REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOHAMED KHEIDER –BISKRA

FACULTÉ DES SCIENCES EXACTES ET DES SCIENCES DE LA NATURE ET DE LA VIE

DÉPARTEMENT D'INFORMATIQUE

MEMOIRE

Présenté pour l'obtention du diplôme
de Magister en informatique

Option : Intelligence Artificielle & Système Distribuée

Une Approche Agent Mobile pour la QoS dans les Réseaux Mobiles Ad hoc

Proposé par : Dr. Okba Kazar

Réalisé par : Maqbol Ahmed

Soutenu le / 06 / 2010 devant le jury composé de:

Pr. Djeddi Noureddine	**Prof. Université de Biskra**	**Président**
Dr. Kazar Okba	**M.C Université de Biskra**	**Rapporteur**
Dr. Bilami Azzedine	**M.C Université de Batna**	**Examinateur**
Dr. Kholladi M^{ed} Kheireddine	**M.C Université de Constantine**	**Examinateur**

Année Universitaire : 2009/2010

Remerciements

Tout d'abord, je tiens à remercier Dieu tout puissant qui m'a donné la volonté, la patience et la santé pour élaborer ce travail.

Ensuite, mes premiers remerciements iront à mon encadreur Dr. Kazar Okba, maître de conférence à l'Université Mohamed Kheider de Biskra, pour m'avoir soutenu durant tout mon mémoire. J'aimerais lui adresser toute ma profonde gratitude et je le remercie du fond du cœur pour toutes ses remarques et suggestions techniques, académiques et professionnelles. Il a toujours su me consacrer des moments de son temps, me guider, me conseiller, et me témoigner son soutien et sa confiance. Je n'oublierai jamais sa gentillesse et sa bonne humeur avec moi.

Je remercie particulièrement Djeddi Noureddine, professeur à l'université de Biskra, qui ma fait l'honneur de présider ce jury. Je remercie également très fort le docteur Bilami Azzedine maître de conférences à l'université de Batna et le docteur Kholladi Mohamed Kheireddine maître de conférences à l'université de Constantine, d'avoir accepté de juger ce travail.

Je voudrais témoigner à chef de département d'informatique Dr. Mohamed Chaouki Babahenini, ma profonde gratitude pour m'avoir faciliter toutes les procédures administratives tout au long des trois années d'études.

Mes prochains remerciements ont une portée de 7000 Km. Ils s'adressent à tous les membres de ma famille résidente au Yemen. Je voudrais leur exprimer toute ma profonde gratitude parce qu'ils m'ont constamment aidé, malgré la distance pour la plupart, par leur soutien moral et leurs encouragements pour achever mon travail.

Une attention toute spéciale s'adresse aux membres de l'équipe IASD pour leur gentillesse et leur bonne humeur. Merci donc à Abdel basset, Soufiane, Amal, Manal, Djaber, Fawzi, Salima et Hanan.

Finalement, je garde une place toute particulière pour mes amis, Mouadh, Al Moutawakil, Al Sharafi, Fathi, Farouk et Arafat. Je voudrais leur exprimer toute ma profonde reconnaissance parce qu'ils m'ont constamment aidé et m'avoir créer des conditions appropriées pour achever ce mémoire.

Résumé

Les réseaux mobiles ad hoc sont des réseaux auto organisés qui ne s'appuient sur aucune infrastructure définie au préalable pour communiquer. Les appareils appartenant à de tels réseaux sont dotés d'une portée de transmission limitée et les paquets de données peuvent traverser de multiples autres appareils avant d'arriver à leur destination.

Par ailleurs, elles sont généralement de petites taille (PC portable, PDA,..) d'où les contraintes de ressources en terme de mémoire et de batteries. Ainsi que, le changement fréquent de topologie de réseau est à relever dans la conception de modèle et/ou de protocole de routage pour améliorer la qualité de service (QoS).

Pour répondre aux problèmes de qualité de service dans les réseaux mobiles ad hoc et au besoin d'adaptation dynamique, nous proposons un modèle à base d'agent mobile pour une gestion hiérarchique et flexible de la QoS. Le modèle adopte sur l'organisation du réseau à trois niveaux (niveau nœud, niveau groupe, niveau réseau) où la gestion de la qualité de service est assurée par la coopération des trois niveaux. Le niveau nœud permet de gérer et d'évaluer les ressources du nœud (l'énergie, la charge de CPU et de Mémoire, le degré de nœud, la bande passante, la moyenne de mobilité et la sécurité). Le niveau de groupe assure les interactions entre les nœuds à l'intérieur du même groupe pour gérer localement les différentes fonctions. L'obtention d'une meilleure QoS au niveau de groupe ou terminal ne signifie obligatoirement pas une meilleure QoS au niveau du réseau, donc, il faut mettre des mécanismes adéquats pour gérer les interactions entre les différents groupes. L'agent mobile est l'un de ces mécanismes important qui peuvent être utilisé pour l'actualisation des routes, la réduction de l'overhead engendré par l'échange des messages de contrôle et peut offrir un meilleur rendement.

Ainsi, nous employons notre modèle pour proposer un nouveau protocole de routage appelons PRAM (Protocole de Routage à base d'Agent Mobile) afin d'améliorer la qualité de service pour les réseaux ad hoc. La réalisation de modèle proposé sous la plate-forme d'agent mobile Aglet a été conduite de montrer l'efficacité d'agent mobile pour la qualité de service dans les réseaux mobile ad hoc.

Mots clés

QoS, réseaux mobile ad hoc, agent mobile, protocole de routage avec QoS, cluster, composant.

Abstract

The mobile ad hoc networks are organized networks car which are not based on any infrastructure defined as a preliminary to communicate. The devices belonging to such networks are equipped with a range of limited transmission and the packages of data can cross different multiples apparatuses before arriving at their destination.

In addition, they are generally of small cuts (portable PC, PDA.) from where constraints of resources term of memory and batteries. The additional challenges such as the frequent change of topology and the limited resources of network are to be raised in the design of model and/or protocol of routing to improve quality of service (QoS).

To meet the problem of quality of service in the ad hoc mobile networks and the need for adaptation dynamic, we propose a model containing mobile agent for a hierarchical and flexible management of QoS. The model adopts on the organization of the network on three levels (level node, level group, and level network) where the management of quality of service is ensured by the co-operation of the three levels. The level node makes it possible to manage and evaluate the resources of the node (energy, the load of CPU and Memoir, the degree of node, the band-width, and the average of mobility). The level of group ensures the interactions between the nodes inside the same group to manage the various functions locally. Obtaining better QoS on the level of group or terminal does not mean obligatorily better QoS on the level of the network, therefore, it is necessary to put adequate mechanisms to manage the interactions between the various groups. The mobile agent is one of these mechanisms important which can be used for the actualization of the roads, the reduction of the overhead generated by the exchange of the messages of control and can offer a better output.

Thus, we employ our model to propose a new protocol of routing invite PRAM (Protocol of Routing containing Mobile Agent) in order to improve quality of service for the ad hoc networks. The realization of model suggested under the platform of mobile agent Aglet was led to show the mobile agent effectiveness for quality of service in the mobile ad hoc networks.

Keywords

QoS, mobile ad hoc networks, mobile agent, protocol of routing with QoS, cluster, component.

ملخص

الشبكات المتحركة ad hoc هي عبارة عن شبكات تقوم بتنظيم نفسها ذاتيا ولا تعتمد علي أي هيكلة أو إدارة من اجل الاتصـــال فيما بين العقد و الأجــهزة التي تنتمي إلي هذه الشبكـــات تكون ذات مـجال إرسـال محدود والمعطيات ترسل عن طريق أجهزة أخري قبل أن تصل إلي المستقبل في نفس الشبكة.

من جهة أخري ، هذه الأجهزة تكون عموما صغيرة الحجم (مثل الكمبيوترات المحمولة ، أجهزة PDA) وبالتالي هناك محدودية في الذاكرة والبطارية. هذه المحدوديات بالإضافة إلي التغيير المستمر لوضعيات العقد توخذ بعين الاعتبار أثناء ما نضع نموذج أو برتوكول الاتصال بين العقد وذلك من اجل تحسين نوعية الخدمات (Qualité de service, QoS).

من اجل وضع حلول لمشكلة تحسين الخدمات في هذه الشبكات واحتياجنا للتكيف الديناميكي، سنضع نموذج يعتمد في الأساس علي العامل المتحرك (Agent Mobile, Mobile Agent) وذلك بغرض التسيير التسلسلي والمرن لتحسين نوعية الخدمات. هذا النموذج يعتمد علي تنظيم الشبكة إلي ثلاثة مستويات (علي مستوي العقدة، علي مستوي المجموعة، علي مستوي الشبكة) بحيث تسير تحسين الخدمات تكون محققة من خلال التعاون بين هذه المستويات. تحسين الخدمات علي مستوي العقدة يسمح بتسيير وتقيم موارد العقدة(الطاقة، الذاكرة والمعالج، رتبة العقدة، شريط المرور، متوسط الحركة). بينما علي مستوي المجموعة نضمن التفاعل أو التواصل بين العقد داخل المجموعة. من المهم أن نشير إلي انه حصولنا علي خدمات أفضل علي مستوي المجموعة أو علي مستوي العقدة لا يعني بالضرورة خدمات جيدة علي مستوي الشبكة، وعليه يجب أن نضع آليات مناسبة من اجل التواصل أو التفاعل بين مختلف المجموعات.

العامل المتحرك يعتبر واحد من هذه الآليات المهمة الذي يستخدم من اجل تحسين الخدمـــات حيث يقوم بتحديث الطرق، تنقيص الازدحام في الشبكة الناتج عن تبادل الرسائل وغيرها من الخدمات. كما انه نستخدم النموذج السابق من اجـل نضع برتوكول جديد للاتصـــال بين العقد، وهذا البرتوكول نسميه برام (برتوكول للاتصال يعتمد علي اساس العامل المتحرك) من اجل تحسين نوعية الخدمات في الشبكـــات المتحركة.

كلمات مفاتيح

نوعية الخدمات، الشبكات المتحركة ad hoc، برتوكول الاتصـــال مع نوعية الخدمات، المجمـوعـات، المركبات.

Sommaire

Introduction générale **1**

Chapitre 1 : Réseaux Mobiles Ad hoc (MANETs) **04**

1.1 Introduction 4
1.2 L'environnement mobile 5
1.3 Fondements des réseaux mobiles ad hoc 7
1.3.1 Historique et évolution des réseaux ad hoc 7
1.3.2 Qu'est ce que les réseaux mobiles ad hoc 8
1.3.3 Modélisation de réseau mobile ad hoc 9
1.3.4 Caractéristique et défis des réseaux mobiles ad hoc 9
1.4 Routage dans les réseaux mobiles ad hoc 12
1.4.1 Problème de routage dans les réseaux ad hoc 13
1.4.2 Contraintes des protocoles de routage dans MANET 14
1.5 Protocole de routage de réseaux mobiles ad hoc 14
1.5.1 Les protocoles proactifs 14
1.5.1.1 Le protocole DSDV (Destination Sequenced Distance Vector) 15
1.5.1.2 Le protocole OLSR (*Optimized Link State Routing Protocol*) 16
1.5.2 Protocole de routages réactifs 16
1.5.2.1 Le protocole DSR (*Dynamic Source Routing*) 17
1.5.2.2 Le protocole AODV (*Ad hoc On-demand Distance Vector*) 18
1.5.3 Les protocoles hybrides 19
1.5.3.1 Le protocole ZRP (*Zone Routing Protocol*) 19
1.5.4 Évaluation des protocoles de routage 20
1.6 Les applications actuelles et futures des MANETs 21
1.7 Difficultés de déploiement des applications sur MANET 22
1.8 Avantages et inconvénients des réseaux mobiles ad hoc 23
1.9 Conclusion 25

Chapitre 2 : QoS dans les Réseaux Mobiles Ad hoc 26

2.1. Introduction 26
2.2. Niveau de service 27
2.3. Notion de qualité de service (QoS : quality of service) 27
2.4. Paramètres de la qualité de service 28
2.5. Modèles de qualité de service 29
2.5.1. Qu'est ce qu'un modèle de QoS ? 30
2.5.2. Le modèle FQMM 30
2.5.3. Le modèle SWAN 31
2.5.4. Le modèle iMAQ 32
2.6. Routage avec qualité de service 33
2.6.1. Protocoles de routage avec QoS pour les MANETs 33
2.6.1.1. Quelque protocoles basées sur la topologie de cluster 34
2.6.1.1.1. CGRS (Cluster-Gateway Switching Routing) 34
2.6.1.1.2. CBRP (Cluster based Routing Protocol) 34
2.6.1.2. Quelque protocoles de routage avec QoS 35
2.6.1.2.1. CEDAR (Core-Extraction Distributed Ad hoc Routing Algorithm) 35
2.6.1.2.2. QOLSR (QoS Optimized Link State Routing Protocol) 36
2.6.1.2.3. TBP (Ticket Based Probing) 36
2.6.1.2.4. DSDV+ (Destination Sequenced Distance Vector avec QoS) 37
2.6.1.2.5. BRuIT (Bandwidth Reservation under InTerferences influence) 38
2.6.2. Evaluation de protocole de routage avec QoS 38
2.6.3. Avantages et inconvénient des protocoles de routage avec QoS 40
2.7. Signalisation pour la QoS 42
2.7.1. Un protocole de signalisation QoS ad hoc 42
2.8. Conclusion 43

Chapitre 3 : La technologie Agent Mobile et les Réseaux Ad hoc 44

3.1. Introduction 44
3.2. Qu'est-ce qu'un agent? 45
3.3. Définition d'un agent mobile 45

3.4. Apport des agents mobiles à l'exécution répartie....46
3.4.1. Infrastructure système....47
3.4.1.1. Support d'exécution....47
3.4.1.2. Support de migration....47
3.4.1.3. Support de communication....47
3.4.1.4. Sécurité....48
3.4.2. Evaluation....48
3.4.2.1. Minimisation du nombre des interactions distantes....48
3.4.2.1.1. Exemple redirection de requêtes....48
3.4.2.2. Minimisation du volume des données transportées sur le réseau....49
3.5. Avantages et inconvénients des agents mobiles....50
3.6. Localisation distribué et adaptative d'agent mobile....52
3.6.1. Principe de localisation des agents....52
3.6.2. Service de localisation....52
3.6.3. La stratégie de migration adaptative....53
3.7. Travaux à base d'agent mobile dans un réseau mobile ad hoc....54
3.7.1. Sécurité d'agent mobile aux réseaux sans fil ad hoc....54
3.7.1.1. Sécurité d'agent mobile....54
3.7.1.1.1. Protection des sites....55
3.7.1.1.2. Protection des agents....56
3.7.1.2. Architecture de SWAT....56
3.7.1.2.1. Protocole et mécanisme de sécurité....57
3.7.1.2.2. L'intergration de système de SWAT....58
3.7.1.2.2.1. L'intergration de l'hôte et l'agent système....59
3.7.1.2.2.2. L'integration de l'hôte et la sécurité....60
3.7.1.2.2.3. L'intégration d'agent système et réseau....60
3.7.1.2.2.4. L'intégration d'agent système et sécurité....60
3.7.1.3. Synthèse....61
3.7.2. Découverte de topologie en utilisant des agents mobiles....61
3.7.2.1. Description des termes pertinents....62
3.7.2.1.1. Lien affinité....62
3.7.2.1.2. Receny....62
3.7.2.1.3. Temps de migration....63
3.7.2.1.4. La moyenne de convergence de connectivité....63

3.7.2.1.5. Moyenne de convergence de lien affinité 64
3.7.2.2. La découverte de topologie : mécanisme de base 65
3.7.2.2.1. Algorithme de navigation 65
3.7.2.2.2. Manipulation l'événement d'oscillation d'agent entre des nœuds.....66
3.7.2.2.3. L'échange d'information et l'interaction entre deux agents.....66
3.7.2.3. Vieillissement de l'information : une méthode prédictive.....67
3.7.2.4. Synthèse.....67
3.7.3. La communication de message basée d'agent mobile dans un réseau ad hoc.....68
3.7.3.1. Description de système.....68
3.7.3.2. Un mécanisme pour la création et la navigation d'agent.....69
3.7.3.2.1. Structure de nœud.....69
3.7.3.2.2. Création d'agent au nœud source et procédure de base de navigation 69
3.7.3.2.3. Comportement d'un agent prés d'emlacementdu nœud de destination.....69
3.7.3.2.4. Réoriantation un agent par le voisin logique de nœud de destination.....69
3.7.3.3. Évaluation des performances.....70
3.7.3.3.1. Evaluation de la validité de système 70
3.7.3.3.2. Evaluation l'efficacité de communication de message basée d'agent.....71
3.7.3.4. Synthèse.....71
3.7.4. Protocole de routage basé d'agent mobile.....71
3.7.4.1. Protocole de routage hybride Ant-AODV 72
3.7.4.2. Résultats de simulation.....72
3.7.4.2.1. La moyenne de délai bout à bout 72
3.7.4.2.2. Fraction de livraison de paquets et Goodput.....73
3.7.4.2.3. Normalisation de routage overhead.....73
3.7.4.2.4. Connectivité 73
3.7.4.3. Synthèse.....74
3.8. Conclusion.....74

Chapitre 4 : Modèle de QoS à base d'Agent Mobile…….... .. 76

4.1. Introduction 76
4.2. Organisation du réseau.....77
4.2.1. Niveaux de gestion de QoS 77
4.2.1.1. Niveau nœud 77

4.2.1.2. Niveau groupe (cluster) 78
4.2.1.3. Niveau réseau 79
4.3. Processus de construction de groupes 80
4.3.1. Découverte des voisins 80
4.3.2. Sélection des chefs 80
4.3.3. Construction des groupes 80
4.4. Algorithme de construction des groupes 80
4.5. Architecture interne de l'agent mobile 81
4.5.1. Architecture interne de l'Agent Nœud 82
4.5.1.1. Architecture interne d'agent nœud de l'état membre 84
4.5.2. Architecture interne de l'Agent Transporteur 85
4.5.3. Architecture interne de l'Agent Routier 85
4.6. Diagramme d'état de transition 86
4.7. Diagramme de classes de modèle 88
4.7.1. Agent nœud 88
4.7.2. Cluster d'agents nœuds 90
4.7.3. Réseau d'agents nœuds 90
4.8. Propriétés du modèle 90
4.9. Conclusion 91

Chapitre 5 : Protocole de Routage et Environnement de développement... 92

5.1. Protocole de routage à base d'agent mobile (PRAM) 92
5.1.1 Introduction 92
5.1.2. Les éléments du SMA 92
5.1.2.1. Agent Nœud 93
5.1.2.2. Agent Transporteur 93
5.1.2.3. Agent Routier 93
5.1.3. Fonctionnalités de PRAM 94
5.1.4. Algorithme de navigation et de mise à jour des groupes 95
5.1.5. Diagramme d'état de transition 95
5.1.6. Propriétés du protocole 97
5.2. Environnement de développement. 97
5.2.1. Introduction 97

5.2.2. Le langage de programmation java....98
5.2.3. La plate-forme d'agent mobile Aglet....98
5.2.3.1. Définition d'un Aglet....98
5.2.3.2. Architecture d'un Aglet....99
5.2.3.3. Cycle de vie d'un Aglet....99
5.2.3.4. Tahiti : un gestionnaire d'agents visuel....100
5.2.4. Présentation de l'application....101
5.3. Conclusion....107

Conclusion générale et Perspective....109
Références....111

Table des figures

Figure 1.1 : La catégorie des réseaux mobiles.....5
Figure 1.2 : Le modèle des réseaux mobiles avec infrastructure.....6
Figure 1.3 : Le modèle des réseaux mobiles sans infrastructure.....6
Figure 1.4 : Modélisation d'un réseau ad hoc en graphe.....9
Figure 1.5 : Changement de topologie dans un réseau mobile ad hoc.....10
Figure 1.6 : Problème de la station cachée.....11
Figure 1.7 : Problème de la station exposée.....12
Figure 1.8 : Mode de communication dans les réseaux mobiles.....12
Figure 1.9 : Réseau ad hoc simple constitué de trois unités mobiles.....13
Figure 2.1 : Le modèle FQMM.....31
Figure 2.2 : Le modèle SWAN.....32
Figure 2.3 : Le modèle iMAQ.....33
Figure 2.4 : L'architecture du protocole INSIGNIA.....42
Figure 3.1 : Le paradigme des agents mobiles.....46
Figure 3.2 : Comparaison entre RMI et Aglets.....49
Figure 3.3 : Temps d'exécution entre objets client et serveur et Aglets.....50
Figure 3.4 : Architecture de hôte de SWAT.....57
Figure 3.5 : Architecture de réseau du SWAT.....59
Figure 4.1 : Organisation de réseau ad hoc.....77
Figure 4.2 : Localisation d'agent nœud.....82
Figure 4.3 : Architecture interne de l'Agent Nœud.....83
Figure 4.4 : Architecture interne d'agent nœud de l'état membre.....84
Figure 4.5 : Architecture interne d'Agent Transporteur.....85
Figure 4.6 : Architecture interne d'Agent Routier.....85
Figure 4.7 : Diagramme d'état de transition d'Agent Nœud.....87
Figure 4.9 : Diagramme de classes de notre modèle.....89
Figure 5.1 : Diagramme d'état transition entre nœud source et nœud destination.....96
Figure 5.2 : Relation entre un Aglet et son Proxy.....99
Figure 5.3 : Cycle de vie d'un Aglet.....100
Figure 5.4 : Illustre le lancement de plate-forme Aglet.....101

Figure 5.5 : Illustre la création Agent Nœud 102
Figure 5.6 : Illustre l'agent nœud du port 5000 dans l'état initial 103
Figure 5.7 : Illustre la sécurité entre des agents nœuds 103
Figure 5.8 : Illustre l'agent nœud du port 5000 après l'élection 104
Figure 5.9 : Illustre l'agent nœud du port 7000 dans l'état initial 104
Figure 5.10 : Illustre l'agent nœud du port 7000 après l'élection 105
Figure 5.11 : Illustre l'agent nœud du port 5001 dans l'état initial 105
Figure 5.12 : Illustre l'agent nœud du port 5001 après l'élection 106
Figure 5.13 : Illustre l'agent nœud du port 7001 dans l'état initial 106
Figure 5.14 : Illustre l'agent nœud du port 7001 après l'élection 107
TAB 1.1 : Réseau sans fil avec infrastructure et réseau sans fil ad hoc 7

Introduction générale

L'évolution récente de la technologie dans le domaine de la communication sans fil et l'apparition des unités de calcule portable, 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 ».

Les réseaux mobile ad hoc (MANETs) appartiennent à une catégorie de réseaux sans fil qui n'a pas besoin d'infrastructure, chaque nœud jouant le rôle de l'hôte ainsi que du routeur. Les terminaux mobiles dans ces réseaux peuvent se déplacer de manière aléatoire et à des vitesses quelconques. Dans ce contexte, il faut prendre en compte des défis ou des caractéristiques lors de déploiement des réseaux mobiles ad hoc.

Le premier défis est la topologie dynamique, où, les hôtes sont mobile et peuvent être connectés entre eux de manière arbitraire, donc, les liens radio changent régulièrement, lorsque les objets se déplacent, s'éteignent, ou lorsque des obstacles apparaissent ou disparaissent.

En plus, les hôtes fonctionnent grâce à une batterie, dont la durée de vie est généralement limitée à quelques heures d'utilisation, les communications doivent donc être réduites au strict minimum. Ainsi, l'hétérogénéité des nœuds ne dispose pas des mêmes propriétés physiques et logicielles comme la capacité de traitement (CPU, mémoire), la taille (petit, grand) et la mobilité (lent, rapide), ainsi que la faible bande passante qui diminue également en raison des interférences des signaux et la déplétion sur le canal sont l'un des défis qui sont à prendre en considération.

Enfin, l'établissement et le maintient de la connectivité du réseau sont réalisés par des hôtes mobiles à cause d'absence de toutes infrastructures ou administration centralisées préexistantes. La transmission des paquets entre les nœuds est assurée à l'aide de protocoles de routage complexe dont la caractéristique essentielle est de surmonter les difficultés introduites par la mobilité.

La notion de qualité de service (QoS) dans les réseaux consiste à privilégier certaines informations par rapport à d'autre, en offrant des services différenciés en fonction des exigences des applications. Dans les réseaux mobiles ad hoc, l'objectif de la qualité de service est d'atteindre un meilleur comportement de la communication, pour que le contenu de cette dernière soit correctement acheminé, et les ressources du réseau utilisé d'une façon optimale.

La QoS est basée en général sur un certain nombre de paramètre comme, le délai de bout en bout, la variance de délai, la bande passante, la perte de paquet et sécurité. Ces paramètres permettent de spécifier la QoS de différentes manières selon les exigences des usagers et des applications. Par exemple, pour les applications temps réel, comme la voix et la vidéo, le délai de bout en bout d'un paquet doit être limité, autrement le paquet est inutile. Les applications non temps réel, comme le transfert de fichier ou la messagerie, quant à elles se focalisent sur la fiabilité des communications.

Les réseaux mobile ad hoc sont idéal pour les applications caractérisées par une absence d'une infrastructure préexistante, tel que les applications militaires et les autres applications de tactique comme les opérations de secours et les missions d'exploration. L'avantage de ces réseaux réside dans la facilité de mise en place et d'ajout de nouvelles stations sur le réseau, l'absence de structure fixe diminue aussi le coût de leur mise en œuvre.

En effet, la conception et la mise en œuvre de modèle et de protocole de routage représentent un problème complexe. Cela est du essentiellement à des particularités qui présentés auparavant. Une des solutions à ce problème passe par l'utilisation d'agent mobile et la mise en place de topologies bien adaptées au routage dans les réseaux ad hoc. C'est dans ce cadre que s'inscrit notre travail.

L'utilisation d'agent mobile est une nouvelle solution aux réseaux mobiles ad hoc où l'agent sautent du nœud à nœud pour collecter toutes les informations relatives à la topologie de chaque nœud et de les distribuer périodiquement à d'autres nœuds. Les agents mobiles sont particulièrement intéressants lorsqu'ils sont utilisés dans le cadre de communications non permanentes. En effet, le maintien du lien de communication peut s'avérer difficile dans les réseaux mobile ad hoc à cause des nœuds qui sont libres de se déplacer aléatoirement. Avec les agents mobiles, un client peut déléguer les interactions avec le service sans maintenir une connexion de bout en bout. En plus, en réduisant le plus possible les communications distantes aux seuls transferts d'agents mobiles, on diminue considérablement les périodes de connexion entre deux nœuds.

En outre, l'agent mobile permet de diminuer la consommation de bande passante et d'optimiser le délai et de réduire la charge des réseaux engendrés. En effet, on obtient une réduction significative de la charge du réseau en terme du nombre total de données transférées ou par les échanges de messages de contrôle, et par conséquent, cela minimise la probabilité de congestion. Cette diminution est constatée dans différents types d'applications nécessitant

d'intenses échanges d'information entre le client et le serveur. En plus, ils peuvent s'adapter facilement aux erreurs de systèmes. Ces erreurs peuvent être d'ordre purement physique, disparition d'un nœud par exemple, ou d'ordre plus fonctionnel, arrêt d'un service.

Ce mémoire se compose essentiellement de cinq chapitres :

Dans le premier chapitre, nous avons présenté les différents concepts liés aux réseaux mobile ad hoc comme, la définition, les caractéristiques des réseaux ad hoc, les protocoles de routage, les applications actuelles et futures des MANETs, les avantages et les inconvénients des réseaux ad hoc, etc.

La notion de qualité de service (QoS), ainsi que les techniques utilisées pour le support de qualité de service dans les réseaux ad hoc : les modèles de qualité de services, les protocoles de routages avec qualité de service, sont illustrés dans le deuxième chapitre.

Le troisième chapitre est consacré au technologie d'agent mobile et l'apport à l'exécution répartie, en mettent la lumière sur les travaux développées qui sont utilisées pour garantir la qualité de service dans les réseaux mobile ad hoc.

Dans le quatrième chapitre, nous proposons un modèle à base d'agent mobile dont le but est d'améliorer la QoS dans les réseaux ad hoc, ce modèle est basée sur l'organisation du réseau à trois niveaux (niveau nœud, niveau groupe, niveau réseau) pour une gestion hiérarchique de la qualité de service. Dans ce chapitre, nous présentons d'abord l'organisation de réseau, puis, l'algorithme de construction des groupes, ensuite l'architecture interne d'agent mobile, on termine par le diagramme d'état de transition et le diagramme de classe de modèle.

Dans le chapitre cinq, nous employons notre modèle pour proposer un nouveau protocole de routage qui nous avons appelé PRAM (Protocole de Routage à base d'Agent Mobile) afin d'améliorer la qualité de service pour les réseaux ad hoc. En plus, nous allons présenter la réalisation de notre modèle en utilisant la plate-forme d'agent mobile Aglet.

Finalement, nous terminons par une conclusion générale avec quelques perspectives qui viennent couronner ce travail, reprenant les points forts et ceux manquants, ainsi que définissant les travaux futurs que nous comptons mener et qui vont dans le sens de l'amélioration de la QoS dans les réseaux ad hoc.

Chapitre 1

Réseaux Mobiles Ad hoc (MANETs)

1.1. Introduction

En général, un *réseau mobile ad hoc* est considéré comme un système autonome dynamique composé de nœuds mobiles interconnectés par des liens sans fil, sans l'utilisation d'une infrastructure fixe et sans administration centralisée. Les nœuds sont libres de se déplacer aléatoirement et s'organisent arbitrairement. Par conséquent, la topologie du réseau peut varier de façon rapide et surtout imprévisible.

Par l'absence d'infrastructures dans de tels réseaux, les nœuds se comportent comme des routeurs de manière à transférer les données d'une source à une destination. Le routage dans ces réseaux, il faut être capable de déterminer le chemin optimal par lequel les paquets vont transités jusqu'à la destination en minimisant l'activité des nœuds. Pour cela, ils utilisent des protocoles de routage qui peuvent être de différentes natures. Beaucoup d'algorithmes et de protocoles ont été proposés, mais peu d'entre eux ont été expérimentés étant donné la complexité de leur mise en œuvre dans un environnement réel.

La notion de Qualité de Service (QoS) dans les réseaux consiste à privilégier certaines informations par rapport à d'autres, en offrant des services différenciés en fonction des exigences des applications. Pour mettre en place de la qualité de service dans les réseaux ad hoc, le calcul des routes doit se baser sur d'autres critères que le nombre de sauts. Plusieurs métriques peuvent être considérées, seules ou combinées : le délai, la bande passante, la sécurité, la connectivité, ou encore la gigue.

Les applications des réseaux ad hoc sont nombreuses, on cite l'exemple classique de leur application dans le domaine militaire et les autres applications de tactique comme les opérations de secours et les missions d'exploration.

L'avantage de ces réseaux réside dans la facilité de mise en place et d'ajout de nouvelles stations sur le réseau, l'absence de structure fixe diminue aussi le coût de leur mise en œuvre. Néanmoins, les réseaux présentent des inconvénients comme, le changement rapide de topologie, la limite de capacité et la sécurité.

1.2. L'environnement mobile

Un environnement mobile est un système compose de sites mobiles qui permet à ses utilisateur d'accéder à l'information indépendamment de leurs positions physiques [1].

Les environnements mobiles offrent aujourd'hui une bonne alternative de communication à moindre coût et à grande flexibilité d'emploi. En effet, les mobiles permettent à un ensemble de machines hôtes d'être interconnectées facilement et rapidement entre elles avec infrastructure définie préalable ou non.

Les applications s'exécutant dans les environnements mobiles ad hoc ont besoin de mécanisme dynamique aux variations du contexte d'exécution comme la fluctuation au niveau des ressources disponibles, la variation de la bande passante, la réduction du niveau d'énergie, etc. tout cela, pour supporter les exigences en termes de QoS des applications. Les réseaux mobiles sans fil, peuvent être classé en deux classes [2] :

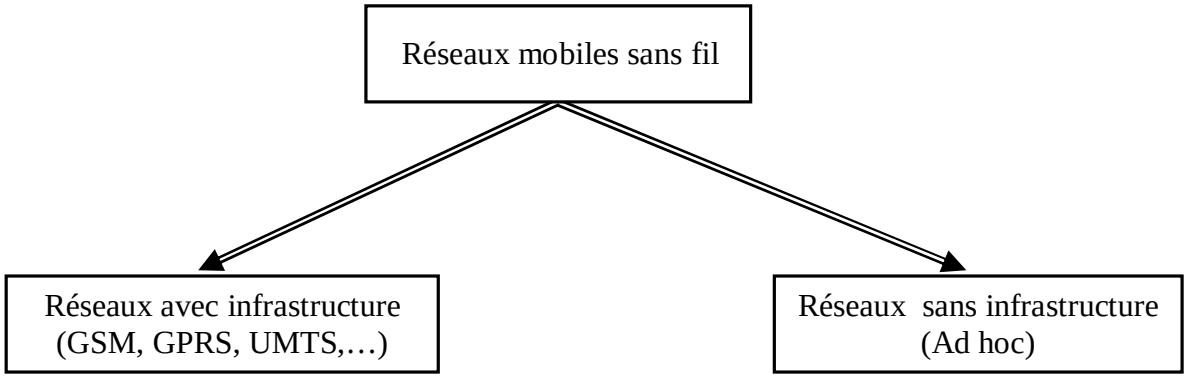

FIG. 1.1 – La catégorie des réseaux mobiles

- Les réseaux avec infrastructure qui utilisent, par exemple, le modèle de communication cellulaire (GSM), exigent d'importantes infrastructures fixes. Dans ce modèle (voir la figure 02), 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. L'autonomie réduite de sa source d'énergie, lui occasionne de fréquentes déconnexions du réseau, sa reconnexion peut alors se faire dans un environnement nouveau voire dans une nouvelle localisation [3].

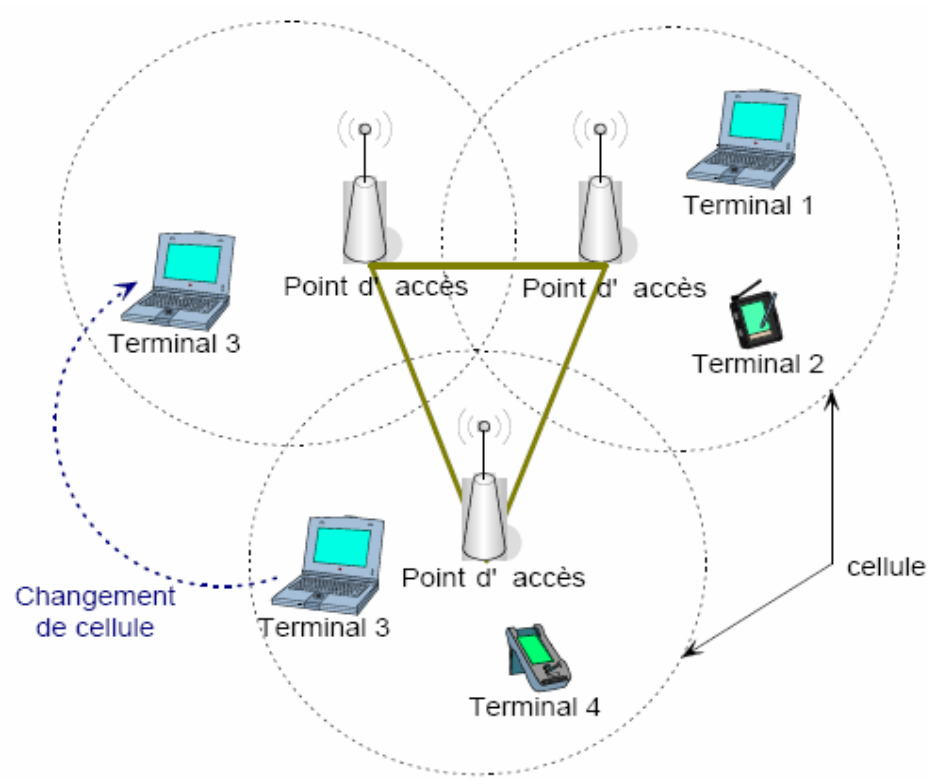

FIG. 1.2 – Le modèle des réseaux mobiles avec infrastructure

- Les réseaux sans infrastructure sont appelés les réseaux Ad hoc et qui ne requièrent aucune infrastructure. Le modèle de réseau sans infrastructure préexistante ne comporte pas l'entité "site fixe", tous les sites du réseau sont mobiles et se communiquent d'une manière directe en utilisant leurs interfaces de communication sans fil (figure 03). L'absence de l'infrastructure ou du réseau filaire composé des 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.

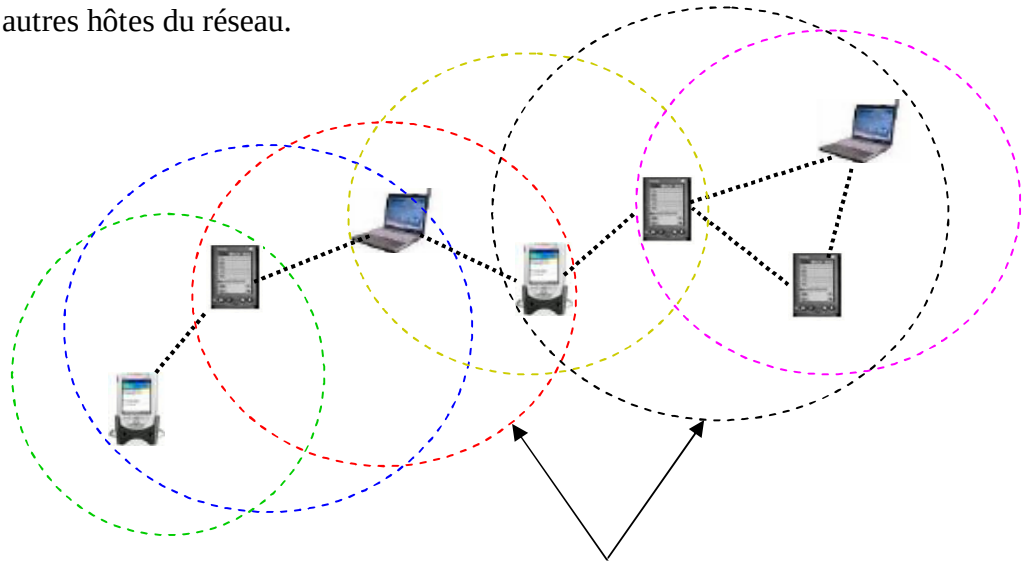

FIG. 1.3 – Le modèle des réseaux mobiles sans infrastructure

Le tableau [3] suivant résume une petite comparaison entre le réseau sans file avec infrastructure et le réseau sans file ad hoc.

	Réseau sans fil avec infrastructure	***Réseau sans fil ad hoc***
Communication	Via un point d'accès	Par voisinage
Cellule	Fixe, gérée par la station de base	Auto-configurable, dépend de la position des terminaux mobiles
Routage intra-cellule	Via un point d'accès	Par voisinage avec algorithme de routage
Routage inter-cellule	Via plusieurs points d'accès	Nécessite une station relais

TAB. 1.1 – Réseau sans fil avec infrastructure et réseau sans fil ad hoc

1.3. Fondements des réseaux mobiles ad hoc

1.3.1. Historique et évolution des réseaux ad hoc

Historiquement, les réseaux mobiles ad hoc ont été principalement employés pour l'amélioration des communications dans le domaine militaire. Dans ce contexte, il n'existe pas d'infrastructure existante pour relier les communications, vue la nature dynamique des opérations et des champs militaires [4].

En 1972, Les premières applications dans les réseaux ad hoc sont apparues avec le projet PRNet (Packet Radio Network). Ce projet a été inspiré par l'efficacité la technologie par commutation de paquet, le partage de la bande passante, le routage store-and-forward, et ses applications dans l'environnement mobile sans fil.

En 1983, le projet SURAN (Survivable Radio Networks) développé par la DARPA pour dresser les principaux problèmes du projet PRNet dans le domaine de la scalabilité, la sécurité, la capacité de traitement et gestion d'énergie.

En 1987, le travail SURAN amené à la conception de la technologie LPR (Low-cost Packet Radio), dotée d'une couche radio DSSS (Direct Sequence Spread- Spectrum) avec un processeur pour la commutation de paquets intégré (Intel 8086). L'évolution des infrastructures du réseau Internet et la révolution de la micro informatique ont permis de rendre faisables et applicables les idées initiales des réseaux radio de paquets.

En 1994, Le programme GloMo (Global Mobile) initié par la DARPA avait comme objectif de supporter les communications multimédia n'importe quand et n'importe où à travers des équipements sans fil.

En 1997, un autre projet a été développé par l'armé américaine, il s'agit de Tactical Internet (IT) [FRE01] qu'est l'une des implémentations des réseaux mobiles ad hoc multi sauts utilisant des débits de plusieurs dizaines de kilobits par seconde.

En 1999, un autre déploiement des réseaux mobile ad hoc de 20 nœuds a été réalisé avec ELB ACTD (Extending the Littoral Battle-space Advanced Concept Technology Demonstration) qui permet de démontrer la faisabilité de concepts militaires pour les communications des bateaux en mer aux soldats sur la terre par l'intermédiaire d'un relais aérien.

Dans les dernières années, la recherche dans ce domaine connaît une grande activité probablement liée au succès de la norme IEEE 802.11 qui permet de réaliser des réseaux ad hoc à moindre coût. L'augmentation des débits de la couche physique laisse également présager des déploiements commerciaux.

1.3.2. Qu'est ce que les réseaux mobiles ad hoc

Un réseau mobile ad hoc, appelé communément MANET (Mobile Ad hoc NETwork) est un réseau sans fil, sans aucune infrastructure fixe, utilisant généralement le médium radio, création et organisation dynamique. Dans de tel réseau, les nœuds ne sont pas tous voisins directs, les voisins de nœuds sont ceux qui se trouvent dans sa zone d'émission. Lorsqu'un nœud émet, tous ses voisins ne peuvent être qu'en mode réception, l'information peut passer d'un nœud à un autre par différents nœuds. Un nœud peut représenter une personne, un portable, un capteur de mesure, ou un dispositif électronique ayant la capacité de communiquer avec d'autres capteurs par des transmissions radio.

Une autre définition, des réseaux ad hoc, a été donnée [5] est la suivante : « Un réseau mobile ad hoc est un réseau auto-organisé formé spontanément à partir d'un ensemble d'entités mobiles communicantes (ordinateurs portables, téléphones mobiles, assistants électroniques) sans nécessiter d'infrastructure fixe préexistante ».

1.3.3. Modélisation de réseau mobile ad hoc

Selon [6], un réseau mobile ad hoc peut être modélisé par un graphe $G_t = (V_t, E_t)$ où : V_t représente l'ensemble des nœuds (i.e. les unités ou les hôtes mobiles) du réseau E_t modélise l'ensemble les connections qui existent entre ces nœuds (voir la figure 1.4) Si $e = (u, v) \in E_t$, cela veut dire que les nœuds u et v sont en mesure de communiquer directement à l'instant t.

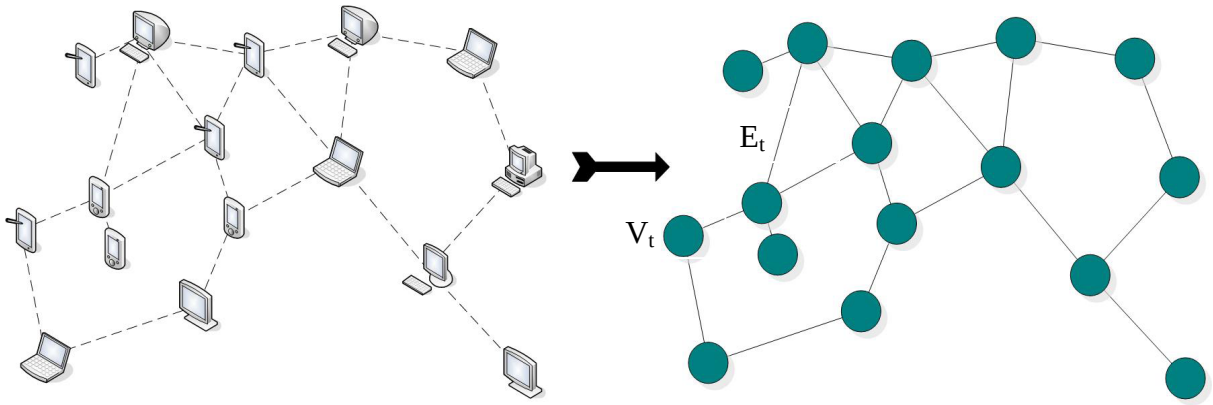

FIG.1. 4 – Modélisation d'un réseau ad hoc en graphe

Notons que, si G_t est un graphe complet bidirectionnel, on parle d'un réseau mobile ad hoc à un saut. Sinon on parle d'un réseau ad hoc multi-sauts. En plus, chaque nœud ne connaît que son propre identifiant : il ne connaît pas à priori les identifiants des nœuds voisins, ni le nombre de nœuds dans le réseau.

1.3.5. Caractéristique et défis des réseaux mobiles ad hoc

Les réseaux ad hoc héritent des mêmes propriétés et problèmes liés aux réseaux sans fil, particulièrement, le fait que le canal radio soit limité en terme de capacité, plus exposé aux pertes (comparé au médium filaire), et sujet à des variations dans le temps. En outre, les liens sans fil sont asymétriques et non sécurisés. D'autres caractéristiques [4, 5, 7,8] spécifiques aux réseaux Ad hoc rendent ceux-ci plus complexes et des contraintes supplémentaires doivent être prises en compte :

- **L'absence d'une infrastructure :** la différence principale entre les réseaux ad hoc et les autres réseaux mobiles est l'absence de toutes infrastructures ou administration centralisées préexistantes. Pour cela, l'établissement et le maintient de la connectivité du réseau sont réalisés par des hôtes mobiles.

- **La mobilité des nœuds et maintenance des routes :** la mobilité continue des nœuds crée un changement dynamique de topologie qui peut varier d'une manière rapide et aléatoire. En effet, Un nœud ad hoc est susceptible de quitter ou de rejoindre le réseau à tout instant donc un protocole doit s'adapter continuellement et rapidement à ces changements afin d'offrir des performances optimales sur la durée. On peut représenter le changement de la topologie comme la figure suivante.

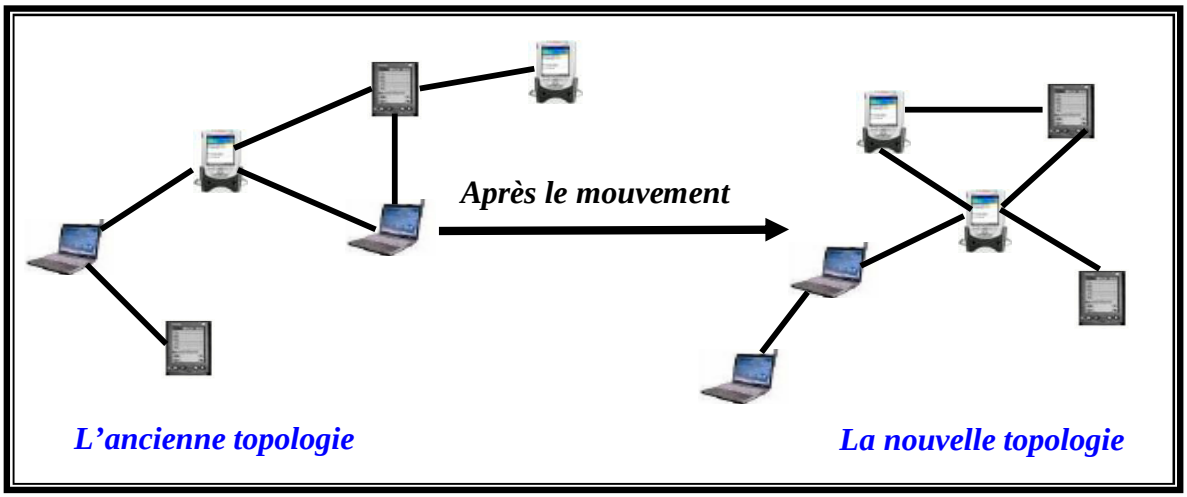

FIG. 1.5 – Changement de topologie dans un réseau mobile ad hoc

- **L'hétérogénéité des nœuds :** les nœuds ad hoc peuvent correspondre à une multitude d'équipements (de l'ordinateur portable au capteur intelligent en passant par le téléphone mobile). Ces équipements ne disposent pas des mêmes propriétés physiques et logicielles comme la capacité de traitement (CPU, mémoire), la taille (petit, grand) et la mobilité (lent, rapide). Dans ce cas, des protocoles doivent donc prendre en compte une telle hétérogénéité.

- **Bande passante limitée :** une des caractéristiques primordiales des réseaux basés sur la communication sans fil est l'utilisation d'un médium de communication partagé. Ce partage fait que la bande passante réservée à un hôte soit modeste. La bande passante disponible dépend à la fois du nombre de nœuds présents dans le voisinage et du trafic de données à transporter, indépendamment des perturbations physiques qui peuvent intervenir.

- **La contrainte d'énergie :** malgré les améliorations des batteries et des technologies de consommation d'énergie, les équipements mobiles disposent de batteries limitées, et dans certains cas très limitées et par conséquent d'une durée de traitement réduite. Les services et les applications supportées par chaque nœud sont limite. D'autant plus qu'une partie de l'énergie est déjà consommée par la fonctionnalité du routage.

- **La taille des réseaux ad hoc :** elle est souvent de petite ou moyenne taille (une centaine de nœuds), le réseau est utilisé pour étendre temporairement un réseau filaire, comme pour une conférence ou des situations où le déploiement du réseau fixe n'est pas approprié (ex : catastrophes naturelles). Cependant, quelques applications des réseaux ad hoc nécessitent une utilisation allant jusqu'à des dizaines de milliers de nœuds, comme dans les réseaux de senseurs. Des problèmes liés au passage à l'échelle tels que : l'adressage, le routage, la gestion de la localisation des senseurs et la configuration du réseau, . . . etc., doivent être résolus pour une meilleure gestion du réseau.
- **Sécurité limitée :** la nature du support physique, l'absence de coordination centrale, la mobilité et l'hétérogénéité du nœud rendent les réseaux ad hoc plus vulnérables que les infrastructures fixes. Les transmissions sans fil peuvent être aisément capturées par un nœud ad hoc dans le voisinage local. Une attaque par déni de service peut être facilement réalisée par un nœud malicieux en s'appropriant la bande passante ou en surchargeant un nœud voisin avec une quantité importante de trafic à router.

D'autres défis, concernent l'accès au support génèrent des problèmes supplémentaires : le problème de la station cachée et celui de la station exposée.

Problème de la station cachée : dans le cas où deux mobiles A et C ne sont pas à portée de communication, et que chacun d'eux communique avec un tiers B, il existe une possibilité de collision sur ce dernier. En effet, sans accord préalable A et C n'ont aucun moyen de prendre conscience de l'autre communication en cours [9]. On dit alors que C est caché de A et que A est caché de C (voir figure 1.6).

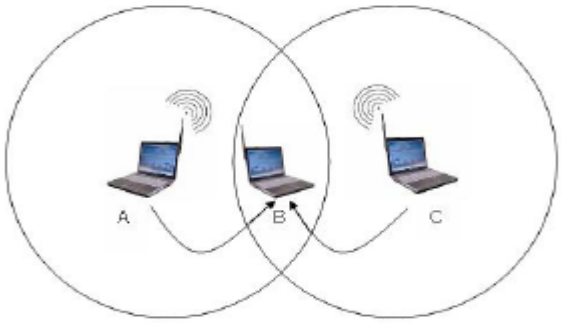

FIG. 1.6 – Problème de la station cachée

Un mécanisme a été pensé afin d'éliminer ce problème : avant l'envoi d'information, l'émetteur transmet un paquet RTS (Request To Send) au récepteur, lui annonçant ainsi une demande de transmission. Le récepteur renvoie un paquet CTS (Clear To Send) s'il est libre. Cette technique permet donc d'obtenir une certaine visibilité de la station cachée.

- **Problème de la station exposée :** ce problème survient quand une station veut établir une transmission avec une deuxième mais doit la retarder car il y a une transmission en cours entre deux autres stations se trouvant dans son voisinage [10].

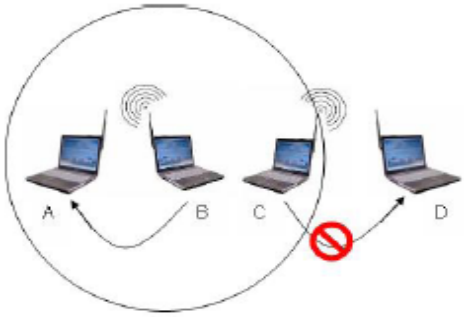

FIG. 1.7 – Problème de la station exposée

Dans la figure 1.7 supposons que les stations A et C peuvent entendre les transmissions de B, mais que la station A n'entent pas C (et vice-versa). Supposons aussi que B est entrain d'envoyer des données vers A et que, au même moment, C veut communiquer avec D. la station C va commencer par déterminer si le support est libre. A cause de la communication entre B et A, C trouve le support occupé et il retarde son envoi bien que celui-ci n'aurait pas causé de collision mais peut gaspiller la bande passante du réseau.

1.4. Routage dans les réseaux mobiles ad hoc

Avant de parler du routage proprement dit, il est bon de rappeler quels sont les principaux modes de communication dans les réseaux mobiles : la communication point à point ou unicast, pour laquelle il y a une source et une seule destination, la communication multipoint ou multicast, qui permet d'envoyer une message à plusieurs destinataires et la diffusion ou broadcast, qui envoie un message à tous les nœuds du réseau. Ces trois modes de communication sont schématisés par la figure suivante [10].

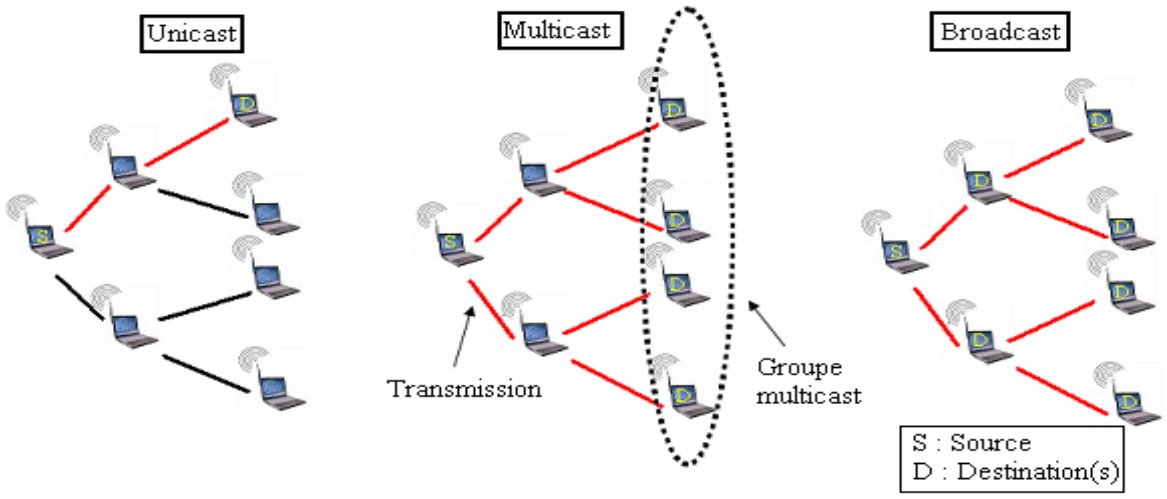

FIG. 1.8 – Mode de communication dans un réseau mobile

En effet, le routage est un point important dans les réseaux mobiles ad hoc. Il faut être capable de déterminer le chemin optimal par lequel les paquets vont transités jusqu'à la destination en minimisant l'activité des nœuds. Le routage (*routing*) est le mécanisme d'ouverture et d'entretien d'une communication entre deux nœuds. L'opération est alors supportée par la source, le destinataire et les relais supportant l'échange [9].

1.4.1. Problème de routage dans les réseaux ad hoc

Comme nous l'avons vu précédemment, l'architecture d'un réseau mobile ad hoc est caractérisée par une absence d'infrastructure fixe préexistante et les nœuds sont libres de se déplacer aléatoirement et s'organisent arbitrairement. Donc, le problème consiste à déterminer un acheminement optimal des informations à travers le réseau au sens d'un certain critère de performance [8]. En effet, il y a une difficulté dans le choix du critère permettent de dire qu'un chemin est meilleur que l'autre.

Un autre problème, qui se pose dans le contexte de routage de 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 nœuds, dans le cas où le réseau serait volumineux [8]. De plus, la gestion du routage devient plus complexe.

D'autre part, lorsqu'une station source peut avoir besoin de transférer des données à une autre station qui ne se trouve pas dans sa portée de communication, par exemple dans le réseau illustré par la figure suivante l'unité mobile W n'est pas dans la portée de communication de l'unité U (indiquée par le cercle d'origine U) et vice versa. Dans le cas où l'unité U veut transférer des paquets à W, elle doit utiliser les services de l'unité V dans l'envoi des paquets, puisque l'unité V contient dans sa portée de communication les unités U et W.

Dans la pratique, le problème de routage est plus compliqué à cause de la non-uniformité de la transmission sans fil et de la possibilité du déplacement imprévisible de tous les nœuds concernés par le routage.

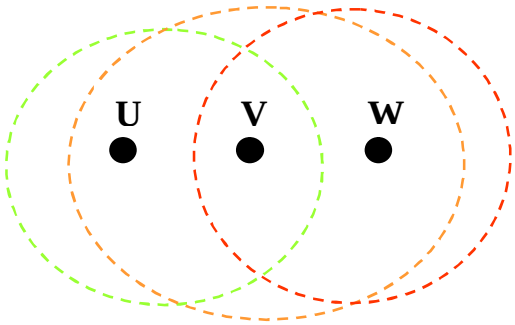

FIG. 1.9 – Réseau ad hoc simple constitué de trois unités mobiles

1.4.2. Contraintes des protocoles de routage dans MANET

Selon [11, 12], il faut prendre en compte des contraintes lors de déploiement d'un protocole de routage pour supporter la topologie dynamique et la mobilité des réseaux ad hoc.

- **Distribution :** les protocoles doivent être entièrement distribués, fournissant la scalability et la tolérance de fautes.
- **Support des liens asymétriques :** les protocoles doivent supporter l'existence des liens unidirectionnels, qui sont fortement exposés dans un environnement radio sans fil.
- **Sécurité :** les protocoles doivent réagir aux menaces et aux vulnérabilités, par des mécanismes qui empêchent toutes les attaques possibles contre un réseau ad hoc, évitent le déni du service et de la consommation agressive de ressources.
- **Minimum de charge de contrôle :** les paquets de contrôle dans un protocole de routage devraient être maintenus aussi minimum que possible, car ils consomment la largeur de bande passante et peuvent causer des collisions avec des paquets de données, diminution de débits et aussi le protocole doit éviter les boucles.
- **Conservation de ressources :** les protocoles devraient optimiser l'utilisation des ressources rares telles que la largeur de la bande passante, la puissance de calcul, la mémoire, et le temps de traitement des terminaux.
- **Qualité de service :** les protocoles doivent être capables de fournir un certain niveau de qualité de service (QoS) surtout en terme du débit et du délai de transmission de données surtout lorsque le réseau est à forte charge comme exigé par les applications temps réel.

1.5. Protocole de routage de réseaux mobiles ad hoc

Une multitude de protocoles de routage a été développée spécifiquement pour les réseaux ad hoc ces dernières années. Ces protocole opèrent où les changements de topologie sont fréquents, et sont exécutés sur des équipements ayant des contraintes de ressources (de batterie, de mémoire, de CPU, . . . etc.). Ils existent [6] cinq grandes familles de protocoles de routage ad hoc : les protocoles proactifs, réactifs, hybrides, hiérarchique et géographique. Nous allons détailler les trois grands premiers le plus connus.

1.5.1. Les protocoles proactifs

Le principe de ce protocole (appelé aussi *table driven*) est le suivant : chaque station maintient une ou plusieurs tables contenant l'information de routage vers tous les autres

station du réseau. Quand la topologie des réseaux ad hoc est modifiée, la maintenance des tables de routage nécessite l'envoi périodique par chaque nœud de messages de signalisation indiquant sa présence à tous ses voisins. Ils se basent principalement sur deux méthodes à s'avoir : la *méthode état de lien et la méthode par vecteur de distance* [9].

Le protocole de routage à état de lien se base sur les informations rassemblées sur l'état des liens dans le réseau. Ces informations sont disséminées dans le réseau périodiquement ce qui permet ainsi aux noeuds de construire une carte complète du réseau. Un noeud qui reçoit les informations concernant l'état des liens, met à jour sa vision de la topologie du réseau et applique un algorithme de calcul des chemins optimaux afin de choisir le noeud suivant pour une destination donnée. Un exemple des algorithmes les plus connus appliqué dans le calcul des plus courts chemins, est celui de Dijkstra [8].

Par contre, dans la méthode par vecteur de distance, Chaque noeud diffuse à ses noeuds voisins sa vision des distances qui le séparent de tous les hôtes du réseau. En se basant sur les informations reçues par tous ses voisins, chaque noeud de routage fait un certain calcul pour trouver le chemin le plus court vers n'importe quelle destination. Le processus de calcul se répète, s'il y a un changement de la distance minimale séparant deux noeuds, jusqu'à ce que le réseau atteigne un état stable [12]. Cette technique est basée sur l'algorithme distribué de Bellman Ford., les principaux sont : DSDV et OLSR.

1.5.1.1. Le protocole DSDV (Destination Sequenced Distance Vector)

DSDV [13] est un protocole de type distance vector qui implique que chaque nœud partage sa table de routage avec ses voisins. Les informations de route comprennent la route, l'adresse de destination, la distance jusqu'à la destination en termes de saut.

L'algorithme considère une route R1 comme meilleure qu'une route R2 si elle a un numéro de séquence plus important c'est-à-dire la route est plus récente donc les informations ont plus de chances d'être correctes ou si le numéro de séquence est le même mais que la distance estimée vers la destination en empruntant la route R1 est plus faible. Chaque station émet sa table de routage régulièrement en incrémentant le numéro de séquence correspondant à la route menant à lui.

Les mises à jour des tables de routage sont transmises périodiquement à travers le réseau afin de maintenir la consistance des informations. Pour cela, la station peut émettre deux types de paquets: la table complète (*full dump*) où la station transmet la totalité de la table de routage aux voisins, ce qui nécessite l'envoi de plusieurs paquets de données ou uniquement

les parties de la table qui ont été modifiées (*incrémental*). Dans ce type, juste les entrées qui ont subi un changement par rapport à la dernière mise à jour, sont envoyées, ce qui réduit le nombre de paquets transmis.

1.5.1.2. Le protocole OLSR (*Optimized Link State Routing Protocol*)

Ce protocole est une optimisation des protocoles à état de liens adaptés à la nature des réseaux ad hoc. La technique clef de ce protocole est appelée MPR (pour MultiPoint Relay). Un MPR est un nœud choisi par son voisin pour transférer les messages de diffusion de ce dernier. Ainsi, au lieu que tous les voisins rediffusent les messages de diffusion, dans ce protocole il n'y a que les MPRs qui vont les rediffuser. Cette amélioration peut largement contribuer à réduire la charge du réseau par rapport aux protocoles du routage du type état de lien traditionnels. Une deuxième amélioration aussi reliée à MPR est que le nombre de messages déclarant les états de lien est diminué. Une troisième amélioration est qu'un nœud MPR déclare seulement les liens avec ses sélecteurs [14].

Dans OLSR, principalement deux types de messages sont introduits : "Hello" et "TC"(Topology Control). Périodiquement, chaque nœud diffuse localement un message Hello contenant des informations sur son voisinage et l'état des liens. Ceci permet à chaque nœud de prendre connaissance de son voisinage à un et deux sauts. L'ensemble MPR est alors construit dans chaque nœud de façon à contenir un sous-ensemble de voisins à un saut qui couvre tous les voisins à deux sauts. Afin de construire les tables nécessaires au routage des paquets, chaque nœud génère périodiquement un paquet TC contenant la liste de ses voisins l'ayant choisi comme MPR. Le message TC est diffusé dans l'ensemble du réseau. Seuls les voisins MPR rediffusent un paquet TC reçu pour éviter l'inondation. Cette technique prometteuse réduit considérablement l'overhead généré par le trafic de contrôle. A la réception d'un message TC, la table de topologie peut être construite. Basé sur la table de topologie, chaque nœud peut calculer la table de routage qui permet d'acheminer les paquets vers n'importe quelle destination dans le réseau [4].

1.5.2. Protocole de routages réactifs

Les algorithmes réactifs [11] suivent une politique radicalement opposée à celle des algorithmes proactifs qui essaient de maintenir au niveau de chaque nœud des tables de routages contenant les meilleurs chemins vers tous les nœuds du réseau, même si elles ne sont jamais utilisées, le protocole réactif (ou on-demande) : les routes ne sont crées que quand elles sont demandées, les nœuds mobiles ne maintiennent pratiquement pas d'informations sur la

topologie du réseau. Lorsqu'un nœud A désire communiquer avec un nœud D par exemple, il commence par demander la construction d'une route vers D, en envoyant un message particulier à tous ceux qui peuvent l'entendre. Ce message, transmis de proche en proche sur tout le réseau jusqu' à son arrivée à D, dans ce cas, le nœud D peut utiliser le chemin inverse pour contacter le nœud A et lui transmettre les informations de routage, la route crée reste valide tant que le nœud D est joignable ou jusqu'à ce que la route ne soit plus utilisée. Le mécanisme de découverte d'une route est basé principalement sur deux algorithmes à s'avoir : la méthode d'apprentissage en arrière (Backward Learning) et la méthode de routage source (Source Routing).

Dans la méthode d'apprentissage en arrière, le nœud source, qui est a la recherche d'un chemin vers la destination, diffuse par inondation une requête dans le réseau. Lors de la réception de la requête, les nœuds intermédiaires (ou de transit) essaient de faire apprendre le chemin au nœud source, et de sauvegarder la route dans la table envoyée. Une fois la destination est atteinte, elle peut envoyer une réponse en utilisant le chemin tracé par la requête, un chemin full duplex est alors établit entre le nœud source et le nœud destination. Une fois le chemin est calculé, il doit être sauvegardé et mis à jour au niveau de la source, tant qu'il est en cours d'utilisation.

Par contre dans la méthode de routage source, le noeud source détermine toute la liste des noeuds par lesquels doit transiter le message. En effet, afin de construire la route, le noeud source doit préciser les adresses exactes des noeuds par lesquels le message transitera jusqu'à atteindre la destination. Ainsi, le noeud source transmet le paquet au premier noeud spécifié dans la route. Notons que chaque noeud par lequel le paquet transite, supprime son adresse de l'entête du paquet avant de le retransmettre. Une fois que le paquet arrive à sa destination, il sera délivré à la couche réseau du dernier hôte [12].

Parmi les protocoles les plus connus de cette catégorie, nous pouvons citer DSR (Dynamic Source Routing) ou encore AODV (Ad hoc On-demand Distance Vector).

1.5.2.1. Le protocole DSR (*Dynamic Source Routing*)

Le protocole DSR utilisant l'algorithme routage par la source, et se base principalement sur deux mécanismes coopératifs : la découverte de route et la maintenance de route. Il permet aussi l'existence de plusieurs routes vers la destination.

Pour *la découverte du chemin,* un nœud source envoie un paquet *route request* à un nœud destination. Ce paquet se propage dans le réseau jusqu' à arriver au nœud destination ou à un

nœud connaissant un chemin vers celui-ci. Le paquet contient l'adresse source, l'adresse de destination, un numéro d'identification, ainsi qu'un champ dans lequel sera accumulée la séquence des nœuds visités durant la propagation de la requête dans le réseau (route record). Quand un nœud reçoit un paquet *route request*, il vérifie s'il connaît un chemin vers la destination. Si ce n'est pas le cas, il ajoute son adresse dans le route record et transmet le paquet à ses voisins. Ce transfert n'a lieu que si l'adresse du nœud n'apparaît pas déjà dans le route record, afin d'éviter les boucles et la multiplication des paquets *route request*. Quand le paquet atteint sa destination, le nœud ainsi atteint envoie un paquet en réponse via le chemin donné dans la route record, si les liaisons sont symétriques, ou via un autre chemin.

La maintenance de la route consiste à envoyer un paquet route error quand une route est inutilisable. Quand un nœud détecte un problème fatal de transmission, ce paquet, contenant l'adresse du nœud qui a détecté l'erreur et celle du nœud qui le suit dans le chemin, est envoyé à l'émetteur original du message. Lors de la réception du paquet route error par la source, le nœud concerné par l'erreur est supprimé du chemin sauvegardé, et tous les chemins qui contiennent ce nœud sont tronqués à ce point là. Par la suite, une nouvelle opération de découverte de routes vers la destination est initiée par l'émetteur [10].

1.5.2.2. Le protocole AODV (*Ad hoc On-demand Distance Vector*)

Ce protocole peut être considéré comme la combinaison de DSR et de DSDV. Il utilise les principes des numéros de séquence pour maintenir la consistance des informations de routage et employer les routes les plus récentes. Il emploie les mécanismes de découverte de chemin et de maintenance de route de DSR en y associant le numéro de séquence et les mises à jour périodiques de DSDV [10].

Si le nœud S veut communiquer avec le nœud D (il n'existe pas de route valide entre eux), le message RREQ (Route REQuest) est envoyé dans le réseau afin de trouver le chemin entre les deux nœuds. Chaque nœud intermédiaire enregistre dans sa table de routage l'adresse du nœud qui lui a transmis le RREQ, établissant ainsi le chemin de retour (Reverse Path). En cas, un nœud reçoit plusieurs copies d'un même RREQ, seule la première est conservée.

Une fois que le message atteint le nœud destination (ou un nœud connaissant un chemin vers celui-ci), ce nœud transmet un message RREP (Route REPly) vers la source par le Reverse Path : il parcourt donc le chemin en sens inverse en modifiant les tables de routage des nœuds par lesquels il passe. En effet, l'information à conserver dans la table de routage

est le nœud suivant dans le chemin (Forward Path) et pas le Reverse Path précédemment enregistré. À l'opposé de protocole DSR qui peut utiliser les liens symétriques ou asymétriques, l'AODV ne supporte que les liens symétriques car les paquets RREP est envoyé par le même chemin que le RREQ.

AODV permet aux nœuds d'envoyer périodiquement des messages Hello, qui sont utilisés par les nœuds afin de signaler leur présence. Ces messages permettent entre autre de détecter des cassures de route. A la rupture d'une route active, le protocole AODV tente de réparer la connectivité localement en diffusant une requête de recherche de route dans le voisinage. Si cette tentative échoue, alors, il envoie un message d'erreur (Route ERRor) vers la source et la route est détruite dans des tables de routage des nœuds intermédiaires. Ensuite, une nouvelle recherche de route est lancée par la source [11].

1.5.3. Les protocoles hybrides

Le principe d'algorithme hybride est de combiner les deux approches (proactif et réactif) afin de profiter les avantages des deux principes, tout en compensant leurs inconvénients. Il utilise un protocole proactif, pour apprendre le proche voisinage (par exemple, voisinage à deux ou à trois sauts), ainsi il dispose des routes immédiatement dans le voisinage. Au-delà de cette zone prédéfinie, le protocole hybride fait appel aux techniques des protocoles réactifs pour chercher des routes. Avec ce découpage, le réseau est partagé en plusieurs zones, et la recherche de route en mode réactif peut être améliorée. L'une des protocoles hybrides est le protocole ZRP.

1.5.3.1. Le protocole ZRP (*Zone Routing Protocol*)

ZRP [15] est un exemple de protocole hybride qui combine des approches proactives et réactives, essayant de ce fait de rassembler les avantages des deux approches. ZRP définit autour de chaque nœud une zone qui contient les nœuds voisins à un nombre donné de sauts du nœud. Des algorithmes proactifs et réactifs sont employés par le nœud pour acheminer les paquets, respectivement, dans et en dehors de la zone. Le routage au sein d'une zone se fait de manière proactive, via le protocole IARP (IntrAzone Routing Protocol) et le routage vers les nœuds extérieurs de la zone se fait de façon réactive, grâce au protocole IERP (IntErzone Routing Protocol).

La recherche des chemins s'effectue comme suit : on vérifie tout d'abord si le nœud destinataire se trouve dans la zone du nœud source (chaque nœud connaissant le contenu de sa zone), auquel cas le chemin est déjà connu. Autrement, une demande d'établissement de route

RREQ est initiée vers tous les nœuds périphériques, ces derniers vérifient si la destination existe dans leurs zones. Dans l'affirmative, la source recevra alors un paquet RREP contenant le chemin menant `a la destination. Dans le cas contraire, les nœuds périphériques diffusent la requête à leurs propres nœuds périphériques qui, à leur tour, effectuent le même traitement.

1.5.4. Évaluation des protocoles de routage

Comme nous l'avons vu, il existe une multitude de protocoles de routage différents, ayant chacun leurs caractéristiques propres. La question posée est la suivante : comment comparer ces méthodes, étant données leurs différences ? Les métriques de routage sont là pour tenter de répondre à cette question : il s'agit de méthodes employées pour évaluer les performances des protocoles des routages d'un réseau ad hoc. En effet, Ces métriques sont très utiles car elles permettent de montrer ce qu'il s'est réellement passé durant la simulation et donc de décrire au mieux le protocole utilisé. Les métriques les plus utilisées sont [16,10] :

- End-to-End Throughput : moyenne des transmissions réussies, c'est à dire la mesure du nombre de paquets transmis avec succès à leur destination dans un intervalle de temps donné.
- End-to-End Delay : mesure du temps mis pour que les paquets arrivent à destination.
- Link Utilisation : probabilité qu'un nœud soit utilisé pour effectuer la transmission.
- Packet Loss : pourcentage des paquets envoyés qui ne sont jamais arrivés à destination ou ayant été corrompus durant le transfert.
- Packet Delivery Fraction : mesure du rapport entre le nombre de paquets reçus par la destination et le nombre de paquets envoyés par la source.
- Routing Overhead : métrique très intéressante, qui mesure le nombre total de paquets de routage transmis pendant la simulation. Elle montre donc à quel point un protocole consomme de la bande passante avec ses messages de routage.
- Path Optimality : calcul de la différence entre le nombre de nœuds par lesquels un paquet est passé pour arriver à destination et le plus court chemin existant physiquement dans le réseau au moment de l'émission du paquet. Si les paquets passent souvent par des chemins proches du plus court, le protocole est dit être un bon protocole.

L'intérêt principal des protocoles proactifs et que l'on peut trouver facilement et rapidement le destinataire sans avoir à lancer une recherche dans le réseau. De plus, les informations collectées pour aider au routage peuvent s'avérer très utiles pour d'autres

applications et, comme les informations sont mises à jour régulièrement, les pertes de routes sont peu fréquentes.

Cependant, ces algorithmes sont tributaires d'une mise à jour des informations régulière et fiable, ce qui induit une charge constante du réseau, due aux messages de contrôle. Dans le cas de réseaux à mobilité forte, cette charge se révèle être une catastrophe car la quasi totalité de la bande passante est consacrée aux messages de contrôle et les applications n'ont plus assez de ressources. De plus, si le réseau est très grand, la quantité d'information à diffuser et à mémoriser devient également un problème.

Par contre, les protocoles réactifs s'avèrent efficace dans les réseaux de taille importante et/ou à forte mobilité. En effet, comme les routes sont construites à la demande, on évite une charge constante et importante du réseau due aux changements de topologie. Mais, si l'algorithme utilisé lors de l'étape de diffusion est inefficace, les performances de ces algorithmes peuvent devenir extrêmement mauvaises.

1.6. Les applications actuelles et futures des MANETs

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. Les applications des réseaux ad hoc peuvent être classées par catégorie comme suit [9,17, 18] :

- **Applications militaires** : le premier domaine d'application des réseaux ad hoc fut le domaine militaire, par le projet DARPA (Defence Advanced Research Projects Agency). Ce projet permet de déployer un mécanisme de communication entre les différents groupes d'unités par l'intermédiaire de véhicules qui communiquent ensemble par liaison radio. L'intérêt des militaires pour une telle technologie s'explique par le caractère particulièrement adapté des réseaux ad hoc aux situations hostiles. En effet, comme le réseau est auto organisé et qu'il ne nécessite pas d'infrastructure, il peut être déployé rapidement, sans difficulté et offrir une bonne tolérance aux pannes.
- **Application de secours et de délivrance :** les réseaux ad hoc à l'avenir pourraient être déployés dans des situations de secours et de délivrance où l'infrastructure fixe a pu être détruite à cause d'un tremblement de terre ou à toute autre catastrophe.

- **Applications commerciales :** il est évident que dans un contexte plus commercial les réseaux ad hoc peuvent également servir pour former des réseaux locaux. En effet, la mise en place du réseau est plus simple, pas de câble à tirer dans le bâtiment d'où une économie intéressante pour une entreprise. Dans le cas de réunions temporaires comme des conférences, l'organisation est simplifiée, toujours par la non nécessité de câblage. Cependant, des progrès sont encore à faire, notamment en ce qui concerne les couches physiques, car ce genre d'application peut nécessiter une bande passante importante, ce qui n'est pas encore fournit par WiFi à l'heure actuelle par exemple.
- **Applications informatique :** dans le cadre de l'informatique omniprésente, les réseaux ad hoc peuvent servir à relier entre eux tous les équipements de la maison ou établir les liens entre les différents composants informatiques des vêtements. Dans ce cas, on parle non plus de LAN (Local Area Network) mais de PAN (Personnal Area Network) et le routage ad hoc peut être utilisé pour faire communiquer tous les éléments grâce à des équipements de faible puissance, et donc de faible portée (ce qui est un avantage, tant au point de vue de la consommation énergétique qu'au point de vue de la santé de l'utilisateur).
- **Réseaux de capteur :** les réseaux ad hoc s'avèrent très utiles dans le cas de mesures en milieu hostile. On parle alors de réseaux de senseurs. Ce sont des équipements possédant des capacités très limitées (mémoire, processeur, bande passante) et de taille réduite. Ces équipements ont de nombreux domaines d'applications : médicales ou militaires par exemple. Ils sont en général utilisés en grande quantité, et les réseaux ad hoc permettent alors la liaison entre tous les objets. On peut citer l'exemple de capteurs météorologiques, de surveillance d'un site, de mesure des constantes d'un être humain ou de contrôle de structures (par exemple, des capteurs coulés dans le béton d'un pont).

En fin, Comme on peut le voir, les réseaux ad hoc ont un très large potentiel dans un futur proche et l'intérêt que porte la recherche pour ce domaine s'en explique donc très largement. De nombreux défis se posent avant de pouvoir utiliser ce type de réseaux dans toutes les applications citées plus haut.

1.7. Difficultés de déploiement des applications sur MANETs

Les réseaux mobiles ad hoc se diffèrent dans quelques aspects, toutefois ils présentent un ensemble de caractéristiques qui influent sur le déploiement d'une telle application. Leurs principales difficultés sont décrites par les points suivants [19] :

- **Caractéristiques des terminaux mobiles :** les applications mobiles fonctionnent sur des dispositifs qui ont des ressources rares à s'avoir : peu de mémoire, temps de traitement lent et de contrainte d'énergie. Sachant que d'autres fonctionnalités comme le routage ont besoin de ces ressources.
- **Gestion des déconnexions :** les terminaux mobiles se relient au réseau pendant des périodes courtes, principalement pour accéder à quelques données ou de demander un service. Par conséquent, il arrive que le client qui demande un service, et le serveur fournissant ce service, ne sont pas reliés en même temps. Dans cette situation, l'application doit réagir, par exemple, il pourrait être possible que le client qui demande un service, de quitter le réseau, et le résultat de la requête est rassemblé dans un terminal. Par conséquent le client peut la récupérer pour sa nouvelle reconnexion.
- **Topologie dynamique** : au différence des réseaux traditionnels, les réseaux mobiles ad hoc s'exécutent dans un contexte extrêmement dynamique. La largeur de la bande passante peut ne pas être stable, les services qui sont disponibles maintenant peuvent ne pas être disponible une seconde plus tard. Par conséquent, les applications doivent prendre en considération ces contraintes.
- **Scalabilité :** la scalabilité est l'un des grands problèmes lors de déploiement d'une application sur MANET. Elle peut être définie comme suivante: l'application doit être capable de maintenir des niveaux acceptables de performances quand le réseau MANET s'agrandie. En effet, l'application doit être assez flexible pour permettre l'ajout d'autres nœuds mobiles sans affecter dans ses performances.

1.8. Avantages et inconvénients des réseaux mobiles ad hoc

Les réseaux mobiles ad hoc sont utiles quand la connexion filaire n'est pas disponible, par exemple lors d'une opération militaire, et plus généralement quand le déploiement rapide d'un réseau est nécessaire. Dans ce cas, les nœuds communiquent en acheminant les messages par routage « multi-sauts ». Le mode ad hoc multi sauts a de nombreux avantages en comparaison des modes de communication avec stations de base [11] :

- ✓ **Pas de câblage :** l'une des caractéristiques des réseaux Ad hoc est l'absence d'un câblage, et ce en éliminant toutes les connexions filaires qui sont remplacées par des connexions radio ou infrarouge.

- ✓ **Déploiement facile** : contrairement aux réseaux cellulaires qui requièrent un important effort de planification pour leur déploiement, les réseaux ad hoc peuvent être déployés facilement et rapidement en permettant des échanges directs entre stations mobiles.
- ✓ **La mobilité** : comme l'indique leur nom, les nœuds peuvent se déplacer librement a condition de ne pas s'éloigner trop les uns des autres pour garder la connectivité du réseau.
- ✓ **Extensible** : un réseau mobile ad hoc peut s'étendre dans sa taille et ceci d'une manière facile. En effet, l'ajout d'un nouveau nœud nécessite quelques configurations pour qu'il fonctionne en sein 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.

Néanmoins, les réseaux mobiles ad hoc présentent des inconvénients peuvent être résumés par les points suivants :

- × **Topologie non prédictible :** le changement rapide de leur topologie dû au aux déplacements des nœuds rendent leur étude très difficile.
- × **Sécurité** : les réseaux ad hoc sont connus pour leur manque d'organisation, de planning et de configuration, ainsi que, les possibilités de s'insérer dans le réseau sont plus grandes, la détection d'une intrusion ou d'un déni de service plus délicate. Donc ils sont généralement considérés difficiles à sécuriser.
- × **Taux d'erreur important** : ce taux d'erreur dépend des collisions qui augmentent avec le nombre de nœuds partagent le médium radio.
- × **Capacités limitées** (puissance de calcul, mémoire, énergie) : dans un tel réseau, il faut trouver un équilibre entre la connexité du réseau et la consommation énergétique. En effet, il faut que la portée d'émission des nœuds soit suffisante pour assurer la connexité du réseau. Mais plus on accroît la portée des mobiles, plus les communications demandent de l'énergie.

1.9. Conclusion

Le concept de réseaux mobile ad hoc est très prometteur comme nouveau mode de télécommunication propre à compléter et à étendre les systèmes de communication existants. Ces réseaux sont constitués de terminaux mobiles qui sont généralement de petites tailles, d'où les contraintes de ressources en terme de mémoire et de batteries et peuvent se déplacer de manière aléatoire et à des vitesses quelconques (aucun infrastructure existante ou administration centralisées). Chaque terminal communique directement avec sa voisine, pour communiquer avec d'autre terminal, il lui nécessaire de faire passer ses données par d'autres qui se chargeront de les acheminer. Pour cela, il est d'abord primordial que les terminaux se situent les unes par rapport aux autres, et soient capables de construire des routes entre elles : c'est le rôle du protocole de routage.

Les réseaux ad hoc sont idéals pour les applications caractérisées par une absence d'une infrastructure préexistante, tel que les applications militaires et les autres applications de tactique comme les opérations de secours (incendies, tremblement de terre…..) et les missions d'exploration.

En fin, le domaine des réseaux ad hoc est en plein essor. Il ouvre des perspectives de recherche importantes et une palette de nouvelles applications surtout très intéressantes en mobilité. Dans le chapitre suivant nous allons décrire et s'intéresser à la qualité de service dans les MANETs.

Chapitre 2

QoS dans les Réseaux Mobiles Ad hoc

2.1. Introduction

Depuis l'apparition et le grand succès commercial de la téléphonie cellulaire, le développement d'équipements mobiles n'a cessé de prendre de l'importance. Grâce aux assistants personnels et aux ordinateurs portables, l'utilisateur devient de plus en plus nomade.

Les débits atteints actuellement avec les réseaux sans fil rendent possible le transfert de flux multimédia soumis à de fortes contraintes. Ainsi, le respect de certaines contraintes telles que la bande passante, le délai ou encore le taux de pertes de paquets devient primordial. Cependant, les solutions qui ont été introduites dans le monde des réseaux filaires deviennent inadaptées pour des réseaux utilisant un médium radio partagé sans aucune administration centralisée.

En effet, l'émergence des services multimédia temps réel, et les champs variés des applications des réseaux ad hoc, la qualité de service dans les réseaux ad hoc est devenu un thème de recherche qui a suscité beaucoup d'intérêts. Dans ce contexte, des travaux pour l'introduction des applications multimédia dans les réseaux ad hoc ont été proposés. Cependant, il est très difficile de garantir une quelconque qualité de service à une application temps réel dans un réseau ad hoc, car il faut prendre en considération les spécificités de ces réseaux, à savoir : la bande passante limitée, le changement dynamique de la topologie en fonction du temps, ainsi que le manque d'information complète sur l'état du réseau. En outre, la communication entre les stations mobiles étant par voix radio, la qualité du lien sans fil reste peu fiable, et susceptible à des variations suivant la configuration et l'état du réseau.

Le fait est que, de nombreuses applications nécessitent le support de la qualité de service. Par exemple, garantir une borne limitée du délai peut être profitable aux applications de téléphonie, garantir le débit peut être ainsi nécessaire pour les applications de vidéo à la demande, les applications militaire exigent des protections strictes, les applications de secours demandent une fiabilité du support avec un débit acceptable, la communication dans un couloire de conférence nécessite une gigue stable avec une consommation minimale d'énergie,…etc.

Les travaux existants sur les protocoles de routage avec QoS dans les réseaux ad hoc supposent que chaque noeud soit capable d'évaluer plusieurs métriques pour ses différents liens avec ses voisins. À partir de ces informations, il est alors possible de calculer, plus ou moins facilement, les meilleures routes avec la QoS demandée. Par exemple, si l'estimation donne un délai entre deux noeuds supérieur à la demande de l'application, les algorithmes doivent trouver une nouvelle route, si elle existe.

Dans ce qui suit nous allons définir de façon claire ce que signifie le terme de qualité de service dans les réseaux mobiles ad hoc et leurs modèles. Puis, nous intéressons au problème spécifique du routage avec qualité de service pour ces réseaux car ce point soulève beaucoup de problèmes auxquels il n'existe pas encore de solution satisfaisante.

2.2. Niveau de service

Le terme « niveau de service » (en anglais Service level) définit le niveau d'exigence pour la capacité d'un réseau à fournir un service point à point ou de bout en bout avec un trafic donné [20]. On définit généralement trois niveaux de QoS [21] :

- **Service à meilleur effort (en anglais best effort) :** le réseau fera de son mieux pour améliorer la QoS fournie mais ne donne aucun engagement pour y parvenir.
- **Service garanti (ou déterministe) :** la QoS demandée doit être garantie par le fournisseur de service. Ce niveau est généralement exigé par les applications temps réel critiques (strictes).
- **Service probabiliste ou statistique :** les paramètres de QoS sont spécifiés par des probabilités ou des contraintes sur la moyenne, variance etc. pour exprimer une certaine tolérance au non respect de la QoS demandée.

2.3. Notion de qualité de service (QoS : quality of service)

Dans les réseaux de télécommunication, l'objectif de la qualité de service est d'atteindre un meilleur comportement de la communication, pour que le contenu de cette dernière soit correctement acheminé, et les ressources du réseau utilisé d'une façon optimale.

La qualité de service QoS (Quality of Service) peut être définie comme le degré de satisfaction d'un utilisateur des services fournis par un système de communication [4]. La QoS est définie dans [22] comme la capacité d'un élément du réseau (ex : routeur, nœud ou une application) de fournir un niveau de garantie pour un acheminement des données.

Un autre définition donné par recommandation du CCITT (Comité Consultatif International Téléphonique et Télégraphique) comme "l'effet général de la performance d'un service qui détermine le degré de satisfaction d'un utilisateur du service" [23]. Cette définition donne une perception de la qualité de service du point de vue utilisateur. D'un point de vue technique, la qualité de service peut être définie comme la capacité de garantir un certain niveau d'assurance, de telle sorte que la fluidité des trafics et/ou des services requis soit au mieux satisfaite pour une application, un hôte ou même un routeur. Cette qualité de service peut également correspondre dans un réseau, à un ensemble de mécanismes permettant de partager équitablement selon les besoins requis des applications, les différentes ressources offertes par le réseau, de manière à donner, autant que possible, à chaque application la qualité dont elle a besoin.

Le RFC 2386 [24] caractérise la QoS comme un ensemble de besoins à assurer par le réseau pour le transport d'un trafic d'une source à une destination. Ces besoins peuvent être traduits en un ensemble d'attributs pré-spécifiés et mesurables en terme de :

- ✓ Délai de bout en bout.
- ✓ Variance de délai (gigue).
- ✓ Bande passante.
- ✓ Pertes de paquets.

En effet, la notion de qualité de service (QoS) dans les réseaux se traduit en terme de paramètres (appelés aussi métrique ou critères) mesurant les performances de transmissions des données de la source vers la destination. Nous parlons dans cette partie les paramètres qui s'avèrent fondamentaux dans les réseaux ad hoc, et qu'ils doivent être en compte lors du déploiement des mécanismes pour l'amélioration de qualité de service dans ce type de réseaux.

2.4. Paramètres de la qualité de service

La qualité de service (QoS) est basée en général sur un certain nombre de paramètres, de natures différentes et qui ont pour but de préciser les besoins des utilisateurs envers les fournisseurs de service. Les paramètres permettant de spécifier la QoS diffèrent selon le type de réseaux et les exigences des usagers et des applications. Par exemple, pour les applications en temps réel, comme l'échange de flux sonore et vidéo, les paramètres les plus considérés sont : la largeur de la bande passante, la gigue, et le délai de bout en bout, la communication

dans un couloire de conférence demande une communication minimale d'énergie où la durée de vie de la batterie dans ce cas est le paramètre clé, la disponibilité du réseau est le paramètre principale pour les opérations d'urgence et de sauvetage. Or, les paramètres [25] les plus importants permettant l'expression des besoins de QoS dans le cas des réseaux ad hoc sont :

- **Le débit (bandwidth) :** est la quantité de données transmises d'une source vers une destination dans une unité de temps. Le débit dépend de la traitement des terminaux, le téléchargement d'une application volumineuse nécessite une assez large bande passante pour récupérer les fichiers de l'application le plus vite possible.
- **Le délai (delay) :** représente la durée séparant l'envoi d'un paquet par un émetteur et sa réception par le destinataire, cette durée résulte essentiellement de la distance physique entre l'émetteur et le récepteur, la qualité des liens, la taille des paquets et la charge du réseau.
- **La gigue (jitter) :** la gigue désigne la variation du délai de bout en bout au cours de la transmission, une forte gigue engendre généralement des perturbations dans l'exécution des applications multimédia.
- **Le taux de perte (packet loss) :** c'est la proportion des paquets qui ne parviennent pas à leur destination par rapport aux paquets émis. Ces pertes dépendent particulièrement de la fiabilité du support des liaisons et de la surcharge locale (congestion).
- **L'énergie :** les exigences en consommation minimale d'énergie sont cruciales pour les réseaux ad hoc car les terminaux sont alimentés par des sources limitées (batteries,….) et, en plus de leurs fonctionnalités classiques, ils assurent l'acheminement des paquets de données qui consomment une partie importante d'énergie.
- **La sécurité :** permet d'exprimer notamment le degré de protection, le contrôle d'accès, l'authentification et la confidentialité. Il faut signaler que beaucoup des chercheurs spécialisés dans ce domaine ne considèrent pas la sécurité comme un paramètre de la QoS mais comme un aspect à part entière.

2.5. Modèles de qualité de service

Actuellement les travaux qui ont été réalisés pour offrir une meilleure qualité de service pour les applications multimédia reposent sur certains aspects liés aux réseaux ad hoc qui sont, les modèles de qualité de service pour les MANETs en plus de l'introduction de la différenciation de service au niveau de la couche MAC [23]. IntServ / RSVP et DiffServ sont

les modèles de qualité de service proposés par l'IETF (The Internet Engineering Task Force) pour les réseaux filaires.

L'application du modèle IntServ dans MANET s'avère inadaptée à l'environnement ad hoc. Ceci est justifié du fait que les capacités des noeuds mobiles sont trop variables et limitées pour supporter un traitement complexe et gérer les réservations ainsi que les états des communications en cours. De plus, une réservation dans les réseaux filaires est différente de celle d'un réseau mobile sans fil, car les liens sont partagés, limités, et susceptibles à des variations spatio-temporelles [4].

Le modèle DiffServ semble plus adapté. Cependant, il a été conçu pour des cœurs de réseaux possédant une bande passante importante et dont la topologie est relativement statique. Ces deux contraintes restent difficiles à satisfaire.

2.5.1. Qu'est ce qu'un modèle de QoS ?

Un modèle de qualité de service définit quels types de service peuvent être fournis dans un réseau et certains mécanismes utilisés afin d'offrir ces services (quelles fonctionnalités doit fournir le protocole de routage, quelle est l'architecture des nœuds, etc.) [26].

En d'autre terme, un modèle de qualité de service décrit un ensemble de services bout en bout, qui permettent aux clients de sélectionner un nombre de garanties qui gouvernent des propriétés telles que le temps, l'ordonnancement et la fiabilité [23]. Le modèle de qualité de service spécifie l'architecture qui doit prendre en considération les défis imposés par les réseaux ad hoc, comme le changement de la topologie et les contraintes de délai et de fiabilité.

2.5.2. Le modèle FQMM

Le modèle FQMM (Flexible QoS Model for MANETs) fut le premier modèle de qualité de service proposé pour les réseaux ad hoc en 2000. Il s'agit d'un modèle hybride combinant les propriétés des modèles IntServ et DiffServ mais adapté aux réseaux ad hoc de petite ou moyenne taille (environ 50 nœud) [27]. Il définit trois type de nœud : nœud d'entrée, intermédiaire, sortie. Les noeuds d'entrée permettent de marquer et classifier les paquets, qui seront ensuite relayés par les noeuds intermédiaires suivant leurs PHB (Per Hop Behavior), jusqu'à arriver au noeud destinataire. FQMM requiert l'utilisation d'un protocole de routage capable d'offrir une certaine qualité de service, c'est à dire capable de rechercher des routes satisfaisant certaines contraintes. La figure 2.1 dans [4] montre deux plans peuvent être distingués, le plans de relayage de données et le plans de contrôle et de gestion.

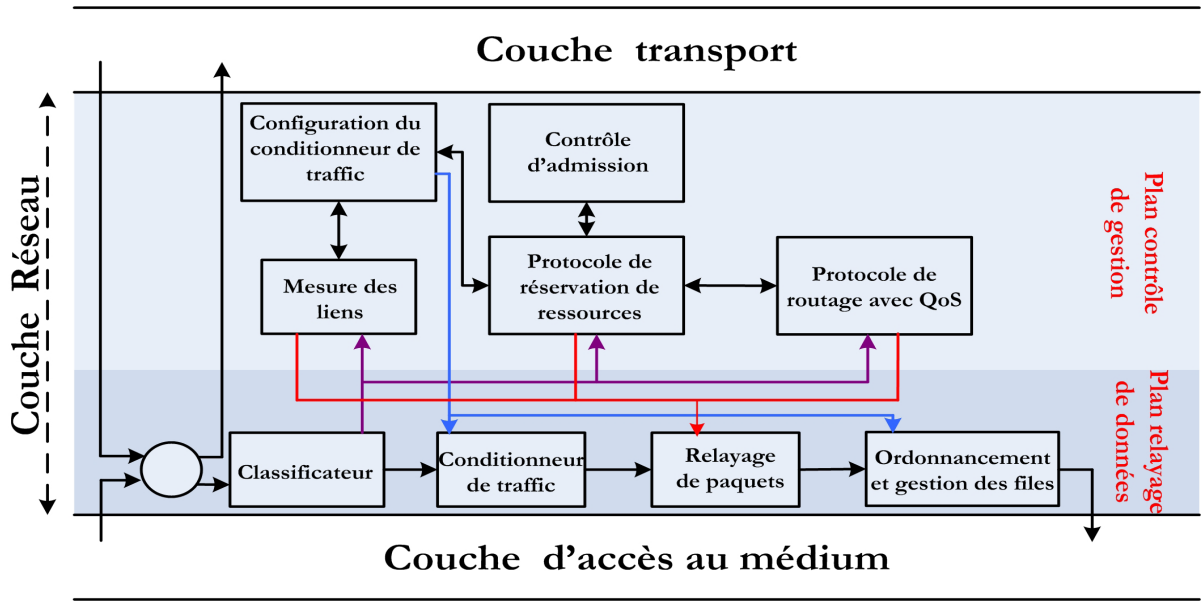

FIG. 2.1 – Le modèle FQMM

L'avantage d'une telle approche est la possibilité d'interfacer le réseau avec l'Internet, vu les mécanismes de qualité de services offerts qui sont proches des protocoles filaires [4].

En revanche, ce modèle souffre de plusieurs problèmes [23] : l'absence de tout contrôle explicite du nombre de services par flux offerts pose un problème de scalabilité, comme dans le cas du modèle IntServ. Il est très difficile de faire un profil dynamiquement négocié du trafic. La résolution de la plupart des problèmes liés au fonctionnement ad hoc (tels que le volume de signalisation, la consommation d'énergie et la bande passante limitée) est laissée à la charge du protocole de routage sous-jacent.

2.5.3. Le modèle SWAN

Selon [28], le modèle SWAN (Service Differentiation in Wireless Ad Hoc Networks) est un modèle réseau sans état qui a été proposé en 2002. SWAN [4, 23, 27] basé sur des algorithmes de contrôle distribués dans le but d'assurer une différenciation de services dans les réseaux mobiles ad hoc de façon simple et robuste. Il offre la priorité (au niveau paquet) aux trafics temps réel en contrôlant la quantité de trafics best effort acceptée par noeud. Pour accepter un nouveau trafic temps réel, le contrôle d'admission sonde la bande passante minimale disponible sur la route (valide et obtenu par un protocole de routage). Une décision à la source est alors prise suivant la bande passante obtenue. Pour maintenir la qualité de service des trafics déjà acceptés, le débit des trafics best effort est régulé en utilisant les mesures de délais au niveau MAC comme paramètre, dans le but de maintenir la qualité de service des trafics déjà acceptés. Dans le cas d'une congestion et afin de permettre à la source

de re-initier le contrôle d'admission, un bit de l'entête IP appelé ECN (Explicit Congestion Notification) est utilisé. Cependant, ce modèle n'apporte aucune garantie quant au maintien de la communication entre deux entités pour un trafic en cours : en fonction des variations de la bande passante, le trafic est maintenu ou coupé. De plus, le protocole de routage utilisé est de type Best Effort, ce qui signifie que lorsqu'un paquet est envoyé, il n'y a aucune vérification quant à son arrivée à destination et donc aucune assurance vis-à-vis de la "livraison" d'un paquet.

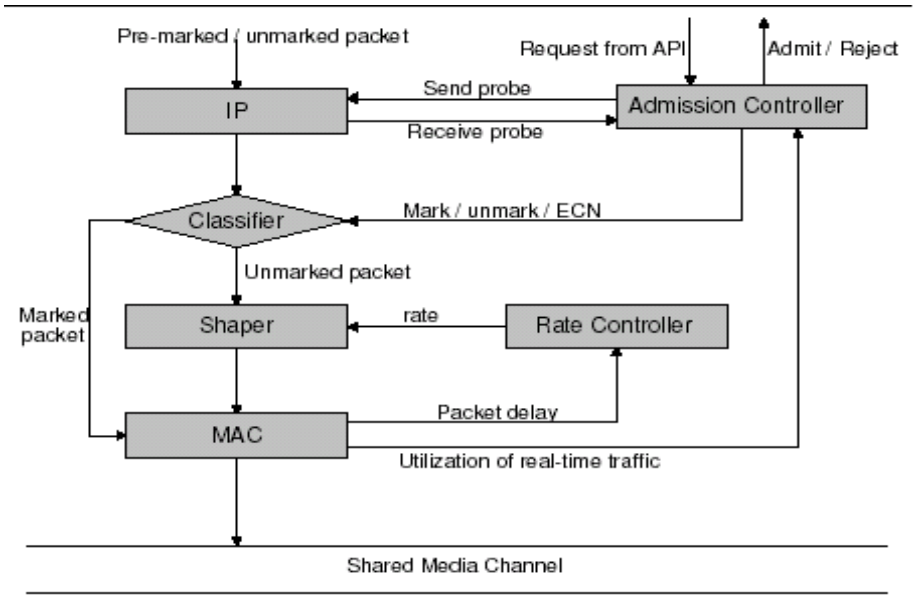

FIG. 2.2 – Le modèle SWAN

2.5.4. Le modèle iMAQ

Le modèle iMAQ (integrated Mobile Ad hoc QoS framework) apporte une solution en matière de qualité de service pour le transfert de données multimédias dans un MANET [26]. Le modèle inclut (voir fig.2.3) une couche ad hoc de routage et une couche de service logiciel (Middleware). Dans chaque noeud, ces deux couches partagent les informations et communiquent afin de fournir les garanties de QoS aux trafics multimédia. Le protocole de routage est basé sur la prédiction de la position des noeuds (predictive location-based) et orienté QoS.

La couche Middleware communique également avec la couche application et la couche réseau et essaye de prévoir le partitionnement du réseau. Pour fournir une meilleure accessibilité aux données, il réplique les données entre les différents groupes du réseau avant d'effectuer le partitionnement [4].

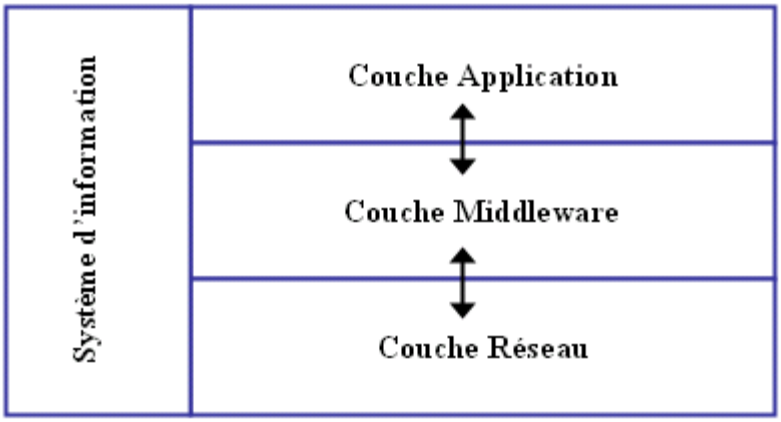

FIG. 2.3 – Le modèle iMAQ

2.6. Routage avec qualité de service

Il est clair que, les MANETs (Mobile Ad hoc Network) posent des problèmes spécifiques ayant une influence importante sur les solutions à mettre en place pour assurer la QoS. Les problèmes principaux sont : la mobilités des noeuds et l'incertitude des liens. En effet, la mobilité des noeuds provoque des changements fréquents de la topologie du réseau, de même que les interférences sur les liens radio, aboutissent à la création ou la disparition de certaines routes. De plus l'incertitude sur les liens témoigne de l'absence de garantie sur le débit de chaque lien, ou même du délai d'acheminement des paquets.

Le routage avec QoS (QoS Routing) peut-être défini comme le mécanisme par lequel les chemins associés aux flux sont déterminés à la fois par la connaissance des ressources disponibles et par les demandes en terme de QoS de ces flux. Pour le routage avec QoS [23]. On ajoute un certain nombre de contraintes (délai, bande passante, fiabilité, etc..) sur les routes afin de déterminer leur éligibilité. Par exemple, on peut vouloir rechercher une route disposant d'une certaine quantité de bande passante pour un trafic vidéo. On peut vouloir rechercher une route assurant que les paquets seront reçus par la destination moins d'un certain temps après leur émission par la source. Toute route satisfaisant un certain critère quantitatif peut être qualifiée de route assurant une certaine qualité de service. Plusieurs protocoles de routage avec QoS pour les MANETs ont été proposés.

2.6.1. Protocoles de routage avec QoS pour les MANETs

Les algorithmes de routage traditionnels discutés dans le chapitre précédent ont été proposés pour router les données sans tenir compte des contraintes spécifiques ou à des demandes des utilisateurs. Ainsi, ils sont inadaptés aux applications qui nécessitent le support de la qualité de service. En effet, le but des protocoles de routage avec support de la qualité de

service est de trouver une route ayant suffisamment de ressources pour satisfaire les besoins de QoS d'une communication, tout en optimisant l'utilisation des ressources disponibles.

2.6.1.1. Quelque protocoles basées sur la topologie de cluster

Le clustring ou regroupement a été utilisé pour différents objectifs comme la mise à l'échelle des réseaux ad hoc, l'abstraction de la topologie pour le contrôle de l'inondation dans les réseaux, la collecte d'information et le partage de la bande passante [29].

2.6.1.1.1. CGSR (Cluster-Gateway Switching Routing)

Ce protocole s'articule autour d'une architecture basée sur un regroupement des stations en clusters. Chacun de ces clusters possède un chef qui se charge des communications à l'intérieur de son propre cluster et maintient les informations de routage lui permettant de joindre les chefs des autres clusters. Le protocole CGRS utilise l'algorithme LLC (Least Cluster Change) pour empêcher le changement fréquent du clusterhead. Dans cet algorithme, les clusterheads peuvent changer leur rôle seulement dans les deux conditions suivantes : deux clusterheads devient des voisins ou un nœud devient déconnecté de n'importe quel groupe.

Cette approche trouve son intérêt dans le cas où les stations d'un même cluster se déplacent peu les unes par rapport aux autres, car il y a peu de changement dans la topologie du cluster. Par exemple, un tel protocole de routage peut être parfaitement adapté à une situation de PAN (Personnal Area Network). Cependant, si les clusters sont amenés à être modifiés trop fréquemment à cause de la mobilité, cette approche s'avère être trop coûteuse pour la mise à jour des informations.

2.6.1.1.2. CBRP (Cluster based Routing Protocol)

Comme son nom l'indique, le protocole CBRP (Cluster Based Routing Protocol) est basé sur la création de groupes (clusters) au sein du réseau. Les noeuds peuvent se voir attribuer des rôles particuliers au sein du réseau grâce à l'échange de message "Hello". Un noeud peut être élu chef de groupe (clusterhead), ou passerelle (gateway) entre groupes suivant sa situation dans le réseau, principalement sa visibilité des autres noeuds. Les routes sont établies à la demande, comme pour un protocole réactif mais uniquement par les chefs de groupe : si un noeud veut envoyer un message, il demande à son responsable de groupe de lui déterminer la route à suivre. La route trouvée est agrégée au fur et à mesure du message de découverte et le nœud destinataire connaît ainsi le chemin de retour [30]. En effet, chaque

nœud du réseau maintient deux tables : une table de ses 1-voisins, et une table des groupes adjacents composée de la liste des groupes adjacents et de leur coordinateur respectif. Ces deux tables sont maintenues à jour grâce aux messages hello diffusés périodiquement par chaque nœud sur deux sauts. Ce message contient l'état du nœud (du point de vue de son appartenance à un groupe), l'identifiant du coordinateur du (ou des) groupe(s) au(x)quel(s) appartient le nœud, une liste des nœuds 1-voisins et une liste des groupes adjacents incluant l'identifiant de leur coordinateur respectif. Lorsqu'un nœud reçoit un message hello, il rafraîchit ses deux tables [31]. Parmi les avantages de CBRP nous trouvons [21] :

- Réduction du nombre de messages de contrôle induits par les processus de découvertes des routes car la découverte est effectuée seulement par les clusterheads.
- Exploitation des liens bidirectionnels, et par conséquent, l'augmentation de la disponibilité des services.

Cependant, CBRP est présent certain désavantage, comme les suivantes [6] :

- L'ensemble des passerelles et clusterheads forme un ensemble redondant et donc crée un trafic de contrôle important.
- La route est constituée d'une liste de nœuds, et non de clusters : il suffit qu'un seul nœud se déplace pour que la route se casse. La hiérarchie n'est pas pleinement exploitée.
- Lorsqu'un nœud relayant un paquet n'obtient aucun acquittement du prochain saut, il enclenche une reconstruction locale de route en essayant d'atteindre le prochain saut via un intermédiaire. La route est donc obligatoirement allongée à chaque reconstruction.

2.6.1.2. Quelque protocoles de routage avec QoS

2.6.1.2.1. CEDAR (Core-Extraction Distributed Ad hoc Routing Algorithm)

Le protocole CEDAR [32] est un protocoles de routage réactifs qui est adapté au dynamisme rencontré dans les réseaux MANETs et assurant une qualité de service au niveau de la bande passante. Il est repose sur le principe d'élection dynamique d'un coeur de réseau stable. Des informations sur les liens stables disposant d'une grande bande passante sont propagées entre les noeuds du coeur. Le calcul des routes est effectué par les noeuds du réseau coeur en utilisant des informations locales. Il est basé sur trois composantes essentielles [33] :

- **Extraction d'un coeur du réseau (Core extraction) :** le réseau choisit les nœuds dynamiquement qui feront parti du réseau de coeur (Dominating Set) pour calculer les routes et maintenir l'état des liens du réseau. L'avantage d'une telle approche est qu'avec

un ensemble réduit de noeuds les échanges d'informations d'état et de route seront minimisés, évitant ainsi des messages supplémentaires circulant dans le réseau. En outre, lors d'un changement de route, seuls les noeuds du cœur serviront au calcul.

- **Propagation d'état de lien (Link state propagation) :** chaque noeud du coeur maintient sa topologie locale a jour. Les liens stables et considérés à forte bande passante sont fortement propagés dans le coeur du réseau tandis que les liens moins performants ne restent diffusés que localement (les noeuds n'ont pas une information sur la topologie globale du réseau).
- **Calcul de route (Route computation) :** celui-ci est basé sur la découverte et l'établissement d'un plus court chemin vers la destination satisfaisant la bande passante demandée. Des routes de 'secours' sont utilisées lors de la reconstruction de la route principale, quand cette dernière est perdue. La reconstruction peut être locale (à l'endroit de la cassure), ou à l'initiative de la source.

Au lieu de calculer une route avec un minimum de saut, l'objectif principal de CEDAR est de trouver un chemin stable pour garantir plus de bande passante. Dans ce protocole de routage, les noeuds du coeur du réseau auront plus de trafics à gérer, en plus des messages de contrôle (pour la découverte et la maintenance des routes). En outre, en cas de forte mobilité, la convergence de l'algorithme est difficile à atteindre [4].

2.6.1.2.2. QOLSR (QoS Optimized Link State Routing Protocol)

QOSLR [33], est une version avec QoS du protocole du routage OLSR, où des extensions sont ajoutées aux messages OLSR pendant le processus de découverte du voisinage. Il est pertinent d'intégrer des paramètres tels que le délai, la bande passante, le coût du lien, la perte de paquet. Les messages de contrôle TC (diffusés par les MPRs pour annoncer l'ensemble des noeuds qu'il peut atteindre) intègrent des informations de métrique additive. Une route avec un délai minimum peut être trouvé en utilisant l'algorithme Dijkstra, si on veut inclure la métrique de bande passante, alors si il existe plus d'une route avec la BP maximale, la route avec le délai minimale sera choisie.

2.6.1.2.3. TBP (Ticket Based Probing)

Les réseaux ad hoc disposent d'une bande passante limitée, or la recherche de routes par diffusion est très coûteuse en terme de bande passante car elle nécessite de nombreux échanges de messages. Le but de Ticket Based Probing [26] est de limiter ce surcoût et de

fournir des garanties de qualité de service. Il diffuse des messages pour la découverte de route, en publiant un certain nombre de tickets logiques. Les tickets représentent le nombre de routes à explorer en parallèle. Cette diffusion est limitée car chaque sonde porte une quantité totale de tickets, définie par la source. Afin d'augmenter la probabilité de trouver une route, on utilise deux types de tickets : les tickets jaunes permettent de rechercher des chemins respectant la contrainte imposée et les tickets verts permettent d'obtenir des solutions de faible coût.

Chaque message de découverte (ou d'observation) de route doit avoir au moins un ticket. Quand un message arrive à un noeud, il peut être divisé en plusieurs messages d'observation, qui sont relayés vers les prochains sauts. Chaque message fils contiendra un sous ensemble des tickets de son message père. Evidemment, un message ayant un seul ticket ne peut être divisé. Lors de l'arrivée d'un message de découverte de route à la destination, le chemin saut par saut est connu et les informations de délai ou de bande passante peuvent être utilisées pour effectuer la réservation de ressources pour la route répondant aux besoins de QoS. Le nombre de tickets généré est fonction de la précision des informations d'états disponibles à la source et les besoins de QoS de la communication. Plus de tickets sont publiés dans le but d'augmenter la chance de trouver un chemin désiré [4].

Malgré le fait que les nœuds ne connaissent que leur voisinage immédiat, Ticket Based

Probing est efficace puisqu'il permet de trouver des routes avec une probabilité proche des algorithmes basés sur l'inondation du réseau et meilleure que des algorithmes recherchant un plus court chemin. Il permet en outre de trouver des routes de plus faible coût que ces deux types d'algorithmes.

2.6.1.2.4. DSDV+ (Destination Sequenced Distance Vector avec QoS)

C'est un protocole de routage avec QoS et réservation de ressources, fondé sur DSDV (proactif, vecteur de distance). Selon [26] ce protocole tente de résoudre les problèmes de station cachée par une allocation dynamique d'unités TDMA (division du temps en unités appelées slots). Lors d'une demande de réservation, le protocole proposé évalue la quantité de bande passante disponible sur la route principale fournie par le protocole de routage DSDV en évaluant le nombre d'unités TDMA disponibles sur chaque lien tout au long de la route.

Afin de résoudre les problèmes de stations cachées, il est nécessaire de ne pas utiliser les mêmes unités pour les transmissions dans deux liens adjacents. Aussi, une politique d'allocation d'unités est utilisée dès l'établissement de la route. Lorsque la demande de route

arrive au destinataire, ce dernier renvoie à l'émetteur une confirmation contenant la politique d'allocation des unités sur le chemin. Les ressources sont alors effectivement réservées au fur et à mesure que ce message traverse le réseau en direction de l'émetteur. Les problèmes liés à la mobilité sont traités en maintenant une route secondaire non optimale en terme de nombre de sauts.

2.6.1.2.5. BRuIT (Bandwidth Reservation under InTerferences influence)

L'objectif du protocole BRuIT [34] est d'intégrer le support de la qualité de service aux réseaux ad hoc en tenant compte le plus justement possible des ressources réellement disponibles, compte tenu des interférences entre transmissions. En effet, ce protocole n'est pas un protocole de routage à proprement parler mais plutôt un protocole de réservation de bande passante s'appuyant sur un protocole de routage réactif basique. Son fonctionnement s'appuie sur deux phases :

- D'abord une phase de "découverte des voisins" qui leur permet de s'échanger leur état de charge respectif (e.g. la valeur totale de leur bande passante déjà réservée). Cette phase permet à chaque noeud de disposer de l'état de charge de son environnement radio.
- Ensuite vient la phase de réservation de ressources qui est effectuée par l'ouverture d'une route sur laquelle les ressources nécessaires au flux seront réservées. Au niveau de chaque noeud, le contrôle d'admission va se faire en fonction de la bande passante disponible et de la charge du medium radio.

Une différence fondamentale entre BRuIT et les autres protocoles de routage existants : alors que, traditionnellement, le routage est effectué par destination, BRuIT fait du routage par flux. En d'autres termes, avec des protocoles de routage classiques dans les réseaux ad hoc, tous les paquets échangés entre deux noeuds A et B sont sur la même route. Avec BRuIT, au contraire, deux connexions différentes ouvertes par A `a destination de B peuvent emprunter des routes différentes. Autrement dit, une entrée dans la table de routage BRuIT n'est plus uniquement déterminée par l'adresse du destinataire, mais par un triplet (adresse source, adresse-destination, identifiant-flux).

2.6.2. Evaluation de protocole de routage avec QoS

Comme nous l'avons vu, il existe plusieurs paramètres (appelés aussi métriques ou critères) qu'ils doivent être pris en compte lors de l'évaluation. Chaque protocole présenté ici

ne traite qu'un aspect particulier de la transmission dans les réseaux ad hoc. De plus, les tests effectués pour évaluer chaque protocole sont très différents [26, 4,34].

Dans le fait, la bande passante dans les réseaux filaires est toujours la ressource la plus critique et par conséquent, les protocoles étudiés proposent des garanties (fermes ou non) sur cette ressource. Dans les réseaux ad hoc, il est important de distinguer les protocoles qui offrent des garanties ferme par des mécanismes de type "réservation" et les autres. BRuIT et DSDV+ apportent de la QoS "ferme" en terme de bande passante et utilise de la signalisation pour réserver les ressources sur la route alors que TBP utilise un mécanisme de sonde qui cherche les routes dont les ressources sont suffisantes pour satisfaire la demande de bande passante.

Un paramètre important à prendre en compte lors de l'évaluation des performances de ces protocoles est leur mécanisme de réaction au changement de topologie. C'est en effet un des problèmes de base des réseaux MANETs. QOLSR, effectuant un routage "saut par saut", il n'y a pas de réservation établie et n'est donc pas sensible au changement de topologie. Tout comme sur Internet, si un lien est rompu le paquet prend un autre chemin. A l'inverse BRuIT et CEDAR doivent recalculer des routes si un lien est rompu. BRuIT remet la route à zéro pour rétablir une réservation complète alors que CEDAR qui ne propose pas de garanties fermes se contente d'effectuer un recalcule local de la route. TBP se repose sur un mécanisme réactif de mises à jour régulières pour établir de nouvelle route. Quand à DSDV+, il propose une solution intéressante au problème en ouvrant deux routes à chaque fois, une principale et une secondaire. Si un lien est brisé sur la première, les paquets emprunteront la seconde.

De plus, l'influence des interférences sur les transmissions est une autre caractéristique des MANETs. En effet les émissions de chaque noeud vont venir interférer sur le médium radio, même au delà de leur zone de portée. La charge du médium radio dépend donc fortement de la bande passante utilisée par les noeuds dans une même région et c'est une donnée très utile pour l'évaluation de la bande passante disponible. Seul BRuIT prend vraiment en compte en proposant un innovant mécanisme de calcul de la charge de l'environnement radio. QOLSR effectue quand à lui des vérifications régulières de la bidirectionnalité des liaisons et considère comme rompu un lien unidirectionnelle. CEDAR aborde le problème sous un autre angle et s'assure de la limitation des interférences en ayant rarement recours à la diffusion en broadcast qui charge plus le médium. À la fin, TBP peut apporter de la QoS sur le délai et le coût alors que QOLSR est plus extensible car on peut paramétrer la métrique via l'ajout des messages de contrôle.

2.6.3. Avantages et inconvénient des protocoles de routage avec QoS

Le routage avec QoS est un élément essentiel pour réaliser une architecture de QoS pour les MANETs. En effet, le meilleur moyen pour garantir de la QoS est la réservation de ressources comme le fait le protocole DSDV+ basé sur la méthode d'accès au niveau MAC, TDMA, qui permet le calcul de la bande passante. Ainsi on obtient un fort taux de satisfaction pour la réservation de ressources [4, 23, 26, 33, 34, 35,36].

CEDAR est incapable de faire une réservation de ressources donc il n'offre pas de garanties sur la bande passante disponible, la mobilité des noeuds étant imprévisible, seule une approximation est envisageable. Ceci est justifié par le fait qu'il ne calcule pas forcément les routes les plus optimales en raison de l'approche minimaliste de la connaissance de la topologie du réseau par le cœur du réseau. Cela empêche donc la découverte de routes possédant une meilleur bande passante entre des noeuds dont le coeur n'a pas forcément connaissance. De plus, le routage à la source limite les performances pour trouver le chemin le plus court. Cependant malgré ses défauts, la connaissance partielle de la topologie du réseau de coeur permet de réduire énormément les overhead dus aux inondations du réseaux, principal obstacle à la mise en place du routage avec QoS car très coûteux.

Il faut insister sur le fait que tout comme TBP, CEDAR limite l'utilisation du broadcast et permet l'utilisation de l'unicast au sein du coeur et aussi entre le coeur et les stations qui s'y rattachent. Cela apporte une plus grande fiabilité et une plus grande performance comparé au problème de terminaux caché/exposé en diffusion et donc limite la perte de paquets.

Cette limitation de la diffusion est aussi réalisée par QOLSR. En effet le concept de sélection des MPR permet de restreindre l'inondation du réseau par les messages de contrôle. De plus comme pour TBP, ce protocole permet d'avoir plusieurs métriques pour définir la QoS. Ainsi on peut trouver une route avec la bande passante maximum ou une route avec un délai minimum. Cependant cette définition de QoS est limitée puisque il est difficile voir impossible de trouver une route optimale avec plusieurs contraintes. L'idée majeure est de rendre prioritaire la bande passante ensuite le délai, donc s'il existe plusieurs routes avec une bande passante maximale, le paquet sera envoyé sur la route où le délai est minimal.

Le protocole TBP réalise une réservation de ressource et en plus limite la diffusion des messages de signalisation. Cependant ce protocole fait partie de la famille des protocoles réactifs ce qui signifie que le délai d'établissement des routes peut devenir important dans un

réseau de forte densité. Ces limitations sont aussi mises en évidence lors de simulations par une distribution des sondes non optimisée, et par une estimation imprécise du délai.

Le protocole BRuIT repose sur un mécanisme de réservation de bande passante, utilisé avec un protocole de routage réactif basique. Sa vraie force réside dans son mécanisme unique de prise en compte des interférences dans le calcul de la bande passante disponible. En cela il apporte une vraie innovation avec un calcul de la charge du medium radio plus efficace et donc une évaluation de la bande passante disponible plus précise. Cela implique la garantie d'un certain débit aux applications qui en ont besoin. BRuIT est donc le seul protocole de routage à intégrer le support de la qualité de service en tenant compte le plus justement possible des ressources disponibles. Toutefois si le modèle théorique de BRuIT est novateur il ne faut pas perdre de vue que ses performances en pratique reste à juger. La principale difficulté restant dans le choix de la valeur k qui définis jusqu'à combien de saut on prend en compte les interférences. Une autre difficulté vient de son utilisation en intérieur, environnement sujet aux problèmes de réflexion/réfraction des ondes (et de multi trajet), où les performances risquent de ne pas bénéficier des innovations introduites par BRuIT.

D'ailleurs, on ajoute des contraintes sur les routes pour déterminer si elles sont admissibles ou non. Par exemple les applications multimédia ont besoin de contraintes de délais et de gigue. Donc les protocoles avec réservation de ressources comme DSDV+, TBP, et BruIT sont les seuls disposés à le faire. Cependant l'efficacité de TBP est optimale pour des réseaux à faible densité et mobilité réduite comme une bibliothèque par exemple alors que BRuIT gère aussi mal la mobilité, dû à un recalcul de route très long (2s) mais est efficace dans les réseaux denses grâce à sa grande fiabilité, donc efficace pour les échanges de fichier.

Quant à DSDV+, il gère bien la réactivité de la mobilité des noeuds et est adapté aux topologies dynamiques qui présentent une grande variation des flux et une inégalité de ressources dans les noeuds. Pour les protocoles ne réservant pas de bande passante comme CEDAR les performances en terme de QoS au niveau de la bande passante sont optimales pour des réseaux de faible mobilité ne demandant pas spécialement des routes optimales, préférant des liens plutôt rapides et stable à une route optimum et moins sure. CEDAR s'implante donc comme un protocole assez polyvalent.

QOLSR ne déroge pas à la règle et est adapté aussi aux réseaux à faible mobilité car dès qu'il y a un changement dans une table de routage, une mise à jour est obligatoire et comme nous l'avons vu avec CEDAR, le maintient de l'état précis des liens est très coûteux en

ressources et représente une barrière à la QoS. Ce protocole s'applique surtout à des réseaux qui ont besoin d'un minimum de contraintes de QoS, comme des réseaux de diffusion de flux multimédia audio et vidéo.

2.7. Signalisation pour la QoS

La signalisation constitue un autre élément essentiel de la qualité de service dans les réseaux. Elle permet de réserver et libérer des ressources du réseau, et de diffuser des informations de contrôle au travers du réseau.

2.7.1. Un protocole de signalisation QoS ad hoc

INSIGNIA [26] est un protocole de signalisation in-band c'est à dire que les données de contrôle sont incluses dans les entêtes des paquets et donc transmises avec les paquets de données au lieu d'être transmises dans des paquets de contrôle spécifiques. Cela est adapté aux réseaux ad hoc car il y a optimisation de l'utilisation de la bande passante (voir fig.2.4).

De plus, INSIGNIA permet de déterminer la quantité de bande passante à attribuer à chaque paquet et de réserver cette bande passante, assurant ainsi une certaine qualité de service. La demande de réservation est effectuée lors de l'envoi du premier paquet de données, et est rafraîchie par le passage des paquets de données.

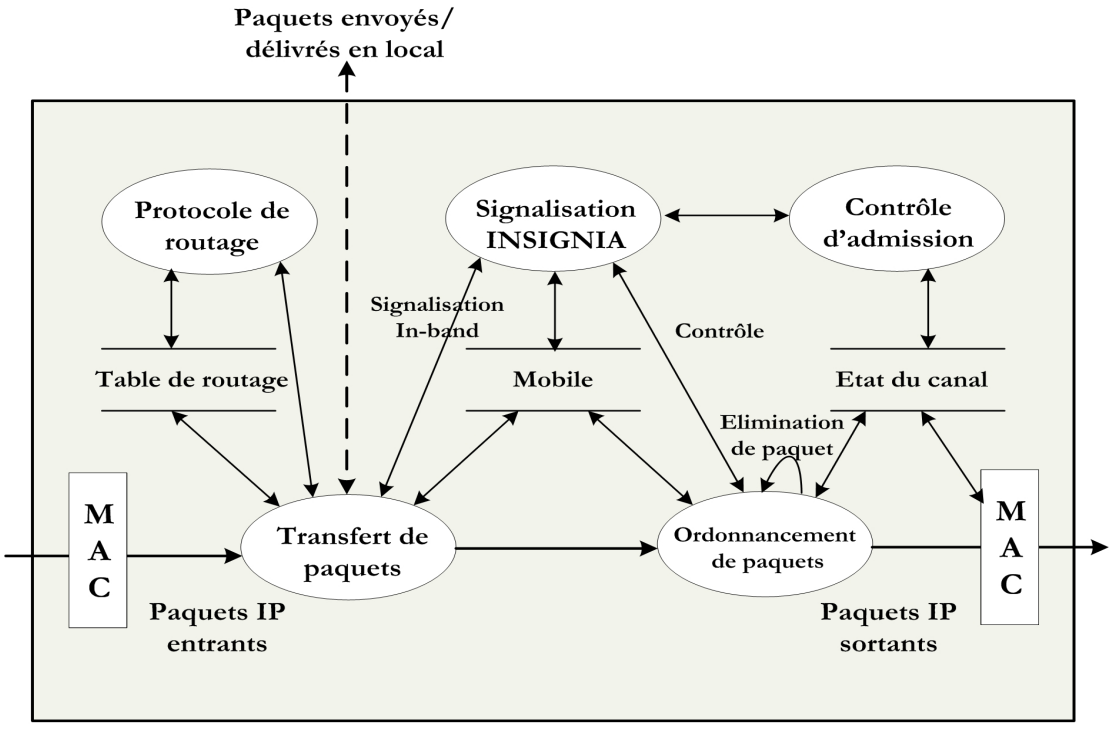

FIG. 2.4 – L'architecture du protocole INSIGNIA

Le destinataire informe périodiquement la source de l'état de la route en envoyant des rapports de QoS (QoS Reporting), qui contiennent des statistiques sur la latence, le taux de perte et le débit. Ces informations peuvent donc servir pour la source à réguler son débit d'émission. Cette réservation est de type soft-state, ce qui implique que les ressources sont libérées automatiquement si elles ne sont pas utilisées durant un certain laps de temps paramétrable. Cette caractéristique est adaptée aux MANETs puisque les ressources seront libérées sans demande spécifique et sans besoin notamment du lien d'allocation, ce qui est pratique dans un réseau où les liens sont peu fiables et la topologie dynamique [23].

L'un des inconvénients d'une telle approche est le fait d'avoir des informations sur les trafics dans chaque noeud ce qui pose des problèmes de capacité des noeuds, et la difficulté de passage à l'échelle avec l'augmentation du nombre de flux. En outre, cette approche n'offre que deux types de service temps réel et best effort. Finalement, INSIGNIA ne supporte que les applications multimédia adaptatives, et la réservation de ressource ne peut être établie que lorsque le trafic est lancé [4].

2.8. Conclusion

Le fait est que, pour obtenir une qualité équivalente à celle fournie par des réseaux filaires s'avère une tâche difficile. De nombreuses contraintes doivent être vaincues afin de tirer les bénéfices d'un réseau ad hoc : l'accès au canal radio, la gestion de la mobilité, la gestion de l'énergie, la sécurité et les solutions pour la qualité de service (QdS ou QoS) comme le délai, la bande passante et le taux de pertes de paquets.

Plusieurs solutions sont aujourd'hui proposées pour fournir de la qualité de service aux réseaux ad hoc, beaucoup reste à faire. Chaque protocole ne traite qu'un aspect particulier de la transmission dans les réseaux ad hoc. De plus, les tests effectués pour évaluer chaque protocole sont très différents. Les topologies, ainsi que les types de trafic utilisés sont propres à chaque protocole, c'est pourquoi il est difficile de comparer ces différentes solutions.

Dans ce chapitre, nous avons présenté le concept de qualité de service dans le cadre des réseaux ad hoc. Nous avons donné quelques technique utilisées pour le support de qualité de service dans les réseaux ad hoc : les modèles de qualité de services, le routage et les protocoles avec qualité de service.

Dans le prochain chapitre, nous allons présenter le paradigme d'agent mobile et l'apport de celle-ci dans les réseaux ad hoc.

Chapitre 3

La technologie Agent Mobile et les Réseaux Ad hoc

3.1. Introduction

La technologie d'Agent Mobile est de plus en plus considérée par la communauté scientifique ainsi que par certains opérateurs pour améliorer les services mobiles. Cette technologie permet aux concepteurs et aux développeurs de systèmes distribués d'utiliser des artefacts informatiques autonomes au niveau de leurs comportements, mais aussi dans l'espace et dans le temps. Ces entités peuvent se déplacer d'un ordinateur à un autre à travers le réseau, sans perdre leur code ni leur état. Cette mobilité peut être contrôlée par le système, on parle alors d'agents mobiles réactifs, ou par les agents eux-mêmes, on parle alors d'agents mobiles proactifs. La mobilité des agents peut améliorer la performance des applications réparties (comme la QoS dans les réseaux mobile ad hoc). En effet, dans une application à base d'agents mobiles, les agents se déplacent vers leurs partenaires ou les services désirés pour accomplir les tâches qui leur sont assignées. Ces déplacements permettent de transformer les communications à distance en communications locales, évitant ainsi les surcharges du réseau induites par plusieurs appels à distance.

Un agent peut être statique ou mobile. Un agent statique est un agent qui s'exécute seulement dans le système où il commence son exécution. S'il a besoin d'information non disponible dans le système ou a besoin d'interagir avec un agent dans un système différent, il utilise un mécanisme de communication tel que RPC.

Par contre, un agent mobile n'est pas lié au système dans lequel il débute son exécution. Il est capable de se déplacer d'un hôte à un autre dans le réseau. Il peut transporter son état et son code d'un environnement vers un autre dans le réseau où il poursuit son exécution. Les agents mobiles sont considérés comme un concept bien adapté pour développer des applications réparties grâce à leurs propriétés de tolérance aux fautes, autonomie et d'adaptation [37].

L'utilisation de la technologie d'agents mobiles dans les réseaux mobile ad hoc permet d'optimiser le délai et réduire la charge des réseaux engendrés par les échanges de messages de contrôle, et par conséquent, minimiser la probabilité de congestion.

Ce chapitre a pour objectif de présenter quelques travaux à base d'agent mobile dans les réseaux mobiles ad hoc comme, la sécurité d'agent mobile en réseaux ad hoc, la découverte de topologie de réseaux ad hoc par agent mobile, la performance de communication messages à bases d'agent mobile dans réseaux ad hoc et le protocole de routage.

3.2. Qu'est-ce qu'un agent?

Dans la littérature spécialisée, on trouve une multitude de définitions des agents. Elles se ressemblent toutes sur les concepts généraux, mais diffèrent selon le type d'application pour lequel est conçu l'agent. Jusqu'à présent, il n'y a pas encore un consensus de la communauté multi-agent sur le terme Agent. La difficulté d'une définition générale est due à l'abstraction de ce concept, tel est le cas de la définition de « intelligence » pour la psychologie ou de « conscience » pour la philosophie, où plusieurs définitions furent données à ces termes. Parmi les définitions on cite:

Première définition due à Ferber [38] : un agent est une entité autonome, réelle ou abstraite, qui est capable d'agir sur elle-même et sur son environnement, qui, dans un univers multi-agents, peut communiquer avec d'autres agents, et dont le comportement et la conséquence de ses observations, de ses connaissances et des interactions avec les autres agents.

Le deuxième définition proposée par [39] : un agent est un système informatique, situé dans un environnement, et qui agit d'une façon autonome et flexible pour atteindre les objectifs pour lesquels il a été conçu.

3.3. Définition d'un agent mobile

L'agent mobile est un agent capable de se déplacer d'une machine à une autre dans un réseau. La mobilité peut être forte ou faible en fonction des éléments impliqués dans le processus de transfert (code, données, pile, tas, compteur,…. etc.) [40].

Le paradigme des agents mobiles propose d'utiliser la migration d'activité en supprimant cette contrainte de connexion constante qui n'est pas évidente ni dans les réseaux de grande envergure ni pour les stations nomades.

On demande aux deux parties d'être connectées seulement durant la phase de migration. Ainsi un agent mobile peut se déplacer dans un réseau de machines offrant des services pour réaliser une tâche complexe. La FIG 3.1 illustre ce paradigme.

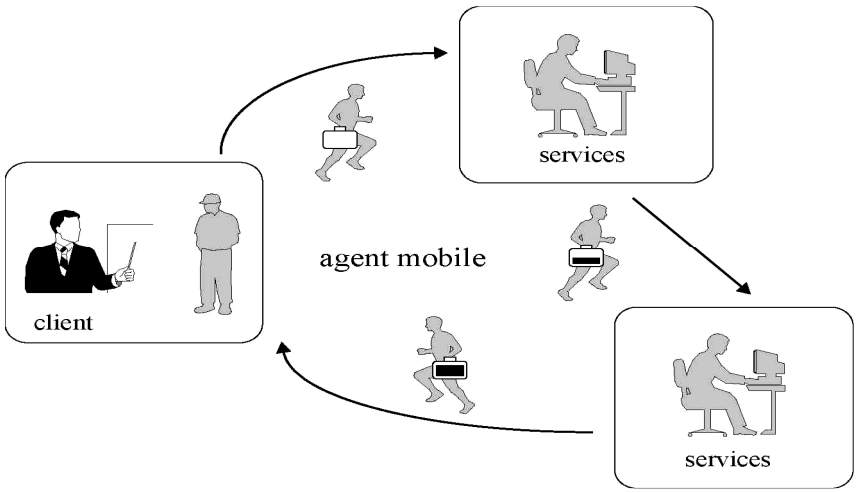

FIG. 3.1 - Le paradigme des agents mobiles.

Un client donne une mission à un agent pour la réaliser, il se déplace dans le réseau de machines accédant localement aux services offerts par ces machines. On peut distinguer trois phases [41] :

- L'activation de l'agent mobile avec la description de sa mission.
- L'exécution de la mission par l'agent qui se déplace pour accéder aux services.
- La récupération éventuelle des résultats de l'agent mobile.

3.4. Apport des agents mobiles à l'exécution répartie

Les paradigmes classiques de construction d'application répartie (par exemple l'appel de procédure distante, l'invocation d'objet distante,..) ont été conçus à l'origine pour s'appliquer à des environnements de taille limitée et parfaitement contrôles par un administrateur unique. La généralisation à grande échelle et à haut débit bouleverse radicalement ce contexte. D'une part, la taille du domaine potentiel d'exécution des application réparties s'est accrue considérablement : les entreprises ou les grands organismes sont délocalisés sur un nombre de sites de plus en plus nombreux. D'autre part, les interactions avec des sites inconnus n'appartenant pas au domaine d'administration des utilisateurs sont de plus en plus fréquentes. Ceci amène à considérer de nouveaux paradigmes de construction d'applications réparties, plus adaptés que les paradigmes classiques pour prendre en compte cette double évolution. L'agent mobile est l'une des pistes explorées actuellement par la communauté de

recherche. Il emprunte au domaine de l'intelligence artificielle les concepts d'autonomie et d'adaptation en y ajoutant celui de mobilité [42].

Le travail [43] montre la situation d'agent mobile par rapport aux autres modèles d'exécution distribuée, examine l'infrastructure système nécessaire au support d'agent mobile et rassemble des résultats expérimentaux sur la mise en œuvre d'agent mobile. Dans la section suivante nous basant sur ce travail.

3.4.1. Infrastructure système

La mise en œuvre d'agent mobile suppose, sur chaque site susceptible d'accueillir des agents mobiles, un support système pour l'exécution, la migration, la communication, et la prise en compte de la sécurité.

3.4.1.1. Support d'exécution

Pour pouvoir s'exécuter sur un site, un agent doit être admis et des ressources doivent lui être allouées, le support d'exécution doit permettre la réception de l'agent (code, état et attributs) et l'activation de code. Le support d'exécution doit prévoir la gestion des erreurs (notification à une certaine adresse) afin que le « propriétaire » (au sens large) de l'agent soit informé de la terminaison anormale de celui-ci.

3.4.1.2. Support de migration

Tout langage de programmation d'agent mobile fournit une primitive de migration volontaire de l'agent sur un site distant (migration proactive). C'est l'agent qui détermine quand, et où, se déplace (suivant une liste statique de sites à visiter ou en fonction de résultat d'exécution). Le système local doit fournir un support pour réaliser cette opération : emballage de l'agent (code, état, attributs), envoi sur le réseau, gestion des erreurs. En cas de succès, les ressources occupées par l'agent localement sont libérées, en cas d'échec, une exception est levée et est traitée par le code de l'agent suivant les instruction du programmeur.

3.4.1.3. Support de communication

Un agent peut avoir besoin de communiquer avec son propriétaire, avec d'autres agent (locaux ou distants), ou avec des services. La communication avec un agent ou un service distant, le choix entre la communication distante et la migration sur le site distant est un problème complexe (compromis entre le coût de communication et le coût de migration) dont la résolution est laissée au programmeur dans les système actuels.

3.4.1.4. Sécurité

Le terme « sécurité » doit être entendu ici au sens le plus large, englobant les notions d'autorisation, d'authentification, de protection, de confidentialité. Le site d'accueil d'un agent doit se protéger contre la sur-utilisation des ressources et les actions malveillantes (intentionnelles ou non)des agents accueillis. Ceci suppose l'identification du propriétaire, du site de provenance, et la vérification du code.

3.4.2. Evaluation

Dans le but d'obtenir une première idée des performances réelles que permet d'obtenir l'agent mobile, il va évaluer des agents mobiles en matière de minimisation du nombre des interactions distantes et du volume des données échangées sur le réseau.

3.4.2.1. Minimisation du nombre des interactions distantes

Lorsque les interaction entre un objet client et un objet serveur sont répétitives (par exemple, pour réaliser une requête complexe sur une base de données), il peut être intéressant de déporter sur le site du serveur un agent mobile réalisant l'ensemble des invocation avant de retourner le résultat globale. Il va évaluer ce bénéfice potentiel selon l'approche de comparaison les Aglets et le mécanisme d'invocation distante RMI de java.

3.4.2.1.1. Exemple redirection de requêtes

Il considère le scénario applicatif suivant : un programme client s'exécutant sur une machine A recherche la liste des hôtels et les numéros de téléphone associés dans une ville. Une première base de données, située sur une machine B, indique la liste des noms des hôtels dans une ville donnée, une seconde base, située sur une machine C, permet d'obtenir le numéro de téléphone associé à un nom. Dans le modèle de programmation client-serveur, une première interaction A-B permet de récupérer une liste de n noms, puis n interactions A-C permettent d'obtenir le résultat demandé. Dans le modèle de programmation par agent mobile, un agent créé sur la machine A se déplace sur la machine B, enregistre dans ses données la liste des n noms, puis se déplace sur la machine C où il effectue les n interaction localement, et enfin revient sur la machine A avec le résultat.

Les deux modèles ont été implantés respectivement avec java RMI et le système d'agent mobile Aglets, les machines utilisées étant situes dans des zone géographiques distante. La taille de l'enregistrement contenant un nom étant de 80 octets.

Les résultats indiqués à la figure suivante montrent qu'à partir d'un seuil de 30 enregistrement, le modèle d'agent mobile est plus performant que le modèle client-serveur (RMI) : le coût de la migration de l'agent sur le réseau est amorti par l'économie du nombre d'appels à distance que le modèle de programmation par agent mobile permet de réaliser [43].

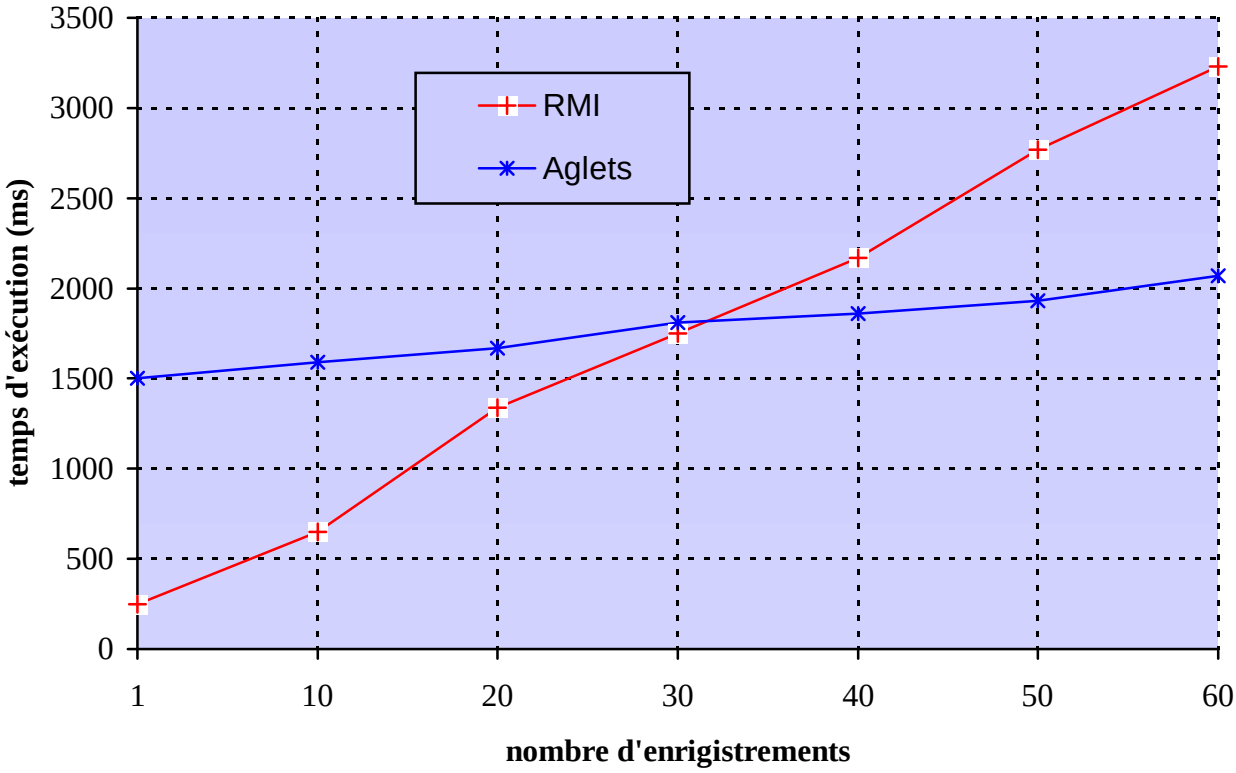

FIG. 3.2 - Comparaison entre RMI et Aglets pour l'application Redirection de requêtes

3.4.2.2. Minimisation du volume des données transportées sur le réseau

Une deuxième source de gain par l'utilisation de code mobile réside dans la possibilité d'aller effectuer le traitement là où se trouvent les données. Les exemples sont nombreux : recherche de mot-clé dans un document volumineux, dégradation d'un flux vidéo pour s'adapter à un terminal mobile, etc. on va évaluer le gain potentiel de base par l'expérience suivante : le client obtient du serveur un tableau de caractères de longueur variable.

Dans le modèle standard d'invocation, ce tableau transite sur le réseau, dans le modèle d'invocation par agent, c'est l'agent (et non le tableau) qui se déplacent sur le réseau. Les résultats sont indiqués à la figure suivante [43].

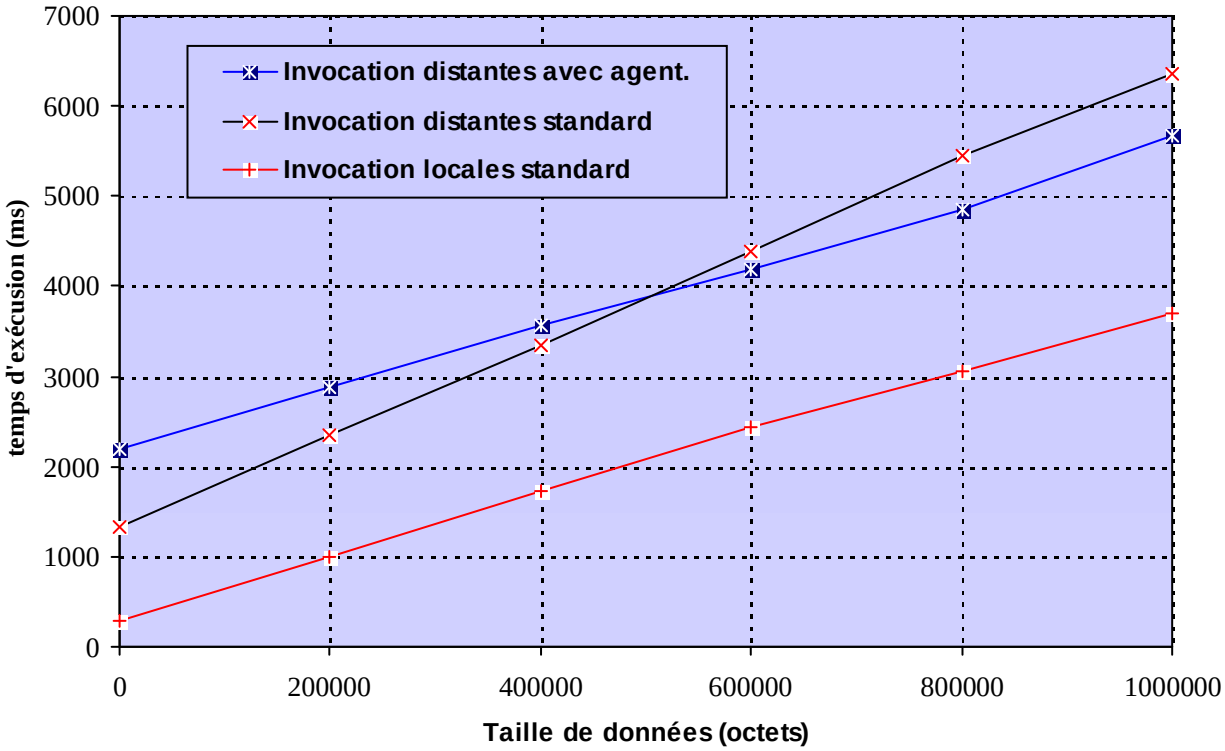

FIG. 3.3 - Temps d'exécution entre objets client, serveur et Aglets

Après analyse du temps de récupération d'un document à partir d'un serveur en utilisant la méthode d'invocation à distante (RMI) et un agent mobile Aglet implantant la compression du document sur le serveur avant transmission sur le réseau. On remarque qu'à partir d'une taille de 550000 octets, l'approche agent mobile est plus performant.

3.5. Avantages et inconvénients des agents mobiles

Selon [44], l'agent mobile possède plusieurs avantages, on peut citer les suivants :

- Le premier avantage d'utilisation les agents mobiles sont la diminution de la consommation de bande passante. En effet, plusieurs études montrent qu'en comparaison de l'envoi à distance de requête (procédures et méthodes), la mise en place des agents mobiles permet d'obtenir une réduction significative de la charge réseau en terme du nombre total de données transférées. Cette diminution est constatée dans différents types d'applications nécessitant d'intenses échanges d'informations entre le client et le serveur (voir la figure 3.2).
- La diminution des temps de latence. Dans le contexte des réseaux à large échelle, la mise en place d'applications reparties, nécessitant de fréquentes interactions entre client et serveur, se heurte aux temps de latence propres aux communications réseaux. Il arrive fréquemment que le temps d'attente de la réponse d'une requête soit plus long que le

temps de traitement nécessaire à la réalisation du service. En rapprochant client et serveur dans un même sous-réseau, voire sur un même site, on les place dans un environnement où les temps de réponse des interactions sont limités, ce qui permet de réduire d'autant les temps de latence.

- Un autre avantage à souligner vient des brèves périodes de communication. En réduisant le plus possible les communications distantes aux seuls transferts d'agents mobiles, on diminue considérablement les périodes de connexion entre deux sites. Cette diminution de la fenêtre d'utilisation des communications réseaux permet de moins se soucier des ruptures de liens physiques qui peuvent intervenir fréquemment dans les environnements sans fil.
- Le maintien du lien de communication peut s'avérer difficile dans les réseaux mobiles ad hoc à cause des nœuds qui sont libres de se déplacer aléatoirement. Avec les agents mobiles, un client peut déléguer les interactions avec le service sans maintenir une connexion de bout en bout.
- La tolérance aux fautes physiques c'est à dire, en se déplaçant avec leur code et données propres, les agents mobiles peuvent s'adapter facilement aux erreurs systèmes. Ces erreurs peuvent être d'ordre purement physique, disparition d'un noeud par exemple, ou d'ordre plus fonctionnel, arrêt d'un service par exemple. Si on prend le cas d'un site perdant une partie de ses fonctionnalités, un service tombant en panne, l'agent pourra alors choisir de se déplacer vers un autre site situé dans un même porte ou d'un autre porte contenant la fonctionnalité désirée. Ceci permet une bien meilleure tolérance aux fautes que le modèle statique classique.
- Les concepteurs préfèrent avoir une méthode permettant de décrire facilement un comportement réel. Avec la méthode classique, il va être très contraignant de décrire des algorithmes d'exploration (de réseau) ou bien encore de caractériser les déplacements des utilisateurs nomades. Avec les agents mobiles, les concepteurs disposent d'une méthode qui permet de décrire naturellement ce genre de comportement. Ainsi, on peut facilement mettre en place un déploiement et/ou une maintenance d'application sur un réseau ou encore suivre les utilisateurs dans leurs déplacements. Ensuite, les agents possèdent une capacité de traitement spécifique leur permettant de s'adapter à leur environnement. La capacité de raisonnement va permettre de concevoir des agents qui seront autonomes, adaptant leurs déplacements en fonction de l'environnement et pouvant moduler leurs fonctionnalités en cours d'exécution.

Néanmoins, le modèle d'exécution répartie d'agent mobile n'est une panacée, même si la protection des sites est quasiment assurée (voir 3.7.1.1), celle des agents reste un réel problème qui n'a pas de solution définitive. De plus, les agents mobiles ont besoin du support fixe d'une plate-forme (c'est à dire, manque d'infrastructure et de standards, même s'il existe un standard pour les systèmes multi agents, la technologie est trop peu mature dans beaucoup de domaines, dont celui de la sécurité, pour faire réellement sortir les agents des laboratoires. On peut ajouter à cela, les difficultés de mise au point dues aux déplacements de l'unité d'exécution qui est difficiles à suivre et aux problèmes de test qui nécessitent d'importants efforts de coordination [45].

3.6. Localisation distribué et adaptative d'agent mobile

Dans ce cas, deux niveaux de mobilité se superposent : celle des sites et celles des agents. Les sites peuvent changer de voisinage à tout moment, et les agents sont amenés à se déplacer de site en site à la recherche d'un service pour la réalisation de leur tâche. Dans ce contexte, il se pose le problème de la localisation et de l'accès au service désiré : l'agent client doit parvenir à rencontrer l'agent serveur sur un même site alors que sites et agents peuvent bouger simultanément.

3.6.1. Principe de localisation des agents

La localisation d'un agent correspond à l'identifiant du site sur lequel il s'exécute (momentanément). Seuls des agents mobiles présents momentanément sur le même site peuvent communiquer entre eux. Autrement dit, les agents ne s'échangent pas de messages à distance et un service de routage des messages inter agents est donc sans objet. Par contre, sous de telles hypothèses, il est nécessaire d'assurer une localisation la plus efficace possible des agents mobiles en prenant en compte le caractère évolutif de l'environnement.

3.6.2. Service de localisation

Selon le modèle [46], les agents peuvent coopérer, soit de façon indirecte : lors de la visite d'un site, un agent peut trouver/déposer des informations dans l'espace mémoire du site visité (tableau blanc), soit de façon direct par appel local de méthode selon un schéma de type client serveur. La localisation des agents nécessite de rappeler ou de préciser les différentes hypothèses faites sur le réseau dynamique. Nous définissons les propriétés minimales suivantes [47] :

- Tout site *s* possède un voisinage n_s constitué de l'ensemble des sites qui sont directement accessibles depuis *s*. Un site dont le voisinage est vide est momentanément isolé (il peut néanmoins rester actif).
- Le mouvement d'un site revient à modifier son voisinage.
- Tout site ne reste pas indéfiniment isolé.
- Il existe des agents dont la mobilité est beaucoup plus importante que celle des sites.

3.6.3. La stratégie de migration adaptative

Une stratégie de migration adaptative est un moyen de guidage d'un agent dans sa phase de migration. Deux approches sont possibles pour ce type de routage : soit réactive lorsqu'un chemin est construit à la demande, soit proactive lorsque le protocole met à jour de façon continue les informations de routage. Selon [46], un agent ne coopère pas directement avec les autres agents pour rencontrer son partenaire. Il profite de la trace laissée par les autres agents sur les sites visités par ceux-ci. Une trace est une information laissée par un agent sur un site visité en vue d'aider un autre agent à retrouver sa cible plus rapidement. Cette information peut être la nouvelle direction de migration d'un agent visiteur, les agents récemment rencontrés par un agent, un endroit spécifique où l'agent cible a été aperçu, ou d'autres informations similaires. Pour le service de localisation il y a deux types de trace :

- Le premier concerne la nouvelle direction de migration d'un agent, ce que nous désignons par trace de visite. Lorsqu'un agent termine la visite d'un site et qu'il s'apprête à migrer, il laisse la trace du site destination où il va entamer sa prochaine visite. Ceci constitue l'équivalent d'un lien de poursuite pour le routage de messages. Cette trace constitue la participation minimale, et obligatoire, que se doit d'acquitter chaque agent auprès du service de localisation.
- Le deuxième type de trace concerne la direction où un agent a été « aperçu». Ce type de trace peut être assimilé à la propagation de rumeurs. Nous désignons ce type de trace par trace de voisinage. Les traces de voisinage fournissent la localisation des agents récemment rencontrés par tout agent visiteur d'un site. Il ne s'agit pas forcément des agents rencontrés dans le voisinage immédiat. Plus précisément, pour un agent visiteur Av, un site S_a est dit dans le voisinage de niveau p d'un site S_b, si le chemin parcouru par Av du site S_a à site S_b est de longueur égale à p. Cette longueur s'exprimant en nombre de migrations nécessaires à l'agent Av pour aller du site Sa au site S_b. Pour un site S_a donné, la trace de voisinage de niveau p, désigne l'ensemble des agents rencontrés dans le voisinage de niveau p de A.

3.7. Travaux à base d'agent mobile dans un réseau mobile ad hoc

Avant de présenter quelques travaux à base d'agents mobiles dans les réseaux mobile ad hoc, il est important de rappeler que ces réseaux sont constitués de sites mobiles variés interconnectés par un support sans fil, sans l'utilisation d'une infrastructure fixe et sans administration centralisée. Ces réseaux ajoutent des contraintes inhabituelles dans les systèmes distribués traditionnels. Ainsi que, les paramètres de QoS qui sont plus considérés sont : la largeur de la bande passante, la gigue, la sécurité, le délai de bout en bout, la communication dans un couloire de conférence demande une communication minimale d'énergie où la durée de vie de la batterie dans ce cas est le paramètre clé.

Dans cette section, nous allons présenter quelque travaux basées d'agent mobile dans les réseaux mobile ad hoc comme, le routage, la découverte de topologie, la sécurité avec le but de s'améliorer sur des systèmes traditionnels en termes d'exécution, évolutivité, fiabilité de bout à bout, et gestion d'erreur.

3.7.1. Sécurité d'agent mobile aux réseaux sans fil ad hoc

La sécurité d'agent mobile sur réseaux mobile ad hoc (Secure Wireless Agent Testbed, SWAT) se composé de PDA à base de plate forme informatique sur un réseau sans fil 802.11b avec routage ad hoc. Le cadre de sécurité utilisé une combinaison de symétrique et cryptographie de clé publique pour soutenir la communication cryptée au réseau et aux couches application d'agent. La fonctionnalité de SWAT est la capacité de garantir le soutien de communication du groupe, via la génération de clés partagées, pour les groupes et les sous-groupes des hôtes et des agents.

La sécurité est surveillée par les agents qui contrôlent des clefs, évaluent le trafic de réseau et analyser les modes de comportements. Dans ce cadre, les agents peuvent révoquer les droits d'accès pour les hôtes suspects ou les agents et de manière adaptative réacheminer le trafic à la couche réseau afin d'améliorer l'intégrité de l'information du système global. Les agents fournissent le cadre d'exécution d'un certain nombre applications décentralisées, y compris, l'authentification d'utilisateur, la collaboration, la transmission de messages et la surveillance de capteur à distance. Dans cette approche nous basant sur [44, 48, 49].

3.7.1.1. Sécurité d'agent mobile

La sécurité d'un point de vue logiciel, consiste à empêcher des accès, et/ou des modifications, non-autorisés aux éléments d'un système informatique. Dans ce cadre, il faut

prendre en considération la distinction entre les demandeurs (utilisateur, processus, périphérique …) et les éléments demandés (processeur, fichier, mémoire .....). Lorsque l'on souhaite avoir un système sûr, on met en place une politique de sécurité qui doit garantir la confidentialité, *i.e* les données des éléments ne sont pas divulguées aux demandeurs non autorisés, et l'intégrité, *i.e* les éléments ne sont pas modifiés par des demandeurs non autorisés [49].

Pour mettre en place la politique de sécurité visée, on utilise généralement des mécanismes d'authentification (mot de passe, certificat ...), de cryptographie (SSH, SSL ...) et de contrôle d'accès (droit utilisateur, pare-feu). L'authentification à pour but d'identifier précisément le demandeur, la cryptographie doit assurer la confidentialité des données échangées et le contrôle d'accès vérifie l'adéquation entre demandeur et éléments demandés. En effet, les agents mobiles représentent un nouveau champ d'investigation pour le domaine de recherche en sécurité, d'une part dans la protection des sites vis à vis des agents malveillants et d'autre part dans la protection des agents vis à vis des sites malveillants [44].

3.7.1.1.1. Protection des sites

La protection des sites contre des attaques menées par des agents malveillants est un problème qui est aujourd'hui bien maîtrisé. En effet, plusieurs solutions permettent maintenant de se prémunir d'éventuelles attaques et voici les méthodes les plus connues :

- Bac à Sable : cette technique consiste à exécuter un agent à l'intérieur d'un environnement restreint, en interdisant l'accès au système de fichiers par exemple. Cette approche peut facilement se mettre en place en utilisant des interpréteurs de code dont leurs possibilités sont limitées.
- Signature de code : la signature de code intervient lors de la création d'un agent, son créateur le signant numériquement afin qu'il puisse s'identifier durant ses déplacements. Cette technique permet d'obtenir une authentification de haut niveau pour les sites.
- Contrôle d'accès : en fonction des agents, le site pourra autoriser l'accès à un ensemble précis de fonctionnalités. le contrôle d'accès permet de mixer les deux premières techniques en offrant aux agents signés plus de fonctionnalité qu'un simple bac à sable sans pour autant accéder à toutes les fonctionnalités.
- Vérification du code : la vérification de code permet d'obtenir une garantie sur la sémantique d'un code à travers l'analyse de sa structure, ou de son comportement pour un agent, en fonction d'une politique de sécurité donnée. Lors de la mise en route de l'agent,

son créateur fournit un ensemble de preuves intégrées qui est transporté par l'agent. Ces preuves garantissant le comportement de l'agent en fonction de critères de sécurité des sites à visiter. Lorsque l'agent commence une nouvelle visite, le site récupère la preuve lui correspondant et vérifie si elle correspond à sa politique de sécurité. Le site choisit alors d'exécuter ou non l'agent.

3.7.1.1.2. Protection des agents

Pour comprendre ce que risque un agent lors de son exécution sur un site malveillant, nous pouvons référencer les éléments transportés pouvant être cible d'attaque :

- Le code : ensemble des instructions composant la tâche de l'agent.
- Les données statiques : données ne changeant pas durant les déplacements (la signature par exemple)
- Les données collectées : ensemble des résultats obtenus au cours des déplacements réalisés par l'agent depuis son lancement.
- L'état courant : ensemble de données servant à l'exécution courante de l'agent.

La sécurité des agents mobiles consiste alors à garantir les critères de confidentialité et d'intégrité de l'ensemble de ces éléments. Du point de vue des données, il est évident qu'un agent ne souhaite pas divulguer des informations critiques à n'importe quel site.

3.7.1.2. Architecture de SWAT

Le SWAT utilise l'extensible d'architecture d'agent mobile (EMAA). Le cadre d'EMAA inclut agents autonomes et asynchrones, ainsi que, le mécanisme de gestion pour la mobilité d'agent, les événements d'agent et la communication entre agent. L'architecture d'EMAA est composé de trois couches principales : *Docks, serveurs, et agents.* Les couches *Docks* fournissent l'environnement d'exécution de l'hôte, dans ce que, tous les autres composants sont exécutés. *Serveurs* fournir les services « lourds » ou fixes, tandis que, *les agents* sont « plus légers » et mobiles, tous les gents ont liste des tâches et des itinéraires. Lorsque l'autonomie est fortement, les agents d'EMAA peuvent recevoir et répondre aux commandes à partir d'une autorité de contrôle, ce qui l'équilibre entre le comportement totalement autonome et la coopération aux centres de commande qui peuvent être commandés par l'humain utilisateurs.

3.7.1.2.1. Protocole et mécanisme de sécurité

Ils ont examiné comment employer des mécanismes de sécurité pour des applications traditionnelles (non-agent) et les réseaux fixes. Les objectives sont :

- Évitez un seul point de défaillance en utilisant des services distribués et décentralisés.
- Pratiquer la sécurité en avant, de ce fait, en excluant l'utilisation des secrets cryptographiques précédents par un attaquant potentiel.
- Employer des contrôles pour assurer l'intégrité des données.
- Considérer la sécurité réactive pour les cas qui échoue la sécurité préventive.
- Déployer un système discrétionnaire de contrôle d'accès pour imposer des ressources privilèges.

Les technologies de sécurité établies qui intégrées dans le SWAT incluent :

- **Cliques** est un protocole pour la gestion et la génération de clé.
- **Spread** est une application client/serveur pour groupe de communication fiable. Le serveur est distribué et chaque client est connecté sur le segment de serveur local.
- **Secure Spread** est une exécution aux protocoles de CLIQUES utilise spread comme plate forme de communication.

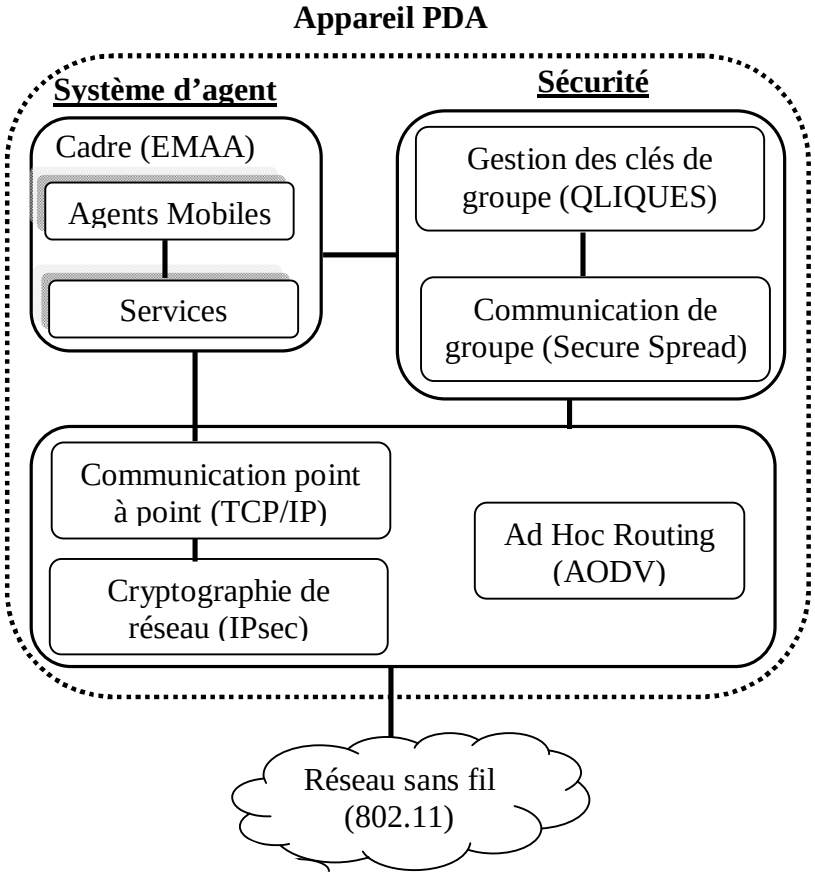

FIG. 3.4 - Architecture de hôte de SWAT

- **Semi-Trusted Mediator (SEM)** est un algorithme pour révocation de l'utilisateur. Dans SWAT, quand un utilisateur est révoqué, le SEM empêche de participer à la génération de clé symétrique.
- **IPSec** est utilisé à déchiffrement du trafic sur la couche réseau, à l'aide des clé(s) symétriques générées par Secure Spread.

3.7.1.2.2. L'intégration de système de SWAT

Chaque hôte du SWAT est composé de plusieurs technologies qui intègrent le système d'agent, le réseau, et l'infrastructure de sécurité. Le schéma 3.4 montre les composants d'un hôte de SWAT et comment ces composants sont reliés. Le cadre d'agent d'EMAA contient les agents mobiles et statiques (services). Les composants de sécurité d'un hôte incluent la gestion des clés de groupe, mis en œuvre par Secure Spread et la révocation des membres du groupe imposée par l'intégration SEM et Secure Spread.

Le cadre d'agent est relié aux composants de sécurité, permettant à un agent (ou au système entier d'agent) de se joindre ou laisser un groupe, la permission de se joindre contrôlé par SEM. Les composants de réseau permettent la sécurité de communication point à point pour le cadre d'agent et la communication fiable de groupe pour les composants de sécurité. Toute la communication de réseau conduite par le réseau sans fil 802.11b du SWAT utilise le protocole de routage AODV.

Quand deux ou plusieurs hôtes sont présentes dans SWAT, les connexions sont faites entre des hôtes par le réseau sans fil du SWAT. Le réseau de SWAT est caractérisé par trois types de trafic : agent système, sécurité, et gestion de réseau ad hoc et de routage. Le trafic d'agent système est généré par EMAA et inclut l'agent de messagerie, l'agent de migration, l'événement de messagerie. Le trafic de sécurité, qui est produit par Secure Spread et SEM, inclut la génération des clés des groupes, l'entrée ou la sortie d'un membre de groupe et la révocation des membres du groupe. La gestion du trafic de réseau ad hoc, produit par le routage AODV, comprend la maintenance et découverte des routes de réseau dans le SWAT.

Les mécanismes de sécurité du SWAT permettent aux hôtes et agents d'appartenir à l'un ou plusieurs groupes. Les groupes fournissent par des canaux chiffrés (cryptées) pour la communication privée. L'utilisation de CLIQUES dans un groupe de hôtes ou agents peuvent de générer une clef symétrique partagée.

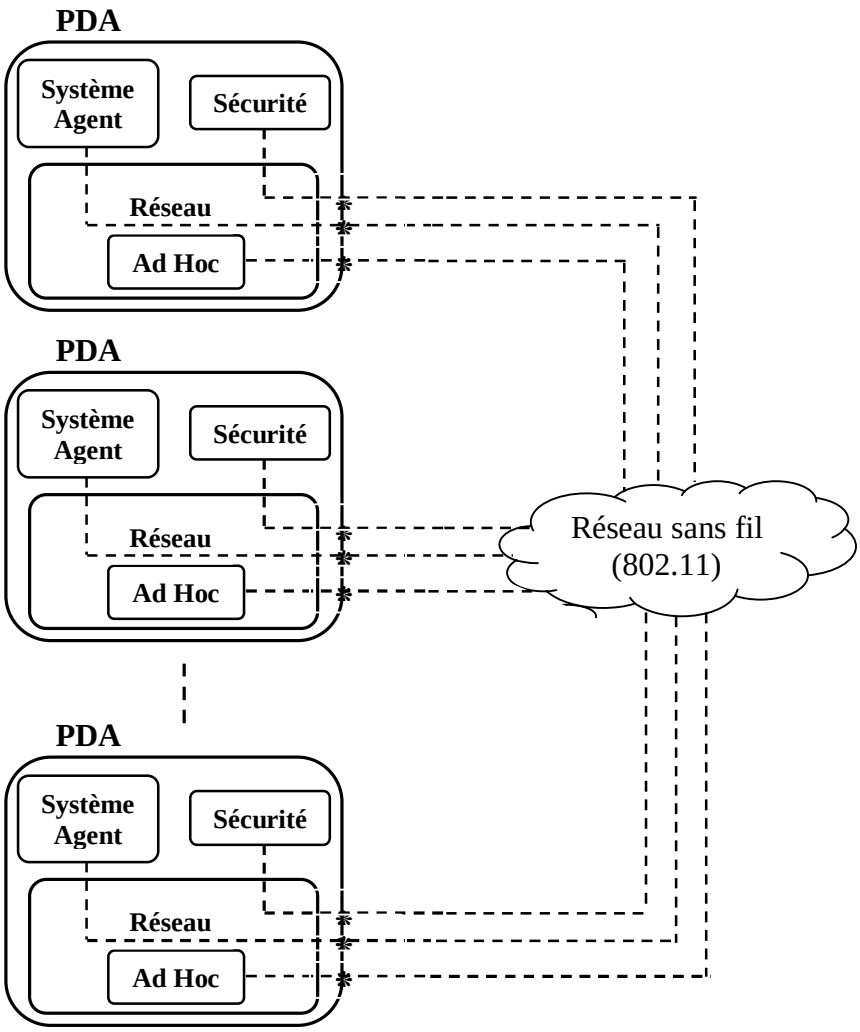

FIG. 3.5 - Architecture de réseau du SWAT

L'utilisation de cette clé, les membres de groupe chiffrent les messages qui peuvent être déchiffrés seulement par les membres de groupe. Si un membre de groupe est retiré à un groupe spécifique, SEM empêcheront ce membre de participer dans la prochaine clé générée par CLIQUES pour ce groupe.

3.7.1.2.2.1. L'intégration de l'hôte et l'agent système

La mobilité d'agent dépend de la capacité d'un hôte de transférer l'état d'exécution d'un agent. L'agent de messagerie du SWAT dépend de la connectivité continue des hôtes fondamentaux. Cette connectivité est réalisée par l'utilisation de protocole de routage ad hoc AODV (SWAT utilise un version modifié de AODV). Des activités au sein de l'équipe SWAT est orienté vers l'amélioration d'exécution AODV pour assurer que la fonctionnalité prévue est préservée tant que plus de fonctionnalités sont ajoutées au SWAT. Le multicast AODV fournit un mécanisme à un groupe d'agents situés sur un groupe d'hôtes pour exécuter la messagerie de groupe d'une façon efficace.

3.7.1.2.2.2. L'intégration de l'hôte et la sécurité

Le SWAT exclut l'attaque passive, en chiffrant tout le trafic de réseau entre les hôtes aux niveaux multiples en utilisant de CLIQUES pour fournir la sécurité contre l'attaque pendant la génération de clés. Le cryptage et décryptage à chaque interface de réseau sont réalisés à l'aide d'exécution IPSec. Les gents de SWAT lorsque exécutant le plan de la messagerie d'agent peut supposer que leur communication est seulement lisible dans le système d'agent. Puisque chaque hôte dans le SWAT est en réseau, il est vulnérable à l'attaque de réseau. Pour cette raison, chaque hôte de SWAT emploie un pare-feu proactif pour filtrer et enregistrer toutes les transmissions inattendues de réseau.

3.7.1.2.2.3. L'intégration d'agent système et réseau

Les changements de topologie de réseau du SWAT conduisent aux redondances de routes entre deux hôtes quelconques peut également changer. Par conséquent, à tout moment, un agent peut avoir un choix dont son message est envoyé au partout de réseau du SWAT. Si, un hôte devient indésirable ou inaccessible, un agent peut être éviter cet hôte. Les agents mobiles emploient un itinéraire pour déterminer à quel hôte ils émigreront au cours d'accomplir des tâches.

Le SWAT introduit les agents de méta raisonnement de réseau qui peuvent exploiter la redondance de route et permettent aux hôtes, Docks, services et d'autres agents de s'adapter aux états changeants de réseau. Par exemple, si un hôte devient compromis, ou supprimé pour des raisons de sécurité, les agents de méta raisonnement de réseau du SWAT peuvent modifier leurs itinéraires ou rerouter les agents de message à augmenter la robustesse et réduire l'effet des hôtes compromis. La discontinuité ou un compromis d'un hôte donne à chaque agent de créer soit un nouvel itinéraire soit un rapport d'échec à son hôte d'origine.

3.7.1.2.2.4. L'intégration d'agent système et sécurité

La sécurité d'agent du SWAT est réalisée par la sécurité de communication de groupe. La génération d'une clef partagée de groupe, s'appuie sur le d'accord de protocole CLIQUES. L'algorithme TGDH (Tree Group Diffe-Hellman), impose une structure arborescente binaire sur les membres d'un groupe, réduire la génération de clé au temps logarithmique. De plus, la structure arborescente permet à des CLIQUES de recalculer une clef à efficacement dans le cas un membre rejoint ou laisse un groupe. SEM enregistre chaque révocation, et empêche les membres révoqués de participer à la génération de prochaine clé. Si nous avons imposé la révocation instantanée, chaque communication devrait être acheminés par le SEM. En raison

de calcule élevé et le coût de réseau de cette approche, le SWAT actuel impose la révocation seulement au moment de la génération de prochaine clé. Au delà des mécanismes de sécurité traditionnels, les agents fournissent la sécurité d'ordinateur par des nouvelles possibilités. Le SWAT avancé une nouvelle approche à la sécurité basée d'agent qui permet aux agents eux-mêmes de raisonner sur la sécurité et comment ils l'affectent.

3.7.1.3. Synthèse

Les agents mobiles, la sécurité, les réseaux ad hoc, émergents comme des composants essentiels pour la prochaine génération d'informatique et la collaboration des infrastructures. Le SWAT démontré comment des agents peuvent être intégrés avec clé publique et les infrastructures de l'encryptions symétriques de clé pour créer les groupes multiples d'agents et faciliter la sécurité de communication inter-agent. Le SWAT présente nouvelles idées sur la façon dont les agents peuvent raisonner et manipulé leur topologie de réseau pour améliorer la sécurité et pour accroître la performance d'application.

3.7.2. Découverte de topologie en utilisant des agents mobiles

Dans le travail posé dans [50], ils ont utilisé un cadre basé sur multi-agents mobiles pour aborder l'aspect de la découverte de topologie dans un environnement fortement dynamique de réseau sans fil ad hoc.

Les agents mobile ou les messagers qui sautent dans le réseau sont une nouvelle solution au problème de la découverte de topologie. Les agents sautent du noeud à noeud, collectent des informations de ces noeuds, rencontrent d'autres agents dans leur voyage, agissent entre eux pour rassembler des mises à jour des parties du réseau qu'ils n'ont pas visité ou avoir visité à un long temps et prend en comptes ces ensembles de données rassemblés à nouvelle visite aux noeuds et aux agents. Un noeud reçoit donc des informations actualisées sur le réseau par la visite des agents dans un intervalle régulier et court.

Le but principal est de collecter toutes les informations relatives à la topologie de chaque noeud dans le réseau sans fil ad hoc et de les distribuer périodiquement (comme mises à jour) à d'autres noeuds par les agents mobiles. L'idée fondamentale est d'utiliser des agents mobiles pour la découverte de topologie a été déjà explorée avec certaines limitations : d'abord, la mobilité de noeud et son effet sur la performance de système n'a pas été mesurée. En second lieu, l'information de convergence (la différence du convergence entre l'information de topologie actuel et l'information de topologie perçue par un noeud dans quelconque point du temps) et ses relations avec le nombre des agents et la fréquence de migration d'agent n'a pas

été bien défini. Troisièmement, les stratégies de navigation utilisées n'assurent pas une distribution équilibrée à l'information de topologie récente parmi tous les noeuds. Dans cette approche nous basant sur [50].

3.7.2.1. Description des termes pertinents

3.7.2.1.1. Lien affinité

Dans un environnement sans fil, chaque noeud n a une portée de transmission sans fil. Ils définissent les voisins de n comme l'ensemble des noeuds à l'intérieur de la portée de transmission R de n. Lorsque le noeud n transmet un paquet, il est diffusé à tous ses voisins. Toutefois, dans l'environnement sans fil, la force des liens de connextion à tous les membres de l'ensemble de voisin de noeud n ne sont pas uniformes. Par exemple, un noeud m dans la périphérie de la portée de transmission de n est faiblement relié à n par rapport à un noeud u qui est plus proche de n. Ainsi, la possibilité de m de laisser la portée de transmission de n due à une mobilité de m ou n est plus que celui de u.

Lien affinité α_{nm} est un lien entre deux noeuds n et m et prédire la durée de vie de ce lien dans un contexte particulier.

Pour simplicité, ils supposent que α_{nm} égal à α_{mn} et la portée de transmission R pour tous les noeuds est égale. Pour découvrir l'affinité α_{nm}, le noeud n envoie une balise périodique et le noeud m échantillons les signaux fort reçus du noeud n périodiquement. Puisque, la force de signal perçue par m est en fonction de R et la distance courante r entre n et m, ils peuvent prévoir la distance courante r au temps t entre n et m. Si M est la moyenne de vitesse des noeuds, le cas plus mauvais à moyenne d'affinité α_{nm} au temps t est (R - r)/M, en supposant que dans le temps t, le noeud m a commencé de se déplacer vers l'extérieur avec une vitesse moyenne M. Si m et n ne sont pas des voisins parmi eux, $\alpha_{nm} = 0$.

3.7.2.1.2. Recency

Un aspect important de l'infiltration d'information de topologie des noeuds mobiles est que l'information diffusée doit être identifiée avec un degré d'exactitude. Puisque la navigation d'agent est asynchrone et il y a un espace de temps évident entre la fourniture d'information par un agent d'un noeud et sa livraison par le même agent à un autre noeud, il devient impératif d'introduire un concept de recency d'information.

Par exemple, soit deux agents A1 et A2 arrivent au noeud n, les deux contenant des informations sur un noeud m qui est loin de noeud n par multi saut. Afin de mis à jour

l'information de topologie de noeud n autour de noeud m, il doit y avoir un mécanisme permettant de savoir qui porte les informations les plus récentes sur le noeud m : agent A1 ou agent A2 ?

Pour mettre en application cela, chaque noeud dans le réseau a un compteur qui est initialisé à 0. Quand un agent laisse un noeud après exécution toutes ses tâches, dans ce nœud au noeud, il incrémente ce compteur par un. Ils nomment ce compteur par *recency token*. Tandis que un agent laisse un noeud (après l'achèvement de tout l'échange de l'information nécessaire et les calculs), il est stocke la nouvelle valeur du *recency token* dans l'identification du noeud dans ses structures de données et continue de navigation. Ainsi, dans un temps quelconque, l'importance du *recency token* de n'importe quel nœud est représente le nombre de fois que le nœud a été visitées par des agents depuis le début du réseau. Ceci implique également que, si deux agents ont un ensemble de données concernant le même noeud, par exemple le noeud X, alors l'agent portant le plus haut valeur du *recency token* de noeud X a des informations plus moderne sur lui.

3.7.2.1.3. Temps de migration (Time-to-Migrate, TtM)

Un agent lors visite un nœud n'est pas autorisé de migrer immédiatement vers un autre noeud. En d'autres termes,un agent sera forcé de rester dans un noeud pendant une période prédéterminée, appelée temps de migration (TtM), avant la migration à un autre noeud. Par le contrôle TtM, la congestion de réseau due au trafic d'agent peut être contrôlée. Par exemple, si TtM = 100 ms, pour un seul agent, il implique que, le milieu sans fil va voire un agent dans chaque 100 ms.

3.7.1.4. La moyenne de convergence de connectivité

Ils ont développé une métrique qui est la moyenne de convergence de connectivité pour mesurer la déviation de la topologie de réseau réelle avec la topologie de réseau perçue par les noeuds individuels à n'importe quel instant de temps. Soit l^a_{nm} est le statut de lien (0 pour la discontinuité et 1 pour la connectivité) entre le noeud n et m comme perçu par le noeud a dans n'importe quel instant de temps et l_{nm} est le statut actuel de lien entre le noeud m et n à celui instant du temps. L'informations sur le statut de lien l^a_{mn} dit il y a une convergence au noeud a, si $l^a_{mn} = l_{mn}$. Ainsi, la convergence de connectivité de lien entre n et m au noeud a est $\gamma^a_{nm} = 1$, si $l^a_{nm} = l_{mn}$ et 0 sinon. La convergence de connectivité de noeud a pour tous les liens dans le réseau, γ^a est défini comme $\gamma^a = (\sum$ pour tous les nœuds paires i-j $\gamma^a_{ij})$ / nombre total des nœuds paires.

À un certain instant de temps, si γ^a =1.0, il implique que l'information de topologie au noeud a est exactement identique que la topologie de réseau actuelle à celle instantanée du temps. Comme un autre exemple, dans un réseau de 10 noeuds, il y a 45 nœuds pairs et le statut de lien 45 possible. Si, à tout nœud a, 44 statut de lien rattache à tout instant du temps avec le statut de lien actuel, alors γ^a = 44/45 = 0.98. La moyenne de convergence de connectivité, γ_{avg} = ($\sum$ pour tous les nœuds k (γ^k)) / nombre des nœuds.

3.7.1.5. Moyenne de convergence de Lien affinité

Comme nous l'avons vu, la moyenne de convergence de connectivité mesure la déviation de la topologie de réseau actuelle avec la topologie de réseau perçue par les noeuds individuels de manière discrète (où le statut de lien est 0 pour la disconnectivité et 1 pour la connectivité). Ils peuvent quantifier le statut de lien basé sur le lien affinité. La quantification pourrait être plus appropriée en formulant une métrique qui leurs aiderait à évaluer la différence entre la topologie de réseau actuelle et la topologie de réseau perçu par les noeuds individuels dans une échelle continue.

Soit α^a_{nm} est l'affinité entre le noeud n et m perçu par le noeud a à l'instant quelconque du temps et α_{nm} est l'affinité actuelle entre le noeud m et n à celui instant du temps. L'information sur le statut de lien α^a_{mn} dit il y a une convergence au noeud a, si $\alpha^a_{mn} <= \alpha_{mn}$. Comme indiqué précédemment, l'affinité est le cas mauvais de la prévision sur la durée de vie d'un lien. Selon [48], si l'affinité d'un lien entre n et m perçu par un noeud a est moins de l'affinité actuelle entre n et m, nous accepterons la perception de noeud a au sujet de la lien affinité entre n et m. Cependant, si $\alpha^a_{mn} > \alpha_{mn}$, nous considérerons cela comme exagération par le noeud a au sujet de l'affinité de lien entre n et m et nous rejetons la perception. Ainsi, la convergence de lien affinité entre n et m au nœud a, λ^a_{nm} =1, si $\alpha^a_{mn} <= \alpha_{mn}$ et 0 sinon.

La convergence de lien affinité de nœud a, λ^a, pour tout les liens dans le réseau, est définie comme : λ^a = (Σ pour tous les nœuds paires i-j (λ^a_{ij})) / nombre totale de nœud paires. À un certain instant de temps, si λ^a = 1.0, il implique que l'information de topologie au noeud a est 100% acceptable, autant que l'affinité basée sur le mécanisme de prédiction est concerné. La moyenne convergence de lien affinité, λ_{avg} = ($\sum$ pour tous les nœuds k (λ^k)) / nombre des nœuds.

3.7.2.2. La découverte de topologie : mécanisme de base

Ils supposent que, un agent émigre chaque K de temps entre les noeuds. Au temps t_0, chacun des noeuds a seulement des informations sur leurs voisins immédiats. Au temps = t_0 + K, un agent saute à un noeud avec l'information qu'il a autour de nœud précédent. Ainsi le noeud courant possède des données sur un nouveau voisin de noeud et ses voisins. Dans le prochain temps de K, un noeud obtient des informations concernant deux noeuds supplémentaires à partir d'un autre agent. Il doit être noté que, par le contrôle de K, il est possible de contrôler le trafic d'agent dans le réseau. En outre, l'agent émigrerait toujours d'un noeud à un seul de ses voisins après chaque temps de K.

3.7.2.2.1. Algorithme de navigation

L'algorithme de navigation pour le débit optimal et la convergence rapide doit s'assurer que, tous les noeuds de membre du réseau sont actualisé ils-mêmes uniformément, indépendamment de leur position et état dans le réseau. Dans cet algorithme, les agents choisissent les noeuds de destination qui ont les moins valeur de recency_token, ce qui implique que, les noeud ont été visité moins fréquemment. Un agent résidant dans le noeud n à tout instant du temps fait le suivant :

- Mis à jour les informations cachées de noeud n avec les informations disponibles dans ces informations caches.
- Choisit tous les noeuds qui sont voisins de n.
- Évalue la valeur minimale du *recency token* de ces voisins à partir de l'information caches.
- Si ce voisin de n n'a pas été visité dans les 3 dernières visites par des autres agents du noeud n, choisir ce voisin en prochaine destination. Cette information d'histoire est stockée dans les noeuds. Sinon, choisir le deuxième voisin de moindre visité, et ainsi de suite. Ceci s'assurera que plusieurs d'agents de même nœud ne choisissent pas la même destination consécutivement.
- Après le choix la bonne destination, il met à jour son identification de noeud de la prochaine destination avec l'identification du noeud de destination, change la table d'histoire du noeud n avec ce nouvel choix de l'identification de nœud.
- Incrémente la valeur du *recency token* du noeud et stocke cette valeur de l'identification de noeud dans sa propre information caches.
- L'agent reprend la navigation.

3.7.2.2.2. Manipulation l'événement d'oscillation d'agent entre des nœuds

Ils considérent un cas où un noeud isolé par d'autres noeuds dans le réseau et en conséquence ne reçoit pas des visites d'agent pendant longtemps. Maintenant, ils supposent qu'il a relié autre fois au réseau après un certain temps. Évidemment, la valeur de recency token de ce noeud serait beaucoup moins que les valeurs des autres, qui reçue continuellement des agents à intervalles réguliers. Maintenant ce noeud isolé obtenir une connexion au réseau serait évidemment à précipitation des agents vers lui. Les agents obtiendraient sur des services, vont à une certaine prochaine destination et reviennent encore à ce nœud jusqu'à la valeur de recency du noeud relié nouvellement n'est plus mineur parmi toutes les recency tokens des autres noeuds dans le réseau. Cela signifie que, si un noeud est isolé et puis rejoint, les agents oscilleraient entre ce nœud et ses voisins jusqu'à sa valeur de recency dépasse la valeur du recency d'un autre voisin. L'oscillation pourrait causer d'autres noeuds dans le réseau de faim dans visites des agents et il est inutile de la perspective de filtrer l'information courante.

Afin d'éliminer cette oscillation, nous avons incorporé la stratégie suivante. Si un noeud constate qu'il ne reçoit pas des visites d'agent pour longtemps plus de temps spécifique, il remet sa valeur de recency token à zéro. Un agent qui visite un noeud, constatant que le noeud a une valeur zéro de recency token reconnaît que le noeud a effectué une remise à zéro. L'agent exécute l'échange d'information standard d'agent nœud et puis assigner une moyenne de toutes ses valeurs de recency token de ce noeud. Ceci empêche l'oscillation du même agent. Cependant, autres agents peuvent à venir de façon indépendante à ce noeud puisqu'ils pourraient encore avoir une valeur de recency pour ce noeud comme moins valeur. Cela est souhaitable afin de filtrer rapidement l'information d'un nouveau noeud.

3.7.2.2.3. L'échange d'information et l'interaction entre deux agents

L'infiltration de l'information partielle de réseau dans les noeuds est un processus asynchrone, car les agents visitent les noeuds asynchrone. Ainsi, il devient nécessaire de développer des stratégies pour l'échange d'information. Le processus asynchrone est constitue en deux étapes :

- La première étape, les recency tokens de tous les noeuds stockés dans l'information caches du nœud actuel sont comparé avec les recency tokens de tous les noeuds stockés dans l'information caches de l'agent. Si le recency token de n'importe quel noeud, par exemple X, dans ce nœud, l'information caché être inférieur par rapport de l'information caches de

l'agent, alors, il est évident que, l'agent est porte l'information plus récent sur le noeud X. Quand un agent est prêt à émigrer (c.-à-d. après un temps d'attente défini plus tôt comme TtM), étape 2 est exécutée.

- La deuxième étape, l'agent copie l'information caches de noeud complètement dans leur propre de l'informations caches. Ceci contient l'information la plus récente puisque l'ensemble de données contient une combinaison de toutes les informations récentes qui pourraient être collectées de la visite d'agents et ceux qui étaient déjà présents dans le noeud. Avec mise à jour de ces valeurs, les agents choisissent leur destination sur la base de l'algorithme de navigation.

3.7.2.3. Vieillissement de l'information : une méthode prédictive

La première caractéristique d'un environnement dynamique est que l'information n'est jamais absolue. Les ensembles de données recueillis par la connectivité des agents de nœuds et les liens d'affinités sont changent constamment. Ils ont défini un concept du vieillissement de l'information sur le lien d'affinité basée sur un algorithme prédictif fonctionnant sur chaque noeud peut prévoir la topologie de réseau actuelle qui est basé sur l'information de réseau courant stockée à ce noeud. Ils appliquent ce mécanisme de prédiction pour prévoir l'état mauvais de topologie et prend l'attention avant de lancer la transfert de données. Dans leurs algorithme, à chaque moment de temps, chaque noeud diminue les valeurs de l'affinité différentes de zéro de tous les liens recueillis à son information cachent par une valeur égale à la vitesse moyenne correspondant à moment de temps.

3.7.2.4. Synthèse

D'après les résultats de simulation situé dans [50], il est clair que, la moyenne de convergence de connectivité s'améliore avec la diminution de la mobilité. La performance serait moins de 80 % pour une mobilité élevée de 30 m/s. Cependant, le temps de migration (TtM) pourrait être abaissée pour produire de meilleurs résultats même à la mobilité élevée. Mais cela a évidemment l'effet de la congestion dans le système. leurs mécanisme de prédiction dans le cadre de TtM =100 milliseconde. Pourrait donner des résultats satisfaisants avec les valeurs de convergence plus de 98 %. Ainsi, le recours aux mécanismes prédictifs même à une basse fréquence de migration d'agent peut élever la performance.

Dans cette étude, ils ont supposé que, les agents ne se perdent pas en transit ou ne souffrent aucun type d'erreurs dans la transmission et la réception. Ils n'ont pas considéré les situations, où un ensemble de nœuds se transfert au loin après création des agents. Dans cette

situation, la population d'agent dans le réseau a comparé au nombre de noeuds actifs augmenterait qui dégraderaient la performance. Dans une telle situation, quelques agents doivent être détruits. En fin, l'utilisation du cadre multi-agents mobile sera capable de faire à chaque noeud dans la topologie de réseau avertie sans consommer la grande partie de capacité de réseau.

3.7.3. La communication de message basée d'agent mobile dans un réseau ad hoc

En effet, il y a un intérêt croissant en employant les agents mobiles comme une partie du solution pour implementer une architecture de réseau plus flexible et plus décentralisée [51]. Dans cette approche, ils essaient de montrer l'utilisation efficace d'agent mobile pour la communication de message hors ligne dans le cadre de réseau ad hoc grand et fortement dynamique. Un agent mobile peut émigrer d'un noeud source avec un message et naviguer de façon autonome dans tout le réseau pour découvrir la destination afin de fournir par ce message. L'agent possède une connaissance approximative sur l'emplacement de destination et profite la mobilité de nœud comme un véhicule pour émigrer physiquement d'un emplacement à autre. Si la destination est devient hors connexion de réseau dans un certain temps, la livraison de message sera reporté et l'agent attend pour sa reconnexion au noeud intermédiaire approprié. Dans cette approche nous basant sur [52].

3.7.3.1. Description de système

Le réseau est modélisé comme un graphe G = (N, L) où N est un ensemble fini de noeuds et L est un ensemble fini de liens unidirectionnel. Chaque nœud n Є N est possède un identifiant unique. Dans un environnement sans fil, ils supposent que, les liens unidirectionnels, la transmission entre deux noeuds ne fonctionne pas nécessairement dans les deux directions, chaque noeud n possède une portée de communication. Si, les deux noeuds n et m sont reliés par deux liens unidirectionnels L_{nm} Є L et L_{mn} Є L, alors, n peut envoyer de message à m via L_{nm} et m peut envoyer de message à n via L_{mn}.

Dans ce travail, ils supposent qu'un noeud maintient la trace de la position physique de ses voisins et avoir un "domicile". D'une façon générale, le domicile est l'endroit le plus préféré d'un nœud, où il réside normalement. Cela signifie que, même si un noeud est mobile, il revient par la suite à son domicile. Cependant, un noeud peut décider de changer son domicile/emplacement dynamiquement.

3.7.3.2. Un mécanisme pour la création et la navigation d'agent

3.7.3.2.1. Structure de nœud

Ils supposent que, chaque noeud connaît son identifiant, domicile, portée de transmission et l'emplacement actuel (par GPS). Chaque noeud transmet périodiquement un message à ses voisins physiques pour les informer à son identification, l'emplacement actuel et domicile. Le noeud de réception ajoute cette information et vérifie si le domicile de noeud de transmission est dans la portée de transmission du domicile de noeud de réception. Si, oui, il ajoute l'identifiant et le domicile de noeud de transmission au voisin logique.

Ils définissent les voisins physiques de n, $M_n \in N$, pour être l'ensemble des noeuds dont les domiciles sont dans l'intérieur de la portée de transmission de N. Quand le noeud n change son domicile, il est communiqué à tous ses voisins logiques dans l'ensemble M_n.

3.7.3.2.2. Création d'agent au nœud source et procédure de base de navigation

Quand un noeud veut envoyer un message à l'autre noeud qui n'est pas son voisin physique, un agent avec l'agent de type = 0 est créé comme transporteur de message. L'objectif de la procédure de base de navigation est de réduire la distance entre le domicile de l'agent courant (emplacement actuel du noeud où l'agent est résident) et le domicile de la destination. Ce critère permettrait à un agent de choisir un voisin physique qui est plus près de domicile de noeud de la destination et d'émigrer là. S'il n'y a aucun voisin physique disponible à cet instant du temps, l'agent attend une durée bien définie et essaye autre fois.

3.7.3.2.3. Comportement d'un agent près d'emplacement du nœud de destination

Le plus souvent, l'agent ne trouverait pas le noeud de destination à son emplacement. Dans ce cas, l'agent doit attendre dans un certain noeud près de domicile de la destination. Cependant, en raison de la mobilité de noeud, ce noeud près de domicile de la destination pourrait s'éloigner. Dans ce cas, l'agent émigrerait à un autre noeud suivant, le même critère de réduire la distance avec le domicile de la destination. Cette boucle de migration attente se poursuivrait jusqu'à le noeud de destination atteigne son domicile et le message est livré.

3.7.3.2.4. Réorientation un agent par le voisin logique de nœud de destination

Un noeud peut décider de ne pas revenir à son domicile dans le futur proche et souhaiter de spécifier un nouvel domicile, où tous les messages de ce noeud doit être réorientés/ redirigé. Un noeud doit informer tous ses voisins logiques au sujet de son nouvel domicile. Puisque, le noeud connaît seulement les domiciles de ses voisins logiques, il lance un agent

avec un agent de type =1, un pour chacune de ses voisin logique, qui transmettre ce message concernant de ce changement. À la réception de ce message, le voisin logique met à jour le champ approprié, enregistrant le nouvel domicile de noeud en train d'étudier.

Quand un noeud de destination change son domicile, le noeud source ne le sait pas. Par conséquent, l'agent qui est créé pour porter le message aux destination atteindrait éventuellement à zone originale de localisation de nœud destination, attendant du noeud destination pour arriver à ce point. Dans cette situation, un voisin logique du noeud destination peut réorienter l'agent vers le nouvel domicile du noeud destination. Tandis que, l'agent est attend le noeud destination dans son zone domicile, l'agent rencontrerait avec les voisins logiques. Si l'agent trouve les informations concernant le nouvel domicile du noeud destination dans n'importe lequel de son voisin logique, l'agent se rediriger lui-même vers cet endroit suivant le même procédure de navigation.

Un noeud de destination qui a établi un nouvel domicile temporaire peut décider après un certain temps pour remettre cela à son original de domicile ou à un autre domicile provisoire. Cependant, avant que ces messages parviennent à tous ses voisins logiques, un agent peut rencontrer un voisin logique de ce noeud destination et se réorienter lui-même vers l'ancien domicile provisoire du noeud destination. Dans cette situation, l'agent serait mal orienté. L'agent doit revenir après un certain temps au domicile original du nœud de destination afin de re-suivre le noeud de destination.

3.7.3.3. Évaluation des performances

L'exécution du système proposé [52] est évaluée sur un environnement simulé. Dans la simulation, ils supposent que, l'environnement est un zone fermé de l'unité 1000 x 1000 dans lequel, les noeuds mobiles sont distribués aléatoirement.

3.7.3.3.1. Évaluation de la validité de système

Ils ont choisi arbitrairement une source de destination paires et étudié le temps pris par un agent initié par une source pour atteindre dans la portée de transmission de domicile au noeud de destination. Le succès de système dépend sur deux facteurs : i) l'agent initié par une source devrait atteindre rapidement aux portée de transmission du domicile du noeud destination. ii) il doit continuer à rester dans la portée de transmission du domicile, en attendant le noeud destination pour atteindre à son domicile.

3.7.3.3.2. Évaluation l'efficacité de communication de message basée d'agent

Ils ont notent que, le nombre moyen de saut pris par un agent pour livrer un message n'est pas élevé. En outre, il ne dépend pas beaucoup du nombre de noeuds de la portée de transmission particulier. Puisqu'un agent émigre toujours à un « voisin approprié », le nombre de saut ne dépend pas beaucoup du nombre de noeuds dans le système. Le nombre de saut diminue avec l'augmentation de la portée de transmission. Quand la portée de transmission est élevée, un agent pourrait prendre un saut « plus long » pour atteindre aux destination plus rapidement. Le nombre maximum des sauts pris par un agent pour délivrer un message est haut dans quelques cas. C'est en raison du fait qu'il y a un changement de domicile du noeud destination et l'agent a été réorienté vers le nouvel domicile de la destination.

3.7.3.4. Synthèse

Si les noeuds dans le réseau sont moins mobile et la taille de réseau est petit, la conception des protocoles de routages existent dans le cadre de réseau ad hoc multi saut sont fonctionnent bien. Cependant, pour un réseau ad hoc grand et fortement dynamique, où chaque noeud a un domicile prédéfini, le système basé d'agent pour la livraison de message est beaucoup plus décisif et efficace.

3.7.4. Protocole de routage basé d'agent mobile

Les protocoles actuelles de routage pour les réseaux mobiles ad hoc (MANETs) souffrent de certains points faibles inhérents (voir chapitre 2). Les agents mobiles ont la capacité de soutenir la communication asynchrone et le traitement de requête flexible. Par conséquent, l'utilisateur mobile peut assigner une tâche à un agent mobile et quand l'agent estime la disponibilité de communication, il va parcourir le réseau et accomplir la tâche déléguée par son utilisateur. De cette façon, un noeud mobile exige moins de connectivité de communication par rapport approches traditionnelles de client/serveur. Une autre raison également important aux agents mobiles dans les réseaux sans fil est qu'ils peuvent réduire le trafic de réseau. Les noeuds mobile fonctionnant sur la puissance de batterie n'ont pas la puissance assez pour exécuter les protocoles nécessaires de routage complexes dans les réseaux ad hoc. Une alternative est d'employer l'agent mobile pour effectuer les opérations de routage et ainsi réduire la complexité et le trafic de réseau. Par conséquent, la durée de vie de la batterie des nœuds est importante. Dans cette approche nous basant sur [53].

3.7.4.1. Protocole de routage hybride Ant-AODV

Les fourmis dans des applications de routage de réseau sont les agents simples qui incarnent l'intelligence et se déplaçant dans le réseau d'un noeud à l'autre et mettant à jour les tables de routage des noeuds qu'elles visitent avec ce qu'elles ont appris dans leur passage jusqu'ici.

La technique de Ant-AODV est combine de deux routages : la routage basé de fourmi et le protocole de routage AODV pour surmonter de certains leur inconvénients inhérents. La technique hybride augmente la connectivité de noeud et diminue le délai bout à bout de retard et de découverte de route. L'établissement de route dans des techniques de routages basé de fourmis conventionnelles est dépend de la visite des fourmis aux noeuds et lui fournissant des routes. Si un noeud souhaite envoyer des paquets de données vers une destination pour laquelle, il n'a pas assez d'une nouvelle route, il faut maintenir les paquets de données dans son tampon jusqu'à une fourmi arrive et lui fournisse une route à cette destination.

Dans Ant-AODV, les agents de fourmis fonctionnent indépendamment et fournissent des routes aux nœuds. Les noeuds ont également la capacité de lancer la découverte de route sur demande pour trouver des routes aux destinations, pour lesquelles, ils n'ont pas assez d'une nouvelle route d'entrée. L'utilisation des fourmis avec AODV augmente la connectivité de noeud qui réduit alternativement la quantité de découvertes de route. Si un noeud lance un RREQ, la probabilité de recevoir des réponses rapidement (par rapport à AODV) des noeuds voisins est élevée en raison de l'augmentation de la connectivité de tous les noeuds qui réduite le délai de découverte de route. Enfin, car les agents de fourmis mettent à jour les routes continuellement, un noeud source peut commuter une plus longue route par une plus court route fourni par les fourmis. Ceci mène à une diminution considérable de la moyenne de délai de bout à bout par rapport aux deux AODV et routage basé de fourmi.

3.7.4.2. Résultats de simulation

Des simulations étendues ont été effectuées dans [53] pour comparer le protocole de routage hybride Ant-AODV proposé en cette approche aux protocoles de routage conventionnels basé de fourmi et d'AODV.

3.7.4.2.1. La moyenne de délai bout à bout

La moyenne de délai bout à bout inclut le tampon de délai pendant la découverte de route, la file d'attente, les délais de retransmission et le temps de propagation et de transfert.

La moyenne de délai bout à bout pour AODV et le protocole hybride de Ant-AODV est très moins. Mais en cas de technique de routage de fourmi, la moyenne de délai bout à bout est élevée.

Dans la comparaison de Ant-AODV et l'AODV, ils peuvent observer que le délai bout à bout est réduit considérablement dans la Ant-AODV par rapport à AODV. Les fourmis aident en maintenant la connectivité élevée dans la Ant-AODV, par conséquent, les paquets ne besoin pas d'attendre dans le tampon d'envoyer jusqu'à les routes sont découverts.

3.7.4.2.2. Fraction de livraison de paquets et Goodput

Goodput est le nombre total de paquets utiles reçus à tous les noeuds de la destination et la fraction de livraison de paquet est le rapport du nombre de paquets de données envoyés au nombre de paquets de données reçus. La fraction de livraison de paquet est très haute pour AODV et Ant-AODV par rapport au routage basé de fourmis. Goodput est également plus haut pour la Ant-AODV et l'AODV par rapport au routage basé de fourmis. La fraction de livraison de paquet et le goodput sont élevés dans la Ant-AODV et l'AODV, parce qu'elles se servent de détection le lien d'échec et les messages de route d'erreur.

3.7.4.2.3. Normalisation de routage overhead

La normalisation de routage overhead est le nombre de paquets de routage transmis par rapport les paquets de données reçus à la destination. Ils constatent que, la normalisation d'overhead est trop haute en cas de protocole de routage basé de fourmis. Parce que, les paquets de données réels livrés sont trop moins et par conséquent le rapport du contrôle d'overhead aux paquets de données livrés devient trop haut. En cas d'AODV, la normalisation d'overhead est le moins. La normalisation d'overhead est plus légèrement dans la Ant-AODV par rapport à AODV en raison du mouvement continu des fourmis dans le réseau.

3.7.4.2.4. Connectivité

La connectivité est la moyenne de nombre des noeuds dans le réseau pour lequel un noeud a des routes expiré. En cas d'Ant-AODV et protocole de routage basé de fourmis, les agents de fourmi parcourent continuellement le réseau et mettent à jour les entrées de table de routage. En raison de ceci, un noeud a assez frais des routes d'un grand nombre de noeuds dans le réseau à n'importe quel moment donné. La connectivité dans Ant-AODV et le protocole de routage basé de fourmis est une connectivité plus double dans AODV.

3.7.4.3. Synthèse

Pendant les simulations, ils ont observé une caractéristique importante des agents de fourmi pour le routage dans MANETs. Après une certaine période (presque 100 secondes de simulation), l'activité de fourmi (la fourmi saut d'un noeud à l'autre et mettant à jour des routes) presque disparaître. Cela pourrait être dû à diverses raisons telles que (i) les paquets de fourmi pourrait être perdu dans transmission sans fil, (ii) le prochain noeud qui était de recevoir les mouvements de paquet de fourmi hors de portée de transmission du nœud d'envoyer, ou (iii) le noeud portant de fourmi aller hors de la cadre sans fil de chaque noeud dans le réseau et le prochain saut de nœud pas valable aux fourmis.

Les points faibles des protocoles de routage sur demande comme AODV et routage basé de fourmi ont été essayés pour surmonter en ce travail en combinant les deux pour renforcer leur potentiel et pour alléger leurs faiblesses. Le protocole hybride Ant-AODV est capable de fournir la réduction de délai bout à bout et la connectivité élevée par rapport à AODV. En raison de la connectivité accrue, le nombre de découvertes de route est réduit et également la latence de découverte de route. Ceci rend le protocole hybride de routage Ant-AODV approprié aux données de temps réel et la communication de multimédia.

3.8. Conclusion

L'utilisation de la technologie d'agents mobiles dans réseaux mobile ad hoc permet de surmonter les problèmes liés à la déconnexion des sites, le délai, le débit,…etc. En effet, un terminal mobile crée un agent mobile et lui demande d'agir pour son compte. L'agent crée, et après sa migration, peut s'exécuter dans le système même si le site mobile créateur fonctionne en mode déconnecté. Une fois le site du client connecté, il va contacter l'agent mobile afin de lui demander de revenir sur son site d'origine. Ainsi cette utilisation permet d'optimiser le délai et réduire la charge des réseaux engendrés par les échanges de messages de contrôle, et par conséquent, minimiser la probabilité de congestion.

Cependant, l'exécution d'un environnement d'agents mobiles, dans ce contexte, pose deux problèmes essentiels. Le premier est relatif à l'apparition et la disparition dynamique des sites pouvant accueillir les agents. Cette situation se produit lorsqu'un agent s'exécute sur un terminal mobile et que son moyen de communication est déconnecté. Le deuxième se pose au niveau de la localisation des agents mobiles du fait de la mobilisation des sites d'accueils.

Ce chapitre a été consacré à technologie d'agent mobile et leur intérêt pour l'utilisation dans les réseaux ad hoc. Nous avons donné les notions de base d'agent mobile, leur apport aux applications réparties surtout les réseaux mobile ad hoc et quelques travaux qui sont basé sur l'agent mobile dans les réseaux mobiles ad hoc.

Notre proposition d'un modèle à base d'agent mobile pour la QoS dans les réseaux mobile ad hoc est exposée dans le chapitre suivant.

Chapitre 4

Modèle de QoS à base d'Agent Mobile

4.1. Introduction

Les déconnexions fréquentes liées à l'environnement mobile ad hoc, les ressources limitées de ces outils mobiles, la nature de déplacement des messagers font la complexité du réseau mobile ad hoc et ainsi de suite. Pour cela, nous allons proposer un modèle basé d'agent mobile pour améliorer les paramètres de la qualité de service et la minimisation de la consommation des ressources. En effet, nous avons évoqué dans le chapitre précédent les raisons pour lesquelles nous avons eu recours au paradigme d'agent mobile pour réaliser les fonctionnalités attendues.

Les agents mobiles sont des entités autonomes, intelligentes et adaptables, nous les utilisons pour assurer la QoS dans un réseau complètement dynamique. Dans ce cas, ils jouent différents rôles. Certains de ces rôle consistent à : collecter les informations produites par un autre agent, mettre à jour la table de routages sur les nœuds mobiles, découverte des nouvelles routes, contrôler les paramètres de QoS durant la connexion, déclancher les activités de maintenance et de l'adaptation en cas de dégradation de la QoS, évaluer les états des nœuds, réserver si possible les ressources adéquates et prendre des décision de manière autonome sur l'utilisation des ressources selon leurs capacités. Pour cela, un agent est associé à chaque terminal, il offre ses fonctionnalités de manière proactive ou réactive selon les conditions de l'environnement.

Il faut mentionnée que notre apport c'est l'utilisation d'une fonction d'optimisation basée sur les six paramètres et d'une topologie de réseau pour le QoS basé sur le notion de cluster avec l'agent mobile. Nous utilisons un agent nœud qui peut changer son état (état membre, état passerelle, état vicaire et représentant) selon le déplacement ou la condition proposée comme la capacité, un agent transporteur pour transférer les données et un agent routier pour découvrir et maintenir des routes.

Dans ce qui suit, nous présentons l'organisation de réseau, l'architecture interne de l'agent mobile, le diagramme de transition et de classes de modèle de QoS à base d'agent mobile, en fin, la propriété de ce modèle et une conclusion.

4.2. Organisation du réseau

Dans notre modèle, la gestion est réalisée de façon distribuée par un ensemble de gestionnaires (agent mobile ou statique) qui communiquent entre eux. Les gestionnaires disposent d'un même degré de responsabilité et chacun est pleinement responsable de la gestion d'un sous-ensemble de noeuds dans le réseau. L'utilisation de gestionnaires multiples permet de limiter localement la charge de gestion et d'améliorer la robustesse du système. En revanche, des mécanismes de coopération doivent être mis en oeuvre pour assurer la cohérence dans les opérations de gestion exécutées par les gestionnaires.

4.2.1. Niveaux de gestion de QoS

Il est évident que, les nœuds sont assemblés dans des groupes différents, chaque groupe a un représentant. La construction des groupes se fait d'une manière distribuée, auto organisable et selon les conditions proposées. Comme illustré dans la figure 4.1, le modèle dispose de trois niveaux de gestion de la QoS, qui sont, au niveau nœud, au niveau groupe (cluster) et au niveau réseau. La gestion de la QoS dans ce modèle est assurée par la coopération des trois niveaux, chaque niveau assure des fonctionnalités spécifiques. Dans la suite, nous décrivons les principales fonctionnalités de chaque niveau.

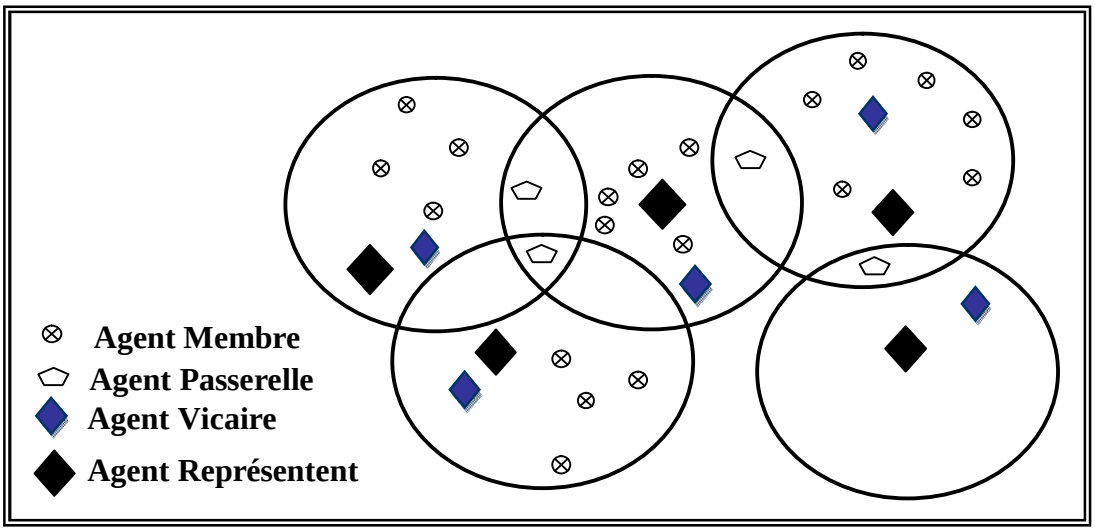

FIG. 4.1 – L'organisation du réseau

4.2.1.1. Niveau nœud

Ce niveau a pour rôle l'adaptation aux ressources disponibles permettant de mieux gérer les ressources du nœud (bande passante, mémoire, CPU, batterie, etc.) dans le but de satisfaire les différents besoins de QoS des applications. Comme nous l'avons vu, ces nœuds

(terminaux mobiles) sont de nature hétérogène (par exemples, ordinateurs portables, téléphone mobiles, PDA), cela implique que les capacités des nœuds sont variées et peut engendrer des liens asymétriques dans le réseau. Donc, une capacité est calculée pour chaque nœud. La définition d'un seuil (**S**) de capacité permet de déterminer l'état d'un nœud, et selon le cas, le seuil peut prendre des valeurs différentes. Nous prenons en compte les paramètres **BP, C, D, M, E, MM**, pour calculer la capacité (C_{ni}) complète du terminal, tel que :

$$C_{ni} = f\ (BP, C, D, M, E, MM)$$

Sachant que :

BP : la Bande Passante maximale.

C : la charge de CPU.

D : le degré de nœud (*i.e* le plus grand nombre de voisins).

M : la charge de Mémoire.

E : niveau d'Energie.

MM : la Moyenne de Mobilité.

Pour simplifier notre étude, nous supposons que ces paramètres sont indépendants et nous introduisons la relation suivante pour mesurer la capacité d'un nœud :

$$\mathbf{C_{ni} = aBP + bC + cD + dM + gE + hMM}$$

Où

a, b, c, d, g, h : sont les paramètres d'adaptation qui permettent de privilégier une ressource ou bien un terminal par rapport à l'autre selon le rôle que l'agent va jouer ou les conditions proposées, tel que : $\mathbf{a + b + c + d + g + h = 1}$

Les nœuds possèdent des capacités fortes utilisées comme représentants par rapport à l'autre moins capacités. Le problème principale à ce niveau est la volatilité des ressources, d'une telle façon que les ressources disponibles au moment **t** ne seront pas obligatoirement disponibles au moment $\mathbf{t+\Delta t}$.

4.2.1.2. Niveau groupe (cluster)

Les nœuds sont ensuite regroupés sous la forme de clusters, selon la topologie de l'environnement de réseau et les conditions proposées. En effet, beaucoup d'algorithmes qui expliquent cette manière. Ce niveau décrit les interactions entre les nœuds à l'intérieur du

même groupe pour gérer localement les différentes fonctions. Au vu de la structuration du réseau, il est logique que certains nœuds aient des statuts spécifiques. Certains se chargent de maintenir les communications dans le groupe qui leur est affecté, tandis que d'autres se chargent de relayer les agents mobiles entre groupes.

Il faut mentionner qu'on a associé dans chaque nœud un agent appelé *agent nœud*, il prend l'état *membre, passerelle, vicaire* et *représentant (chef)*. Dans le niveau nœud, il prend l'état par défaut (état membre). Dans le niveau cluster, on le trouve dans l'état membre, vicaire et représentant (i.e. dans le cluster, il y a plusieurs agents, donc, un seul agent prend l'état représentant, un autre agent prend l'état vicaire, tandis que, les autres prennent l'état membres). Dans le niveau réseau, on trouve l'état membre, passerelle, vicaire et représentant. Nous décrivons brièvement le rôle de chaque état d'agent :

- *Un agent dans l'état représentant* est l'agent chargé de diriger les communications du groupe et la coopération avec les autres représentants des groupes voisins, soit directe ou un indirecte par l'agent passerelle. Il est élu dans chaque groupe selon la valeur de C_{ni}.
- *Un agent dans l'état vicaire* est l'agent qui fait toutes les tâches de l'agent représentant en cas de son absence (déplacer de son groupe, tombe en panne, dépasse le seuil qui détermine le niveau de la QoS par lui-même ou par les autres agents).
- *Un agent dans l'état passerelle* est un agent chargé de faire communiquer les groupes entre eux. Il appartient donc au moins à deux groupes différents et doit être capable de joindre les agents représentants de chacun de ces groupes par communication directe.
- *Un agent dans l'état membre* ne joue plus le rôle de routier, il coopère avec les agents de son groupe pour la gestion de la QoS au niveau du cluster. Il ne se charge pas d'un rôle supplémentaire comme l'agent représentant et l'agent passerelle. En cas de dégradation de la QoS demandée par les applications s'exécutant sur le terminal, l'agent associé coopère avec les agents voisins pour la maintenance locale de la QoS.

4.2.1.3. Niveau réseau

Dans ce niveau, un réseau ad hoc est organisé comme un ensemble de clusters de nœuds. En effet, l'obtention d'un meilleur QoS au niveau de groupe ou terminal ne signifie obligatoirement une meilleure QoS au niveau du réseau, donc, il faut mettre des mécanismes adéquats pour gérer les interactions entre les différents groupes, l'agent mobile est l'un de ces mécanismes important qui peut être utilisé pour l'actualisation des routes, la réduction de l'overhead engendré par l'échange des messages de contrôle et peut offrir un meilleur

rendement. La QoS dans ce niveau est assurée principalement par les agents représentant, les agents passerelles et aussi les agents membres. Les agents mobiles ont la possibilité d'émigrer leur code, leur état et leurs données pour continuer l'exécution dans d'autres terminaux. Cela permet à ces agents de parcourir le réseau en sauvegardant les routes entre les différents groupes. Chaque fois qu'un agent mobile arrive à un représentant une route est actualisée.

4.3. Processus de construction de groupes

4.3.1. Découverte des voisins

En effet, chaque terminal mobile peut posséder de capacité différente et envoyer sa table de routage aux voisins. Cette table de routage peut contenir l'information comme, la capacité de terminale C_{ni}, l'identifiant, le nombre de voisins, la sécurité etc. Lorsque chaque terminal possède l'information nécessaire sur lui-même et sur leurs voisins, l'étape suivant va exécuter.

4.3.2. Sélection des chefs

Dans cette étape, chaque ensemble de nœuds élise leur chef qui possède la grande capacité C_{ni} et également détermine le vicaire qui est la moins capacité par rapport le chef.

4.3.3. Construction des groupes

Après la sélection des chefs, les groupes vont établir où chaque groupe contient un ensemble de terminal mobile et bien sûr avec le chef et vicaire. Le nombre des nœuds dans chaque groupe n'est nécessairement le même mais ne dépasse pas le nombre maximum.

4.4. Algorithme de construction des groupes

En effet, plusieurs algorithmes ont été proposés dans la littérature pour la construction et la maintenance des groupes. Nous allons essayer de proposer un algorithme de construction des groupes selon la fonction d'optimisation qui nous présentons auparavant.

- Chaque agent nœud calcule la capacité de nœud C_{ni} (selon la fonction d'optimisation), créer un agent transporteur, donner sa table de routage et l'envoyer à tous les agents nœuds adjacents.
- Chaque agent nœud reçoit le(s) agent transporteur(s) vérifie le mot de passe pour accepter ou non, ajouter les informations à sa table de routage et comparer sa capacité avec celle des autres agents.

- Après un temps prédéfini l'ensemble des agents décide de faire l'élection entre eux.
- Lors de construction de groupe, il faut prendre en considération, chaque groupe constitue un nombre limite d'agent nœud.
- L'agent qui possède la grande capacité élu comme représentant du groupe (chef) et l'agent qui possède le moins capacité par rapport l'agent représentant élu comme agent vicaire, tandis que, les autres considère comme membres.
- Après de déterminer tout les groupe, les agents membres qui appartient au moins à deux groupes déclarent agents passerelles.
- Chaque agent membre peut quitter son groupe à un autre groupe.
- On peut ajouter un nouvel agent nœud à un groupe, mais, il faut prendre l'accord de l'agent représentant, soit pour la sécurité, soit pour le nombre maximum de nœuds.
- Si, l'agent représentant déplacer à un autre groupe ou atteinte de seuil de qualité de service, l'agent vicaire le remplacer, s'il présent, sinon, les agents membres font la reconstruction de groupe.

4.5. Architecture interne de l'Agent Mobile

Dans notre modèle, il y a trois type d'agents, Agent Nœud (AN) qui prend l'état membre, passerelle, vicaire et représentant, l'Agent Transporteur (AT) qui est responsable de porter de données d'un nœud source à un nœud destination et l'Agent Routier (AR) qui est chargé de découvrir des routes et réaliser la maintenance.

Nous présentons dans cette partie l'architecture interne de, l'agent nœud, l'agent nœud dans l'état membre, l'agent transporteur et l'agent routier. Cette architecture est à base de composants pour obtenir l'adaptabilité, l'évolution, la réutilisation de code, etc. L'architecture d'agent est par assemblage de composants où chacun implémente une partie des fonctionnalités de l'agent en ajoutant un raisonnement centralisé afin de fournir l'autonomie et la flexibilité aux applications.

Il faut noter que, le comportement d'agent, soit réactif pour répondre immédiatement aux événements nécessitant une réaction rapide (par exemple, coupures des liens, mobilité forte, transmet un message RREP, etc.), soit proactif qui permet à l'agent d'équilibrer entre les exigences des applications en terme de qualité de service (par exemple, réception de données avec le prend en considération la sécurité, évaluation les ressources de nœud, taux de perte négligeable, etc.), soit hybride qui composée les deux (réactif et proactif).

On peut également distinguer entre deux type d'agents : les agents légers et les agents lourds [44]. Les agents légers exécutent une tâche qui effectue de courtes phases de calcul local et ils vont donc migrer fréquemment et rapidement. Il s'agit d'agents de petite taille dans le sens où leur code exécutable est réduit le plus possible et les informations qu'ils véhiculent sont peu volumineuses. Cette petite taille va sa donner la faculté d'un déplacement très bref dû à un temps de transmission très court grâce à leur faible coût en bande passante. Cette caractéristique est fort appréciable dans les réseaux mobiles ad hoc visés où le lien entre deux nœuds a une durée de vie limitée et réalisé par une connexion sans fil peu généreuse en bande passante.

A l'opposé des légers, les agents lourds réalisent une tâche imposant de longues phases de traitements locaux et en conséquence ils effectuent de rares déplacements qui sont relativement lents. Ces agents sont dits « lourds » car la taille du code exécutable ainsi que celle des données transportées seront beaucoup plus volumineuses que celles des agents légers. Cette propriété va poser un problème dans les réseaux mobiles ad hoc qui ne peuvent pas garantir la continuité des liens de communication et ne peuvent proposer qu'un faible bande passante, mais on peut les utiliser dans quelque cas qui soumis à certain condition, par exemple transmission de données à un saut. Dans la figure suivante, nous présentons l'agent qui appelé agent nœud, sa localisation dans le nœud et la ressource **R** qui est interne par rapport le nœud et externe par rapport l'agent.

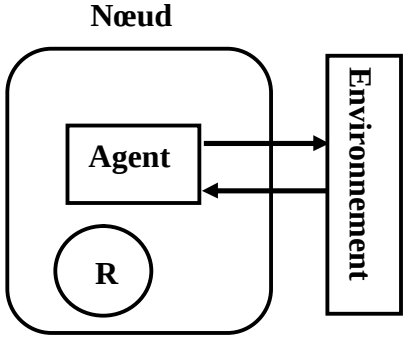

FIG. 4.2 - Localisation d'agent nœud

4.5.1. Architecture interne de l'Agent Nœud

La figure 4.3 présenté l'architecture interne de l'agent nœud et les principaux composants qui permettent l'implémentation d'agent sont les suivants :

- **Le composant de raisonneur** : le raisonneur joue le rôle de connecteur entre les micros composants, le régulateur et le point de passage obligé de tout appel de méthode des composants. Il offre également des primitives de changement des composants.

- **Le composant de décision** : il permet à l'agent de sélectionner la meilleure action à exécuter parmi l'ensemble des plans envoyés par le composant de raisonneur, cette sélection est basée sur des mesures multicritères de la qualité de service [20].
- **Le composant analyseur** : un analyseur implémente une politique d'adaptation qui est un ensemble de règles définies pour un ensemble de composants d'un agent et un ensemble d'évènements liés aux variations du contexte d'exécution. Lorsqu'il détecte un changement dans l'environnement, il notifie le composant de raisonneur qui émet à son tour une notification individuelle à chaque agent vivant sur son site [54].
- **Le composant de sécurité** : ce composant a pour fonction d'assurer la sécurité d'agent contre tous les accès malveillants, il les vérifier par des mécanismes bien déterminé comme, confidentialité, authentification, clé symétrique ou/et asymétrique, etc.

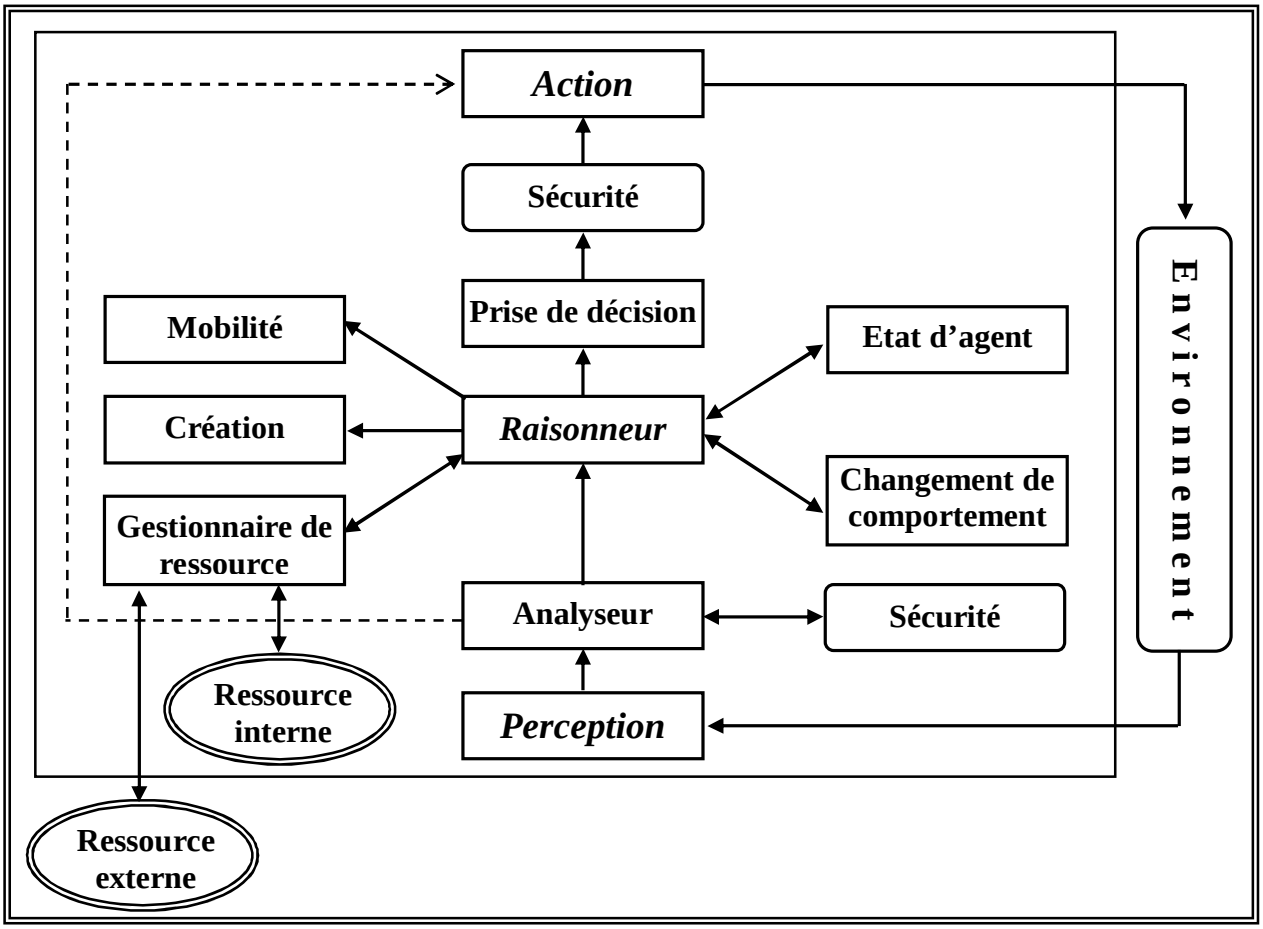

FIG. 4.3 - Architecture interne de l'Agent Nœud

- **Le composant de gestionnaire des ressources** : leur objectif est la gestion et l'évaluation local de la ressource de nœud (batterie, l'espace mémoire, la charge de CPU, etc.) afin d'obtenir une qualité de service plus favorable dans chaque nœud.

- **Le composant de mobilité** : est un composant important d'un agent où un agent peut créer un agent mobile qui lui permet de s'exécuter dans différents noeuds d'un réseau. On distingue deux types de mobilité, soit l'exécution à distance où l'agent est transféré vers le site de destination avant le début de l'exécution (code et données), il y reste jusqu'à la fin de l'exécution de la tâche, soit la migration où cette migration est réalisée grâce à des performatifs de migration pendant l'exécution de l'agent; leurs invocations provoquant la capture de son état, son encapsulation, et son transfert au site de destination.

4.5.1.1. Architecture interne d'agent nœud de l'état membre

Cette architecture est un cas particulier de l'architecture d'agent nœud présentée auparavant. En effet, l'architecture interne de tous les états d'agent nœud (membre, passerelle, vicaire et représentant) est similaire. La différence est seulement dans le rôle de chaque état et en particulier dans le protocole de routage entre le nœud source et le nœud destination (voir le chapitre suivant). Donc, nous exposons dans la figure 4.5 comme exemple de l'architecture interne d'un seul état qui est l'état membre.

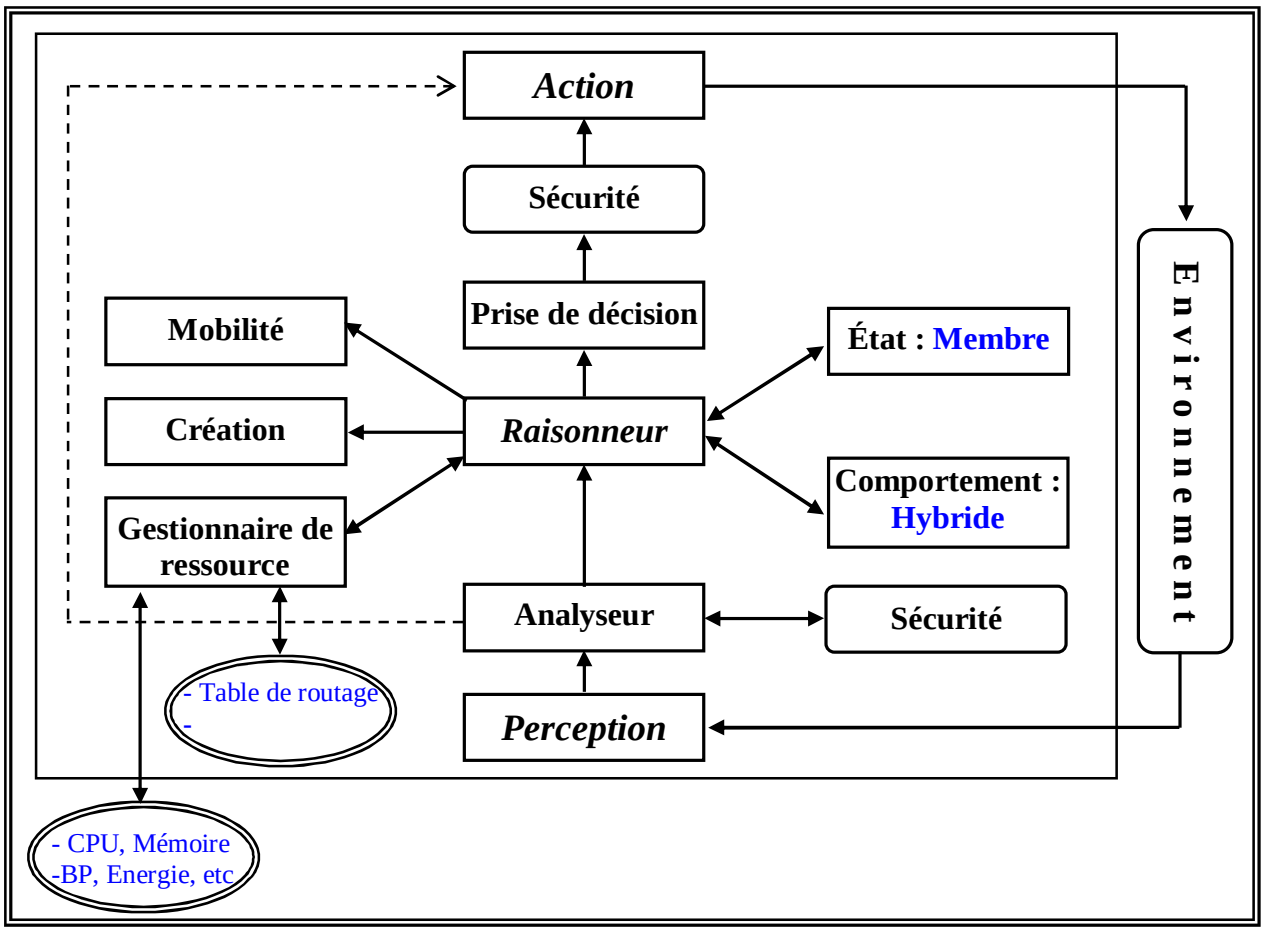

FIG. 4.4 - Architecture interne d'agent nœud de l'état membre

4.5.2. Architecture interne de l'Agent Transporteur

Le cycle de vie d'un agent transporteur est initialisé soit actif, en attente, suspendu ou se déplace. Dans cette architecture le composant de négociation joue un rôle important dans la communication avec les autres agents lors le déplacement.

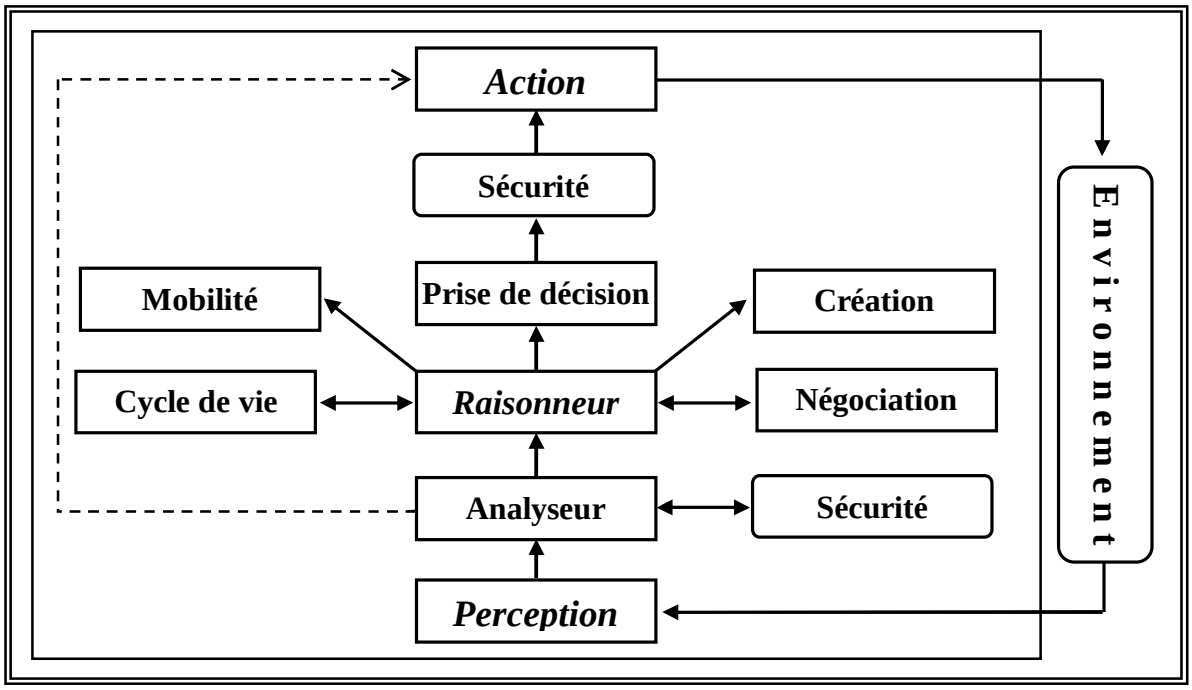

FIG. 4.5 - Architecture interne d'Agent Transporteur

4.5.3. Architecture interne de l'Agent Routier

L'agent routier est un agent réactif et léger, son objectif est de découvrir et maintenir une route entre le nœud source et le nœud destination. En effet, il est moins important par rapport aux autres, son architecture est simple, il a une base de connaissance et moins sécurité.

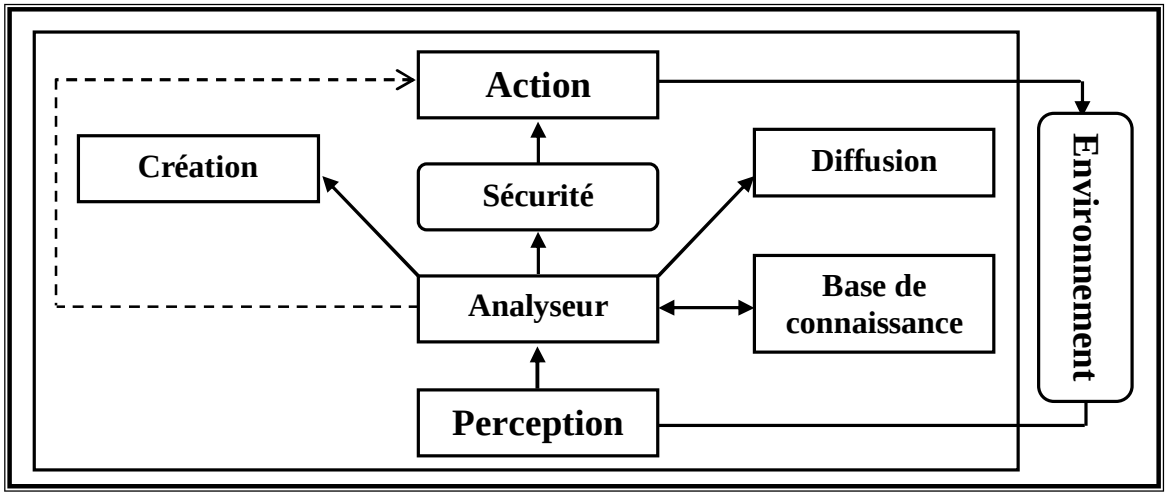

FIG. 4.6 - Architecture interne d'Agent Routier

4.6. Diagramme d'état de transition

Les diagrammes d'états de transitions permettent de décrire les changements d'états d'un objet ou d'un composant, en réponse aux interactions avec d'autres objets/composants ou avec des acteurs.

On suppose qu'on organise notre réseau comme nous présentons dans la figure 4.1, après la création d'un agent nœud dans chaque terminal, il calcule les ressources du nœud, créer agent transporteur, livrer les données, envoyer à tous les agents nœuds voisins, entrer dans un cluster. Le cluster n'est pas crée dans deux cas, le premier cas est l'état initial de réseau (il n'y a pas aucun groupe et chaque agent nœud avoir sa table de routage et les tables de routage de tous agents nœuds voisins), le deuxième cas, après un temps égal par exemple **Δt**, plusieurs nœuds sont hors de tous les clusters et ils veulent créer un nouveau cluster. Dans ce cas, il faut collecter leurs informations (table de routage et surtout la capacité de chacun) entre eux, puis, élire l'agent nœud qui a la grande capacité (les paramètres qui déterminent la capacité présentés en dessus) comme un agent représentant, les autres agents nœuds sont considéré comme des agents membres sauf ceux qui appartient au moins à deux clusters ils appellent agents passerelles. Si l'agent représentant est absent, c'est à dire, il se déplace à un autre cluster ou il dépasse le seuil de qualité de service, alors, il devient un agent membre et l'agent vicaire le remplace dans le cas où il est présent.

La figure 4.8 explique seulement les états de transitions d'agent nœud d'un état à l'autre (état membre, état passerelle, état vicaire et représentant) lors de l'organisation du réseau.

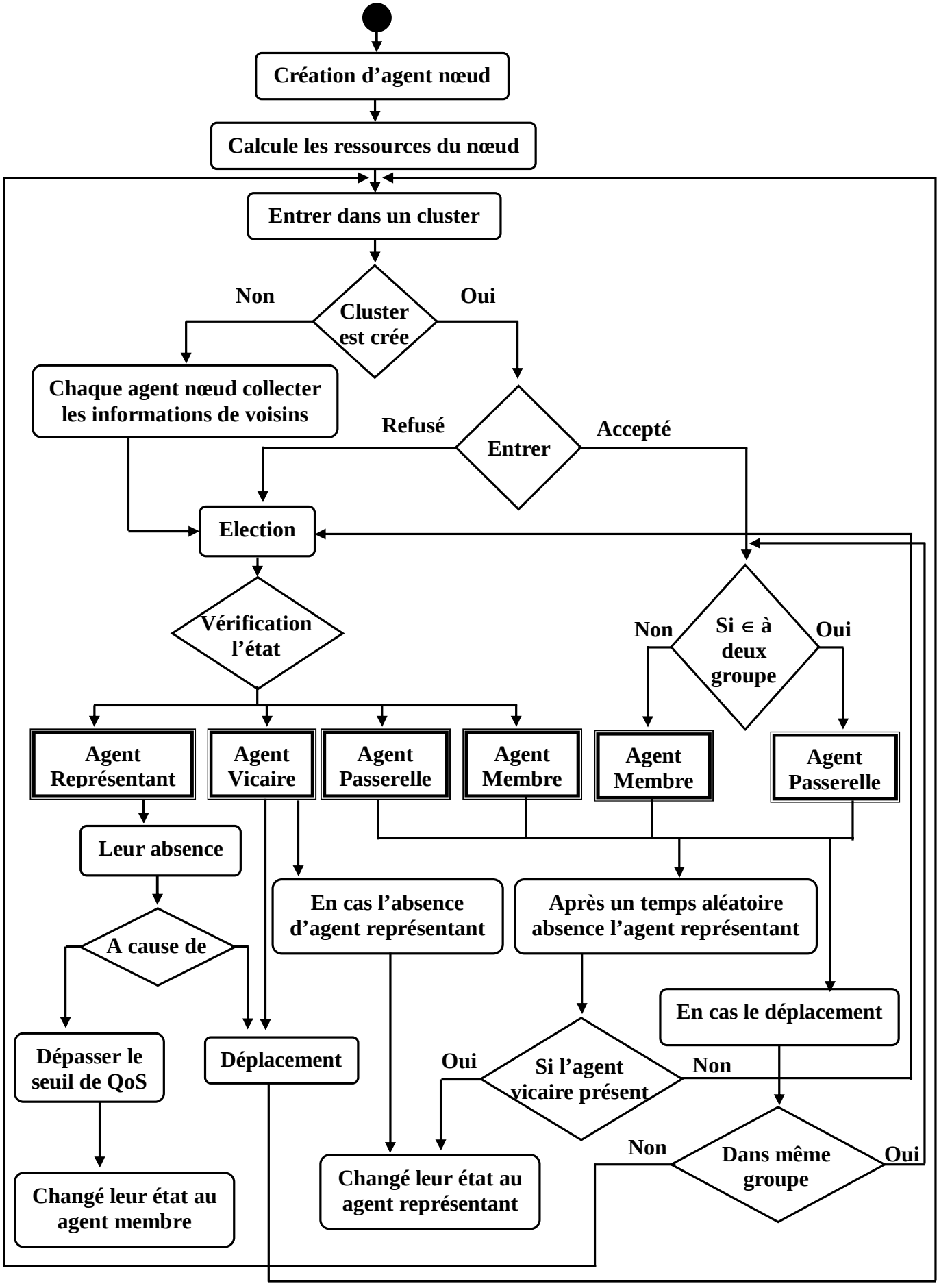

FIG. 4.7 – Diagramme d'état de transition d'agent noeud lors organisation de réseau

4.7. Diagramme de classes de modèle

4.7.1. Agent nœud

L'élément de base correspond au agent nœud représenté par la classe *AgentNoeud*. Il est identifié à l'aide d'un identifiant unique représenté par l'attribut *AgentID*. Il est également caractérisé par l'attribut *Name*. L'attribut *EtatAgent* détermine l'état d'agent nœud : état représentant, vicaire, passerelle et état membre. On remarque que, l'agent nœud ne reste toujours en même état à cause de la mobilité qui lui permet de changer son état. En effet, l'agent nœud peut prendre les quatre états précédents dans la durée de vie et le temps qui lui prend en chaque état est différent.

Du fait de sa mobilité et de ses contraintes en énergie et en bande passante, l'agent nœud est susceptible de quitter ou de rejoindre le réseau à tout instant. L'attribut *JoinTime* indique la date à laquelle l'agent à rejoint le réseau, tandis que l'attribut *LeaveTime* précise la date à laquelle il a quitte le réseau. L'attribut *LeaveTime* peut facilement être renseigné si l'agent nœud quitte proprement le réseau, c'est-à-dire si un protocole d'annonce lui permet d'informer les autres agents nœuds (par exemple agent représentant) de son départ. En revanche, si l'agent nœud quitte le réseau de manière abrupte, le départ d'agent nœud est uniquement pris en compte lorsque celui-ci n'a pas donné présence de vie au delà d'une certaine période de temps. L'attribut *SeuilQoS* permet de déterminer le niveau de capacité d'agent nœud et selon cet attribut peut décider de continuer dans un même état (état représentant) ou le changer. L'attribut *DégN* détermine le nombre de voisins de nœud.

Un autre attribut *NbDep* est le nombre de déplacement d'agent nœud qui est déterminé dans un temps défini, grâce les attributs *JoinTime* et *NbDep,* on peut calculer la moyenne de mobilité d'agent nœud. Les attributs *UsageMem* et *UsageCpu* indiquent respectivement l'état d'espace mémoire et l'état de charge CPU. L'attribut *Bandepass* exprime l'état de la bande passante de l'agent nœud. La sécurité d'agent nœud est représentée par l'attribut *SecurAgent.* Un dernier attribut permet d'obtenir la capacité générale d'agent appelé *CapAgent.* Par ailleurs, les comportements (méthodes) d'agents comme, *Calculcap* permet d'obtenir la capacité de nœud (C_{ni}) selon la fonction d'optimisation. Les méthodes *getUsageMem, getUsageCpu, getDégN,* donnent respectivement l'espace d'utilisation de mémoire actuel, le taux d'utilisation courant de CPU et le nombre de voisins de chaque agent nœud dans le réseau.

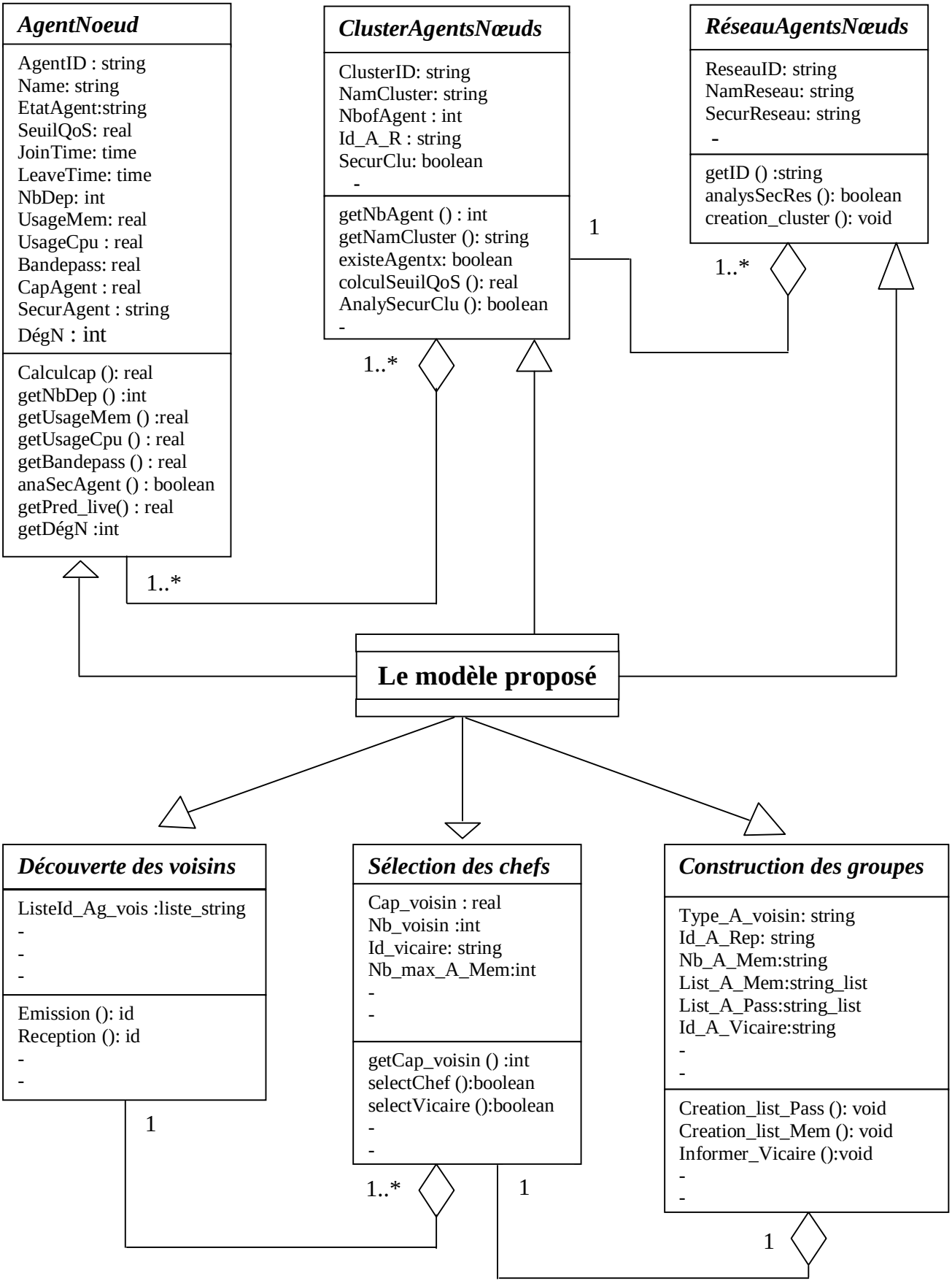

FIG. 4.8 - Diagramme de classes de notre modèle

4.7.2. Cluster d'agents nœuds

Les agents nœuds sont ensuite regroupés sous la forme de clusters. Un cluster des agents nœuds est représenté par la classe *ClusterAgentsNœuds.* Elle dispose de l'attribut *ClusterId* et *NamCluster* pour identifier un cluster par identification unique. Le nombre d'agents nœuds présent dans le cluster est également spécifié, à l'aide de l'attribut *NbofAgent.* La classe *ClusterAgentsNœuds* est reliée à la classe *AgentNoeud* par une relation d'agrégation. La cardinalité indique qu'un cluster est au moins composé d'un agent nœud mais qu'un agent nœud peut appartenir à plusieurs clusters. L'attribut *Id_A_R* permet de donner l'identification de l'agent représentant. Un cluster d'agents nœuds est caractérisé par l'attribut *SecuClus* qui empêche tous les accès malveillants à un cluster.

4.7.3. Réseau d'agents nœuds

Un réseau d'agents nœuds est organisé comme un ensemble de clusters d'agents nœuds. Il est représenté par la classe *RéseauAgentsNœuds* dans notre modèle. Elle est identifié à l'aide d'un identifiant unique représenté par l'attribut *ReseauID*. Il est également caractérisé par l'attribut *NamReseau* et L'attribut *SecurReseau*.

4.8. Propriétés du modèle

- Notre modèle est consacré pour la qualité de service dans les réseaux mobiles ad hoc et plus précisément pour la routage avec QoS qui est la problématique principale dans ce type de réseau.
- Exploitation facile du réseau, pour cela nous décomposons le réseau en un ensemble de niveaux pour obtenir une vue hiérarchique qui rendra le déploiement des protocoles prenant en compte la qualité de service plus rapide et plus aisé.
- Sécurité de réseau, nous utilisons le concept de sécurité qui est intégré dans l'agent au niveau de chaque nœud et au niveau de chaque cluster.
- Nous employons la technologie d'agent mobile issue du domaine de l'intelligence artificielle qui a donné des avantages importants par rapport aux autres technologies et plus approprié pour les environnements dynamique. Pour cela, on associé à chaque terminal un agent qui permet d'introduire un degré d'intelligence à chaque nœud du réseau. En outre, l'utilisation de la technologie d'agent mobile permet de surmonter les problèmes liés à la déconnexion des nœuds. En effet, un terminal mobile crée un agent mobile et lui demande d'agir pour son compte. L'agent créé, après sa migration, peut

s'exécuter dans le système même si le site mobile créateur fonctionne en mode déconnecté. Une fois le nœud du client connecté, il va contacter l'agent mobile afin de lui demander de revenir sur son site d'origine.

- Les tâches attribuées aux terminaux de manière inéquitable selon les capacités hétérogènes, cela implique une économie d'énergie.
- Nous avons bénéficié à travers l'architecture interne de l'agent des avantages de deux paradigmes « agent » et « composant » qui offrent la flexibilité à notre modèle et par conséquent, ils peuvent donner un bon fonctionnement pour la fonction de routage avec qualité de service et les autres fonctions.
- En comparaison de ce modèle avec les autres travaux expliqué dan le chapitre précédent. On peut noter que, chaque travail traite presque un seul paramètre de QoS, tandis que, ce modèle offre plus de paramètre de QoS.

4.9. Conclusion

Dans ce chapitre, nous avons présenté, le modèle de qualité de service à base d'agent mobile, ce modèle définit quels types de service peuvent être fournis dans un réseau ad hoc et certains mécanismes utilisés afin d'offrir ces services. L'utilisation de topologie de cluster permet d'obtenir une vue hiérarchique du réseau afin de gérer au mieux la QoS associée à chaque niveau. Il prend en considération certaine paramètre de qualité de service, comme, la sécurité, le taux de perte, le délai, l'énergie, etc. Cependant il est très difficile de garantir certaine qualité de service, car les contraintes imposé par ce type de réseaux, notamment : la mobilité des nœuds, la topologie dynamique et variable, l'hétérogénéité des terminaux.

L'accent dans ce modèle est mis sur la gestion efficace des ressources afin d'améliorer la QoS dans un réseau où sa topologie peut varier d'une manière rapide et aléatoire. De plus, nous avons modélisé l'agent par un assemblage de composants afin d'en faciliter l'implémentation, ainsi de bénéficier des avantages du paradigme de composant. En outre, nous avons présenté le diagramme de classe de notre modèle qui est constitué d'agent ad hoc, cluster d'agents et réseau d'agents ad hoc.

Chapitre 5

Protocole de Routage et Environnement de développement

5.1. Protocole de Routage à base d'Agent Mobile (PRAM)

5.1.1. Introduction

Comme nous l'avons vu dans les chapitres précédents, l'étude et la mise en œuvre des protocoles de routage intégrant de la qualité de service pour les MANETs constituent une problématique d'actualité. En effet, l'opération de routage se heurte à de nombreuses difficultés car la recherche de routes s'appuie sur des informations dynamiques. Des mécanismes (réguliers ou utilisés seulement lors de la recherche de routes) doivent exister pour obtenir une route valable. En claire, les nœuds ne peuvent s'appuyer sur une information statique, et doivent obtenir lors de la recherche de route une information la plus « fraîche » possible. Trouver un chemin n'est qu'une partie du problème, il faut pouvoir assurer la stabilité des communications car la mobilité des nœuds peut entraîner de nombreuses reconfiguration des chemins. Ainsi, durant la communication, l'ensemble des relais d'une communication va changer plus ou moins fréquemment [8].

Dans ce chapitre, nous avons utilisé notre modèle pour proposer un nouveau protocole de routage ***PRAM*** (**P**rotocole de **R**outage à base d'**A**gent **M**obile), qui inspiré essentiellement du protocole CBRP (voir chapitre 2) afin d'assurer ou d'améliorer certaines QoS, et de fournir plus d'adaptabilité aux variations des conditions d'exécution. Il s'agit de bénéficier conjointement des avantages de la technologie « agent mobile » et de la topologie virtuelle de « cluster » pour prendre en compte les contraintes de ce type de réseau, et d'offrir une intelligence pour une gestion distribuée et adaptative de qualité de service.

5.1.2. Les éléments du SMA

Le modèle qui nous avons proposé dans le chapitre précèdent est constitué de trois agents principale : agent nœud qui prend l'état représentant, l'état vicaire, l'état passerelle et l'état membre, agent transporteur et agent routier.

5.1.2.1. Agent Nœud

L'agent nœud est un agent qui est implémenté ou installé dans chaque terminal mobile, après le décomposition de réseaux en cluster (groupe) où chaque cluster constitue de plus ou moins de terminal mobile. Chaque agent nœud prend l'un des états suivant : état membre, état passerelle, état vicaire et représentant.

Chaque agent nœud maintient une table de routage, cette dernière est représentée par des structures de données conceptuelles possédant des informations nécessaires. La structure de la table de routage est la suivante :

ID_Voisin	ID_Groupe	Etat	C_{ni}	Seuil_QoS
@ IP_V	@ IP_R	M/P/V/R	%	%

Sachant que :

- ID_Voisin : est un identificateur d'un voisin, nous utilisons l'adresse IP d'un nœud pour l'identifier.
- ID_Groupe : est un identificateur d'un groupe, nous utilisons l'adresse IP et le nom d'un nœud représentant comme l'identificateur d'un groupe.
- Etat : ce champ désigne l'état d'agent nœud, il peut être soit membre, passerelle, vicaire ou bien représentant.
- C_{ni} : représente la capacité du nœud qui est calculé par l'agent nœud.
- Seuil_QoS : représente le degré de capacité, si la capacité d'un terminal mobile atteigne une valeur constante, il faut informer les autres pour réduit leur charge ou peut l'éliminer ou le remplacer.

5.1.2.2. Agent Transporteur

L'agent transporteur est l'agent chargé de transporter les données entre les différents nœuds. Lorsque un agent nœud décide d'envoyer les données à un autre agent nœud, il le crée. L'agent transporteur prend les données puis déplacer d'un nœud à l'autre jusqu'arrive au nœud de destination. Avant de livrer des données, chacun vérifie le mot de passe de l'autre pour assurer la sécurité.

5.1.2.3. Agent Routier

Un agent routier est un agent simple, son rôle est similaire à un message RREQ dans le protocole AODV. L'agent nœud crée l'agent routier, dans le cas où il aurait besoin de connaître une route vers une certaine destination et qu'une telle route n'est disponible. Ainsi,

l'agent représentant envoie périodiquement l'agent routier aux autres agents voisins pour mettre à jours leurs tables de routage. Dans le paragraphe suivant, nous allons détailler le rôle de chaque agent et la coopération entre eux.

5.1.3. Fonctionnalités de PRAM

Dans notre protocole que nous appelons PRAM (Protocole de Routage à base d'Agent Mobile), les trois agents coopèrent entre eux pour assurer la qualité de service.

Lorsque un nœud source S veut envoyer des données à un nœud destination D, dans ce cas, si le nœud destination est un voisin (i.e. on peut l'atteindre par un seul saut) au nœud source, alors, l'agent nœud du source envoie des données directement par un agent transporteur ou bien par un message. Dans le cas contraire (le nœud destination dans un groupe différent ou on n'atteinte que par plus d'un saut) l'agent nœud du source crée et envoi l'agent routier à l'agent représentant pour lui donner une route qui le mène au nœud destination. Quand l'agent représentant a reçu l'agent routier, s'il connaît l'adresse de nœud destination, il le retourne au nœud source avec la permission d'envoyer, il crée un autre agent routier et l'envoyer au nœud destination pour informer et réserver la bande passante, sinon il l'envoie aux autres représentants soit directement ou indirectement par l'agent passerelle, et ainsi de suite jusqu'à ce qu'il trouve le nœud de destination. Dans ce cas, l'agent routier établi la route inverse jusqu'au nœud source, l'agent nœud du source crée un agent transporteur, livrer des données et l'envoyer au nœud destination.

Il est important de mentionner que, pour assurer la sécurité dans le réseau. Les agents utilisent un mot de passe ou n'import quel codage (dans notre protocole, nous avons utilisé un mot de passe) avec la croissant de sécurité d'un niveau à un autre. Par exemple, pour assurer la sécurité, on utilise le cryptage asymétrique où chaque groupe possède deux clés, une est partagée par tous les agents de réseau, l'autre est privé de l'agent représentant. Donc l'agent représentant possède deux clés (publique et privé). Lorsque un agent nœud S veut communiquer avec un agent nœud D, il y a deux cas, si l'agent nœud S est un voisin à un agent nœud D (i.e. dans même groupe), alors, chacun vérifie le mot de passe de l'autre. Dans le cas contraire (agent nœud S dans un groupe et agent nœud D dans un autre groupe), la communication entre les agents représentants, il faut utiliser les mots de passes privés et publique.

Quand, la capacité de nœud est atteinte à une valeur distante (par exemple, Seuil_QoS 5%), si ce nœud est un agent représentant, alors, il faut faire la réélection entre les nœuds de

groupe, dans le cas l'agent vicaire n'est présent. Si, ce nœud est un seul et son état un agent passerelle, alors, il y a deux solution, soit, un autre agent membre le remplace, soit, on fait la reconstruction des groupes qu'ils associent. Si, ce nœud est un agent membre on l'élimine directement. Dans tous les cas, il faut informer tous les voisins pour faire les mises à jours à leurs tables de routages.

5.1.4. Algorithme de navigation et de mise à jour des groupes

L'algorithme de navigation et de mise à jour des clusters doit s'assurer, certaines contraintes de QoS, tous les nœuds du réseau sont actualisés, indépendamment de leurs positions et état dans le réseau.

- Chaque agent représentant envoie périodiquement deux agents routiers, un à tous les agents membres qui sont dans même groupe, l'autre à tous les agents représentants voisin.
- Le premier agent routier saut d'un nœud à un nœud dans un même groupe, collecte des information de ces nœuds et fait la mise à jour, en revanche, le deuxième saut d'un agent représentant aux agents représentants voisins pour faire la mise à jours aussi.
- Chaque table de routage d'agent membre ou passerelle contient l'information nécessaire sur les voisins (i.e. on dit un nœud A est un voisin de nœud B, si, un seul saut entre eux).
- On suppose que, l'agent membre avant de déplacer, peut informer son chef et chaque agent représentant peut se déplacer à l'intérieur de groupe sans problème.
- Au cours de l'opération de transfert des données par un agent transporteur, on suppose que, le nœud destination est déplacé avant l'arrivée, donc, l'agent transporteur communique avec l'agent représentant et réoriente vers un nouvel emplacement.
- On détecte qu'un nœud est en cas d'oscillation (parfois quitte le réseau complètement et le relié après un certains temps différent), alors, on envoie l'agent et il le réside dans le nœud plus proche, s'il détecte le nœud a présenté, il le livrer le(s) donnée (s).

5.1.5. Diagramme d'état de transition

Le diagramme d'état de transition permet de décrire la fonctionnalité de notre protocole PRAM et par conséquent, les comportements de trois agents : agent nœud, agent transporteur, agent routier. Le diagramme suivant résume les étapes qui surviennent entre le nœud source et le nœud destination.

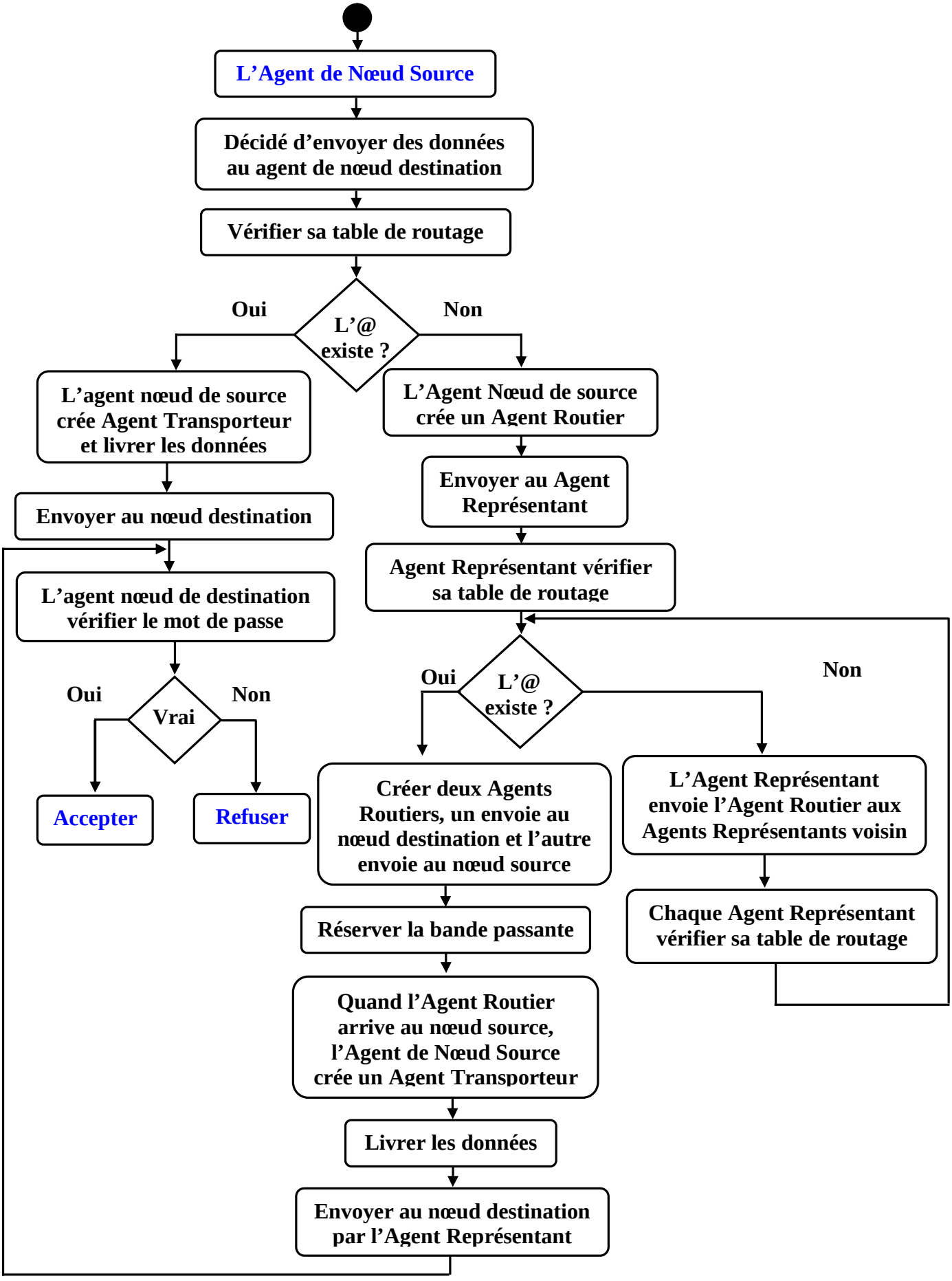

FIG. 5.1 - Diagramme d'état de transition illustre l'envoie des données entre Nœud Source et Nœud Destination

5.1.6. Propriétés du protocole

- Notre protocole prend en compte la différenciation de terminaux. Cette tendance est née suite à l'hétérogénéité des équipements mobiles vendus sur le marché. Cette propriété consiste à préférer les terminaux à haute capacité pour la fonction de routage par rapport aux terminaux de faible capacité.
- Une nouvelle métrique du protocole de routage est alors considérée, où, un agent de nœud à savoir le mot de passe pour éviter les agents malveillants.
- La taille de groupe varie entre un et trois sauts pour s'adapter au changement de la topologie et de la qualité de service.
- Ce protocole utilise une topologie virtuelle permettant l'organisation du réseau sous forme de groupes dynamiques. Il est également employé la technologie d'agent mobile.
- Le protocole PRAM est un exemple de protocole hybride, où, le routage au sein d'un groupe se fait de manière proactive et le routage vers les nœuds extérieurs du groupe se fait de façon réactive.
- Les comportements des agents dans ce protocole sont de manière centralisées et distribuées, chacun prend en charge une partie des fonctionnalités du routage. Et en conséquence, on bénéficie les avantages de deux manières.

5.2. Environnement de développement

5.2.1. Introduction

Généralement, le programme de notre modèle est composé de quatre étapes : la première étape évalue les ressources du nœud comme, la charge du CPU et de la mémoire, le degré du nœud, le niveau d'énergie, la bande passante, la moyenne de mobilité et la sécurité.

La deuxième étape va être la construction des groupes avec l'élection (i.e. déterminer le représentant, passerelles, vicaires et membres de groupe), la troisième étape est la communication entre des agents nœuds pour tester l'implémentation du protocole de routage, la dernière étape c'est la mesure du débit, le temps de réponse, la gigue et le taux de perte.

Dans cette partie, nous allons aborder l'aspect implémentation de notre modèle qui est proposé dans le chapitre précédent. Nous allons réaliser deux étapes : dans la première étape nous allons évaluer les ressources du nœud comme le charge de CPU et Mémoire, le degré de

nœud et on réalise la sécurité entre système multi-agents. La deuxième étape va être la construction des groupes avec l'élection (i.e. déterminer le représentant, passerelles, vicaires et membres de groupe).

5.2.2. Le langage de programmation Java

Pour la mise en œuvre de notre modèle, nous avons opté pour le langage Java. En effet, Java est un langage de programmation orienté objet de très haut niveau, d'utilisation simple et très proche de C++ (la syntaxe représente une version améliorée de C++). Il possède certaines caractéristiques qui nous a amené à développer notre système à l'aide d'il, ces caractéristiques sont :

- Java s'affranchit des plates formes : il fonctionne en mode interprété par opposition aux langages compilés et peut s'exécuter sur de nombreux système d'exploitation.
- Java propose un mode de fonctionnement adapté aux applications réseaux.
- Java est multi-thread : l'utilisation d'un agent indépendant, et autonome nécessite l'emploie d'un langage permettant la concurrence et le parallélisme. Ce langage la doit aussi possède un système de gestion de synchronisation, et comme on a dit dans le deuxième chapitre un agent est un programme autonome toujours en attente, et prêt a rependre aux changements de son environnement; l'utilisation d'un langage multi-thread s'avérer nécessaire.
- La portabilité : java est un langage qui permet l'exécution du système à base d'agents mobiles sur différentes systèmes d'exploitation et sur différentes machines.
- Java offre la possibilité de créer des applications multitâches de façon simple.

5.2.3. La plate-forme d'agent mobile Aglet

En 1996, Big Blue (IBM) met à disposition "Aglets Worbench" qui regroupe une librairie sous forme d'un package Java, ainsi qu'un serveur d'aglets (Tahiti). "Aglets Workbench" sera renommé en 1998, ASDK (Aglets Software Developement Kit). La ASDK est destiné à faciliter la création d'Agent mobile [55].

5.2.3.1. Définition d'un Aglet

Les Aglets [40] vient des mots (Agents Applets), sont des objets Java mobiles qui peuvent se déplacer d'une machine à une autre. Ainsi, un Aglet qui s'exécute sur un hôte peut stopper son exécution, se déporter vers un hôte distant et continuer cette exécution dans son nouvel environnement.

5.2.3.2. Architecture d'un Aglet

Le modèle d'objet d'Aglet définit un ensemble de techniques permettant de créer des agents mobiles dans un réseau du type étendu. Les principaux éléments sont [40, 56] :

- Aglet : est un objet mobile de Java qui visite les serveurs où les agents sont autorisés, dans un réseau informatique. Un Aglet est autonome puisqu'il peut reprendre son exécution dès son arrivée à destination et réactif car il peut répondre (réagir) à des événements de son environnement.
- Proxy : un proxy est un représentant d'un Aglet. Il sert de bouclier à l'Aglet contre l'accès direct à ses méthodes publiques. Le Proxy fournit également la transparence à l'emplacement pour l'Aglet. C'est à dire qu'il peut cacher le vrai emplacement de l'Aglet.

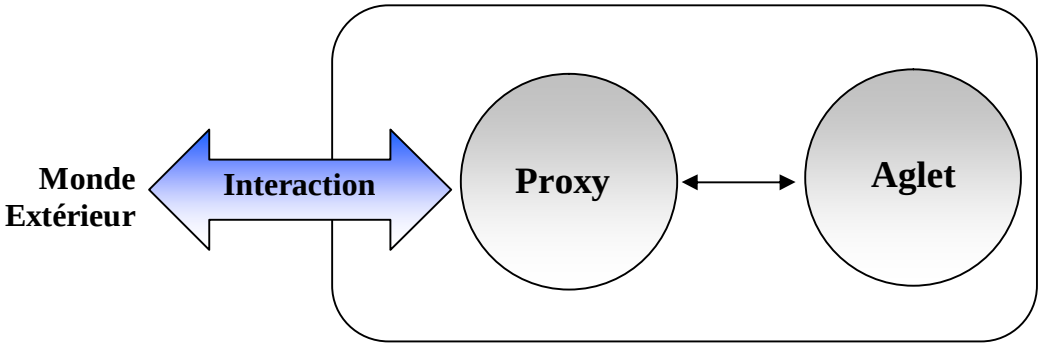

FIG. 5.2 - Relation entre un Aglet et son Proxy

- Contexte : est un objet qui fournit des moyens pour mettre à jour, et contrôler des Aglets dans un environnement uniforme d'exécution où le système hôte est immunisé contre des Aglets malveillants. Un noeud dans un réseau informatique peut accueillir des contextes multiples.
- Hôte : un hôte est une machine capable d'héberger plusieurs contextes. L'hôte est généralement un noeud dans un réseau.

5.2.3.3. Cycle de vie d'un Aglet

Les types de comportement des Aglets ont été implémentés de manière à répondre aux principaux besoins des agents mobiles. Les principales opérations affectant la vie d'un Aglet sont [56,57] :

- Création: la création d'un Aglet a lieu dans un contexte. Le nouveau Aglet est assigné un identificateur, inséré dans le contexte, et initialisé. L'Aglet commence à exécuter dès qu'il sera avec succès initialisé.

- Cloning: le clonage d'un Aglet produit une copie presque identique de l'Aglet initial dans le même contexte, la seule différence est que les Aglets sont relancés.
- Disposition: la destruction d'un Aglet, le stoppera dans son exécution actuelle et la retirera de son contexte actuel.

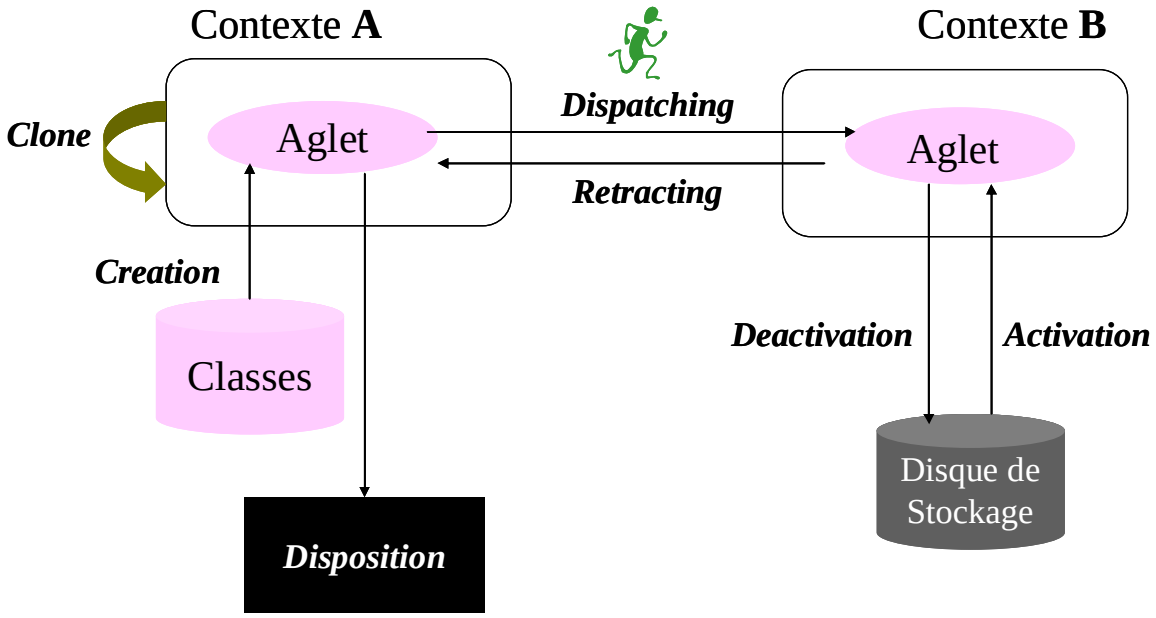

FIG. 5.3 - Cycle de vie d'un Aglet

- Dispatching: la déportation d'Aglet d'un contexte vers un autre, le retirera de son contexte actuel et l'insérera dans le contexte du destinataire, où son exécution sera relancée.
- Retracting: la récupération d'un Aglet, le retirera de son contexte actuel et l'insérera dans le contexte où le retrait a été demandé.
- Deactivation/Activation: la désactivation d'un Aglet permet de l'enlever temporairement de son contexte actuel, et de l'enregistrer dans la mémoire secondaire. L'activation de l'Aglet va le restaurer en mémoire secondaire.

5.2.3.4. Tahiti : un gestionnaire d'agents visuel

Tahiti [40] utilise une interface graphique unique pour suivre et contrôler l'exécution des Aglets. Il est possible en utilisant le glisser – déposer de faire communiquer deux agents ou de les faire migrer vers un site particulier. Tahiti dispose d'un gestionnaire de sécurité paramétrable qui détecte toute opération non autorisée et empêche l'agent de la réaliser.

Le lancement de Tahiti se fait par l'exécution du fichier de commandes « agletsd.bat » (sous plate-forme Windows). Ce dernier exécute le démon Tahiti (le serveur) ainsi qu'un

Viewer d'Aglet qui porte le même nom. C'est grâce à ce dernier que nous pouvons faire des opération précises sur les Aglets : les exécuter, les détruire, les envoyer sur un serveur distant, etc.

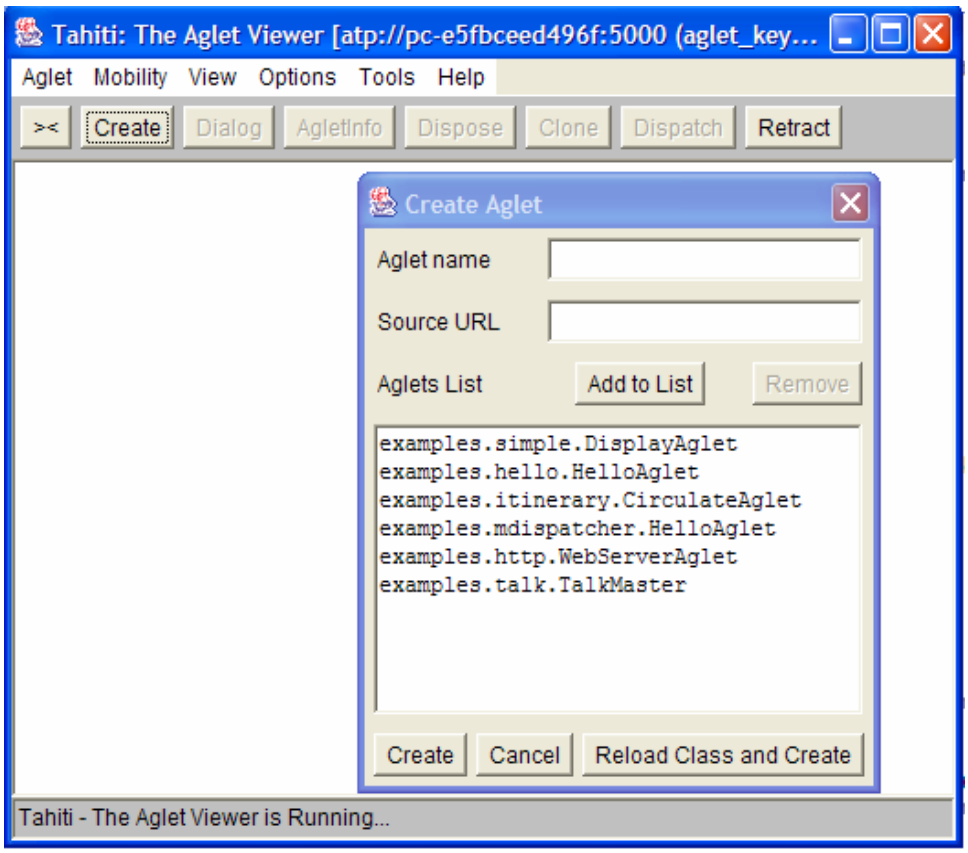

FIG. 5.4 – Illustre le lancement de plate-forme Aglet

Au lancement de Tahiti, le serveur écoute sur le port 4434 par défaut. Il est possible de créer ou choisir un autre port en spécifiant le numéro du port sur la ligne de commande, au lancement de Tahiti.

5.2.4. Présentation de l'application

Notre application est réalisée entre deux nœuds (ordinateurs portables), nous allons ouvrir dans chaque nœud deux ports (i.e. on considère notre application est réalise entre quatre ordinateur). Dans chaque nœud nous lançons un agent nœud, voici le lancement (création) de notre agent nœud dans la plate-forme Aglet.

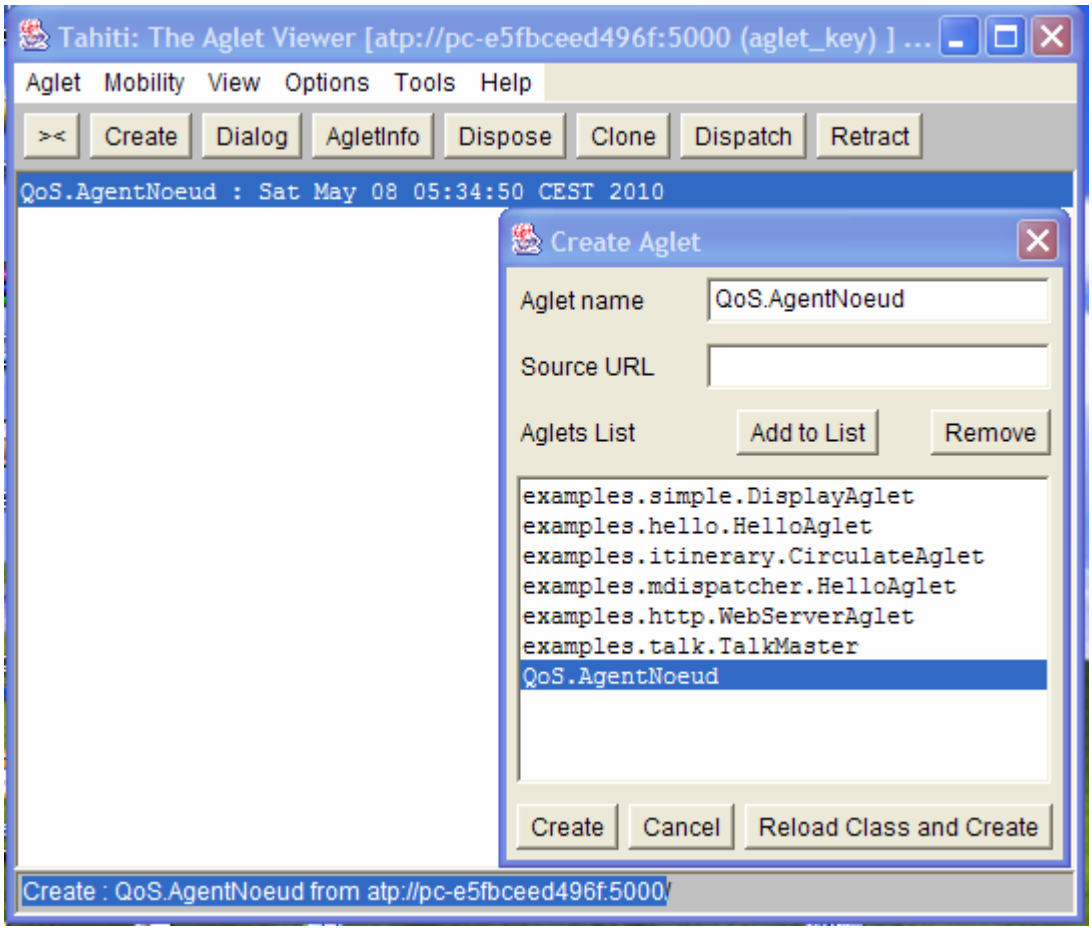

FIG. 5.5 – Illustre la création Agent Nœud

Après le lancement d'agent nœud, il évalue les ressources de nœud, détecte ses voisins, crée un agent transporteur, donne cette évaluation en plus de l'information nécessaire et envoi (l'agent nœud envoie l'agent transporteur) à tous les agents voisins (voir les figure suivantes).

Dans la figure suivante, on remarque qu'après l'évaluation des ressource de nœud 5000, l'agent nœud a des informations et considère lui-même dans l'état représentant, la figure montre la charge de CPU (2 %), la charge de Mémoire (71 %), le degré de nœud (30 %), c'est-à-dire, le nombre de voisins est = 3, dans notre application le nombre maximum de voisin = 10. Selon la fonction d'optimisation qui est proposée dans le chapitre précédent, la capacité de nœud est (C_{ni} = 70 %). On remarque aussi que l'agent nœud détecte ses voisins, dans cet exemple l'agent nœud qui possède le port 5000, ses voisin sont des agent nœuds qui possède les ports 7000,7001 et 5001). L'agent nœud de port 5000 dans l'état initial (fig.5.6) n'avoir que leurs informations, après l'échange des informations (tables de routage) entre eux, il a les informations de tous les agents voisins.

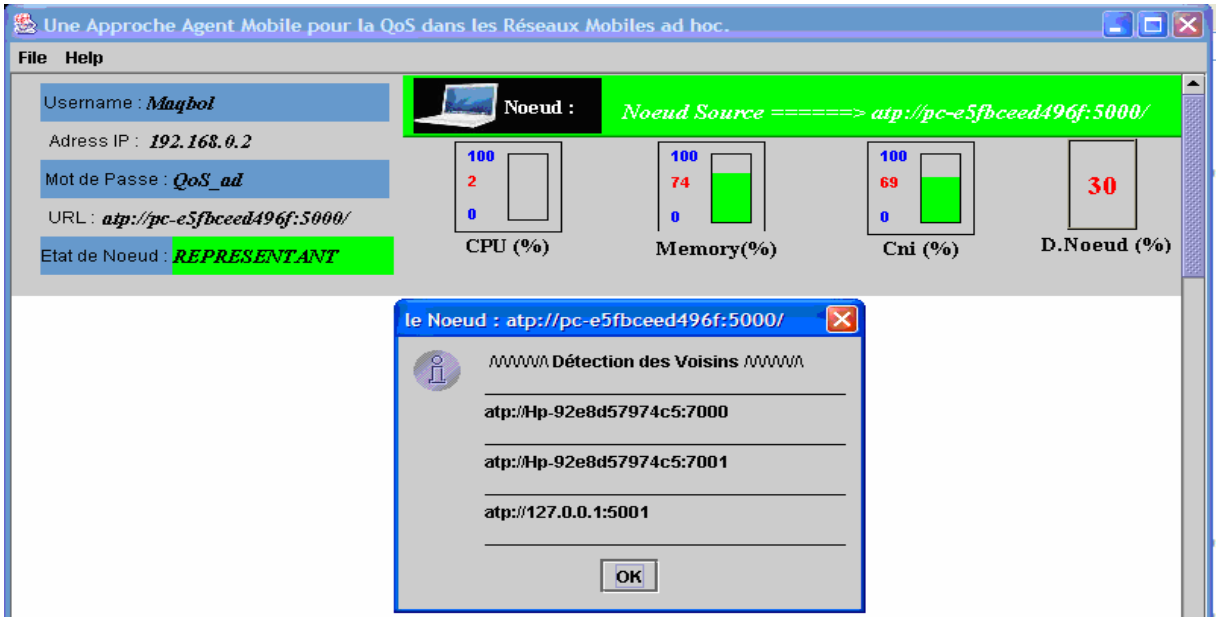

FIG. 5.6 – Illustre l'agent nœud du port 5000 dans l'état initial

Pour chaque agent transporteur du chaque nœud peut accéder à tous ses voisins, il faut qu'il possède le mot de passe (mécanisme pour assurer la sécurité entre tous des agents nœuds) où chaque agent nœud vérifie le mot de passe de l'autre (voir les deux figures suivantes par exemples).

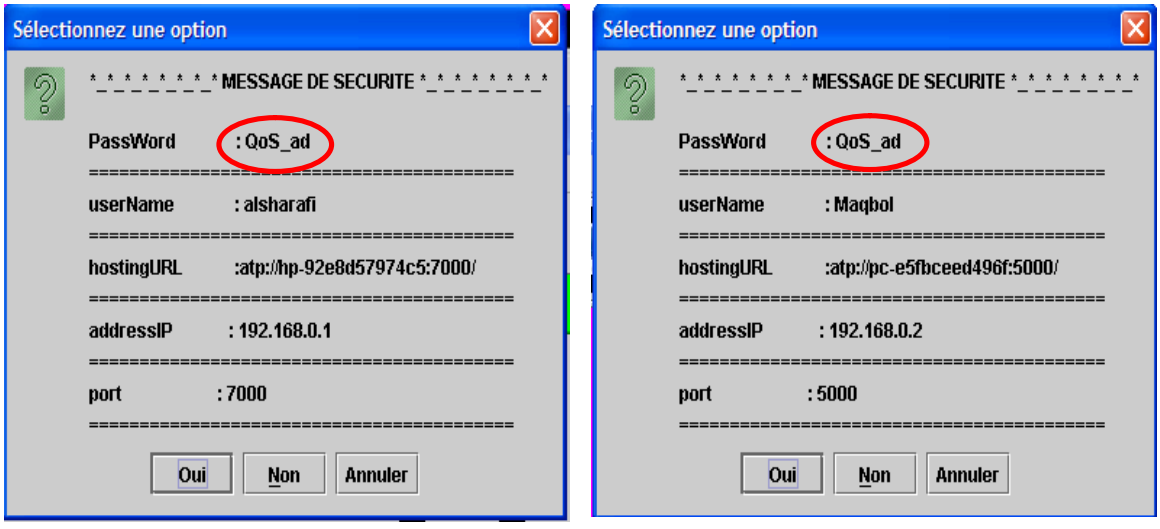

FIG. 5.7 – Illustre la sécurité entre des agents nœuds

Dans chaque nœud, on trouve l'information nécessaire sur lui-même et sur ses voisins. Dans notre application nous utilisons deux nœuds mais on suppose que nous avons quatre nœuds (dans chaque nœud, on peut ouvrir plus d'un port, où, chaque port est considéré comme un nœud). Voici les résultats qui sont affiché dans chaque port après l'échange des informations avec les voisins et après l'élection.

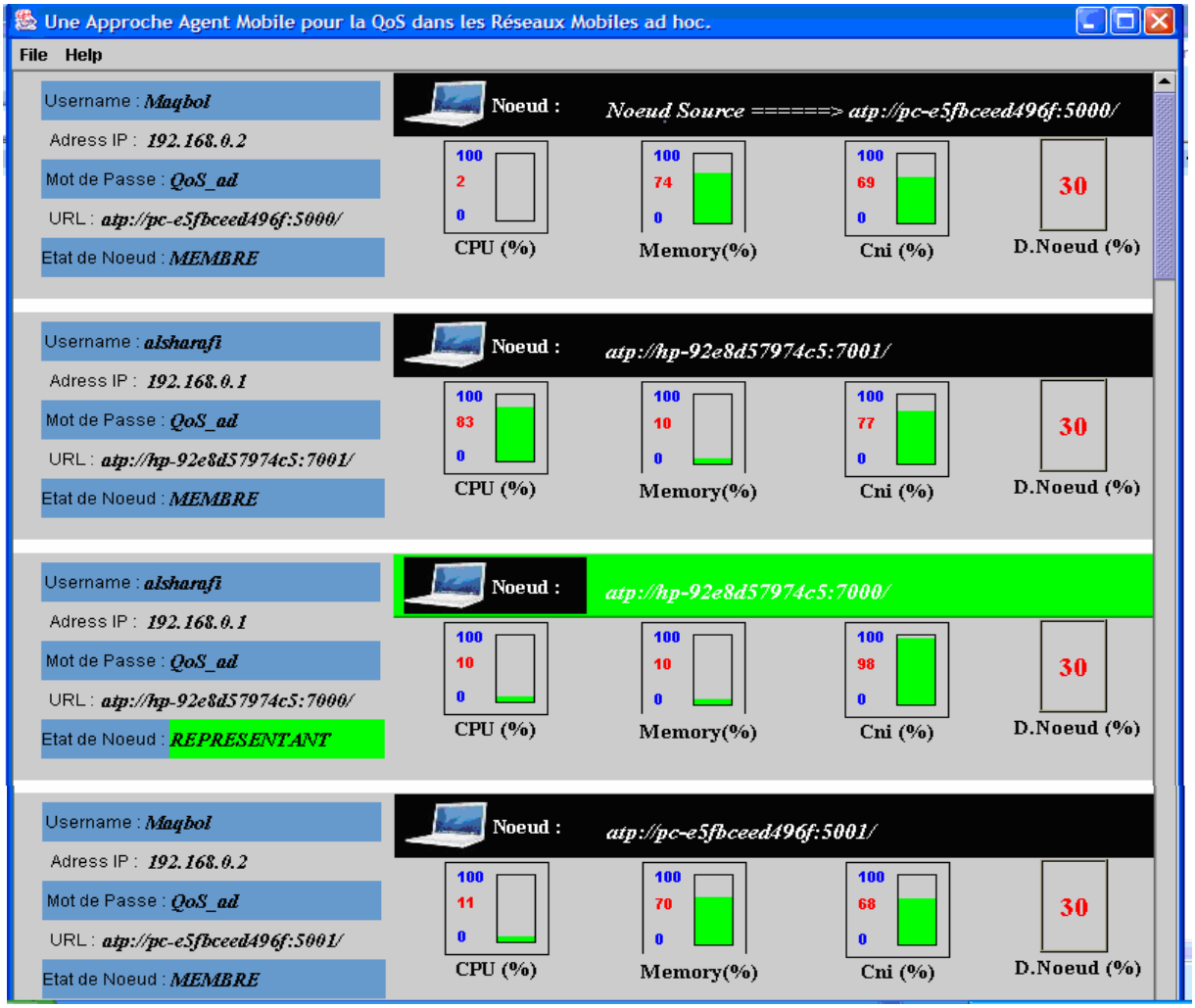

FIG. 5.8 –Illustre l'agent nœud du port 5000 après l'élection

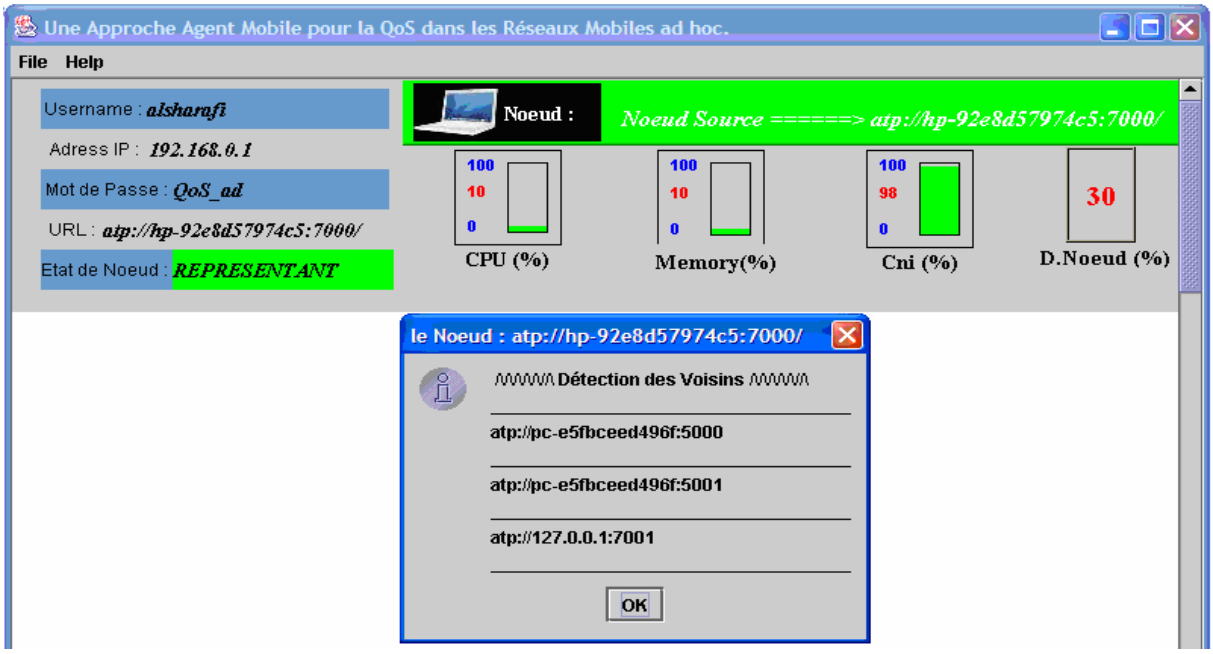

FIG. 5.9 –Illustre l'agent nœud du port 7000 dans l'état initial

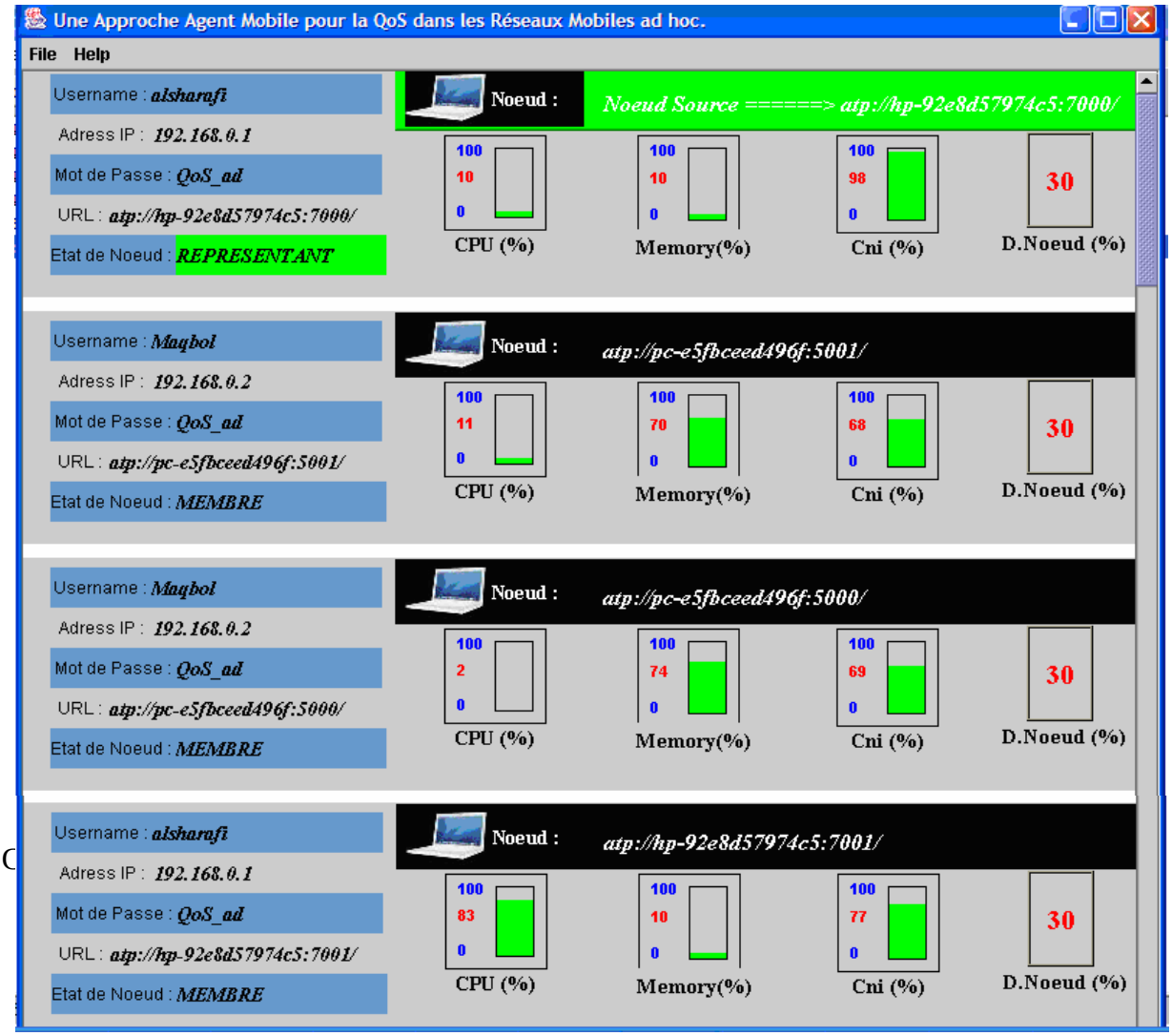

FIG. 5.10 – Illustre l'agent nœud du port 7000 après l'élection

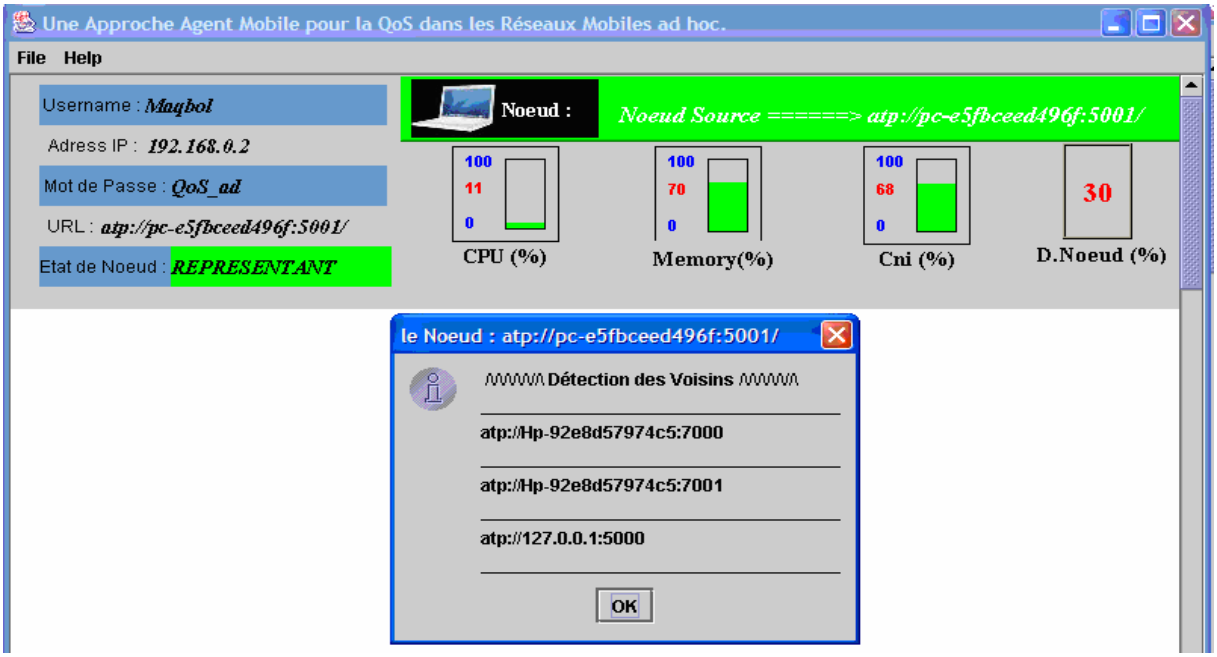

FIG. 5.11 – Illustre l'agent nœud du port 5001 dans l'état initial

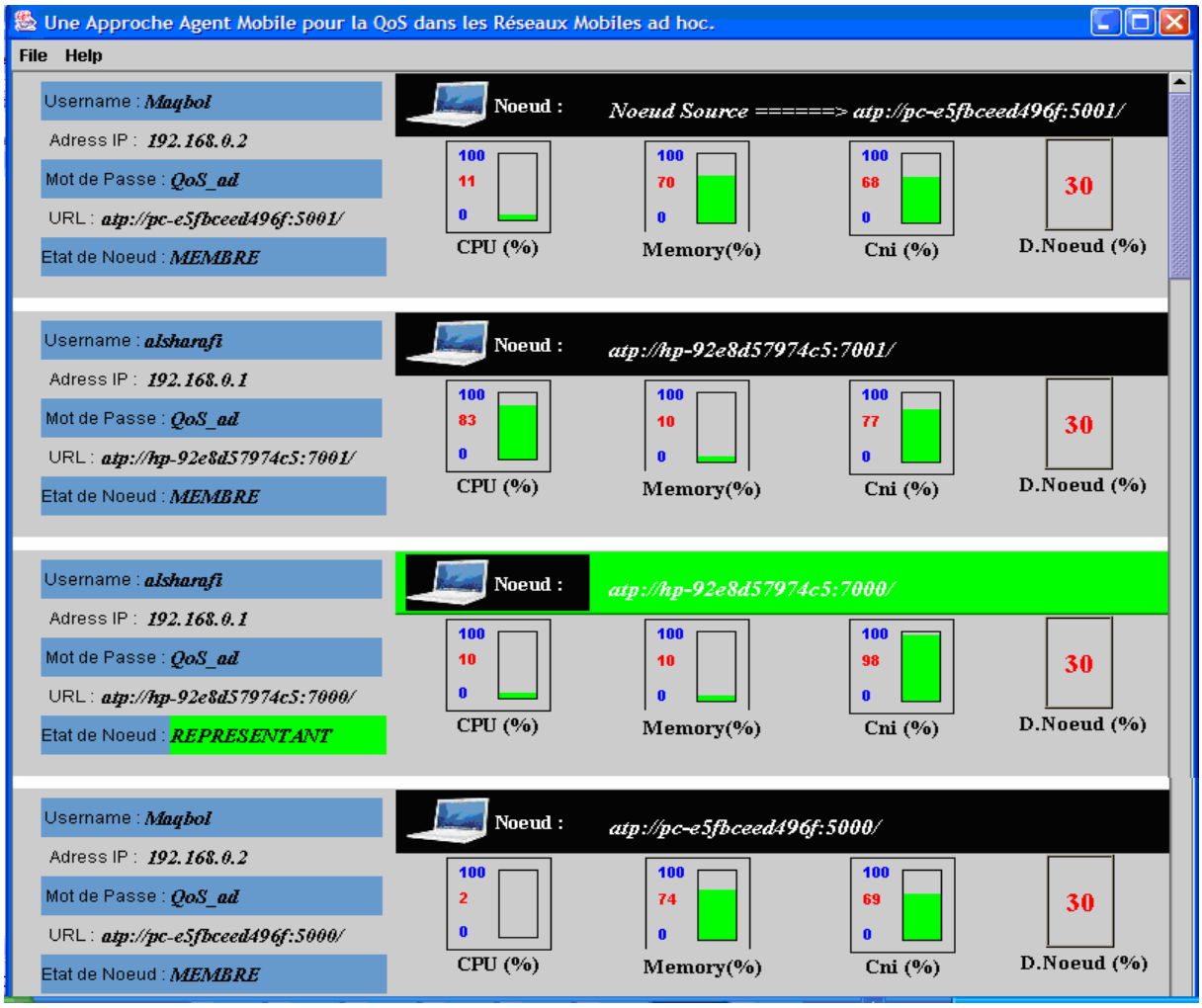

FIG. 5.12 – Illustre l'agent nœud du port 5001 après l'élection

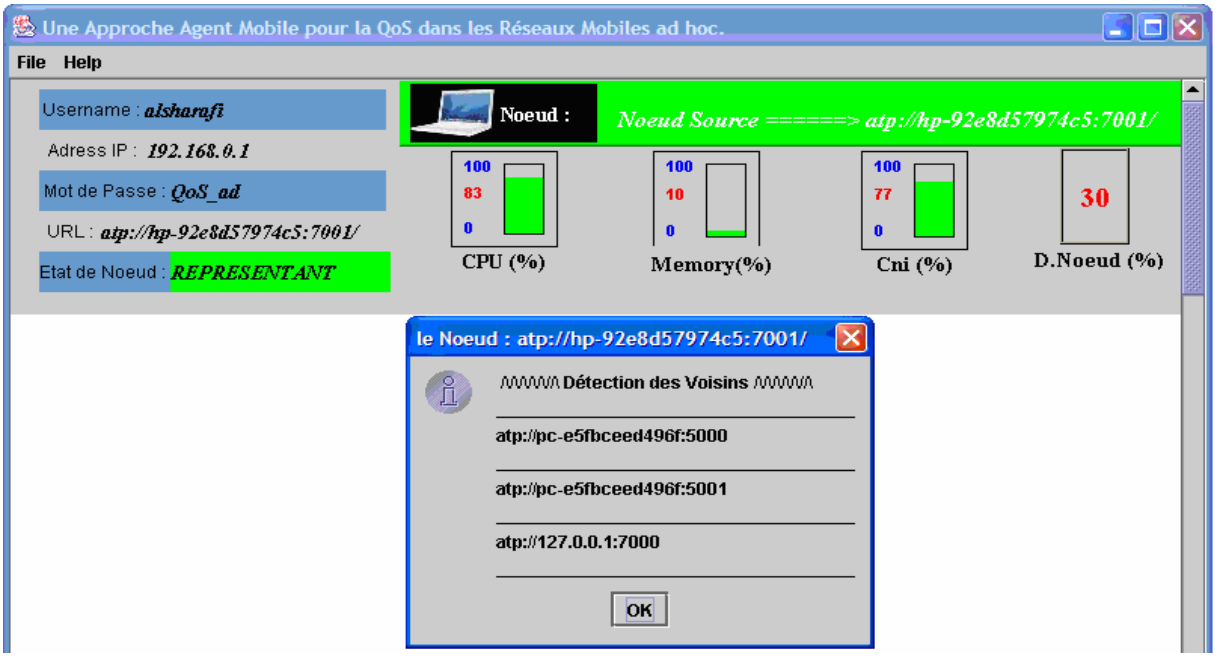

FIG. 5.13 – Illustre l'agent nœud du port 7001 dans l'état initial

FIG. 5.14 – Illustre l'agent nœud du port 7001 après l'élection

Dans les figures précédentes, on obtient un seul nœud (port) représentant dans tous les nœuds celui qui a la plus grande capacité. Dans notre application le nœud du port 7001 est le représentant, tandis que les autres sont membre, les valeurs dans chaque nœud sont les même dans chaque port parce que chaque agent nœud évalue ses ressources et les envoie à tous les voisins.

5.3. Conclusion

La conception et la mise en œuvre de protocole de routage qui répond aux défis de réseau mobile ad hoc représentent une tâche difficile.

Dans ce chapitre, nous avons proposé une solution au problème de routage avec QoS basée sur le modèle présenté dans le chapitre précédent afin d'améliorer la qualité de service

et prendre en compte certaine contrainte de réseau ad hoc. On a également présenté, les éléments du système multi-agents mobiles et leurs fonctionnements, ainsi, la fonctionnalité de notre protocole, l'algorithme de navigation et des mises à jour des groupes, le diagramme d'état de transition entre le nœud source et le nœud destination lors du transfert de données entre eux, etc.

Par ailleurs, le développement de notre application nous a permis de faire une étude réal pour améliorer la qualité de service dans les réseaux mobiles ad hoc par des agent mobiles, nous réalisons l'étape d'évaluation des ressources de terminal mobile par agent mobile et faire l'élection entre eux. En effet, dans notre application nous avons remarqué l'efficacité d'agent mobile pour faire l'évaluation des ressources et le déplacement entre les nœuds mobiles, ainsi, nous avons découvert le comportement des agents mobiles et les réseaux ad hoc et les différentes tendances des chercheurs dans ces domaines.

Conclusion générale et perspective

L'objectif de ce mémoire est d'apporter une solution aux problèmes liés à la qualité de service (QoS) dans les réseaux mobiles ad hoc. Nous avons tenté de comprendre le comportement de réseaux ad hoc, ainsi que les différents problèmes liés à la gestion de la qualité de service dans cet environnement mobile.

Dans un premier temps, nous avons d'abord présentés l'état de l'art de réseau mobile ad hoc. En effet, ce réseau peut être définis comme une collection d'entités mobiles interconnectées par un lien sans fil formant un réseau temporaire sans l'aide de toute administration ou de tout support fixe et avoir une taille importante. Bien que le changement fréquent de la topologie, la bande passante limitée, l'hétérogénéité des terminaux et par conséquent les capacités différentes, les ressources d'énergies et de sécurités limitées. L'utilisation de réseau mobile ad hoc croit d'une manière considérable. Elle est idéale dans les situations où l'installation d'une infrastructure n'est pas possible, parce que l'infrastructure est trop chère ou trop vulnérable. En raison du manque de support d'infrastructure dans les réseaux ad hoc, chaque nœud fonctionne comme un routeur et participe dans l'envoie de données pour d'autres nœuds. La plupart des recherches dans les réseaux ad hoc se concentrent sur le développement des protocoles de routages dynamiques pour trouver des itinéraires entre les nœuds.

Ensuite, nous avons identifié la notion de qualité de service (QoS) et les paramètres qui définissent la qualité de service d'un réseau ad hoc. Ces paramètres sont, en effet, essentiels pour juger la qualité de service. Par ailleurs, nous avons évoqué les mécanismes de QoS, en présentant les modèles de qualité de service proposés pour les réseaux ad hoc et le concept de routage avec qualité de service.

Un autre volet de notre mémoire consacré au paradigme d'agent mobile dans les réseaux mobiles ad hoc, où, on a présenté quelque travaux traitant certain défis par l'utilisation d'agent mobile comme, la sécurité, la découverte de topologie, le protocole de routage, etc.

Enfin, nous avons proposé notre solution par un modèle à base d'agent mobile qui permet d'améliorer la qualité de service. Ce modèle est adopté sur infrastructure hiérarchique en trois niveaux, où chaque niveau a des fonctions prédéfinies. Le premier (niveau nœud) pour la gestion des ressources sur terminal mobile, le deuxième (niveau cluster) gère la QoS dans le cluster et le troisième (niveau réseau) pour gérer les interactions entre les différents groupes. En plus, nous avons employé ce modèle pour proposer un nouveau protocole de

routage PRAM (Protocole de Routage à base d'Agent Mobile), on bénéficie conjointement des avantages de la technologie d'agent mobile et la topologie virtuelle de cluster afin de prendre en compte les contraintes de ce type de réseau. Dans cette solution nous avons essayé de montrer théoriquement ou pratiquement l'augmentation de quantité de données transmises d'une source vers une destination dans une unité de temps. Ainsi, les agents mobiles parcourent continuellement le réseau et mettent à jour les entrées de table de routage. En raison de ceci, un nœud a assez de frais des routes d'un grand nombre de nœuds dans le réseau à n'importe quel moment donné. Donc, la connectivité est plus élevée, et cela mène à réduire les découvertes des routes et réduit le délai de bout en bout. En plus, les paquets qui ne parviennent pas à leur destination par rapport aux paquets émis sont trop moins à cause de la fiabilité du support des liaisons et de la surcharge locale. Ainsi que, la sécurité est bien, grâce l'utilisation des agents mobiles un mot de passe entre eux.

Toutefois, La solution que nous proposons et réaliser n'est sûrement pas la solution miracle au problème de qualité de service dans réseaux ad hoc, nous ne pouvons pas couvrir tous les points de notre sujet mais nous espérons avoir réussi à ouvrir des ports à des études en futur pour développer ou faire des amélioration à cette solution. il est évident qu'il y a des améliorations qui restent à faire pour aboutir à un niveau de qualité de service plus acceptable.

L'évaluation de tout paramètres de qualité de service est l'une de perspectives de notre travail et il serait intéressant de réaliser cette étude dans des conditions vraies réelles. Il y a un certain nombre de paramètres que nous voudrions explorer, par exemple, le temps de réponse, le taux d'erreurs, la charge du réseau complètement, ainsi que la durée de vie d'énergie.

Dans notre travail, dans le cas de la mobilité des nœuds est très forte, notre modèle devient très coûteux à cause de la reconstruction des groupes. De ce fait, il est important d'étudier une solution pour la reconstruction des groupes par une manière simple, efficace et rapide.

Finalement, nous voudrions ajouter une méthode bien structurée pour garantir la sécurité très forte et pour l'analyse des ressources.

Références

[1]: Ouahiba Fouail, « Découverte et fourniture de services adaptatifs dans les environnements mobiles », Thèse doctorat, l'Ecole nationale supérieure des télécommunications, paris, 2004.

[2]: N. Badache, T. Lemlouma, « Le Routage dans les Réseaux Mobiles Ad Hoc », Mini projet, Université Houari Boumedienne, Algérie, Septembre 2000.

[3]: Nabil Kouici, « Gestion des déconnexions pour applications réparties à base de composants en environnements mobiles », Thèse doctorat, l'Institut National des Télécommunications dans le cadre de l'école doctorale SITEVRY en co-accréditation avec l'Université d'Évry Val d'Essonne, paris, 2005.

[4] : Rabah Meraihi, « Gestion de la qualité de service et contrôle de topologie dans les réseaux ad hoc », Thèse doctorat, l'Ecole nationale supérieure des télécommunications, paris, 2006.

[5]: Rémi. Badonnel, « Supervision des Réseaux et Services Ad-Hoc », Thèse doctorat, Université Henri Poincaré – Nancy 1, 11 décembre 2006.

[6] : Fabrice Theoleyre, « Une auto-organisation et ses applications pour les réseaux ad hoc et hybrides », Thèse doctorat, l'Institut National sciences appliquées de Lyon, 2006.

[7]: I. Chlamtac, M. Conti, J.J.-N. Liu, « Mobile ad hoc networking: imperatives and challenges», 2003.

[8]: Mariam Dawoud, « Analyse du protocole AODV », DEA d'Informatique, Université libanaise, Université Paul Sabatier – I.R.I.T, N° d'ordre: 6/2006.

[9]: Julien Cartigny, « Contributions à la diffusion dans les réseaux ad hoc », Thèse doctorat, Université des Sciences et Technologies de Lille LIFL, Numéro d'ordre : 3378, 2003.

[10]: Van der Meerschen Jérôme, «Hybridation entre les modes ad hoc et infrastructure dans les réseaux de type Wi-Fi», Mémoire de fin d'études en vue de l'obtention du grade d'Ingénieur Civil Informaticien en Sciences Appliquées, Université Libre de Bruxelles, 2006.

[11]: Kheldoun Ahmed, « Adaptation de la méthode CTH* aux réseaux mobiles Ad Hoc (MANET) », Mémoire présenté pour obtenir le grade de magister en informatique, Institut National de Formation en Informatique (I.N.I) Oued-Smar Alger, 2008.

[12]: Nadia Mansouri, « Protocole de routage multichemin avec équilibrage de charge dans les réseaux mobiles Ad Hoc », Rapport de Projet de fin d'études, l'Ecole supérieure des télécommunications de Tunis, 2007.

[13]: C.E. Perkins, P. Bhagwat, « Highly dynamic destination-sequenced distance vector (DSDV) for mobile computers», ACM SIGCOMM '94 Computer Communications Review. October 1994.

[14]: Xiaoyun Xue, « mécanismes de sécurité pour des protocoles de routage des réseaux ad hoc », Thèse doctorat, l'Ecole nationale supérieure des télécommunications, paris, 2006.

[15]: Z.J. Haas, M.R. Pearlman, « The performance of query control schemes for the zone routing protocol», In ACM SIGCOMM'98, 1998.

[16]: M. Dahl, C. Thrane, U. Sørensen, M. Clemmensen, « A Distributed Database System in a dynamic peer-to-peer environment », PROJECT TIME SCOPE Date: Sep 2. - Dec. 19. 2005. Department of Computer Science.

[17]: Guerroumi Mohamed, « Accessibilité aux données dans les réseaux mobiles ad hoc », Mémoire présenté pour obtenir le grade de magister en informatique, Université Houari Boumedienne, Algérie, 2005.

[18]: Michaël Hauspie, « Contributions à l'étude des gestionnaires de services distribués dans les réseaux ad hoc », Thèse doctorat, Université des Sciences et Technologies de Lille, Numéro d'Ordre : 3506, 2005.

[19]: C. Mascolo, L. Capra and W. Emmerich, « Principles of Mobile Computing Middleware », University College London. In Q. Mahmoud (ed), Middleware for Communications. John Wiley, 2004.

[20]: http://www.commentcamarche.net/contents/internet/qos-qualite-de-service.php3

[21]: Bouachiba Fouad, « Une Approche à base d'Agent Adaptables pour la QoS dans les Réseaux Mobiles Ad hoc », Mémoire présenté pour obtenir le grade de magister en informatique, Centre universitaire de Tebessa, 2008.

[22]: QoS Forum. QoS, « protocols and architectures », White paper of QoS Forum, July 1999. Http: //www.qosforum.com.

[23]: Mohamed Brahma, « Étude de la QoS dans les Réseaux Ad hoc : Intégration du Concept de l'Ingénierie du Trafic », Thèse doctorat, Université de Haute Alsace UFR des sciences et techniques, N°d'ordre : 06MULH0844, 2006.

[24]: E. Crawley, R. Nair, B. Rajagopalan, H. Sandick, « A Framework for QoS-based Routing in the Internet», IETF RFC2386.1998

[25]: Zoubir Mammeri, « qualité de service dans les réseaux : problématique, solutions et challenges », ACTE D'ETR'05, France, 2005.

[26]: Claude Chaudet, « Qualité de service et réseaux ad hoc – un état de l'art », rapport de recherche, N° : 4325 12 novembre 2001, INRIA, France.

[27]:http://igm.univmlv.fr/~dr/XPOSE2007/mdouis_LaQualiteDeServiceDansLesReseauxAdHoc/qos_models.html.

[28]: G. Ahn, A. Campbell, A. Veras, and L. Sun, « Supporting Service Differentiation for Real-time and Best-effort Traffic in Stateless Wireless Ad hoc Networks », Juillet 2002. In the IEEE Transactions on Mobile Computing.

[29]: Hajer Ferjani, Mouna Ayari, « DRC : mécanisme de clustering pour la gestion par politiques dans les réseaux ad hoc », Ecole Nationale des sciences de l'informatique, Tunisie, 2006.

[30]: Jean-Pierre CHANET, « Algorithme de routage coopératif à qualité de service pour des réseaux ad hoc agri environnementaux », Thèse doctorat, Université Blaise Pascal - Clermont II, N° d'Ordre : 1745, 2007.

[31]: Françoise SAILHAN, « Localisation de ressources dans les réseaux ad hoc », Thèse doctorat, Université paris VI, 2005

[32]: R. Sivakumar, P. Sinha and V. Bharghavan, « CEDAR : a Core-Extraction Distributed Ad hoc Routing algorithm», INFOCOM '99, Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies Proceedings, IEEE , Volume : 1 ,1999.

[33]: http://www.kovaxlabs.com/docs/rapport_QoS_Routing_AdHoc.pdf

[34]: http://enslyon.free.fr/rapports/info/Sebastien_Hinderer_1.pdf

[35] : Claude Chaudet, Isabelle Guérin Lassous, « Routage QoS et réseaux ad hoc : de l'état de lien à l'état de nœud », rapport de recherche, N° : 4700 Janvier 2003, INRIA, France.

[36]: Chouaib BOULKAMH, Mohamed Nadjib OUNES, Khaled LAGGOUN, Maamar SEDRATI, Azeddine BILAMI, « Impact de la Charge de Contrôle de Routage Sur la QdS dans un Réseau Ad hoc », 4th International Conference on Computer Integrated Manufacturing CIP'2007 03-04 November 2007, Département d'informatique, Université de Batna.

[37]: Salah El falou, « programmation répartie, optimisation par agent mobile », Thèse doctorat, Université de CAEN/BASSE-NORMANDIE, U.F.R. sciences, Ecole doctorale SIMEM, 2006.

[38]: Jacque Ferber, « Les systèmes multi-agents: vers une intelligence collective », 1995.

[39]: N. R. Jennings, M. Wooldridge, and K. Sycara, « A roadmap of agent research and development », Int Journal of Autonomous Agents and Multi-Agent Systems», 1998.

[40]: Bettahar Aoued, « Les Aglets d'IBM », Université de Montréal, BETA08036508, Cours IFT6802 – H2003.

[41]: Stéphane PERRET, « Agents mobiles pour l'accès nomade à l'information répartie dans les réseaux de grande envergure », Thèse doctorat, Université Joseph Fourier - Grenoble I ,1997.

[42]: Jean-Paull Arcangeli, Guy Bernard, Abdelkader Hameurlain, Jean-François Monin, « agent et code mobile », Technique et science informatiques, RSTI série TSL. Volume 21-n° 6/2002.

[43]: Guy Bernard et Leila Ismail, « Apport des agents mobile à l'exécution répartie », RSTI série TSI. 2002.

[44]: Christophe CUBAT DIT CROS, « Agents Mobiles Coopérants pour les Environnements Dynamiques », Thèse doctorat, Institut National Polytechnique de Toulouse (ENSEEIHT), 2 décembre 2005.

[45]: http://larim.polymtl.ca/~sg/MA2Fran.htm

[46]: Cubat dit Cros (Christophe), « Agents mobiles coopératifs pour les environnements dynamiques», In : Les Nouvelles Technologies de la Répartition NOTERE'2004,Saidia, Maroc, juin 2004.

[47]: Tra Goore Bi, Ibrahim Lokpo, Gérard Padiou, « Localisation décentralisée et adaptative d'agents mobiles dans les réseaux dynamiques », Institut National Polytechnique Félix Houphouet-Boigny, Département Mathématiques et Informatique Yamoussoukro, Côte d'Ivoire, Institut de Recherche en Informatique de Toulouse, UMR CNRS 5505 ENSEEIHT, BP 7122, 2 rue Charles Camichel, F-31071 Toulouse cedex 7, RENPAR'16 / CFSE'4 / SympAAA'2005 / Journées Composants, Le Croisic, France, 5 au 8 avril 2005.

[48]: Evan Sultanik, Donovan Artz, Gustave Anderson, Moshe Kam, William Regli, Max Peysakhov, Jonathan Sevy, Nadya Belov, Nicholas Morizio, Andrew Mroczkowski, « Secure Mobile Agents on Ad Hoc Wireless Networks », Department of Computer Science, Department of Electrical and Computer Engineering, College of Engineering, Drexel University, 2003.

[49]: Pascal Chour, Frédéric Cuppens, Yves Deswarte, Ludovic Mé, Refik Molva, and Yves Roudier, « Sécurité des réseaux et systèmes répartis », Hermes Science, 2002.

[50]: Romit RoyChoudhuri, S. Bandyopadhyay, Krishna Paul, « Topology discovery in ad hoc wireless networks using mobile agents», Department of Computer Sc and Engg Haldia Institute of Technology Haldia, West Bengal, India, PricewaterhouseCoopers, Saltlake Technology Center Sector V, Calcutta 700 091, India, Cognizant Technology Solutions, Sector V, Saltlake Calcutta 700 091 India.

[51]: Stéphane PERRET, « Agents mobiles pour l'accès nomade à l'information répartie dans les réseaux de grande envergure », Thèse doctorat, Université Joseph Fourier - Grenoble I ,1997.

[52]: S. Bandyopadhyay, Krishna Paul, « Evaluating the Performance of Mobile Agent-Based Message Communication among Mobile Hosts in Large Ad Hoc Wireless Network », PricewaterhouseCoopers Limited Sector V, Saltlake Calcutta 700 091 INDIA, Techna Digital Systems SDF Building, Saltlake Calcutta 700 091 INDIA.

[53]: Shivanajay Matwaha, Chen Khong Tham, Dipti Srinivasan, « Mobile Agents based Routing Protocol for Mobile Ad Hoc Networks », Department of Electrical and Computer Engineering, National University of Singapore 10, Kent Ridge Crescent, Singapore 119260.

[54]: Sébastien Leriche, Jean-Paul Arcangeli, « Une architecture pour les agents mobiles adaptables », Equipe IAM (Ingénierie des Applications Mobiles) Université Paul Sabatier / IRIT – Toulouse, 2004.

[55]: http://www.trl.ibm.co.jp/aglets.

[56]: http://stromboli.it-sudparis.eu/~bernard/ipr/projets97-98/agents-rapport/aglet.htm.

[57]: Kimble Cheron, Professor Steven A. Demurjian, and Mitch Saba, « Software Agents and Aglets », Computer Science & Engineering Department The University of Connecticut 191 Auditorium Road, Box U-155 Storrs, CT 06269-3155.