec leurs positions
```

```
# -----  
for {set i 0} {$i < $val(nn)} {incr i} {  
  $ns initial_node_pos $node($i) 30  
}
```

```
# changement dans les positions des noeuds pour obtenir un réseau fusionée
```

```

#-----
$ns at 0.0 "$node(0) setdest 700.0 380.0 500.0 "
$ns at 0.0 "$node(1) setdest 800.0 220.0 500.0"
$ns at 0.0 "$node(2) setdest 800.0 350.0 500.0"
$ns at 0.0 "$node(3) setdest 720.0 200.0 500.0"
$ns at 0.0 "$node(4) setdest 880.0 100.0 500.0"
$ns at 0.0 "$node(5) setdest 600.0 200.0 500.0"
$ns at 0.0 "$node(6) setdest 580.0 110.0 500.0"
$ns at 0.0 "$node(7) setdest 750.0 300.0 500.0"
$ns at 0.0 "$node(8) setdest 520.0 180.0 500.0"
$ns at 0.0 "$node(9) setdest 850.0 280.0 500.0"
$ns at 0.0 "$node(10) setdest 690.0 150.0 500.0"
$ns at 0.0 "$node(11) setdest 740.0 60.0 500.0"
$ns at 0.0 "$node(12) setdest 450.0 250.0 500.0"
#####
$ns at 0.0 "$node(13) setdest 550.0 350.0 500.0"
$ns at 0.0 "$node(14) setdest 650.0 380.0 500.0"
$ns at 0.0 "$node(15) setdest 650.0 60.0 500.0"
$ns at 0.0 "$node(16) setdest 880.0 150.0 500.0"
$ns at 0.0 "$node(17) setdest 550.0 300.0 500.0"
$ns at 0.0 "$node(18) setdest 800.0 290.0 500.0"
$ns at 0.0 "$node(19) setdest 650.0 250.0 500.0"

#-----
# informer les noeuds quand la simulation se termine
#-----
for {set i 0} {$i < $val(nn)} {incr i} {
$ns at 6.0 "$node($i) reset";
}
# Finalisation de Simulation
#-----
$ns at 6.01 "finish"
$ns at 6.01 "puts \"NS EXITING...\"; $ns halt"

#-----
# Exécution de la Simulation
puts "Starting Simulation..."
$ns run

```

## ***Acronyme***

<b><i>AODV</i></b>	<i>Ad hoc On-Demand Distance Vector</i>
<b><i>DSR</i></b>	<i>Dynamic Source Routing</i>
<b><i>EM</i></b>	<i>Exclusion Mutuelle</i>
<b><i>IEEE</i></b>	<i>Institute of Electrical and Electronics Engineers</i>
<b><i>IP</i></b>	<i>Internet Protocol</i>
<b><i>K-EM</i></b>	<i>K-Exclusion Mutuelle</i>
<b><i>MAC</i></b>	<i>Medium Access Control</i>
<b><i>MANET</i></b>	<i>Mobile Ad-hoc Networks</i>
<b><i>MSG</i></b>	<i>Message</i>
<b><i>NbJeton</i></b>	<i>Nombre de Jeton</i>
<b><i>NAM</i></b>	<i>Network AniMator</i>
<b><i>NMM</i></b>	<i>Nombre de Messages Moyen</i>
<b><i>NS-2</i></b>	<i>Network Simulator 2</i>
<b><i>OLSR</i></b>	<i>Optimized Link State Routing protocol</i>
<b><i>RDM</i></b>	<i>Random Direction Model</i>
<b><i>RED</i></b>	<i>Random Early Detection</i>
<b><i>RWM</i></b>	<i>Random Waypoint Model</i>
<b><i>SB</i></b>	<i>Station de Base</i>
<b><i>SC</i></b>	<i>Section Critique</i>
<b><i>TAM</i></b>	<i>Temps d'Attente Moyen</i>
<b><i>TCL</i></b>	<i>Tools Command Language</i>
<b><i>TCP</i></b>	<i>Transport Control Protocol</i>
<b><i>sitlead</i></b>	<i>Site Leader</i>
<b><i>rpres</i></b>	<i>Ressources Présent</i>
<b><i>jpres</i></b>	<i>Jeton Présent</i>
<b><i>TPATT</i></b>	<i>TemPs d'ATTente</i>
<b><i>UDP</i></b>	<i>User Datagram Protocol</i>
<b><i>UM</i></b>	<i>Unités Mobiles</i>
<b><i>ZRP</i></b>	<i>Zone Routing Protocol</i>