



Sécurité dans les réseaux Ad-Hoc

Sous la direction de:

Mr Abderrahim BENSLIMANE
Mr Abderrezak RACHEDI

Par:

Mr El khadar Mohammed Amine
Mr BABINET Florent
Mr CHUENTE DJOMOU Aubin

Remerciements

Un grand merci à Mr A. Rachedi et à Mr. Benslimane sans qui l'avancement de ce projet n'aurait pas été possible. Merci pour tous vos conseils avisés qui nous ont permis de mener ce projet à bon terme. Un merci non moins reconnaissant à tout le personnel de l'IUP pour nous avoir toujours prodigué de si bons conseils dans les moments difficiles. Encore un grand merci à tous et toutes.

Table des Matières

| | |
|---|----|
| Remerciements | 2 |
| Résumé | 5 |
| I. Introduction générale | 6 |
| II. Etat de l'art | 8 |
| 1. Les Réseaux ad hoc..... | 8 |
| a. Définition..... | 8 |
| b. Caractéristiques principales des réseaux « Ad Hoc »..... | 8 |
| c. Architecture des Réseaux sans fil..... | 9 |
| d. Routage dans les réseaux « Ad Hoc »..... | 10 |
| 1) Protocole OLSR | 11 |
| 2) Protocole AODV | 11 |
| 3) Comparaison des performances..... | 11 |
| 2. Vulnérabilité et Challenges | 12 |
| 3. Objectifs du projet..... | 13 |
| 4- L'architecture à mettre place..... | 13 |
| a. Modèle de confiance..... | 14 |
| b. Algorithme de Clustering..... | 15 |
| III. Etude pratique | 16 |
| 1. Travail réalisé..... | 16 |
| a. Algorithme mis en place..... | 16 |
| b. Implémentation logicielle | 16 |
| c. Création du Script Tcl de simulation | 17 |
| d. Fichier Awk..... | 18 |
| 2. Evaluation des résultats obtenus..... | 18 |
| 3. Organisation du travail..... | 20 |
| 4. Problèmes rencontrés..... | 20 |
| 5. Bilan des approches et acquis..... | 21 |

| | |
|--|-----------|
| IV. Conclusion..... | 23 |
| V. Bibliographie..... | 24 |
| Annexe 1 : Algorithme mis en place..... | 25 |
| Annexe 2 : Script TCL..... | 30 |
| Annexe 3 : Script AWK..... | 34 |

Résumé

Les réseaux mobiles ad hoc sont constitués de plusieurs noeuds mobiles qui communiquent entre eux par des liaisons sans fil et en l'absence d'une infrastructure fixe et centralisée. Ces caractéristiques font de ces types de réseaux assez originaux mais aussi vulnérables. Ainsi, la sécurité de ceux-ci a suscité la réaction de nombreuses recherches ces dernières années. On propose dans ce présent rapport, une nouvelle architecture basée sur un modèle de confiance et un algorithme de clustering. Cette architecture permettra de diviser un réseau par groupe (cluster) ; dans chacun de ces derniers, un CA sera élu comme chef de cluster et chaque nœud aura un rôle spécifique afin d'amener tous les noeuds à coopérer. Le CA de chaque cluster sera protégé par un ensemble de nœuds (minimum 1) appelé DDMZ (Dynamic Demilitarized Zone). L'idée principale de la DDMZ est de sélectionner des noeuds appelés RA ; ces noeuds ont un degré de confiance élevé (égaux au CA), et qui sont à un saut du CA. Ces noeuds ont pour rôle de filtrer les données envoyées au CA. Certains RA auront aussi le rôle de GW, c'est à dire d'échanger les informations entre clusters et dans ce cas il appartiendra aux deux clusters. Les autres nœuds avec un degré de confiance élevé et qui sont à plus d'un saut du CA sont appelés MN. Les autres nœuds ayant un degré de confiance moins élevé et situés à deux sauts ou plus du CA seront des VN.

I. Introduction générale

Un réseau ad hoc mobile (MANET) est un système autonome de nœuds mobiles reliés par des liens sans fils dont l'union forme un graphe arbitraire. Les nœuds du réseau jouent le rôle de routeurs et sont libres de se déplacer aléatoirement et de s'organiser arbitrairement. En conséquence, la topologie du réseau peut changer rapidement et de manière imprévisible. Un tel réseau ne nécessite pas d'infrastructure fixe et représente une option attractive pour connecter spontanément des terminaux mobiles. Les champs d'application sont divers : déploiement d'un réseau durant une opération militaire sur un champ de bataille ou durant une opération de sauvetage dans un lieu difficilement accessible... etc. Quelle que soit l'application visée, un réseau MANET possède des exigences spécifiques en terme de sécurité, du fait de ses particularités : liens sans fils, contraintes d'énergie, limitation éventuelle de la bande passante et de la puissance de calcul, non connectivité permanente d'un nœud avec (tous) les autres nœud.... Jusqu'à présent, les nombreux travaux traitant de la sécurité des MANETs s'articulent principalement selon les trois problématiques suivantes : la mise au point de mécanismes d'authentification et de gestion de clefs adaptés et la sécurisation des protocoles de routage ad hoc, la définition de mécanismes renforçant la collaboration, et la définition des mécanismes et protocoles de sécurité adaptés aux réseaux ad hoc mobiles.

Problématique : cependant, différents points restent non abordés, ou insuffisamment traités dans les travaux existants. Le sujet de ce projet consiste à définir une architecture hiérarchique distribuée pour sécuriser les réseaux ad hoc mobiles et prenant en compte les points suivants :

- Le routage dans les réseaux ad hoc
- La sécurité des données (intégrité et authentification)
- La mobilité des nœuds.
- Le contrôle d'accès à un réseau ad hoc
- La délégation du pouvoir de contrôle d'accès.

Durant le premier semestre, nous avons réalisé une étude approfondie du niveau de sécurité dans les réseaux sans fils, en particulier dans les réseaux ad hoc. On a fait une étude des différentes architectures des réseaux ad hoc déjà mises en place. Nous avons aussi étudié l'état actuel des solutions de sécurisation proposées par les différentes équipes de recherche travaillant sur ce sujet à travers le monde.

Nous avons commencé à nous familiariser avec les différents outils dont on aura besoin pour le projet ; à savoir Network Simulator (simulation d'exemple de protocoles tels que TCP, UDP) et le langage C++.

II. Etat de l'art

1. Les réseaux ad hoc

a. Définitions

Un réseau sans fil « ad-hoc » est un réseau radioélectrique sans aucune infrastructure pré existante. Ils sont capables à se créer et s'organiser dynamiquement lorsque plusieurs équipements se trouvent à portée radio les uns des autres. Il se forme au gré de l'apparition et du mouvement des noeuds.

C'est un réseau capable de rendre transparentes, aux utilisateurs mobiles, les modifications de topologie qu'il subit. C'est donc un système autonome de noeuds mobiles. Ce dernier peut fonctionner d'une manière isolée ou s'interfacier à des réseaux fixes au travers de passerelles.

A partir de cette définition générale, il est intéressant de mettre en avant les caractéristiques principales qui différencient un réseau ad hoc d'un réseau classique.

b. Caractéristiques principales des réseaux « ad hoc »

- **Mobilité** : La mobilité des noeuds constitue à l'évidence une caractéristique très spécifique des réseaux ad hoc. Cette mobilité est intrinsèque au fonctionnement du réseau. Dans un réseau ad hoc, la topologie du réseau peut changer rapidement, de façon aléatoire et non prédictible.

- **Equivalence des noeuds du réseau** : Dans un réseau classique, il existe une distinction nette entre les noeuds terminaux (stations, hôtes) qui supportent les applications et les noeuds internes (routeurs par exemple) du réseau, en charge de l'acheminement des données. Cette différence n'existe pas dans les réseaux ad hoc car tous les noeuds peuvent être amenés à assurer des fonctions de routage.

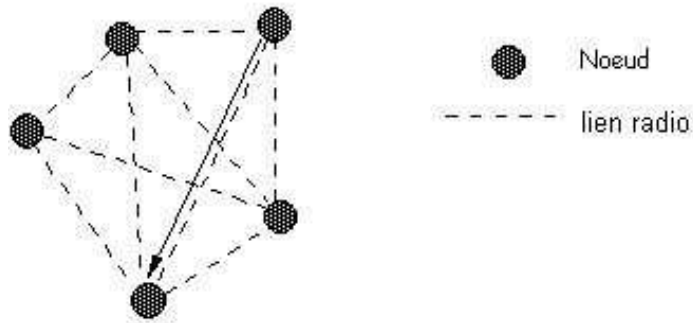
- Liaisons sans fil
- Autonomie des noeuds
- **Vulnérabilité** : Les réseaux sans fil sont par nature plus sensibles aux problèmes de sécurité. Pour les réseaux ad hoc, le principal problème ne se situe pas tant au niveau du support physique mais principalement dans le fait que tous les noeuds sont équivalents et potentiellement nécessaires au fonctionnement du réseau.

c. Architecture des Réseaux sans fil

On distingue plusieurs architectures de réseaux sans fil mais dans notre cas nous allons nous attarder sur les deux architectures propres aux réseaux ad hoc : Le réseau complet et réseaux à routage interne.

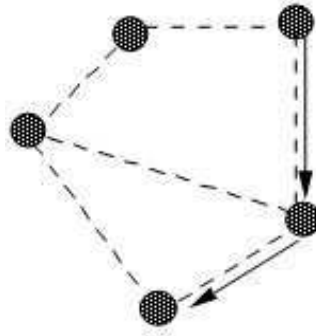
Réseau complet :

C'est un maillage intégral (full mesh), chaque noeud est directement relié à tous les autres.



Réseau à routage interne :

C'est une façon de relier des noeuds dans une topologie ne pouvant se réduire à un cas plus simple.



d. Routage dans les réseaux « Ad Hoc »

Définition du routage :

Le routage est une méthode d'acheminement des informations à la bonne destination à travers un réseau de connexion donné. Le problème de routage consiste à déterminer un acheminement optimal des paquets à travers le réseau au sens d'un certain critère de performance. Le problème consiste à trouver l'investissement de moindre coût en capacités nominales et de réserves qui assure le routage du trafic nominal, et garantit sa survivabilité en cas de n'importe quelle panne d'arc ou de nœud. Le problème qui se pose dans le contexte des réseaux ad hoc est l'adaptation de la méthode d'acheminement utilisée avec le grand nombre d'unités existant dans un environnement caractérisé par de modestes capacités de calcul et de sauvegarde et de changements rapides de topologies.

Classification :

Suivant la manière de création et de maintenance de routes lors de l'acheminement des données, les protocoles de routage peuvent être séparés en deux catégories, les protocoles proactifs et les protocoles réactifs. Les protocoles proactifs établissent les routes à l'avance en se basant sur l'échange

périodique des tables de routage, alors que les protocoles réactifs cherchent les routes à la demande.

1) Le protocole OLSR (Optimized Link State Routing Protocol).

OLSR est un protocole de routage destiné aux réseaux Ad Hoc mobiles. C'est un protocole proactif c'est à dire qu'il échange régulièrement des informations sur la topologie du réseau afin d'en déterminer les tables de routage. (Alors que AODV crée les tables de routage sur demande).

L'originalité de OLSR repose sur le concept de relais multipoints (MPR Multipoint Relay) qui permet de contrôler les liens entre les mobiles par des paquets spéciaux, les 'Hello'. Il autorise également les réseaux denses en économisant une grande partie de la bande passante du réseau. OLSR présente enfin l'avantage de s'adapter parfaitement aux protocoles de l'Internet et de donner à chaque mobile la topologie du réseau à tout instant.

2) Le protocole AODV (Ad hoc On-Demand Distance Vector).

L'algorithme sur demande de cheminement du vecteur de distance (AODV) est un autre protocole de cheminement conçu pour les réseaux mobiles ad hoc. AODV est capable de l'Unicast et du cheminement de Multicast. C'est un algorithme qui construit les itinéraires entre les noeuds seulement sur demande par des noeuds de source. Il maintient ces itinéraires aussi longtemps qu'ils sont nécessaires par les sources.

3) Comparaison des performances.

OLSR et AODV bien que de nature très différentes, sont très similaires en termes de performances. Dans un réseau très mobile, avec de fréquent changement de topologie, AODV a un petit avantage sur OLSR car les routes sont mises à jours plus rapidement. OLSR doit attendre plusieurs paquets Hello perdus avant de modifier l'état du lien et envoyer des informations de mises à jours. Par contre, dans un réseau plus statique, OLSR encombre moins le réseau

qu'AODV qui émet beaucoup plus de messages à chaque découverte de route. En effet dans ce cas OLSR n'émet presque pas de message de mises à jour de la topologie. Dans un réseau très dense, OLSR charge moins le réseau que AODV. Dans des réseaux moyens, OLSR et AODV sont équivalents. Lors de communications courtes, OLSR a un énorme avantage sur AODV car les routes sont disponibles immédiatement.

Dans la plupart des cas, les messages de contrôles de AODV sont légèrement plus nombreux que ceux de OLSR. AODV émet d'autant plus de paquets que le réseau est grand. OLSR est un peu supérieur à AODV car s'il est équivalent dans la plupart des réseaux, il est meilleur dans certains cas particuliers : comme des réseaux denses, ou des réseaux où le trafic est important et composé de nombreuses et courtes connexions.

2. Vulnérabilité et Challenges

Comme nous l'avons souligné plus haut, dans les caractéristiques des réseaux Ad hoc, ces réseaux sont par nature plus sensibles aux problèmes de sécurité. Le principal problème se situe dans le fait que tous les noeuds sont équivalents (hétérogénéité des noeuds) et potentiellement nécessaires au fonctionnement du réseau. Les possibilités de s'insérer dans le réseau sont plus grandes (réseau ouvert) surtout que certains noeuds sont malveillants et non coopérants, et l'absence de centralisation pose un problème de remontée de l'information et la détection d'intrusions ou d'un déni de service est plus délicate.

Cela ne va pas aussi sans d'autres problèmes, notamment ceux concernant la disponibilité des services ou encore la surcharge du réseau. Pour que le mécanisme de retransmission des paquets puisse fonctionner, il faut que les noeuds restent en marche, mais cela pose des problèmes aux utilisateurs qui préfèrent les laisser éteints pour économiser leurs batteries. Il faut inciter les utilisateurs à maintenir en marche leurs terminaux et à relayer les paquets, donc

trouver un moyen de les faire coopérer en leur épargnant une perte d'énergie trop importante. Dans le cas contraire, on aurait une indisponibilité des services.

L'autre problème majeur est en rapport avec la surcharge du réseau. Un réseau peut être surchargé et ne plus pouvoir transmettre d'informations utiles, soit à cause d'une action malveillante, soit par un envoi trop important d'informations de la part des nœuds. Il nous faut un mécanisme qui décourage l'abus d'utilisation des services et limite un engorgement du réseau dû à un trafic sans intérêt.

Le fait que les nœuds soient fortement mobiles, cela peut engendrer des modifications fréquentes de topologie. Ces modifications peuvent causer des pertes de paquets entre les nœuds. Ainsi, la stabilité du réseau pourra être mise en péril.

3. Objectifs du projet

- Mise en place d'un nouveau protocole de sécurité basé sur le système de clustering pour les réseaux ad hoc qui adapte le principe de la division du réseau sous forme de groupes (clusters), avec un seul chef par groupe.
- Ce protocole devra assurer la diminution de la charge du réseau, assurer une bonne stabilité du réseau, mieux gérer la mobilité des nœuds, et surtout la disponibilité du réseau.
- Optimiser l'algorithme de cette mise en place afin de permettre de maintenir le plus longtemps possible cette architecture.
- Etudier les failles et les problèmes de cette nouvelle architecture.

4. L'architecture mise en place

Pour la mise en place de ce nouveau protocole de sécurité, nous avons établi une nouvelle architecture hiérarchique basée sur la division du réseau sous

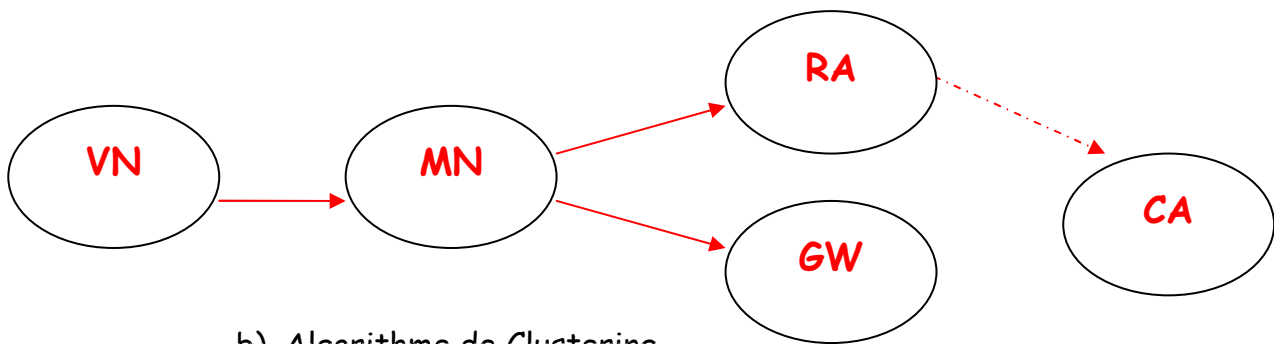
forme de groupes (clusters), avec un seul chef (CA) groupe. Cette architecture est basée sur un modèle efficace de confiance et un algorithme de Clustering.

a) Modèle de confiance

Ce modèle de confiance, basé sur PGP (Pretty Good Privacy), est composé de plusieurs nœuds, chacun ayant un niveau de confiance ($T_m \in [0,1]$) :

- **CA_k** (Certification Authority) : Il correspond au CA du Cluster k. Il gère la clé publique des nœuds appartenant au même Cluster k. Il possède le niveau de confiance le plus élevé ($T_m=1$).
- **RA_{i,k}** (Registration Authority) : Il correspond au RA du cluster k assuré par le nœud i. Son principal rôle est de protéger le CA contre les attaques des nœuds étrangers selon une zone de protection DDMZ. Son niveau de confiance est aussi de 1. ($T_m=1$).
- **GW_{i,j}** (Gateway Node) : il assure la connexion entre deux clusters i et j. Ces deux clusters doivent être certifiés par deux CA différents. Ce nœud doit aussi avoir un bon niveau de confiance. ($T_m \in [0,7 - 1,0]$). Il peut aussi jouer le rôle de RA.
- **MN_{i,k}** (Member Node) : Il correspond à un nœud membre i du cluster k. Étant un nœud visiteur, son statut change (passant de Nœud visiteur à nœud membre), grâce à son bon comportement dans le réseau. Il peut être recommandé par le CA du cluster k à d'autres CAs. Son niveau de confiance est de $T_m \in [0,5 - 0,7]$.
- **VN_{i, k}** (Visitor Node) : Il correspond au nœud visiteur i dans un cluster k. Il a une certification faible, car le CA et les RAs ont besoin de plus d'informations sur son comportement. Par ailleurs, il ne peut pas communiquer à l'extérieur du cluster k. Il a le niveau de confiance le moins élevé $T_m \in [0,1- 0,5]$.

Le schéma ci-dessous nous présente comment s'effectue la transition des différents nœuds dans un cluster :



b) Algorithme de Clustering

L'algorithme de Clustering distribué utilise un niveau de confiance et de mobilité relative pour la sélection de la tête du Cluster (Cluster Head). Cette dernière deviendra CA du Cluster. Considérons deux noeuds A et B en compétition pour l'élection de CA dans un Cluster. Le cluster A envoie un paquet hello à B et celui lui renvoie un message lui certifiant qu'il est bien présent. Le Noeud A lui renvoie deux paquets successifs. Une fois que le Noeud B assigne les deux paquets reçus par A, ce dernier calcule la relative la Relative Mobilité(RM) de B par rapport à A. Celle ci dépend des puissances de réception des paquets par le noeud B et de la variance de A. La même opération est effectuée pour tous les noeuds existants lors d'une compétition il y a plus de deux noeuds pour l'élection d'un CA.

Cet algorithme sera implémenté selon plusieurs critères :

- Seuls les noeuds de confiance $T_m=1$ peuvent être candidat pour devenir CA
- Pour chaque cluster, il existe un seul CA
- Tous les noeuds voisins de confiance du CA peuvent devenir RA dans le Cluster
- Les autres noeuds sont à une distance maximum d-hop du CA.

III. Etude pratique

1. Travail réalisé

a. Algorithme mis en place

Après réflexion avec le tuteur, on a décidé de mettre en place cet algorithme afin d'implémenter cette architecture pour la formation des clusters. Dans cet algorithme, après la phase d'initialisation, précisée plus haut, dans laquelle chaque nœud devrait envoyer des messages « Hello » afin de repérer ses voisins et de calculer sa relative mobilité par rapport aux autres noeuds, on passera à une phase de compétition et d'élection. Durant ces différentes phases, l'implémentation doit être faite de telle manière que durant l'élection d'un CA, une fois après avoir choisi les noeuds qu'il estime comme potentiels CAs pendant la phase d'élection, chaque noeud envoie des messages « JOIN » pour signaler à ces derniers son choix. Il classe ces noeuds selon l'ordre de performance dans un tableau.

Une fois après avoir envoyé ces messages « JOIN », il attend une réponse venant de son destinataire. Si la réponse est « ACCEPT », il le choisit comme candidat pour devenir CA. Si la réponse est « REJECT », le destinataire lui dit qu'il a déjà choisi un autre noeud comme candidat et précise sa position par rapport à lui. Si le noeud choisi par l'autre ne lui correspond pas, il choisit le deuxième noeud dans le tableau. Si le nœud reçoit un message join il attend de définir son statut avant de lui répondre et si il se propose comme CA il attend de recevoir un join pour mettre son statut en CA_NODE. Dans l'annexe (1) nous présentons l'algorithme qui nous a permis d'implémenter ces différentes étapes.

b. L'implémentation logicielle

Afin d'expliquer brièvement ce que peut réaliser NS, nous pouvons dire qu'il s'agit d'un simulateur à événements discrets qui permet d'exécuter tous types de scénarios sur des topologies définies par l'utilisateur. Il permet la

description et la simulation de réseaux IP. Le réseau est représenté (modélisé) par ses sources de trafic (applications), ses protocoles (UDP, TCP), ses routeurs (avec leurs files d'attente) et les liens qui les relient. Le réseau est ensuite simulé, ce qui produit des traces et des statistiques. Des outils périphériques permettent l'animation du réseau (NAM : Network Animator) ou la conversion vers d'autres outils (comme par exemple xgraph, gnuplot, tracegraph pour dessiner des courbes).

c. Script TCL de simulation :

Nous avons créé un script final en Tcl qui nous permet de simuler au mieux l'architecture mise en place. Voir annexe (2)

Le fichier Tcl contient des options et configurations que nous pouvons modifier par rapport à ce que nous voulons faire et par rapport à toute optimisation possible.

Les scénarios de simulation sont générés avec les paramètres listés dans le tableau ci après.

| Paramètres | Valeur dans notre simulation |
|---------------------------|-------------------------------|
| Nombre de noeuds | 50 |
| Taille du réseau | 1000x1000m² |
| La portée de Transmission | 250 (m) |
| Puissance de Transmission | 0.281838 (watt) |
| Temps de Simulation | 80(s) |
| Vitesse de Mobilité | 7-15(m/s) |

Le mouvement de noeuds mobiles est généré aléatoirement. Nous avons choisi cette taille de réseau car pour 50 nœuds cela nous permet d'avoir une concentration moyenne de nœuds satisfaisants. Pour la vitesse nous avons choisi une vitesse entre 7 et 15 car cela correspond à la vitesse moyenne des utilisateurs.

d. Fichier AWK

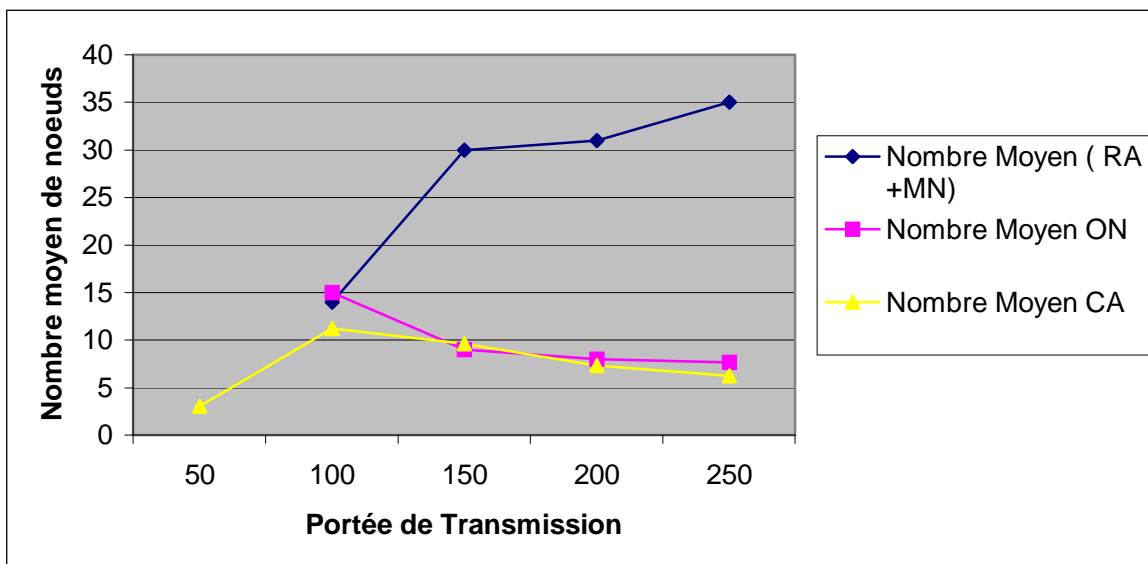
Le fichier Awk nous permet de faire un calcul afin de définir le nombre de paquets envoyé entre les différents nœuds. Ce nombre de paquets nous a permis d'évaluer le trafic circulant dans le réseau. Voir annexe (3)

2. Evaluation des résultats obtenus

Pour évaluer les performances de cette algorithmme il faut se baser sur plusieurs critères: disponibilité du réseau, mobilité du réseau, charge du réseau et stabilité du réseau

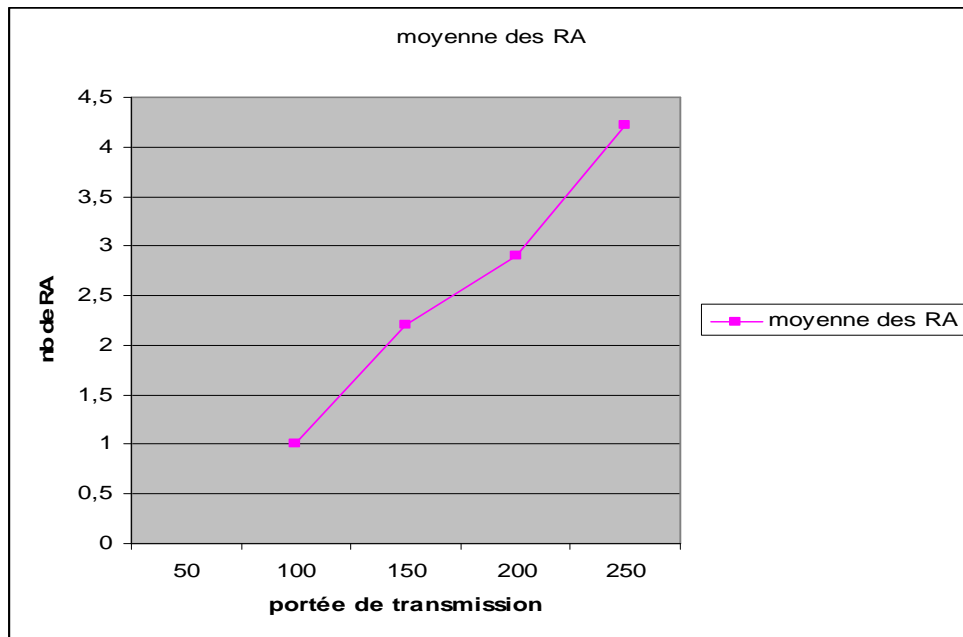
Les résultats présentés sont obtenus à l'aide de 10 simulations dans le but d'obtenir un meilleur résultat.

- Disponibilité du réseau



Sur cette simulation, nous pouvons voir les moyennes des différents statuts des nœuds du réseau.

Nous pouvons remarquer que le nombre de nœuds isolés a tendance à diminuer quand on augmente la portée de transmission. De même que les CA, les nœuds RA et MN ont eux tendance à augmenter.

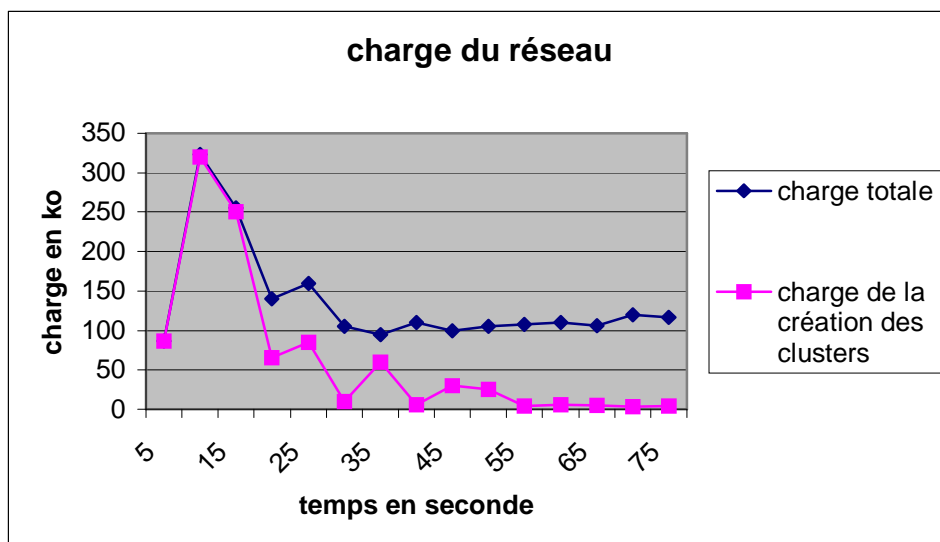


Cette simulation nous montre la robustesse de la DDMZ. Le nombre de RA par cluster augmente quand on augmente la portée de transmission et donc la résistance du cluster aux attaques devient plus importante.

- Mobilité du réseau

Après plusieurs tests, nous pouvons dire que des nœuds ayant une vitesse élevée ont tendance à partir du cluster rapidement et donc que le nombre de déclenchement d'élection se succède plus rapidement et cela a aussi un impact sur la charge du réseau. Nous pouvons aussi dire que ce système ne serait pas tout à fait fiable sur des nœuds à haute vitesse.

- Charge du réseau



Nous remarquons que la création des clusters surcharge le réseau pendant 20 secondes à peu près avec une montée à 325ko/s et que après cela elle n'influence pas le réseau. Les autres charges du réseau sont les paquets de reconnaissance CA-RA; elle représente 100 ko/s. La quantité de données envoyées avec les hello correspondent à 86 ko.

Donc d'après notre simulation, nous pouvons en déduire que l'algorithme de clustering est efficace pour sécuriser les réseaux avec des grandes portées de transmission et une mobilité moyenne ou faible mais qu'il commence à donner des signes de faiblesse avec une mobilité rapide ou une trop faible portée de transmission.

3. Organisation du travail

Afin de mieux mener à bout ce projet, nous avons essayé de s'organiser de manière à ce que tout le monde puisse y mettre sa main à la patte. Ainsi, il y a un étudiant qui se sentait plus à l'aise au niveau de la programmation C++ qui a implémenté une grande partie de l'application suivant aussi les idées et propositions que les autres ont pu lui apporter. Les deux autres se sont chargés de programmer les fichiers tcl et awk qui ont permis de faire les tests et simulations sur Network Simulator. Ces derniers s'associaient aussi afin de faire les différents comptes rendus qui permettaient le suivi du projet. Nous pensons que le fait que tous les membres du groupe n'avaient pas cette facilité au niveau programmation en C++ a été un petit handicap pour le cheminement de ce projet.

4. Problèmes rencontrés

Au cours de la réalisation de notre projet nous avons dû faire face à de nombreux problèmes tant au niveau technique, bien entendu, qu'au niveau organisationnel et « logistique ».

Parmi ceux-ci, figuraient des problèmes qui sont survenus à cause de la méconnaissance, a priori, de certains outils ou fonctionnalités. Certaines

hypothèses n'ont pu être validées ou rejetées qu'après avoir été testées. Il existe donc une différence notable entre la théorie et la pratique.

Plus concrètement, s'agissant de NS, outre la familiarisation avec ce logiciel, d'autres difficultés sont apparues. Il a fallu comprendre la manière dont a été conçu ce logiciel et saisir les interactions entre les différents fichiers et modules qu'il regroupe. La recompilation de NS n'étant pas possible à l'IUP (pour des raisons de droits), nous avons dû l'installer chez nous. Des complications émanant de bibliothèques manquantes nous avons contraint à effectuer cette manœuvre sous Linux (Ubuntu). En conséquence, nous avons dû installer par la même occasion ce système d'exploitation sur nos postes. Plusieurs « bugs » ont surgi durant notre projet, nous obligeant à réinstaller plusieurs fois NS et nous pénalisant dans notre progression.

Les autres problèmes que nous avons rencontrés sont du fait que nous avons quelques lacunes au niveau de programmation. Nous avons eu du mal à vraiment maîtriser les principes de la programmation en langage C++.

De plus lors de résolution de problèmes nous nous sommes trop focalisés sur un point et nous n'avons pas pris assez de recul et donc nous avons perdu du temps.

Plus généralement, nous pouvons dire que l'aspect recherche documentaire est un point très important durant le projet à ne pas négliger. Chercher, trier, classer, organiser les informations pertinentes, tout en ne s'égarant pas dans la masse colossale d'informations présentes, pour finalement trouver des informations viables n'est pas chose aisée.

5. Bilan des approches et acquis

➤ Les perspectives

Dans cette partie nous allons traiter des perspectives que notre projet offre. Premièrement, il permet d'exploiter des résultats de simulations de communications au sein de réseaux mobiles ad hoc. Pour ce faire, plusieurs paramètres peuvent être modifiés (nombre total de noeud, puissance d'antenne,

énergie embarquée...). On s'est rendu compte que plus la portée de transmission est grande plus on a de RA et moins on a de CA et de ON ; ainsi si on a un nombre réduit de ON, on augmente la sécurité de communication dans le réseau.

Tout ceci permet d'effectuer des études concernant l'influence de ces paramètres sur la sécurité stabilité, mobilité, charge et disponibilité du réseau.

➤ Optimisation possible

- Ajustement des timers au plus proche pour réduire le temps d'exécution
- Réglage des problèmes des pertes de paquets
- Mise en place des GW(Gateway Node)
- Actualiser les noeuds en fonction des paquets beacon CA et RA

➤ Les différents apports

Ce projet nous a été grandement profitable à bien des égards. En effet, sa réalisation nous a permis d'acquérir des compétences techniques grâce à la maîtrise du logiciel NS et de Linux. Nous avons, par son biais, approfondi le savoir technique que nous avons emmagasiné durant notre scolarité et commencé à le convertir en savoir-faire. Bien entendu nous possédions déjà un certain nombre de savoir-faire dans divers domaines mais cette expérience en a enrichi l'éventail. Pour rester sur l'aspect technique nous pourrions dire que nous avons accru notre connaissance sur les protocoles de routage, plus particulièrement ceux utilisés dans les réseaux mobiles ad hoc.

VI - Conclusion

Les réseaux informatiques basés sur la communication sans fil peuvent être classés en deux catégories : les réseaux avec infrastructure fixe préexistante, et les réseaux sans infrastructure. Dans la première catégorie, le modèle de la communication utilisé est généralement le modèle de la communication cellulaire. Dans ce modèle les unités mobiles sont couvertes par un ensemble de stations de base reliées par un réseau filaire, et qui assurent la connectivité du système. La deuxième catégorie essaie d'étendre les notions de la mobilité à toutes les composantes de l'environnement, toutes les unités du réseau se déplacent librement et aucune administration centralisée n'est disponible. Les réseaux de cette catégorie sont appelés : les réseaux ad hoc.

Etudier la sécurité dans les réseaux ad hoc, sujet d'actualité, était pour nous une occasion de découvrir et d'approfondir nos connaissances à tous les niveaux et bien sur dans le monde sans fil.

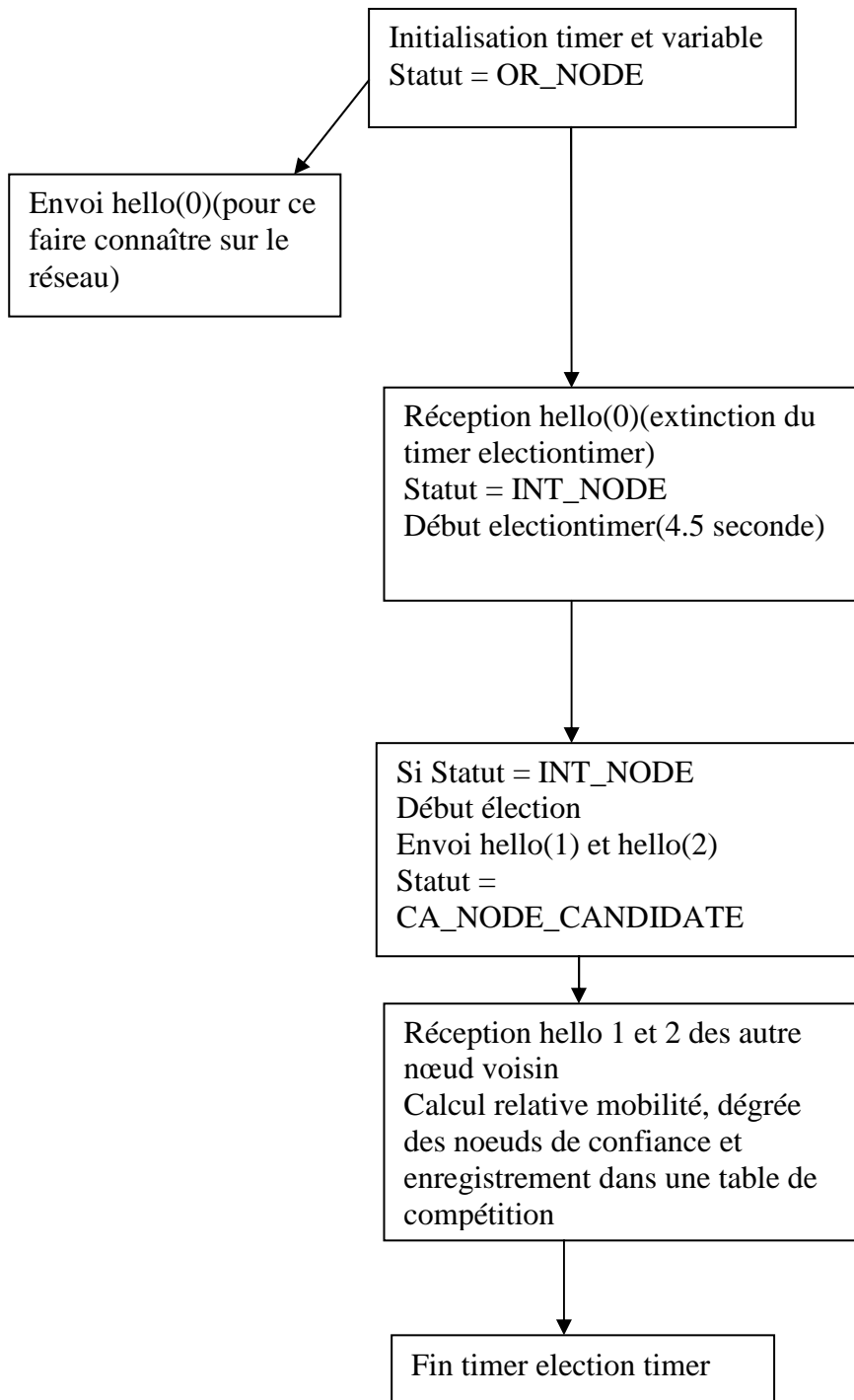
L'étude que nous avons faite nous a permis de conclure que la résolution des problèmes de sécurité, mobilité et routage induits par le nouvel environnement des réseaux mobiles ad hoc, requiert des compétences en sus des techniques réseaux classiques, comme en algorithmique, méthodologie de l'évaluation d'algorithmes de télécommunications et en modélisation de trafics et d'architectures de réseaux, théorie analytique de l'information, cryptographie, etc. des réseaux ad hoc et offrir une meilleure adaptation à la mobilité de ces environnements.

Bibliographie

- ❖ Abderrezak Rachedi and Abderrahim Benslimane ,A secure Architecture for mobile Ad hoc Networks , LIA/CERI University of Avignon , Agroparc
- ❖ Quest for Security in Mobile Ad Hoc Networks, Proceedings of ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC), Long Beach, CA, 2001
- ❖ Sergio Marti, T.J Giuli, Kevin Lai and Mary Baker, Mitigating Routing Misbehaviour in Mobile Ad Hoc Networks
- ❖ Adrian Perrig, Ran Canetti, J.D. Tygar and Dawn Song, Efficient Authentication and Signing Multicasts Streams over Lossy Channels, IEEE Symposium on Security and Privacy, September 2005
- ❖ Franck Stajano, Security for Ubiquitous Computing, John Wiley and Sons, 2002, ISBN 0-470-84493-0, <http://www-lce.eng.cam.ac.uk/fms27/secubicomp/>
- ❖ P. Basu and N. Khan and T. Little. A mobility based metric for clustering in MANET. In Proceedings of Distributed Computing Systems Workshop, :43-51, 2001.
- ❖ M. Gerla and J. T.-C. Tsai. Multicluster, Mobile Multimedia Radio Networks. Wireless Networks. (1995) 255-256
- ❖ S. Yi and R. Kravets. Quality of Authentication in Ad Hoc Networks. ACM, MobiCom2004. (2004)
- ❖ S. Capkun and J. P. Hubaux and L. Buttyan. Mobility Helps Peer-to-Peer Security. IEEE Transactions on Mobile Computing. 5 (2006) 48-60
- ❖ Eitan Altman and Tania Jimenez. Ns For beginners. 2003

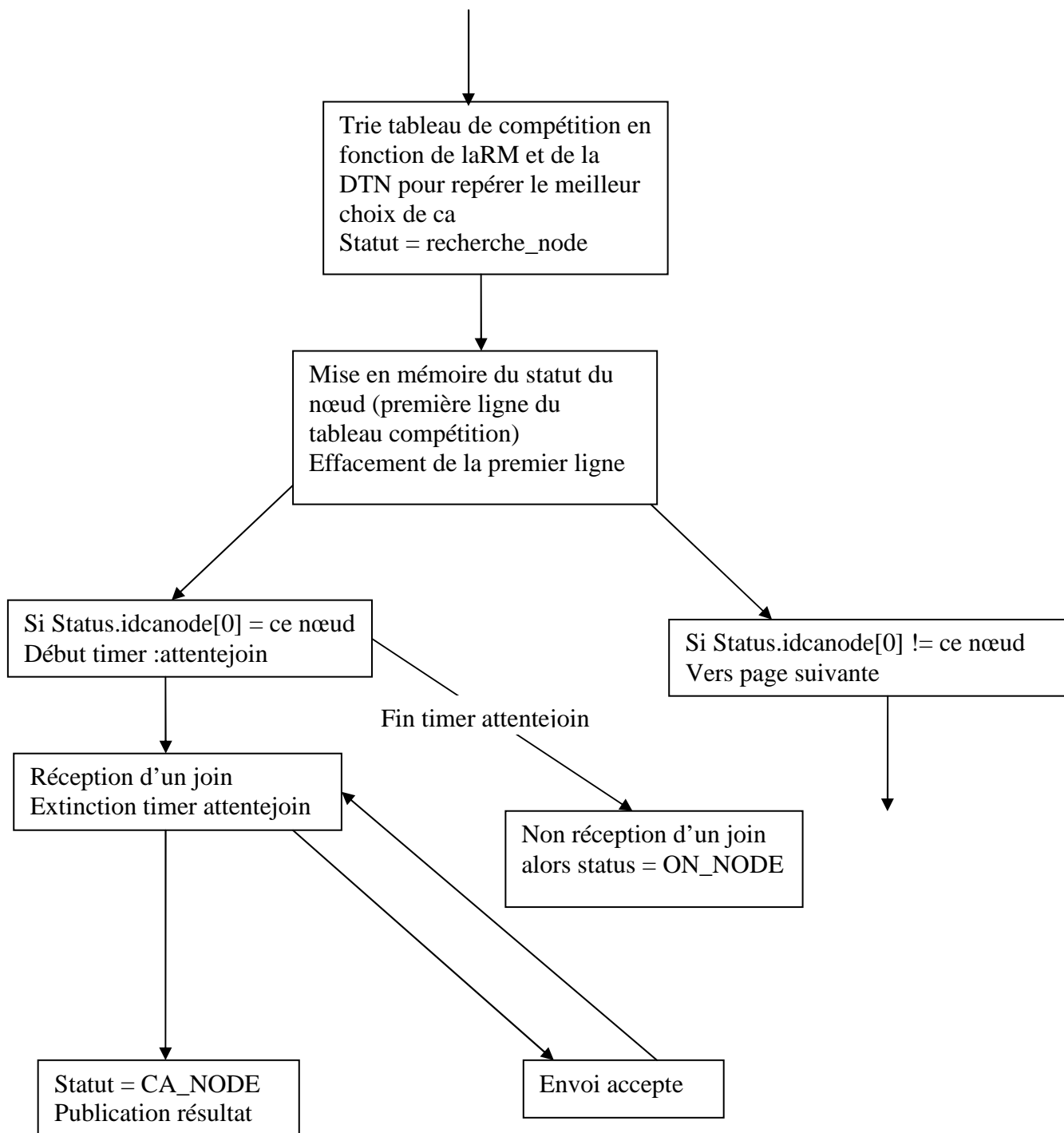
Annexe 1

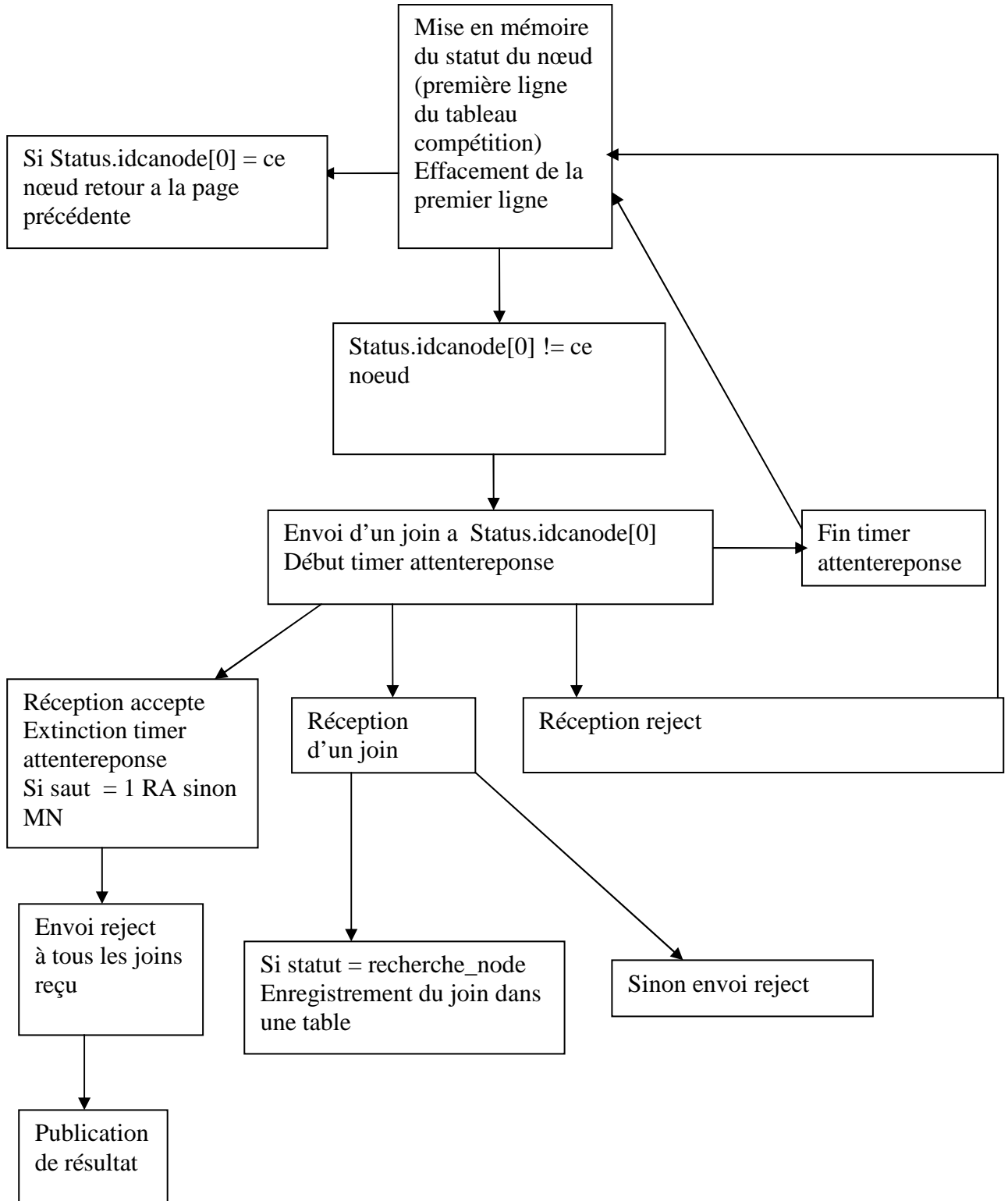
Phase initialisation



Fin de la phase initialisation

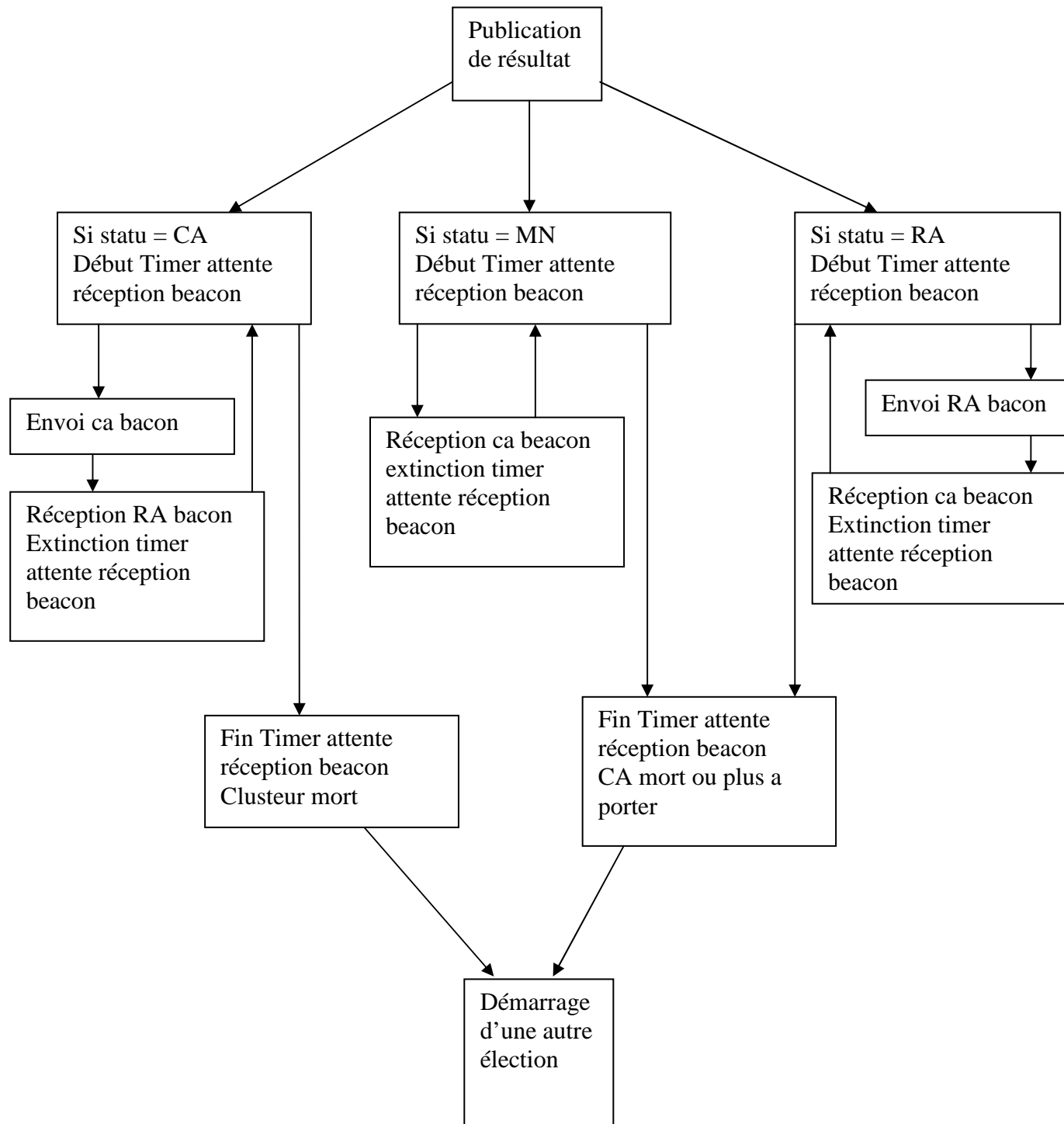
Début de la phase de compétition





Fin de la phase de compétition

Durée de vie d'un cluster



Annexe 2

Fichier TCL

```
set val(chan) Channel/WirelessChannel ;#Channel Type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model
set val(netif) Phy/WirelessPhy ;# network interface type
set val(mac) Mac/802_11 ;# MAC type
set val(ifq) Queue/DropTail/PriQueue ;# interface queue type
set val(ll) LL ;# link layer type
set val(ant) Antenna/OmniAntenna ;# antenna model
set val(ifqlen) 50 ;# max packet in ifq
#set val(sc)
#"/home/rachedi/NS2/ns-allinone-2.29/ns-2.29/indep-utils/cmu-scen-gen/setdest/scen-4-tes"
#set val(sc) "/zac/scen5-2500-300"
#set val(sc) "scen1-50"
set val(nn) 50 ;# number of mobilenodes
set val(rp) DSDV ;# routing protocol
#set val(rp) DSR ;# routing protocol
#set val(rp) AODV ;# routing protocol
set val(x) 5000
set val(y) 500

#-----
# unity gain, omni-directional antennas
# set up the antennas to be centered in the node and 1.5 meters above it
Antenna/OmniAntenna set X_ 0
Antenna/OmniAntenna set Y_ 0
Antenna/OmniAntenna set Z_ 1.5
Antenna/OmniAntenna set Gt_ 1.0
Antenna/OmniAntenna set Gr_ 1.0

# Initialize the SharedMedia interface with parameters to make
# it work like the 914MHz Lucent WaveLAN DSSS radio interface
Phy/WirelessPhy set CPTthresh_ 10.0
Phy/WirelessPhy set CSTthresh_ 1.559e-11
Phy/WirelessPhy set RXTthresh_ 3.65262e-10
Phy/WirelessPhy set Rb_ 2*1e6
Phy/WirelessPhy set Pt_ 0.28183815

#
Phy/WirelessPhy set freq_ 914e+6
Phy/WirelessPhy set L_ 1.0
#-----
proc create-god { nodes } {
    global ns_ god_ tracefd
    set god_ [new God]
    $god_ num_nodes $nodes
}

# Initialize Global Variables
```

```

set ns_ [new Simulator]
set tracefd [open ca.tr w]
$ns_ trace-all $tracefd

set namtrace [open ca.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)

# set up topography object
set topo [new Topography]

$topo load_flatgrid $val(x) $val(y)

# Create God
create-god $val(nn)

# New API to config node:
# 1. Create channel (or multiple-channels);
# 2. Specify channel in node-config (instead of channelType);
# 3. Create nodes for simulations.

# Create channel #1 and #2
set chan_1_ [new $val(chan)]
set chan_2_ [new $val(chan)]

# Create node(0) "attached" to channel #1

# configure node, please note the change below.
$ns_ node-config -adhocRouting $val(rp) \
    -llType $val(ll) \
    -macType $val(mac) \
    -ifqType $val(ifq) \
    -ifqLen $val(ifqlen) \
    -antType $val(ant) \
    -propType $val(prop) \
    -phyType $val(netif) \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace ON \
    -movementTrace OFF \
    -channel $chan_1_

#set node_(0) [$ns_ node]
for {set i 0} {$i < $val(nn)} {incr i} {
    set node_($i) [$ns_ node]
    $node_($i) set X_ [expr int(rand()*499.0*rand()+1.0)]
    $node_($i) set Y_ [expr int(rand()*499.0*rand()+1.0)]
    $node_($i) set Z_ 0.0
    $node_($i) random-motion 1
    $node_($i) setdest [expr int(rand()*499.0*rand()+1.0)] [expr int(rand()*499.0*rand()+1.0)]
    [expr int(rand()*799.0*rand()+1.0)] [expr int(rand()*15.0*rand()+7.0)]
}

```

```

#####
##
#for {set i 0} {$i < $val(nn)} {incr i} {
#$ns_ initial_node_pos $node_($i) 20
#}
#####
####
#
# Define traffic model
#
#puts "Loading scenario file..."
#source $val(sc)
#
#Provide initial (X,Y, for now Z=0) co-ordinates for mobilenodes
#
# Setup traffic flow between nodes
# TCP connections between node_(0) and node_(1)
for {set i 0} {$i < $val(nn)} {incr i} {
    set cluster_($i) [new Agent/Clustering]
    $ns_ attach-agent $node_($i) $cluster_($i)
}
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns_ at 1.0 "$cluster_($i) start"
}
# Tell nodes when the simulation ends
#
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns_ at 80.0 "$node_($i) reset";
}
$ns_ at 80.0 "stop"
$ns_ at 80.01 "puts \"NS EXITING...\" ; $ns_ halt"
proc stop {} {
    global ns_ tracefd
    $ns_ flush-trace
    close $tracefd
}

puts "Starting Simulation..."
$ns_ run

```

Annexe 3

Fichier Awk

```
BEGIN{
    paquets_sent = 0;
    lasttime = 0;
}
$1=="s"{
    i=$2;
    if (i-lasttime < 5.0)
    {
        paquets_sent = paquets_sent + 1;
    }
    else {
        lasttime =i;
        print "Temps: "lasttime " le nombre de paquets est " paquets_sent
        paquets_sent =0;
    }
}
END {
    print " le nombre de paquets est " paquets_sent
}
```