République algérienne démocratique et populaire
Ministère de l'enseignement supérieur et de la recherche scientifique

Université des sciences et technologies d'Oran « Mohamed BOUDIAF »
Faculté des sciences
Département d'informatique

Mémoire présenté pour l'obtention du diplôme de
MAGISTER EN INFORMATIQUE
OPTION SYSTEMES RESEAUX ET BASES DE DONNEES

DETECTION D'INTRUSIONS DANS LES RESEAUX AD HOC

Par
BENYETTOU Lahouari

Soutenue publiquement le 06 juillet 2011 devant un jury composé de :

Dr. MESSABIH B.	MCA	Président
Dr. CHOUARFIA A.	MCA	Encadreur
Dr. MEKKAKIKA M.Z.	MCB	Co-Encadreur
Dr. MEKKI R.	MCA	Examinatrice
Dr. BELKADI K.	MCA	Examinateur
Mr. MAAMAR S.	MAA	Invité

Résumé

L'absence d'infrastructure dans les réseaux sans fil ad hoc nécessite l'intégration, au sein de chaque nœud, différents modules permettant d'assurer sa propre sécurité, et participant ainsi à la sécurité du réseau Ad hoc.

Nous proposons une approche de détection d'intrusions au niveau de la couche 2 (liaison des données). L'objectif est de faire face aux attaques par brouillage virtuel où un nœud malicieux cherche à simuler une occupation du canal de communication, empêchant ainsi ces voisins à emmètre ou recevoir des données. Notre démarche consiste en premier lieu à détecter le nœud malicieux selon un comportement douteux en fonction des RTS émis (Nœud malicieux par émission de faux paquets RTS). Une fois localisé, une alerte est diffusée à l'ensemble des nœuds voisins pour l'élimination de ce dernier de toutes les tables de voisinage afin de l'isoler.

Pour se faire, nous avons mis en œuvre un module de coopération entre les différents nœuds. Les algorithmes de détection/réaction et mise-à-jours sont implémenté sous NS. Les résultats d'implémentation et des simulations obtenus montrent l'impact de l'attaque sur le réseau Ad hoc et les performances qu'offre notre approche.

Mots-clés: Ad hoc, RTS/CTS, Brouillage virtuel, Faux RTS, Détection d'intrusions, IDS.

Remerciements

Tout d’abord, je tiens à remercie mon encadreur, Dr. Abdallah CHOUARFIA, d’avoir accepté de diriger ce mémoire, pour ça confiance et ses encouragements continus.

Je remercie également mon co-encadreur, Dr. Zoulikha MEKAKIA MAAZA, qui m'a permis de réaliser ce travail et a su me conseiller très efficacement. Je tiens à lui exprimer ma reconnaissance, pour sa patience, sa présence, son dévouement et sa disponibilité à l’élaboration de ce travail.

Mes sincères remerciements, au Dr. Belhadri MESSABIH, pour avoir accepté de présider ce mémoire, à l’ensemble du jury, au Dr. Rachida MEKKI et au Dr. Khaled BELKADI ainsi qu’à Mr Soulaimane MAAMAR.

Mes remerciements vont aussi à tous les enseignants de la post-graduation systèmes réseaux et bases de données, en particulier la responsable de la P.G. Dr. Hafida BELBACHIR qui ne cesse d’encourager et de conseiller ses étudiants.

Je remercie également tous ceux qui ont contribué directement ou indirectement à l’aboutissement de ce présent mémoire. Merci à tous les membres de ma famille, à mes sœurs et à mes frères, à mes parents qui m’ont beaucoup soutenu et à qui je dédie tout ce travail.

Sommaire

Liste des figures & Tableaux

Introduction Générale

Les réseaux sans fil Ad hoc sont des réseaux ne disposant d'aucune infrastructure préexistante et formés de nœuds mobiles interconnectés par des liaisons sans fil. Leurs architectures évoluent au gré de l'apparition et du mouvement des nœuds. L'absence d'une gestion centrale des fonctionnalités du réseau rend ces réseaux beaucoup plus vulnérables aux attaques que les réseaux avec infrastructure (WLAN, LAN).

Si les MANET se différencient des réseaux classiques, cellulaires ou filaires, par les caractéristiques de leur topologie, les services demandés au réseau par les utilisateurs restent identiques, notamment en matière de sécurité. Les mécanismes de prévention d'intrusion, tels que le chiffrement et l'authentification, ne sont pas suffisants en matière de sécurité, la détection d'intrusions peut être considérée comme une deuxième ligne de défense. La majorité des techniques de détection d'intrusion proposée sont déployées dans la couche réseau.

Notre contribution à la sécurité traite les vulnérabilités au niveau de la couche MAC (Medium Access Control) en particulier des paquets de contrôle RTS (Request to Send).

Un nœud malveillant peut exploiter ces vulnérabilités au niveau du protocole de la couche MAC dans le but de créer des attaques de type brouillage virtuelle afin de perturber l'accès au canal de communication grâce au phénomène du faut blocage.

Dans le cadre de notre travail de recherche, nous proposons un système de détection d'intrusions ou IDS (Intrusion Detection System) pour les MANET, afin de contrer les attaques par brouillage virtuel basé sur les faux paquets RTS.

A cet effet, notre présent mémoire est organisé comme suit :

- Le chapitre1 présente une vue générale est détaillé des réseaux sans fil et sans infrastructure (Ad hoc).
- Le second chapitre introduit les IDS, il recense les différentes approches adoptées pour les réseaux Ad hoc, une étude comparative résume l'ensemble des travaux.
- Le chapitre 3 présente l'approche élaboré « Agent IDS » pour la détection et le traitement d'attaque de types faux RTS ainsi que son implémentation, les résultats font l'objet de discussion.

Enfin une conclusion générale clôture ce présent mémoire.

Chapitre I
Réseaux Ad Hoc

1.1 Introduction

Les avancées des performances des réseaux sans fil et le besoin de créer rapidement des réseaux sans infrastructure préexistante, dans des situations urgentes et des endroits parfois hostiles, ont favorisé le développement réseaux mobiles Ad hoc. Un réseau mobile Ad hoc appelé aussi réseau spontané, est composé de plusieurs entités mobiles et autonomes, capables de s'auto-organiser et de communiquer entre elles sans l'existence d'une infrastructure centralisée.

L'élargissement du domaine d'application des réseaux Ad hoc nécessite plus de sécurité pour assurer l'intégrité et la confidentialité des données qui circulent dans le réseau. En effet, les réseaux mobiles Ad hoc sont confrontés à de nombreux problèmes liés à leurs caractéristiques qui rendent les solutions de sécurité développées pour les réseaux filaires ou sans fil avec infrastructure inapplicables dans le contexte Ad hoc. Par conséquent, de nombreux thèmes de recherches ont surgi au cours des dernières années pour remédier à ces vulnérabilités et assurer les services de sécurité dans les réseaux mobiles Ad hoc.

Ce chapitre est organisé comme suit : dans la section 1.2, nous présentons les réseaux mobiles Ad hoc, leurs caractéristiques et leurs domaines d'application, ainsi que leurs protocoles d'accès au médium et de routage. La section 1.3 est dédiée aux vulnérabilités et la sécurité des réseaux mobiles Ad hoc, nous classifions les différentes attaques dans les réseaux mobiles Ad hoc. Enfin, la section 1.4 conclut le chapitre.

1.2 Les réseaux sans fil Ad hoc

1.2.1 Définition

Un réseau Ad hoc est un ensemble de nœuds mobile interconnectés par des liaisons sans fil formant un réseau dynamique sans infrastructure préexistante ou une architecture centralisée. Les nœuds sont libres de se déplacer, de joindre ou de quitter le réseau à tout moment, créant ainsi un changement spontané de la topologie. Chaque nœud dans ce type de réseau communique directement avec les autres nœuds qui se trouvent dans son rayon de communication (portée radio). La communication avec les nœuds hors portée radio se fait à travers des nœuds intermédiaires, qui s'approprient le rôle d'un routeur et acheminent les messages à destination. Ce processus se fait grâce au protocole de routage. À cet effet, plusieurs protocoles de routage ont été proposés et standardisés par le groupe MANET (Mobile Ad hoc NETwork).

Par ailleurs, les environnements adaptés à l'utilisation de ces réseaux sont caractérisés par l'absence ou la détérioration d'infrastructure réseau préexistante, telles les opérations de secours après un sinistre, les missions d'exploration ou les applications militaires. D'autres contextes, dont les spécificités de mobilité, de sécurité, de durée d'utilisation, et de rapidité de déploiement rendent impossible l'accès à une infrastructure réseaux existante, sont autant de cas d'utilisation idéale des réseaux ad hoc. Pour préciser les domaines d'utilisation, nous donnons ci-après les principales applications des réseaux ad hoc.

1.2.2 Les caractéristiques des MANET

Un réseau ad hoc est un système autonome constitué de nœuds mobiles. Ces derniers communiquent avec leurs voisins par des liaisons sans fil point à point. Quand les zones d'émission/réception de deux nœuds en communication sont disjointes, les nœuds intermédiaires sont alors sollicités pour assurer le routage. A partir de cette définition générale nous présentons les caractéristiques principales qui différencient un réseau ad hoc d'un réseau doté d'une architecture fixe.

1.2.2.1Les propriétés des nœuds dans un réseau Ad hoc:

Dans les réseaux ad hoc les propriétés des nœuds sons caractériser par :

La mobilité de tous les nœuds est une caractéristique intrinsèque des MANET. Le déplacement des nœuds provoque des modifications aléatoires et non prédictibles de l'architecture du réseau. De ce fait, les techniques de routage des réseaux classiques, basées

sur des routes préétablies par des équipements spécialisés et dédiés, ne peuvent plus fonctionner correctement.

L'équivalence des nœuds est une spécificité des MANET. Dans un réseau classique, il existe une distinction nette entre les nœuds terminaux (stations, hôtes) qui supportent les applications et les nœuds internes du réseau (routeurs), chargés de l'acheminement des données. Cette différence n'existe pas dans les réseaux ad hoc car tous les nœuds peuvent être amenés à assurer des fonctions de routage.

Le nombre de nœuds mobiles présents dans un MANET varie selon les besoins ou la position de chaque nœud. D'une façon plus générale aucune limitation n'est faite sur la taille ou le nombre de nœuds d'un réseau ad hoc.

Les ressources énergétiques des nœuds mobiles, alimentés par des sources d'énergies autonomes (batteries) sont limitées. Ces équipements intègrent des modes de gestion d'énergie et il est important que les protocoles mis en place dans les réseaux ad hoc prennent en compte cette caractéristique.

L'absence de serveur centralisé rend complexe le contrôle et la gestion d'une architecture qui se forme et évolue au gré de l'apparition et des déplacements des nœuds. En conséquence, il n'existe aucune hiérarchie entre les nœuds et aucun service réseau ne peut prétendre être centralisé.

Les liaisons physiques s'appuient sur les technologies de communications sans fil, indispensables à la mise en place d'un réseau ad hoc. Malgré des progrès très importants, leurs performances sont encore aujourd'hui en deçà de celles des technologies des réseaux LAN filaires.

Les vulnérabilités des 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 nœuds sont équivalents et potentiellement nécessaires au fonctionnement du réseau. Les possibilités de s'insérer dans le réseau sont plus grandes, la détection d'une intrusion ou d'un déni de service est plus délicate et l'absence de centralisation rend plus complexe la collecte d'informations pour la détection d'intrusions.

1.2.3 La norme IEEE 802.11

La norme IEEE 802.11 [1] définit deux modes opérationnels: le mode infrastructure, qui nécessite la présence d'un équipement spécialisé, appelé point d'accès, pour contrôler les communications entre les hôtes, et le mode ad hoc, basé sur des communications point à

point entre les hôtes d'un réseau. Le mode infrastructure représente la configuration la plus répandue. Celui-ci est utilisé par exemple pour permettre aux postes nomades d'accéder aux réseaux d'entreprises ou aux réseaux publics.

La normalisation des réseaux locaux par L'IEEE couvre les couches physique et liaison de données du modèle de référence O.S.I.[2]. La couche liaison de données retenue par l'IEEE est composée de deux sous-couches: la sous-couche LLC (Logical Link Control) et la sous-couche MAC (Medium Access Control).

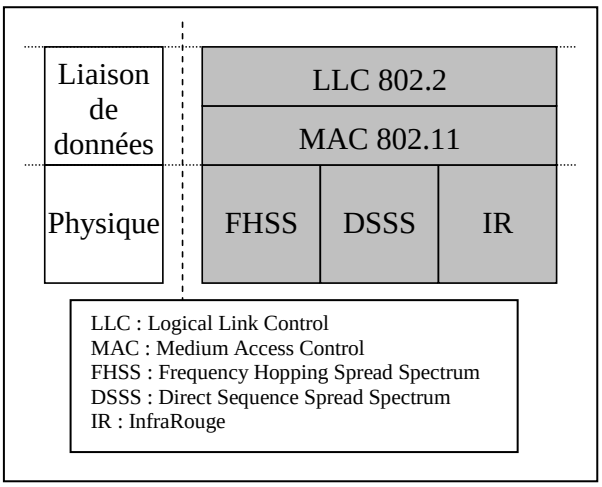

Figure. 1.1 Modèle IEEE 802.11

La couche LLC ou IEEE 802.2 est commune à toutes les normes 802 de l'IEEE. La couche IEEE 802.11 est une couche commune à plusieurs couches physiques 802.11, représentées sur la figure 1.1.

1.2.3.1 Description de la couche MAC IEEE 802.11 en mode ad hoc

Suivant les recommandations de l'IEEE, la fonction principale de la couche MAC est de gérer les accès au support. Deux méthodes, le PCF (Point Coordination Function) et DCF (Distributed Coordination Function) sont proposées dans la norme IEEE 802.11. Le mode PCF est utilisé pour supporter les trafics synchrones tels que les trafics en temps réel. Ce mode est utilisé dans le cas des réseaux avec infrastructure, car un point d'accès est nécessaire. Le mode ad hoc est uniquement basé sur le DCF. Le principe de base pour contrôler l'accès au média repose sur le CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance). Ce mode d'accès est une adaptation du protocole CSMA/CD (Carrier Sense Multiple Access with Collision Detection) utilisé dans les réseaux de type ETHERNET. Le protocole CSMA/CD autorise une station à émettre quand le média est libre et ensuite écoute le support pour détecter une éventuelle collision résultante d'une émission simultanée d'une autre station. Les liaisons radios n'étant pas full-duplex, la

détection d'une collision pendant l'émission est impossible. Le protocole 802.11 utilise des trames d'acquittement, appelées ACK (ACKnowledgement), pour confirmer qu'une collision ne s'est pas produite et donc que la trame émise a été correctement reçue.
Le contrôle de l'accès au support, suivant le DCF, utilise des espaces inter-trames ou IFS (Inter-Frame Space).

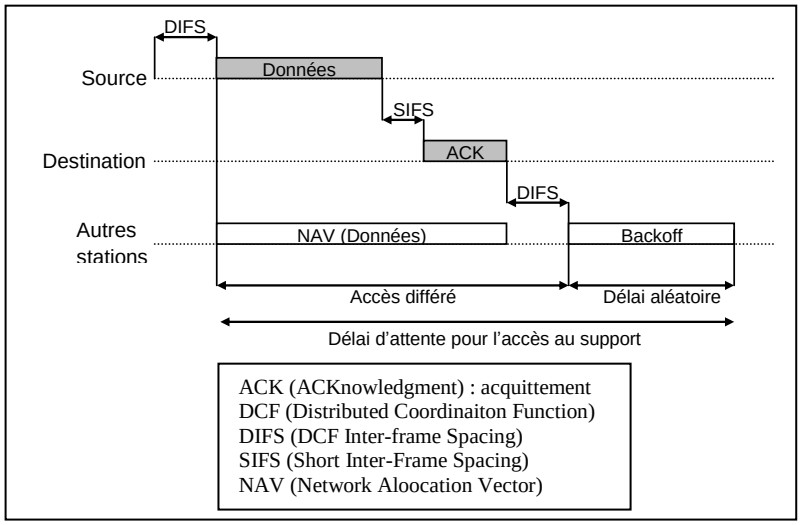

Figure. 1.2 Contrôle de l'accès au support

Avant d'émettre une trame de données, une station écoute le support pour déterminer si une transmission est en cours. Si aucune transmission n'est détectée pendant un intervalle égal à DIFS (Distributed Inter-Frame Space), la station transmet sa trame de données immédiatement. Toutefois plusieurs stations pourraient émettre simultanément. Pour prendre en compte cette situation la station destination confirme à l'émetteur, après un délai SIFS (Short Inter-Frame Space) plus court que l'intervalle DIFS, la bonne réception de sa trame par l'émission d'une trame d'acquittement ACK (Voir la figure 1.2). Quand une transmission est en cours et qu'une station a des données à transmettre, celle-ci attend que le canal reste libre pendant un intervalle de temps égal à DIFS augmenté de la valeur courante de son temporisateur de backoff (Bacltioff Timer) après la fin de la transmission pour pouvoir émettre ses données.

L'algorithme de backoff exponentiel permet de résoudre le problème de l'accès au support quand plusieurs stations ont des données à transmettre en même temps.

Après la fin d'une transmission, pour pouvoir émettre, une station teste que le support reste libre pendant une période égale à DIFS augmentée de la valeur courante du temporisateur de backoff. Lorsque le support est libre, les stations décrémentent leur temporisateur jusqu'à ce que le support soit occupé ou que la valeur du temporisateur soit nulle . Si le

média redevient occupé avant que le temporisateur atteigne la valeur 0, le temporisateur de backoff se bloque jusqu'à la prochaine libération du média plus une période égale à DIFS. Dès que la valeur du temporisateur de backoff est nulle, la station émet sa trame. Malgré ce mécanisme, deux stations peuvent émettre en même temps. Dans ce cas, chaque station régénère une nouvelle valeur de temporisateur calculée en fonction du nombre de tentatives de retransmission de la trame et de la valeur d'une variable aléatoire CW (Contention Window) jusqu'à une valeur maximale. Lorsque la transmission se produit, le temporisateur de backoff est alors initialisé avec sa valeur minimale.

La durée d'attente aléatoire (DAA) du backoff est calculée de la manière suivante :

DAA= CW* random(0,CW)*SlotTime

random(0,CW) est une variable aléatoire uniforme comprise entre 0 et CW-1

CW est la taille de la fenêtre de contention, ***CW = [CWmin, CWmax]***

Lors de la première tentative de transmission, ***CW = CWmin***; et à la fois suivante (en cas de collision) CW est doublée jusqu'à ce qu'elle atteigne ***CWmax.***

Cet algorithme permet aux stations d'accéder au support avec la même probabilité, mais ne permet pas de garantir un délai minimal nécessaire aux applications multimédia.Le schéma de la figure 1.3 résume une communication réussie dans le cas où aucune collision ne se produit :

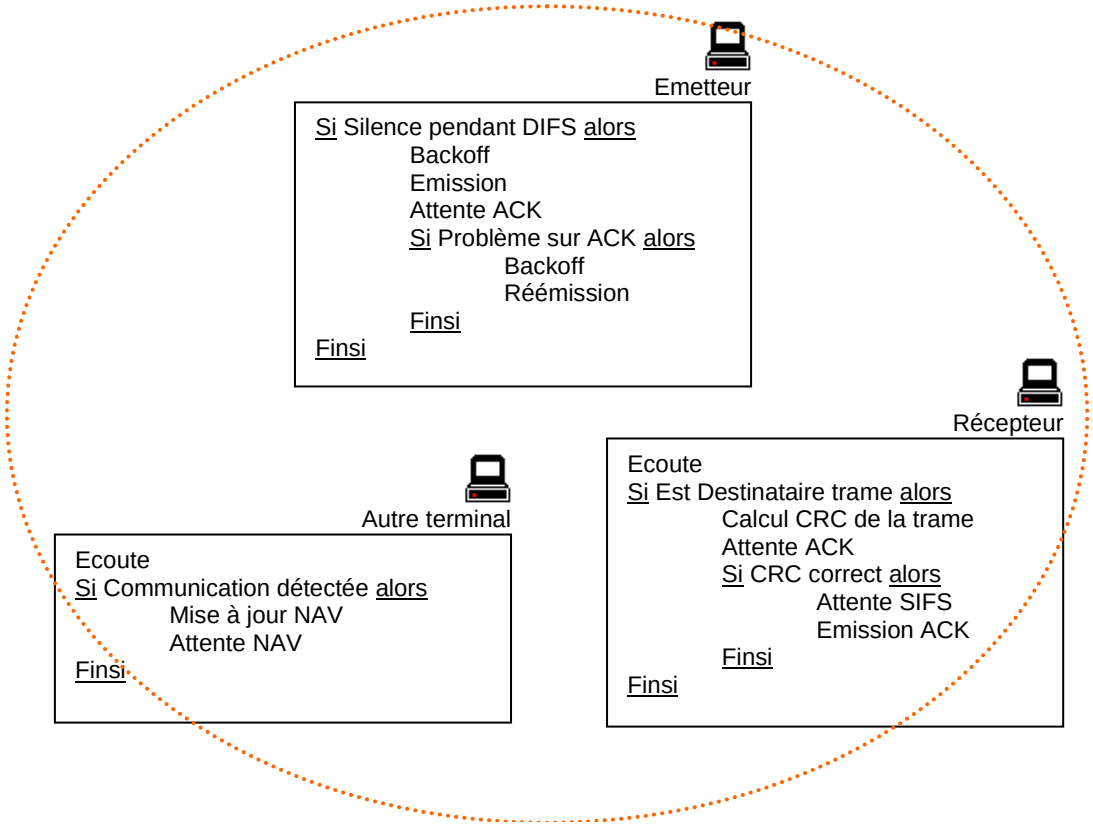

Figure 1.3 : Le protocole CSMA/CA dans sa forme la plus simple

Le problème des stations cachées se produit dès que trois stations forment un réseau et que deux d'entre elles ne sont pas à la portée l'une de l'autre, mais elles ont des zones de couverture qui se recoupent (voir la figure 1.4). Si les stations A et C ne font que la détection de porteuse en écoutant le canal, n'étant pas en mesure de s'entendre, elles vont s'autoriser à émettre des paquets en même temps à une station B située dans l'intersection des zones de couverture, dans ce cas, il va y avoir une collision entre les paquets et la station B ne pourra recevoir aucune des communications. On dit que les stations A et C sont cachées l'une par rapport à l'autre.

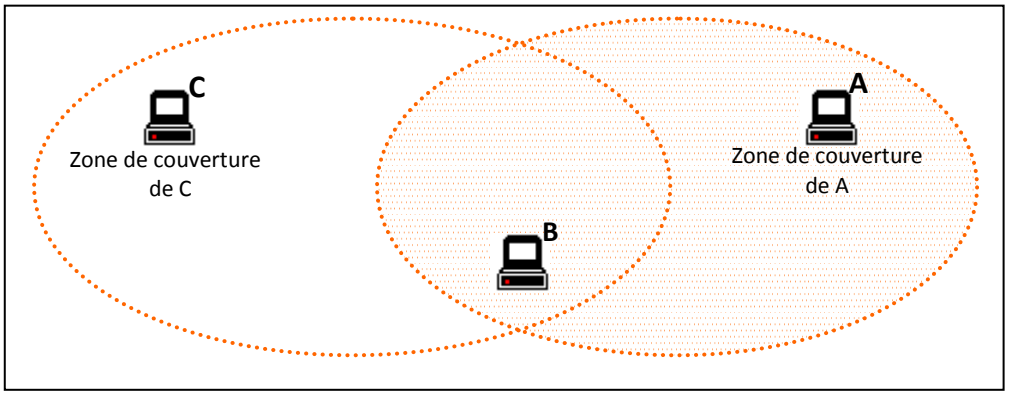

Figure. 1.4 Stations cachées

L'écoute du support est réalisée à la fois au niveau de la couche physique avec le PCS (Physical layer Carrrier Sense) et au niveau de la couche MAC avec le VCS (Virtual Carrier Sense). Le PCS analyse l'activité du support pour détecter la présence d'autres stations 802.11. Le VCS est un mécanisme de réservation basé sur l'échange de trames RTS/CTS (Request To Send/Clear To Send) entre une station source et une station destination avant tout envoi de données. Ces trames possèdent un champ, appelé durée de vie ou Duration, qui indique aux autres stations du réseau le temps pendant lequel le média est réservé pour la transmission des données. Une trame RTS est émise par la station source vers la station destination avant tout envoi de données. A la réception d'une trame RTS, la station destination répond par l'émission d'une trame CTS dont le champ durée de transmission sera lu par les autres stations pour mettre à jour leur NAV (Network Allocation Vector). Après la réception du CTS, la station source est assurée de la disponibilité et de la stabilité du support pour la transmission de ses données et de leur acquittement.

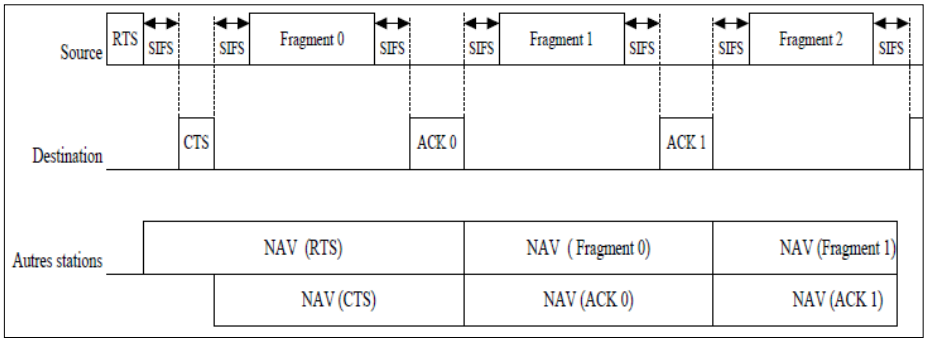

Figure 1.5 Fragmentation des trames 802.11

Ce mécanisme additionnel, consommateur de bande passante, n'est pas obligatoire pour transmettre toutes les trames. Il peut être activé uniquement pour l'émission des trames dont la longueur est supérieure à la valeur de la variable, appelée RTS_Threshold, dont la retransmission n'est pas souhaitée. Ce mécanisme est aussi utilisé pour résoudre le problème de la station cachée. La figure 1.6 résume l'usage du RTS/CTS et du NAV.

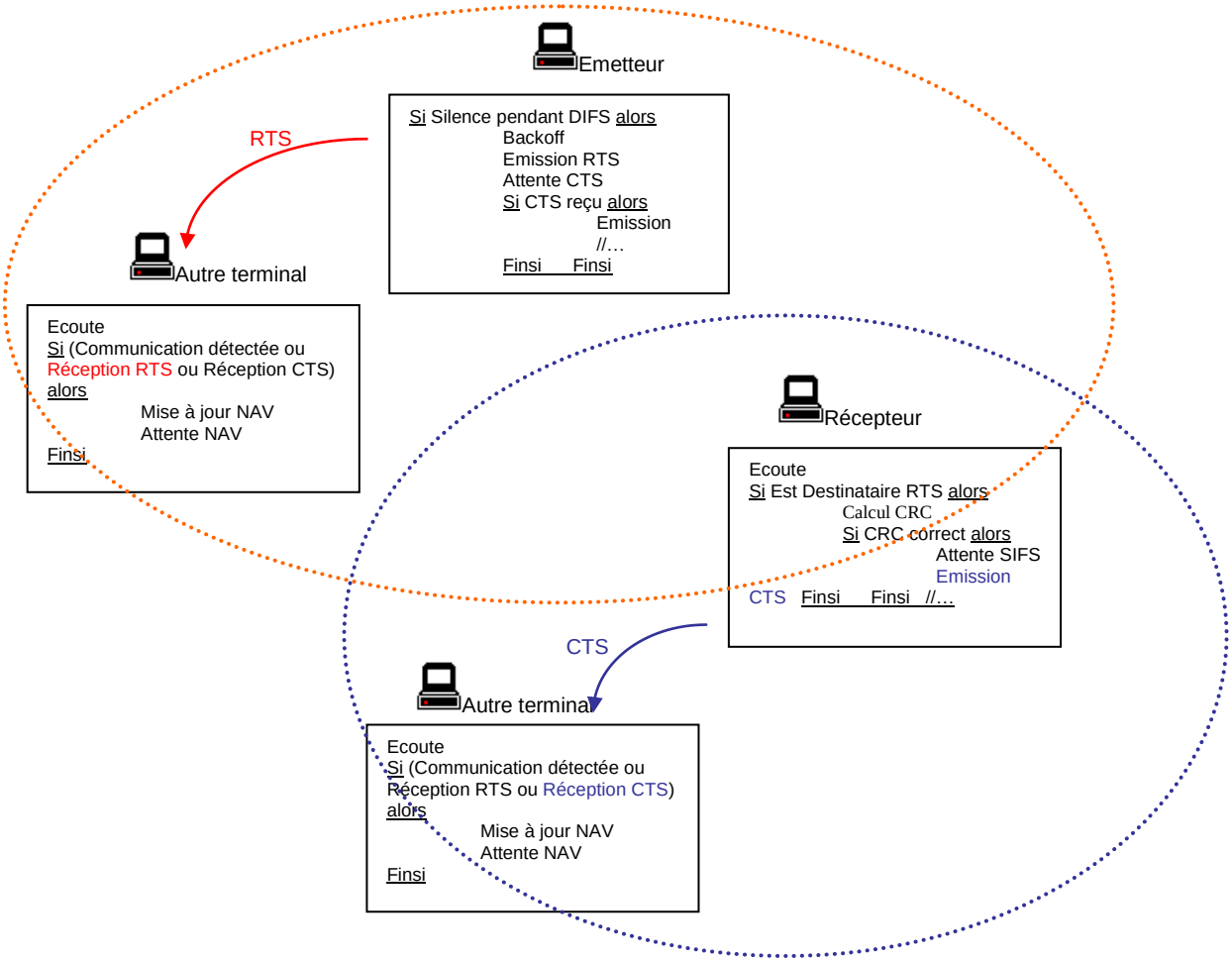

Figure 1. 6 : Le protocole CSMA/CA avec le mécanisme RTS/RTC

Le mécanisme RTS/CTS est expliqué plus en détails dans le chapitre 3.

La fragmentation consiste à diviser les trames de taille importante en fragments plus petits. Ceci à pour conséquence d'améliorer les performances du réseau car plus une trame est petite plus le risque d'erreur de transmission est faible. Tout d'abord, les trames RTS/CTS donnent la durée du prochain fragment et de son acquittement. Elles ne sont utilisées qu'une seule fois avant la transmission du premier fragment (Voir la figure 1.5). Ensuite, les champs durés de vie du premier fragment et de la trame ACK permettent aux autres stations du réseau de mettre à jour leur temporisateur NAV. La station source garde le contrôle du support pendant toute la durée de transmission de la trame fragmentée. La station destination acquitte chaque fragment correctement reçu par une trame ACK. Si un ACK n'est pas correctement reçu, la station source suspend l'émission des fragments et tente d'accéder de nouveau au support pour reprendre la transmission à partir du dernier fragment non acquitté.

1.2.3.2 L'établissement d'un réseau IEEE 802.11 en mode ad hoc

Dans un réseau ad hoc ou IBSS (Independent Basic Service Set), les stations communiquent entre-elles sans utiliser les services d'un point d'accès.

Pour contrôler les fonctionnalités d'accès au canal radio par le protocole CSMA/CA, et celles du mode d'économie d'énergie, les stations d'un IBSS doivent maintenir une horloge synchronisée. La synchronisation des horloges locales des stations est contrôlée par l'émission périodique de trames balises (Beacon frames). Ces trames balises doivent être transmises à intervalles réguliers appelés TBTT (Target Beacon Transmission Time). Aussi, à l'approche de la fin d'un intervalle TBTT, chaque station suspend tout autre trafic pour permettre l'émission ou la réception de trames balise. A la fin de l'intervalle TBTT, chaque station attend la fin d'un délai aléatoire supplémentaire, appelé Beacon Backoff, pour transmettre sa trame balise. A la fin de ce délai, si aucune trame balise n'a été reçue, la station envoie sa trame balise avec la valeur de son horloge locale temporisateur TSF (Timing Synchronisation Function).

Les autres stations comparent la valeur de leur temporisateur avec celle transmise dans la trame balise et ajustent leur temporisateur sur la plus grande des deux valeurs. De cette façon, l'horloge locale d'une station n'est jamais reculée et toutes les stations sont synchronisées sur l'horloge la plus rapide des stations de l'IBSS. La station qui à émis la dernière trame balise n'est pas autorisée à passer en mode économie d'énergie.

Quand une station veut s'intégrer à un IBSS, identifié par son SSID (Service Set identity)

indexSSID : Service Set identity déjà existant, elle écoute le support pour détecter des trames spécifiques. Cette écoute (scanning) peut être réalisée de deux manières différentes: en écoute passive, la station attend de recevoir une trame balise (Beacon); en écoute active, la station émet une trame de requête (Probe Request) et attend de recevoir la réponse (Probe Response). Celle-ci est générée par la station qui a émis la dernière trame balise (Beacon Frame).

1.2.4 Les protocoles de routage pour les réseaux Ad hoc

Un protocole de routage a pour fonction de déterminer le chemin entre deux nœuds en fonction d'une stratégie prédéfinie. Le routage dans les réseaux Ad hoc présente des défis plus complexes en comparaison avec le routage dans les réseaux filaires traditionnels. En effet, une stratégie intelligente de routage est nécessaire pour supporter la nature et les paramètres du réseau (la mobilité, le nombre de nœuds, la densité du trafic, la qualité du service et la superficie du réseau).

Dans un réseau ad hoc, le protocole de routage, distribué sur l'ensemble des nœuds, vise de plus à minimiser le temps d'établissement de la route, aussi appelé temps de *latence*, ainsi que l'utilisation des ressources nécessaires à cette opération.

Lorsque deux nœuds échangent directement leurs paquets de données sans passer par des nœuds mobiles intermédiaires, la connexion est dite *directe*.

Si le chemin entre les nœuds source et destination nécessite la présence de plusieurs nœuds intermédiaires, la connexion est alors qualifiée de *multi hop*.

Dans un réseau à architecture fixe, les routes vers les différents réseaux sont prédéfinies et maintenues par les équipements d'interconnexion fixes, appelés routeurs. L'architecture dynamique d'un réseau ad hoc, qui résulte du mouvement, de l'apparition des nœuds ou de l'état de la connexion physique, nécessite une mise à jour régulière des tables de routage situées dans chaque nœud. Pour acheminer un paquet entre deux nœuds mobiles d'un réseau ad hoc, le mécanisme de base est l'inondation. Celle-ci consiste à transmettre le paquet à l'ensemble des nœuds du réseau. L'inondation est réalisée par diffusions successives à l'ensemble des voisins de chaque nœud.

Des mécanismes complémentaires de contrôle peuvent être utilisés pour éviter les bouclages ou la duplication des paquets. Ce mécanisme d'inondation, très coûteux en ressources réseau, ne peut s'appliquer qu'à de très petits réseaux. Les protocoles de routage viseront, eux, à limiter la propagation des paquets par inondation. Selon le rôle joué par les nœuds dans la diffusion des messages, lorsque tous les nœuds ont des fonctionnalités

identiques, le protocole est qualifié d'*uniforme*, si certains nœuds ont des fonctionnalités particulières dans la diffusion des messages, le protocole est *non uniforme*.
Comme dans les réseaux à architectures fixes, les deux techniques à *état de liens* et à *vecteur de distance* sont utilisées. Les protocoles de routage à vecteur de distance possèdent une table de routage qui, à chaque nœud du réseau, associe l'adresse du prochain nœud, le vecteur, et le nombre de nœuds intermédiaires, *la distance*.

Les protocoles de routage à état de liens utilisent, eux, une base de données qui leur permet de construire la topologie du réseau et de connaître ainsi le chemin vers tous les nœuds du réseau. Une métrique basée généralement sur plusieurs paramètres relatifs aux liaisons est utilisée pour sélectionner la meilleure route.
Si la mise à jour de la table de routage est effectuée de façon périodique, le protocole est alors qualifié de *proactif*. De cette façon, l'ensemble des routes, mêmes celles inutilisées, sont mises à jour et demeurent immédiatement disponibles. Cette technique génère de nombreux paquets sur le réseau mais permet de minimiser le temps de découverte d'une route lors de son utilisation.

Pour diminuer la charge réseau due aux paquets de mise à jour, certains protocoles de routage déclenchent la recherche d'une route uniquement quand celle-ci est demandée. Le délai d'obtention d'une route est alors plus long. Les protocoles qui utilisent ce mode de fonctionnement sont dits *réactifs*.
Pour diminuer le nombre de messages de contrôle nécessaires à la découverte des routes, les protocoles de routage non uniformes sélectionnent certains nœuds pour créer des architectures hiérarchiques et dynamiques. Ainsi, pour les protocoles à sélection de voisins, chaque nœud décharge la fonction de routage à un sous ensemble de voisins directs. Tandis que pour les protocoles à partitionnement, le réseau est découpé en zones dans lesquelles le routage est assuré par un unique nœud maître. Certains de ces protocoles, qualifiés *d'hybrides*, utilisent conjointement le routage à état de liens et le routage à vecteur de distance.

Dans [3], les principaux protocoles de routage sont présentés selon leur mode de fonctionnement sur la figure 1.7

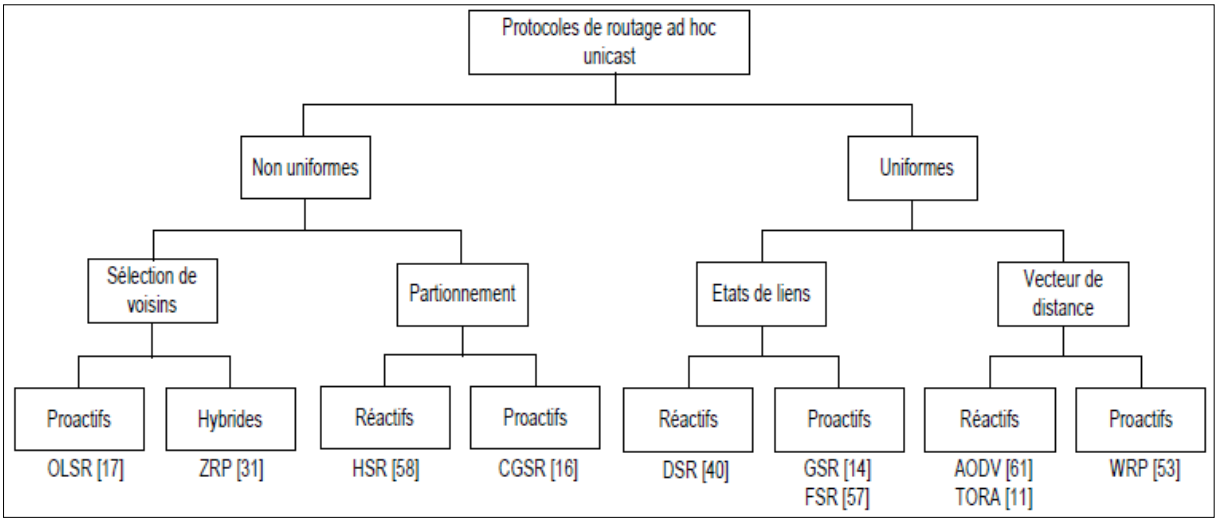

Figure 1.7 Classification des protocoles de routage Ad hoc.

Nous prenons exemples des protocoles AODV [4] et OLSR [5], possèdent, chacun, des stratégies de routage différentes:

- AODV (Ad hoc On demand Distance Vector) est un protocole réactif, uniforme et orienté destination. La route retenue est bidirectionnelle et correspond au plus court chemin (en nombre de nœuds) entre la source et la destination. Chaque nœud maintient une table de routage dont les entrées mémorisent, pour une destination:l'identifiant de cette destination,
    - l'identifiant du prochain nœud vers cette destination,
    - le nombre de nœuds jusqu'à cette destination.

  La demande de route est diffusée par la source à travers tout le réseau. Celle-ci permet à tous les nœuds de mémoriser une route vers la source. Quand la destination reçoit cette demande, la transmission de sa réponse permet aux nœuds intermédiaires de mémoriser la route recherchée. Pour maintenir les routes, chaque nœud vérifie périodiquement la présence de ses voisins. De plus, un numéro de séquence date chaque requête ou réponse pour éviter les multiples traitements et actualiser les routes.

- OLSR (Optimized Link State Routing Protocol) [5] est l'autre proposition retenue par l'IETF pour le routage dans les réseaux ad hoc. OLSR est un protocole proactif qui repose sur l'échange régulier d'informations sur la topologie du réseau.

L'algorithme est optimisé par la réduction de la taille et du nombre des messages échangés: seuls des nœuds particuliers, les MPR, MultiPoint Relay, diffusent des messages de contrôle sur la totalité du réseau.

Les définitions suivantes sont utilisées dans la description du protocole:

Nœud: hôte d'un réseau ad hoc implémentant le protocole OLSR.

Interface: point d'accès au réseau ad hoc. Un nœud peut avoir plusieurs interfaces, chacune ayant une adresse IP propre.

Voisin immédiat: le nœud X est un voisin immédiat du nœud Y si Y est à portée du nœud X (une des interfaces de X peut envoyer des messages sur l'une des interfaces de Y).

MPR (MultiPoint Relay) **:** nœud sélectionné par un de ses voisins immédiats (appelé MS, MPR Selector) pour retransmettre ses messages de mise à jour. L'ensemble des MPR d'un nœud est choisi parmi les voisins immédiats, de manière à permettre d'atteindre tous les nœuds situés exactement à deux sauts.

Lien: couple d'interfaces capables de communiquer (i.e. recevoir des messages). L'état d'un lien peut être:

- symétrique SYM, si les deux interfaces peuvent s'entendre,
- asymétrique ASYM ou HEARD, si les nœuds ont des puissances d'émission différentes,
- MPR, si l'émetteur a sélectionné le noeud comme MPR, dans ce cas le lien doit être symétrique.
- Lost, quand le lien est perdu.

Tous les nœuds envoient périodiquement des messages HELLO à leurs voisins immédiats (temporisateur HELLO_INTERVAL) sur chacune de leurs interfaces. Ces messages ne sont pas relayés vers d'autres nœuds.

OLSR utilise un seul format de message, transporté par le protocole UDP. L'entête précise si le message doit être seulement transmis au voisinage immédiat ou bien à l'ensemble du réseau.

Chaque nœud garde en mémoire la description de son voisinage: interfaces voisines, voisins à 2 sauts, MPR et MS. Cette description est mise à jour à chaque réception

d'un message HELLO, et les informations obsolètes sont effacées.

Le routage OLSR est basé sur l'acheminement des messages par les nœuds qui ont un voisinage symétrique. Un lien ne peut participer à une route que s'il est symétrique. Le routage vers les stations éloignées de plus d'un saut (1+N sauts) se fait grâce aux MPR, qui diffusent périodiquement des messages TC (Topology Control) contenant la liste de leurs MS. Un numéro de séquence permet d'éliminer les doublons. Ces messages servent à maintenir dans chaque station une table de la topologie.

La table de routage est construite et mise à jour à partir des informations contenues dans la table des interfaces voisines et la table de la topologie, en utilisant un algorithme de plus court chemin. La métrique prise en compte est le nombre de sauts.

1.3 Les vulnérabilités et la sécurité des réseaux Ad hoc

Les réseaux ad hoc sont vulnérables aux mêmes attaques que les autres types de réseaux. Celles-ci sont par contre plus aisées à mettre en œuvre car dans un réseau ad hoc, on ne peut ni contrôler l'accès au support de transmission, ni définir les limites du réseau. De plus, les réseaux Ad hoc sont confrontés à des vulnérabilités propres à leurs mécanismes et caractéristiques, la nature dynamique et coopérative des réseaux Ad hoc et surtout l'absence d'un point de concentration et de centralisation pour l'authentification et le contrôle sont susceptibles aux attaques qui peuvent exploiter le comportement coopératif durant le processus de routage, de gestion de topologie ou d'accès au médium.

1.3.1 Classification des attaques

La taxonomie des attaques contre les réseaux ad hoc proposée dans [5] et représentée sur la figure 1.8 identifie deux familles d'attaques: les attaques passives et les attaques actives.

Les attaques passives sont basées sur l'accès, par un nœud non autorisé, aux trames qui transitent sur le réseau pour collecter de l'information sans altérer les données échangées. L'écoute passive consiste à prendre connaissance du contenu des messages échangés entre deux nœuds. L'analyse du trafic consiste à déduire, en fonction des différents échanges réseaux, des informations sur l'organisation ou la configuration du réseau.

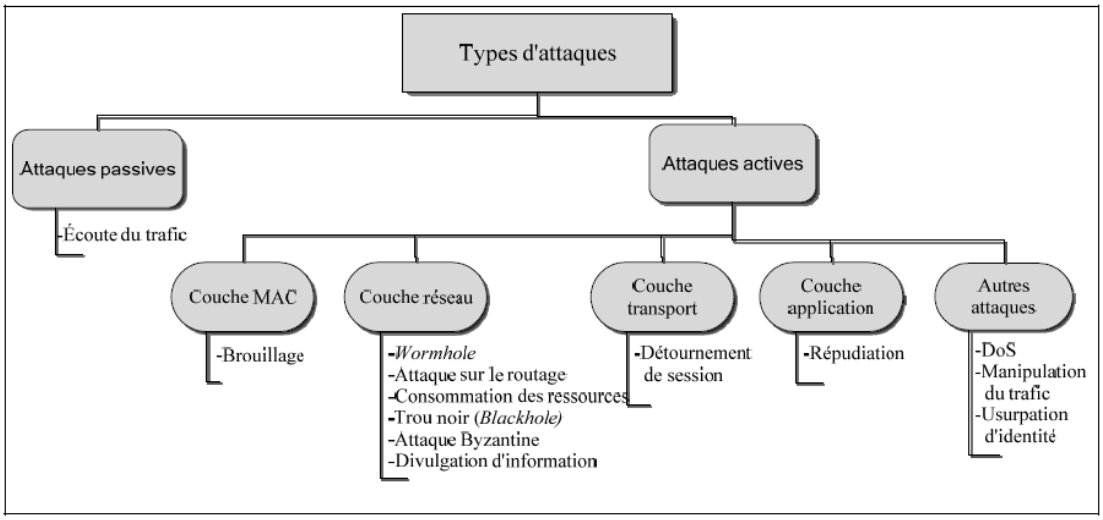

Figure. 1.8 Classifications des attaques dans les réseaux Ad hoc.

Les attaques actives ont pour objectif de permettre à un nœud autorisé de modifier un message, des données ou encore un flot de données. La mascarade est une attaque qui consiste pour un attaquant à se faire passer pour un autre utilisateur afin de bénéficier de ses droits d'accès aux différentes ressources. Pour réaliser une attaque par rejeu, l'attaquant commence par enregistrer une séquence de trafic compatible avec la politique de sécurité et ensuite régénère cet échange à la place d'une des parties pour tromper l'autre. La modification partielle du contenu d'un message échangé entre deux entités, ou la génération de trafic non légitime sont aussi des techniques exploitées par des attaquants pour compromettre le fonctionnement d'un réseau.

Dans un réseau ad hoc les attaques peuvent être dirigées contre les services d'une station ou ceux du réseau. Dans ce dernier cas, elles ciblent principalement le protocole de routage afin de perturber les communications entre les nœuds [6],[7],[8]. Les principales conséquences de ces attaques, présentées dans [9], sont résumées ci-dessous:

- L'introduction d'une boucle de routage.
- La création d'un trou noir qui consiste à rediriger le trafic vers un nœud qui ne retransmet pas les informations.
- La division du réseau en plusieurs sous-réseaux afin de bloquer les échanges entre les nœuds appartenant à des sous réseaux différents.
- La non retransmission par un nœud de certains messages.
- L'arrêt d'un nœud en raison de son manque d'énergie.
- Le déni de service d'un nœud qui est empêché d'émettre ou de recevoir des paquets.

1.3.2 Objectifs de la sécurité

Pour sécuriser un système, représenté par un ou plusieurs équipements informatiques connectés en réseau, la première étape consiste à établir la politique de sécurité. Celle-ci donne les règles d'application des services de sécurité pour protéger les ressources du système. Les ressources systèmes comprennent les données hébergées localement, les applications accessibles aux utilisateurs, ainsi que l'ensemble des ressources matérielles locales, comme l'espace disque ou le processeur.
Dans [2], l'ISO a défini les services de sécurité suivants :

L'authentification, qui garantit l'identité de l'entité, de l'utilisateur ou du processus. C'est une étape incontournable pour le contrôle de l'accès aux ressources réseau. Sans 'authentification, un nœud malicieux peut facilement usurper l'identité d'un autre nœud dans le but de bénéficier des privilèges attribués à ce nœud, ou d'effectuer des attaques sous l'identité d'un nœud pour nuire à ça réputation.

Le contrôle d'accès, qui garantit le respect des droits d'accès aux services et données hébergés par le système. Seuls les nœuds autorisés peuvent former, détruire, rejoindre ou quitter un groupe (cluster).

La confidentialité des données, qui garantit que l'information ne doit être ni rendue accessible, ni divulguée à un utilisateur, une entité ou un processus non autorisé. Dans le contexte des réseaux Ad hoc, la confidentialité consiste à refuser l'accès aux informations échangées entre deux nœuds dans le réseau par tout nœud malveillant ou non désiré. Or, les réseaux Ad hoc sont caractérisés par la diffusion générale des informations, ce qui constitue un vrai challenge pour la confidentialité.

L'intégrité des données, qui garantit que l'information ne doit pas être altérée ou détruite de manière non autorisée. Dans les réseaux Ad hoc, le bon fonctionnement du réseau repose essentiellement sur l'échange des messages de contrôle fournissant les informations pour le routage. Dans ce contexte, il est important, en premier lieu, de garantir l'intégrité des fonctionnalités de routage et des messages de contrôle contre toutes les modifications non autorisées.

La disponibilité des services, qui garantit que l'accès aux services offerts par le système est possible. La disponibilité est difficilement applicable dans les réseaux Ad hoc. En effet, à cause de la mobilité des nœuds, un protocole de routage ne pourrait pas maintenir toutes les routes vers tous les nœuds et surtout les nœuds qui quittent le réseau. Mais certains types d'attaques pourraient avoir effet sur la disponibilité des nœuds participant au réseau ou la disponibilité de certains services (attaques qui épuisent les batteries des nœuds, attaques par déni de service, etc).

La non répudiation, qui garantit que toute entité responsable d'une action sur le système ne peut nier l'avoir effectuée. Cette fonction donne l'assurance de l'identité du nœud originaire d'un message. La non-répudiation est utile dans la détection et l'isolation des nœuds malicieux. Par exemple, si un nœud *A* reçoit un faux message de la part d'un nœud *B*, la non-répudiation permet au nœud *A* de dénoncer *B* et informer les autres nœuds que *B* est compromis.

1.3.3 Défense contre les attaques dans les réseaux Ad hoc

Plusieurs solutions ont été proposées pour pallier les problèmes de sécurité dans les réseaux Ad hoc. Nous avons recensé les exemples de solutions dans le tableau1

Attaques	**Définition**	**Solutions proposées**
Wormhole	Un attaquant pourrait rediriger le trafic entre deux zones géographiquement éloignées pour créer un vertex dans la topologie et ainsi avoir une bonne position géographique pour contrôler le trafic qui passe par lui.	Packet Leashes *[7]*
Attaque de routage	Un nœud malicieux pourrait perturber le fonctionnement d'un protocole de routage en modifiant les informations de routage, fabriquer les fausses informations de routage ou usurper l'identité d'un autre nœud.	SEAD [10] ARAN [11], ARIADNE [7], SAODV [12].
Brouillage (Jamming)	C'est une attaque classique sur la disponibilité du canal de communication grâce à la génération massive d'une grande quantité d'interférence radio.	FHSS, DSSS [13].
Brouillage virtuel (Virtual jamming)	Un attaquant pourrait exploiter les vulnérabilités au niveau de la couche MAC et former de faux paquets de contrôle avec une fausse adresse de destination, afin que les nœuds récepteurs de ces paquets bloquent leurs transmissions inutilement, dans le but de créer une attaque de type déni de service dans le réseau.	DATA-Send (*DS*) [14] *RTS* validation*[15]*
Attaque trou noir *(Backhole attack)*	Le but de cette attaque est la falsification des informations de routage ou le détournement du trafic.	[16]
Attaque sur les Ressources	Les réseaux MANET sont caractérisés par des ressources limitées (batterie et bande passante). Une attaque sur les ressources pourrait avoir des conséquences sur la disponibilité.	SEAD [10]
Attaque Byzantine	Grâce à cette attaque, un nœud malicieux altère les messages et pourrait créer des problèmes de boucle de routage, routage de paquets vers des chemins non optimaux, sélectionné les paquets à rejeter… Ce type d'attaque est difficile à détecter car le réseau semble fonctionner correctement.	OSRP [17]
DoS	Ce type d'attaque consiste à envoyer délibérément des messages pour causer une saturation de la bande passante et paralyser le réseau.	SEAD[10], ARIADNE [18], SAODV [12].
Divulgation d'information	L'échange des informations confidentielles doit être protégé contre l'écoute ou l'accès non autorisé.	SMT(Papadimitratos et Haas, 2003),SRP [19].
Répudiation	Ce type d'attaque a une conséquence sur l'intégrité des communications entre les nœuds dans le réseau.	ARAN [11]
Usurpation d'identité	L'usurpation d'identité a pour but la falsification des informations relatives aux identités. Ce qui pourrait conduire à l'isolement de nœuds, l'échange de fausses informations de routage et l'atteinte à la confidentialité et l'intégrité.	ARAN [11], SAODV [12].
SEAD: Secure Efficient Ad hoc Distance vector routing protocol; SAODV: Secure Ad hoc On-demand Distance Vector routing; ARAN: Authenticated Routing for Ad hoc Networks; ARIADNE: A Secure On-Demand Routing Protocol for Ad hoc Networks; FHSS: Frequency-Hopping Spread Spectrum; OSRP: On-demand Secure Routing Protocol; SMT: Secure Message Transmission Protocol; SRP: Secure Routing Protocol for Mobile Ad hoc Network; DSSS: Direct-Sequence Spread Spectrum.		

Tableau 1.1 présentation des différentes attaques avec leur solution

1.4 Conclusion

Nous avons présenté dans ce chapitre les différentes caractéristiques des réseaux mobiles Ad hoc, ainsi que leur domaine d'application et la technologie de la couche MAC 802.11. De plus, nous avons détaillé les besoins en sécurité dans ces réseaux. En outre, nous avons discuté des différents domaines de recherche liés à la sécurité dans les réseaux mobiles Ad hoc. Les mécanismes de sécurité préventifs forment la première ligne de défense du système contre les différentes menaces. On ne peut toutefois considérer cette protection comme absolue, permanente et incontournable. Une surveillance permanente du système protégé peut permettre de renforcer l'action des mécanismes de sécurité. D'où, la nécessité d'avoir des systèmes de détection d'intrusions pour limiter les activités malveillantes en tenant compte des caractéristiques de ces réseaux.

Chapitre II
Système de Détection d'Intrusion

2.1 Introduction

La sécurité des réseaux informatiques consiste à protéger les services offerts contre toute manipulation non autorisé en employant des mécanismes de préventions tels que le contrôle d'accès, le chiffrement et l'authentification. Néanmoins, cette protection n'est pas suffisante en matière de sécurité, et ne peut être considérer comme absolue et incontournable. En effet, il est parfois possible, pour un utilisateur malveillant, de contourner les mécanismes de prévention et donc de violer la politique de sécurité que mettent en œuvre ces mécanismes. Une surveillance permanente du système protégé peut permettre de renforcer la sécurité. Cette tâche de surveillance, aussi appelée audit suppose un mécanisme d'enregistrement des événements du système au sein de «journaux» et une phase d'analyse des journaux afin d'identifier une éventuelle violation de la politique. La nécessité d'automatiser les tâches d'audit des systèmes informatiques, a conduit au développement des systèmes de détection d'intrusions IDS pour (Intrusion Detection System).

Dans ce chapitre, nous présentons les différentes étapes du processus de détection d'intrusion ainsi que les caractéristiques des IDS. Ensuite, nous présentons les différentes architectures d'IDS pour les réseaux Ad hoc et les travaux IDS-ad hoc existant.

2.2 Système de détection d'intrusion

Un système de détection d'intrusions regroupe à la fois des mécanismes de surveillance, de contrôle et d'analyse permettent de repérer des activités anormales ou suspectes sur la cible analysée (un réseau ou un hôte).

Historiquement, J.P Anderson [20] a été le premier à montrer l'importance de l'audit de sécurité pour détecter des violations de la politique de sécurité appliquée à un système. Il propose d'utiliser les traces d'audits de sécurité pour construire un modèle statistique représentatif du comportement usuel d'un utilisateur du système afin d'en détecter toute action inhabituelle. Cette première approche qui consiste à détecter une déviation par rapport à un comportement normal préalablement défini est appelée l'approche comportementale ou encore anomaly detection.

Une autre approche consiste non plus à se référer aux comportements normaux et détecter les écarts par rapport à cette référence, mais à modéliser les menaces par des signatures et à rechercher la présence de ces signatures dans les traces d'audits. Cette approche est appelée l'approche par scénarios ou misuse detection. Dorothy E. Denning et al [21] ont proposé le premier système de détection d'intrusion hybride, appelé IDES (Intrusion Detection Expert System) [21], regroupant les deux approches.

2 .2.1 Architecture d'un IDS

Le groupe de travail IDWG (Intrusion Detection Worhing Group) de l'IETF a proposé un modèle fonctionnel d'IDS constitué de trois composants de base [22]. La figure 2.1 illustre les interactions entre ces trois composants. Un capteur (ou senseur) est chargé de collecter des informations sur l'évolution de l'état du système et de fournir une séquence d'événements qui traduit l'évolution de l'état du système. Un analyseur détermine si un sous-ensemble des événements produits par le capteur est caractéristique d'une activité malveillante. Un manager collecte les alertes produites par le capteur, les met en forme et les présente à l'opérateur. Éventuellement, le manager est chargé de la réaction à adopter. Nous détaillons par la suite chacun de ces trois composants.

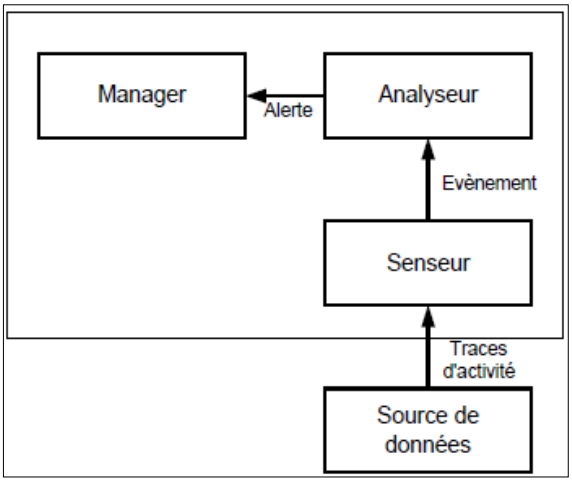

Figure 2.1 IDWG : Modèle d'architecture IDS de IDWG

2.2.1.1 Le capteur

Le capteur (senseur) est chargé de collecter les données brutes et de les mettre en forme pour transmettre des événements à l'Analyseur. En général un prétraitement est effectué sur les données brutes collecté pour filtrer un certain nombre de données considérées comme non pertinentes afin de limiter la quantité d'information à analyser. De plus, le capteur réalise généralement une mise en forme des données brutes acquises afin de présenter à l'analyseur des données utilisant un certain format d'événements. On distingue classiquement trois types de capteurs en fonction des sources de données utilisées pour observer l'activité du système :

- Les capteurs système qui collectent des données produites par les systèmes d'exploitation des machines, notamment par le biais des journaux d'audit système ou par celui des appels système invoqués par les applications. On désigne les IDS utilisant des capteurs système par l'acronyme HIDS (Host-based IDS).
- Les capteurs réseau qui collectent les données en écoutant le trafic réseau entre les machines par le biais d'une interface spécifique. On parle alors de NIDS (Network-based IDS).
- Les capteurs applicatifs qui collectent les données produites par une application particulière, avec laquelle des utilisateurs sont susceptibles d'interagir, comme un serveur web ou un serveur de base de données. L'application doit alors être instrumentée à cet effet.

2.2.1.2 L'analyseur

L'objectif de l'analyseur est de rechercher dans le flux d'événements transmis par le capteur, des signes d'activités indésirables ou non autorisées et génère les alertes destinées au Manager. Deux grandes approches ont été proposées, l'approche comportementale (anomaly detection) et l'approche par scénarios (misuse detection) :

L'approche par scénarios ou misuse detection identifie une intrusion par la présence d'une suite d'événements, dans les enregistrements d'audit, appelée signature, définie comme étant révélatrice d'une attaque connue. Cette approche nécessitent une connaissance a priori des attaques à détecter : une alarme est émise lorsque la trace d'une attaque connue est détectée. Les approches par scénarios utilisent des techniques différentes, dont les principales sont :

- **Les techniques à base de règle** (i.e. les systèmes experts). Dans cette technique la base de connaissances du système expert contient des règles concernant non seulement les scénarios d'attaque que l'on veut détecter sur le système cible, mais aussi les vulnérabilités connues de ce système. La construction de cette base repose entièrement sur l'expérience et le savoir faire de l'administrateur de sécurité.
- **Les techniques fondées sur Analyse par comparaison (Pattern Matching)** : Le principe de cette approche est de faire correspondre à chaque signature d'attaque un motif (Pattern) qui est sous forme d'une chaîne de caractères. Durant l'analyse du flux de données qui est aussi une chaîne de caractères, le système de détection d'intrusion tente de reconnaître les motifs d'attaques connus [23].
- **Les techniques fondées sur algorithmes génétiques**, dans ce type de technique les algorithmes génétiques sont utilisés pour optimiser la recherche de scénarios d'attaque qui sont enregistré dans des fichiers d'audit [24]. L'utilisation des algorithmes génétique permet d'éviter une recherche exhaustive en sélectionnant les meilleurs éléments d'une population (ici les scénarios d'attaque).

L'intérêt de l'approche par scénario est la prise en compte des comportements exacts des attaquants potentiels, cependant cette approche possède les inconvénients suivants :

- la base de règles doit être bien définit, ce qui n'est pas toujours le cas
- les performances du système expert sont en fonction des règles fournis par l'expert humain. Or les connaissance de ce dernier en matière d'attaque son limiter
- les nouvelles attaques ne peuvent être détectées (faux négatifs) tant que leur signature n'a pas été établie.

- la génération de fausses alarmes (faux positifs) quand les signatures ne sont pas suffisamment précises.

L'approche comportementale, une attaque est identifiée par la mesure d'une déviation sensible du comportement d'une entité par rapport à un comportement de référence, réputé sain et défini auparavant. Cette approche nécessite une phase d'apprentissage, durant laquelle les cas de références, représentant le profil normal, sont construits. Une alarme est émise quand une déviation trop importante de ce profil est observée. L'entité, dont le comportement est observé pour la détection, peut être l'utilisateur, le système, ou le réseau. Pour modéliser le profil normal, les approches comportementales utilisent en générale:

- des **modèles statistiques** qui consistent à utiliser des formules statistiques pour modéliser un profil de comportement et détecter ensuite les comportements intrusifs. Chacune des ces mesures est associée à un seuil ou à un intervalle de valeurs, dans lequel une activité est considérée comme normale. Tout dépassement de seuil ou situation de valeurs à l'extérieur des bornes de l'intervalle indique une activité anormale
- des **systèmes experts**. la base de connaissances du système expert contient des règles pour représenter le comportement des entités. La différence majeure entre un système expert et un modèle statistique est que ce dernier utilise des mesures statistiques pour identifier des comportements dans les données d'audit alors que le système expert utilise un ensemble de règles pour représenter ces comportements.

L'apprentissage dans les approches comportementales repose sur les techniques:

- **des générateurs de forme prédictive**. Cette approche prédit les formes les plus probables en se basant sur les formes observées. Durant la phase d'apprentissage, cette approche détermine des règles temporelles qui caractérisent le comportement normal des utilisateurs.
- **des réseaux de neurones**. Cette approche est constitué de plusieurs éléments de traitement simples appelés unités et qui interagissent en utilisant des connections pondérées. Au départ le réseau constitue le profil normal d'un utilisateur. Ainsi, après chaque commande utilisée par cet utilisateur, le réseau essaye de prédire la commande suivante, en tenant compte des n commandes antérieures. Si la commande réelle dévie de celle prédite, alors une alarme est envoyée.

L'intérêt de l'approche comportemental est de permettre la détection d'attaques inconnues. Néanmoins, elle présente aussi quelques défauts tel que:

- le choix des différents paramètres du modèle statistique sont en fonction de l'expérience de l'officier de sécurité
- le déclenchement des faux positifs si le profil de l'entité observée est incomplet ou si des modifications brutales de l'environnement interfèrent sur ce profil,
- des faux négatifs si l'apprentissage du profil est faussé, par exemple, par une modification lente du comportement d'un utilisateur dans l'intention de faire apprendre au système un comportement intrusif.

L'approche hybride, qui combine entre les deux techniques précédentes.

Quelle que soit l'approche choisie comportementale, par scénarios ou hybride, tous les IDS se trouvent confrontés au problème de la détection de nouvelles attaques, plus précisément à la détection d'attaques inconnues suite à la complexité croissante des réseaux et des attaques auxquels ils sont sujets [24].

2.2.1.3 Le manager

Le manager est responsable de la présentation des alertes à l'opérateur (fonction de console de management). Il peut également réaliser les fonctions de corrélation d'alertes, dans la mesure de leur disponibilité. Enfin, il peut assurer le traitement de l'incident, par exemple au travers des fonctions suivantes :

- confinement de l'attaque, qui a pour but de limiter les effets de l'attaque ;
- éradication de l'attaque, qui tente d'arrêter l'attaque ;
- recouvrement, qui est l'étape de restauration du système dans un état sain ;
- diagnostic, qui est la phase d'identification du problème, de ses causes et qui peut éventuellement être suivi d'actions contre l'attaquant (fonction de réaction).

2 .2.2 Fréquence d'utilisation d'un système de détection d'intrusion

Un système de détection d'intrusion peut effectuer la surveillance des évènements selon deux cas:

- **Surveillance périodique** *(offline):* l'analyse se fait périodiquement à la recherche de nouvelles violations ou d'anomalies. Les résultats sont comparer à la fin de journée pour avoir une dépense plus fiable du point de vue temps de calcule (repense passive).

- **Surveillance en temps réel** *(online):* Il détecte l'attaque au moment où elle se produit, l'analyse se fait de manière continue. Elle permet de limiter les dégâts en temps réel d'une attaque ont appliquons des contre-mesures (repense active).

2 .2.3 Efficacité d'un système de détection d'intrusions

Les IDS sont très importants dans une stratégie de sécurité, c'est pour quoi le choix d'un l'IDS est très décisif et doit être basé sur ces caractéristiques, les tâches qu'il devra accomplir, et sur l'architecture du réseau selon DEBAR [25] on peut déterminer l'efficacité d'un IDS par les mesures suivantes :

- **Exactitude :** Qualité des informations fournies par l'IDS : Le taux de faux positif ; si le taux est élevé alors ce système considère des actions légitimes comme atypiques avec un taux élevé
- **Performance :** Réponse des IDS dans un environnement surchargé elle est mesurée par le taux de traitement en temps réel des traces d'audits.
- **Perfection :** La possibilité de mettre à jour la base des signatures ou de modifier certaines signatures.
- **Tolérance aux pannes :** un IDS doit être résistant aux attaques, en particulier dans le cas des attaques de déni de service très favorisées par les hackers (un nœud en panne mais considéré comme élément sein par les autres nœuds connectés sur ce réseau).

Pour une détection d'intrusions efficace, il est important de prendre en considération les caractéristiques présenté par [24] :

- **La distribution** : un grand nombre d'attaques réseaux se caractérisent par des comportements anormaux à différents éléments du réseau (serveur, routeur,…). Il est donc très important de distribuer les fonctions de détection à plusieurs entités qui surveillent différents points du réseau.
- **L'autonomie***:* des échanges excessifs d'informations entre les entités distribuées peuvent congestionner le réseau. Il serait donc plus judicieux de laisser l'entité, surveillant un élément réseau, effectuer une analyse locale et détecter les comportements intrusifs locaux. Ainsi, les unités distribuées doivent être autonomes.
- **La délégation** *:* La dynamicité des réseaux nécessite de pouvoir modifier, à n'importe quel moment, les fonctions de détection d'intrusions pour les adapter aux

changements se produisant dans le réseau surveillé. Cela est possible grâce au modèle de délégation. Les tâches déléguées sont envoyées aux entités autonomes. Chaque entité aura à exécuter sa propre tâche. Lorsque de nouvelles tâches doivent être ajoutées, ceci est fait dynamiquement.

- **La communication et coopération** : la complexité des attaques coordonnées ne facilite pas leur détection par une seule entité. En effet, chaque entité n'ayant qu'une vue locale restreinte du réseau, il lui est très difficile de détecter ce type d'attaques. La détection de ce genre d'attaques, nécessite une corrélation des différentes analyses effectuées à différents points du réseau. Les différentes entités doivent alors se communiquer leurs analyses et coopérer afin de détecter efficacement les attaques coordonnées.
- **La réactivité** : l'objectif majeur de la détection d'intrusions est de réagir rapidement lorsqu'une attaque se produit afin de limiter les dommages qui peuvent être causés.
- **L'adaptabilité** : les politiques de sécurité d'une entreprise peuvent changer. Dans ce cas l'administrateur doit changer et/ou rajouter de nouvelles politiques afin de modifier et réadapter les tâches de détection d'intrusions.

2.3 Les nouveaux défis de l'IDS dans les MANET

Les solutions d'IDS proposées pour les réseaux classique déploient souvent des capteurs réseau dans les principaux points de concentration du trafic, tels que les commutateurs, les routeurs et pare-feu. Ces IDS à base de capteurs réseau (NIDS) sont sécurisés physiquement, et utilisent les techniques de détection basée sur les signatures pour détecter les attaques. Par ailleurs, ces solutions ne peuvent entre employées pour les réseaux ad-hoc. En effet, Les réseaux ad hoc manquent de points de concentration où le trafic réseau peut être contrôlé. Cela limite l'efficacité d'un IDS basé sur le capteur réseau, puisque seul le trafic généré au sein de la portée de transmission radio peut être surveillé. De plus, dans un réseau ad hoc caractérisé par une topologie dynamique et des changements imprévisibles, il peut être difficile de s'appuyer sur l'existence d'un nœud centralisé pour effectuer l'analyse et la détection.

Par conséquent, l'IDS ad hoc a besoin d'une architecture pratique et évolutive pour recueillir suffisamment de preuves en temps réel afin de détecter les attaques de manière efficace.

En outre, les liaisons sans fil entre les nœuds mobiles sont beaucoup plus fiables que ceux dans un réseau filaire, par conséquent, le mécanisme de détection doit être capable de tolérer la perte de messages afin d'avoir suffisamment de données à analyser et à maintenir la précision de détection.

2.3.1 Les contraintes imposées par les MANET pour les IDS

Ces contraintes sont imposées par les nœuds, l'architecture dynamique du réseau, et les liaisons sans fil:

2.3.1.1 Les ressources limitées des nœuds

Les nœuds d'un réseau ad hoc sont des systèmes portables et mobiles. De ce fait, leurs capacités sont limitées, notamment leur ressource en énergie, leur puissance de calcul et leur capacité de stockage. L'IDS doit donc être conçu pour limiter l'utilisation des ressources locales.

2.3.1.2 La flexibilité, la mobilité et les limites du réseau

Le nombre et la position des nœuds, qui constituent l'infrastructure du réseau, évoluent selon les comportements des utilisateurs et le contexte d'utilisation. Il revient donc à chaque nœud d'assurer sa propre sécurité, et d'accorder une confiance mesurée aux autres nœuds. La technique de filtrage des paquets utilisée pour contrôler les accès aux points d'entrée d'un réseau filaire n'est pas applicable aux réseaux ad hoc. Ainsi, nous pouvons considérer que les attaques sont toujours internes.

L'IDS doit prendre en compte la dynamique permanente du réseau et l'absence de frontière identifiable entre l'extérieur et l'intérieur du réseau.

2.3.1.3L'absence de système central permanent

Dans un réseau ad hoc, il n'existe pas de nœud central permanent capable de collecter l'ensemble des informations nécessaires à la détection des intrusions. A un instant donné, celles-ci sont distribuées sur l'ensemble des nœuds actifs.

L'IDS doit permettre la collecte d'informations situées sur les différents nœuds du réseau.

2.3.1.4Les performances limitées du réseau

Dans les réseaux sans fil, les performances, notamment en matière de débit, sont aujourd'hui en dessous des débits disponibles dans les réseaux filaires.

Les échanges de données nécessaires à la détection des intrusions devront donc être réduits à leur strict minimum.

2.3.1.5 La politique de sécurité

D'une façon générale un IDS se base sur une politique de sécurité, définie de façon explicite ou implicite, pour identifier les attaques. Pour les réseaux ad hoc, [25] propose de définir la politique de sécurité par rapport à des communautés d'utilisateurs.

2.3.2 Les caractéristiques d'un IDS pour MANET

Dans un MANET, la mobilité des nœuds, la distribution des fonctions et des données et les caractéristiques des connexions sans fil imposent des contraintes spécifiques pour la détection d'intrusions. A partir des caractéristiques générales des IDS, et des contraintes imposées par les réseaux ad hoc, nous établissons dans cette section les spécifications d'un IDS pour MANET.

2.3.2.1Principes de détection

L'architecture de l'IDS doit être indépendante de la méthode de détection. La sélection d'un principe de détection ne pourra se faire qu'à l'issue de tests comparatifs.

2.3.2.2Sources de données

L'architecture de l'IDS doit être indépendante de la source des données et répondre aux critères de portabilité et d'indépendance vis-à-vis des différents systèmes d'exploitation utilisés sur les nœuds.

2.3.2.3Fréquence d'utilisation

La détection des tentatives d'intrusions doit se faire à l'exécution de l'attaque pour permettre aux utilisateurs de prendre les mesures nécessaires et renforcer ainsi l'action des mécanismes de sécurité.

2.3.2.4Comportement après détection

Sous le contrôle de l'utilisateur, la réponse doit être active en local pour accroître le niveau de sécurité et informative vers les autres nœuds présents dans le réseau.

2.3.2.5Distribution des nœuds

L'architecture de l'IDS doit prendre en compte le caractère spontané des réseaux ad hoc ainsi que l'absence de nœud central permanent.

2.3.2.6Débits limités des liens inter nœuds

Les technologies WLAN offrent encore aujourd'hui des débits inférieurs à ceux des LAN. L'IDS doit s'appuyer sur les technologies les moins consommatrices de ressources réseaux.

2.3.2.7Mobilité des nœuds

L'IDS doit posséder les mécanismes lui permettant de prendre en compte la mobilité des nœuds.

2.3.2.8Normalisation

L'architecture de l'IDS doit adopter les normes actuelles notamment pour pouvoir coopérer avec d'autres IDS.

2.4 Détection d'intrusions dans les réseaux Ad hoc

L'absence des nœuds pour une surveillance centralisée et le manque de confiance entre les nœuds d'un réseau mobile ad hoc, rendent un système de détection d'intrusions centralisé irréalisable. Pour remédier aux limites des IDS classique inapproprié aux réseaux ad hoc, plusieurs travaux de recherche ont été réalisés dans ce domaine. Dans cette partie, nous allons illustrer les modèles les plus importants des systèmes de détection d'intrusions proposés pour les réseaux Ad hoc.

2.4.1 Les IDS autonomes

Dans cette architecture, chaque nœud exécute un IDS qui détecte les attaques de façon indépendante. L'IDS autonome n'a confiance qu'en soi même, donc, il n'existe aucune coopération, ni de partage d'informations avec les autres nœuds du réseau (*voir Figure 2.2*). Ainsi, toute décision de détection d'intrusion est basée sur des informations disponibles au niveau du nœud.

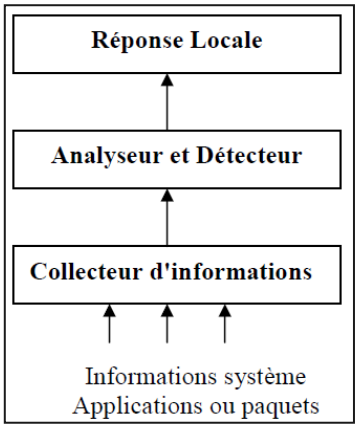

Figure 2.2 Architecture générale d'un IDS individuel

Bien que l'efficacité de cette solution soit limitée, cette architecture est mieux adaptée dans un environnement où tous les nœuds ne sont pas capables d'exécuter des IDS [27].

2.4.2 Les IDS distribués et coopératifs

L'approche utilisé pour les IDS distribués et coopératifs repose sur le principe de base des réseaux ad hoc, la détection des intrusions, comme tous les autres services du réseau ad-

hoc, doit aussi être distribuée sur l'ensemble des nœuds du réseau. Chaque nœud est autonome, et de ce fait, il ne peut s'appuyer que sur ses ressources propres pour détecter les intrusions dont il est la cible. La détection de certains types d'intrusions peut nécessiter la collecte d'informations complémentaires disponibles uniquement sur d'autres nœuds. Dans ce cas, les nœuds sont amenés à coopérer pour s'échanger des données ou encore des alertes. Zhang et Lee [28] ont proposé la première architecture d'IDS distribuée et coopérative pour les réseaux ad hoc présenté dans la figure 2.3

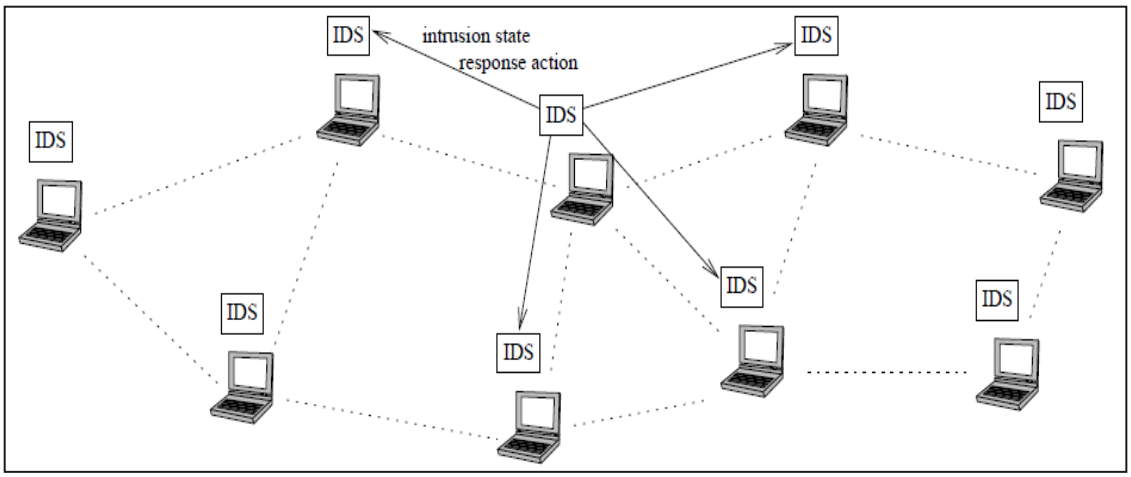

Figure 2.3 Architecture d'IDS distribué et coopératif

2.4. 3 Les IDS réparties en groupes

Les IDS réparties en groupe (Cluster-based IDS) ont été proposés afin de minimiser la surcharge du réseau et d'économiser l'énergie. Dans cette architecture, le réseau Ad hoc est divisé en un ensemble de groupes (clusters) ayant chacun un seul chef de groupe (Cluster-head) qui agit comme une petite station de base. Ainsi, la coopération est limitée entre le chef de groupe élu et chacun des membres du même groupe. Les activités malveillantes sont reportées au chef de groupe pour les analysés et détecté les intrusions. Parallèlement, tous les chefs de groupe peuvent coopérer pour former un IDS globale [29].

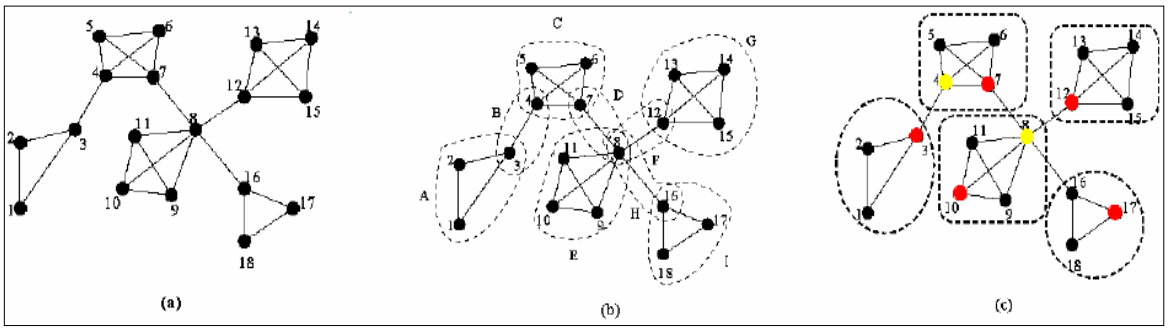

Figure 2.4 formation des clusters dans l'IDS en groupe

Toute fois, cette architecture est avantageuse que si les clusters sont stables pendant une longue période. Si les clusters sont régulièrement modifiés en raison d'itinéraires, la détection d'intrusion sera inefficace [30].

2.4. 4 Les IDS hiérarchiques

Les IDS hiérarchiques ont été proposées pour les réseaux sans fil multicouches. Cette architecture utilise le même principe que les IDS réparties en clusters pour la création des groupes avec chef de groupe, seulement les différents chefs de groupes ne coopèrent pas directement, mais forment un autre groupe d'un niveau supérieur avec un chef de groupe qui agit comme une station de base. Cette opération est répétée de façon hiérarchique jusqu'à atteindre la couche la plus supérieure représentée par une station de base centrale.

Cette architecture est souvent utilisée pour la détection d'intrusions dans les réseaux de capteurs ad hoc. Dans la figure 2.5 nous présentons l'architecture hiérarchique proposée par [30]

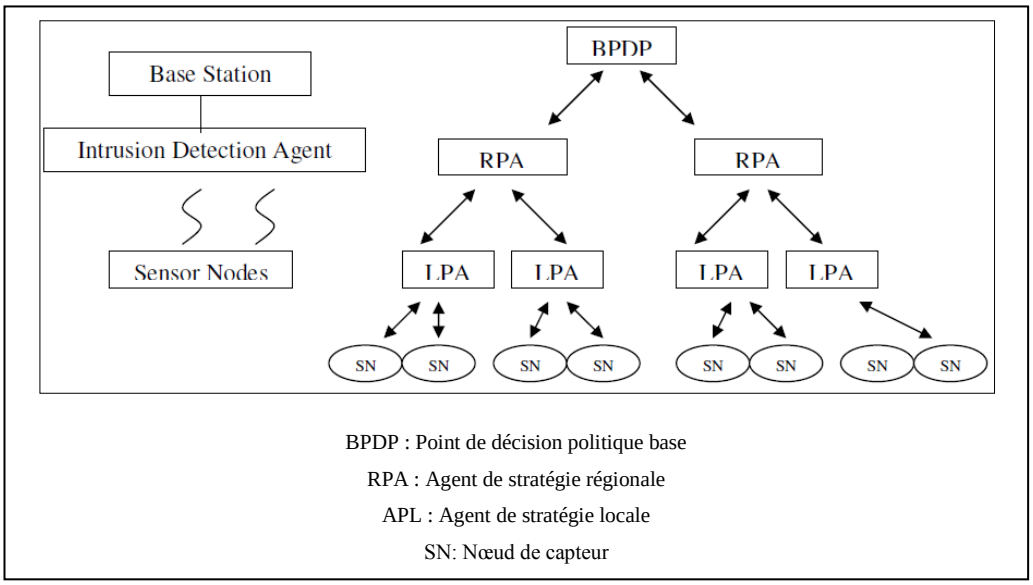

Figure 2.5: Architecture d'IDS Hiérarchique

2.5 Travaux antérieur sur les IDS Ad hoc

La détection d'intrusion dans les réseaux ad hoc a fait l'objet de plusieurs travaux de recherche ces dernières années. En effet, même avec une sécurité préventive, il est toujours possible que certains nœuds soient compromis. Dans ce cas, la détection d'intrusion peut être utilisée comme un autre niveau de défense et un élément essentiel pour un réseau ad hoc hautement sécurisé.

Zhang et Lee [31] ont proposé la première architecture d'IDS distribuée et coopérative. Dans cette architecture, chaque nœud dispos d'un agent IDS et tous les nœuds participent à la détection d'intrusion. L'agent IDS fonctionne indépendamment, collecte des traces d'audit local et les journaux d'activité, surveille les activités locales et communique avec les autres agents via un canal sécurisé. Il peut détecter les intrusions au niveau local et lancer des réponses. Si une anomalie est détectée ou si la preuve n'est pas concluante, les nœuds voisins peuvent participer collectivement à une action de détection d'intrusion globale. Par ailleurs, les critères explicites pour détecter les intrus et la coopération entre les nœuds voisins pour une détection au niveau globale ne sont pas spécifiés dans [31].

Huang et Lee [9] ont étendu leurs travaux antérieurs en proposant un IDS à base de cluster, afin de prendre en considération les contraintes de ressources auxquels les réseaux mobiles ad hoc sont confrontés. Ils ont utilisé l'ensemble de données prélevées des tables de routage et un algorithme de classification avec arbre de décision pour différencier un comportement normal d'un comportement anormal

Tseng et al [4] décrivent plusieurs attaques possibles dans la base de protocole AODV. Ils ont utilisé une machine d'états finis pour détecter des comportements anormaux et identifier les attaques. Ils suggèrent également l'utilisation d'un champ supplémentaire contentent le saut précédent pour déterminer le chemin source des messages de contrôle AODV.

Deng et al [32] ont proposé une approche hiérarchiquement distribuées et une autre approche complètement distribuée. La méthode de détection utilisée dans les deux architectures repose sur l'algorithme de classification SVM (Support Vector Machines). Les données d'audit sont issues de la couche réseau. Ils concluent qu'une approche hiérarchique distribuée peut être une solution plus prometteuse par rapport à une approche de détection d'intrusion complètement distribué.

Kachirski et Guha [33] proposent un système de détection d'intrusion à base de cluster en utilisant les technologies d'agents mobiles. Le système proposé utilise des agents mobiles effectuant chacun un rôle particulier. Les résultats de chaque nœud sont regroupés dans des points de cluster afin de limiter la tâche de contrôle de paquets sur quelques nœuds et minimiser le temps de traitement de l'IDS par chaque nœud.

Liu et al [34] ont proposé une méthode de détection d'anomalie entièrement distribuée. L'approche proposée sélectionne des fonctionnalités de la couche MAC au profil d'un comportement normal des nœuds mobiles, pour ensuite appliquer l'analyse d'écart [29] sur les vecteurs des signatures de fonctionnalités construites à partir de l'apprentissage.

Anjum et al [35] proposent une approche de détection d'intrusions basée sur les signatures d'attaque connus dans les réseaux ad hoc. Cette approche étudie la capacité des divers protocoles de routage pour faciliter la procédure de détection d'intrusion. Les auteurs concluent que le choix du système de détection dépend du protocole de routage que nous voulons utiliser.

Mitrokotsa et al [23] ont proposé un IDS complètement distribués pour les réseaux mobiles ad hoc. La méthode de détection basée sur le comportement, utilise l'ensemble de fonctionnalités de la couche MAC d'un comportement normale comme entrée aux réseaux de neurones Emergent SOM. Les données d'audit prélevé de la couche MAC sont ensuite utilisées pour effectuer la détection d'intrusions. L'approche proposée est également capable d'identifier la source d'attaque.

Plusieurs autres solutions ont été proposées pour la détection d'intrusions dans les réseaux Ad hoc que nous résumerons dans Le Tableau 2.1

Nom de l'IDS	Origines des données	Type de détection	Prétraitement des donnés	Attaque	Type de réponse	Année
A Cooperative Intrusion Detection System for Ad hoc Networks [9]	Couche Routage	Comportementale	Répartie en groupes	Blackhole, Routing Loop, Selfishness, Sleep Deprivation, Denial-of-Service	Passive	2003
Effective Intrusion Detection Using Multiple Sensors in Wireless Ad hoc Networks[33]	Système	Comportementale	Répartie en groupes	Denial-of-Service	Passive	2003
A Specification-based Intrusion Detection System for AODV[33]	Couche Routage /AODV	Scénarios	Distribué	Forged Sequence number, Forged Hop count, Tunneling attack	Passive	2003
A General Cooperative Intrusion Detection Architecture for MANETs[36]	Couche Routage /AODV	Scénarios	Hiérarchique	Packet Dropping Attacks	Active	2005
Intrusion Detection of Packet Dropping Attacks in Mobile Ad hoc Networks[23]	Couche MAC	Comportementale	Distribué	Packet Dropping Attacks	Passive	2006
Detecting Intrusion attacks in Ad hoc Networks[37]	Couche Routage /AODV	Scénarios	Distribué	Resource Consumption, Packet dropping, Fabrication attack	Active	2007
Power-Aware Hybrid Intrusion Detection System (PHIDS) using Cellular Automata in Wireless Ad hoc Networks[38]	Hybride/ HIDS+NIDS	Comportementale	Répartie en groupes	Denial-of-Service	Passive	2008
Hierarchical Design based Intrusion Detection System for Wireless Ad hoc Sensor Network[27]	Couche Routage/AODV	Scénarios	Hiérarchique	Denial-of-Service	Active	2010

Tableau 2.1 Solutions proposées pour la détection d'intrusions dans les réseaux Ad hoc

2.6 Conclusion

Dans ce chapitre nous avons introduit et présenté les composantes nécessaires à la détection d'intrusions dans les es réseaux mobiles Ad hoc. Après avoir situé le contexte spécifique de la détection d'intrusions dans les MANET et les caractéristiques des principaux IDS pour ces réseaux. Nous avons conclus que le choix de l'architecture de l'IDS pour un réseau mobile ad hoc dépend de l'origine des données d'audit et des types d'intrusions que nous voulons analyser. Le prochain chapitre a pour objet de présenter notre modèle d'IDS distribué et coopératif pour les réseaux sans fil ad hoc. L'analyse des modèles d'IDS distribués réalisée dans l'état de l'art nous permettra d'orienter la suite de nos travaux.

Chapitre III

Contribution à la Détection d’Intrusions dans les Réseaux Ad Hoc

3.1. Introduction

La majorité des techniques de détection d'intrusion proposée dans les réseaux mobiles ad hoc sont déployées dans la couche réseau. Dans ce chapitre, nous nous concentrons sur la détection d'intrusions dans les réseaux ad hoc au niveau de la couche MAC. En effet, Les vulnérabilités de cette couche peuvent être exploitées pour perturber les couches supérieures. Les exploitations de ces vulnérabilités sont appelées les attaques inter-couches. Dans [39] les auteurs proposent la définition suivante pour l'attaque inter-couches : c'est une attaque qui se focalise sur la couche MAC et dont l'impact se propage jusqu'aux autres couches supérieures. Certains chercheurs ont traité les vulnérabilités des paquets de contrôle RTS (Request to Send), CTS (Clear to Send) et ACK, et même de certaines attaques de type inter-couches, mais ils se sont focalisés sur des solutions de cryptographie, d'authentification ou l'addition des paquets de contrôle. Dans ce chapitre nous proposons de renforcer la sécurité par un système de détection d'intrusion pour contrer l'attaque par brouillage virtuel basé sur les faux paquets RTS.

3.2 Positionnement bibliographique

Dans cette section, nous présentons le mécanisme RTS/CTS de la technologie IEEE 802.11 et le problème du faux blocage.

3.2.1 Mécanisme RTS/CTS

Le mécanisme RTS/CTS est utilisé par le protocole MAC IEEE 802.11 dans le but d'éviter le problème bien connu des nœuds cachés [40]. L'idée de base du mécanisme RTS/CTS consiste en la transmission du paquet RTS par le nœud émetteur vers le nœud récepteur. Lorsque le nœud récepteur reçoit le paquet RTS, il va répondre en émettant un paquet CTS, dans le but d'informer tous les nœuds voisins qui vont recevoir ce paquet de la durée de transmission et d'éviter le problème des nœuds cachés. Comme illustré dans la figure 1(a), le nœud C reçoit le paquet RTS et bloque sa transmission par un certain NAV(RTS) [1].

$$NAV(RTS) = 3.SIFS + T_{CTS} + T_{DATA} + T_{ACK} \quad (1)$$

où TCTS, TDATA et TACK sont les temps de propagation des paquets CTS, DATA et ACK respectivement. Le temps *SIFS* est l'abréviation de « Short InterFrame Spacing » [2].

Il en est de même pour le nœud D, qui bloque sa transmission pendant un certain *NAV(CTS)*, une fois qu'il a reçu le paquet CTS. Le NAV(CTS) est calculé par le nœud B, fondé sur le NAV(RTS) contenu dans le paquet RTS.

$$NAV(CTS) = NAV(RTS) - (SIFS + TCTS) \quad (2)$$

L'émetteur A envoie le paquet DATA une fois qu'il a reçu le paquet CTS du nœud B. Le nœud B répond par un paquet d'acquittement (ACK) lorsqu'il reçoit correctement le paquet DATA, comme illustré dans la figure 3.1(b).

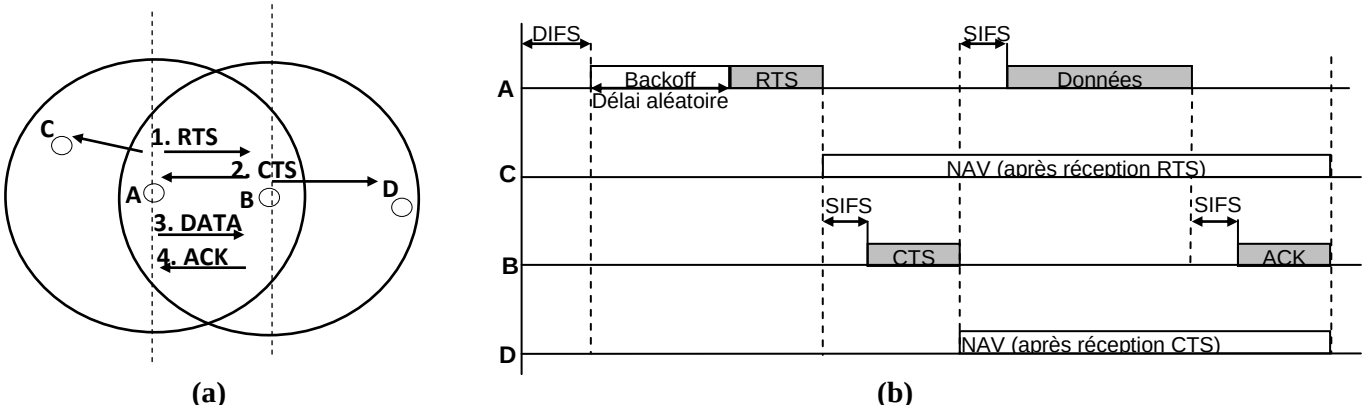

Figure.3. 1 Le mécanisme RTS/CTS dans le protocole MAC IEEE802.11

1 *NAV* (Network Allocator Vector), qui est la durée de transmission calculée par le nœud émetteur A
2 Dans IEEE 802.11, *SIFS* = 10*us*

3.2.2 Problème du faux blocage avec le mécanisme RTS/CTS

Le problème du faux blocage est introduit par Ray et Starobinski [15] [41]. Le faux blocage se produit lorsque le nœud récepteur du paquet RTS bloque sa transmission inutilement. La figure 3.2 illustre le problème du faux blocage avec le mécanisme RTS/CTS.

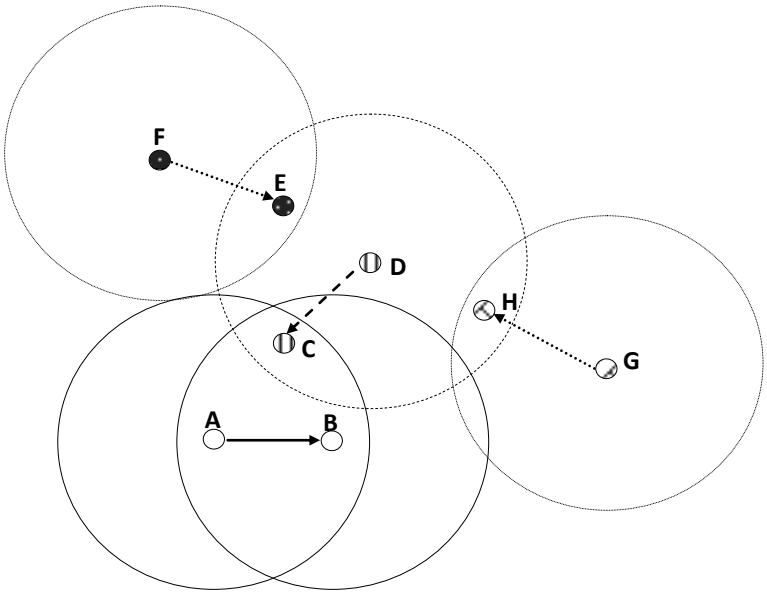

Figure 3. 2 Problème du faux blocage

Lorsque le nœud A veut communiquer avec le nœud B, il lui envoie le paquet RTS pour l'informer de la durée de transmission et réserver le canal de communication. N'importe quel nœud dans le voisinage du nœud A qui reçoit le paquet RTS va se bloquer pendant la durée NAV indiquée dans le paquet RTS, sauf le nœud B qui va répondre par un paquet CTS. Si le nœud D veut communiquer avec le nœud C ou si le nœud C veut assurer la bonne réception du paquet RTS émis par D, ce dernier ne peut pas répondre par un CTS car il est bloqué. Le problème avec ce scénario est que la situation de faux blocage risque de se propager et de s'étendre à d'autres nœuds, comme les nœuds E et H [39].

3.2.3 Brouillage virtuel

Le problème du faux blocage peut être exploité par un attaquant dans le but de créer une attaque de type déni de service dans le réseau. Rahman et Gburynski [40] ont traité sur le problème du faux blocage basé sur l'émission volontaire d'un faux paquet RTS, dont le but est de gêner et de retarder la transmission dans une partie du réseau.

Cette attaque est appelée brouillage virtuel (virtual jamming). Parmi ces solutions, nous citons la solution fondée sur l'addition d'un paquet de contrôle comme le paquet DATA-Send (*DS*) dans MACAW (Multiple Access with Collision Avoidance for Wireless) [41]. Ce paquet (*DS*) permet aux voisins de l'émetteur du paquet RTS de savoir que le paquet CTS a bien été reçu par l'émetteur du paquet RTS.

Une autre solution a été proposée par Ray et Starobinski [15] pour réduire l'impact de cette attaque, est appelée RTS validation. Cette solution a été améliorée dans [40]. L'idée principale de cette solution est basée sur la décision du nœud récepteur du paquet RTS avant que ce nœud ne bloque sa transmission, il doit écouter le canal et vérifier son statut, puis enfin décider de son blocage après un certain temps (RTS_Defer_Time). Une fois que le nœud reçoit le paquet RTS, il doit bloquer sa transmission pour un certain temps défini RTS_Defer_Time et ne pas se bloquer pour NAV(RTS) comme c'est le cas dans le protocole classique. Après RTS_Defer_Time, le nœud vérifie le statut du canal, puis il bloque sa transmission si le canal est occupé. Dans le cas où le canal est libre, il ignore le paquet RTS.

Dans [40], les auteurs ont proposé une extension de la solution «RTS validation » et l'ont appelée « validation aléatoire de RTS» (the random RTS validation). Le principe de cette solution est le même que celui de la précédente solution (simple RTS validation) : la différence entre les deux solutions se résume au niveau de la variable aléatoire RTS_Defer_Time sélectionnée dans un certain intervalle. Rachedi et Benslimane dans [39] ont définit de nouveaux algorithmes d'attaque qui exploitent les vulnérabilités des paquets CTS et ACK ; ils ont proposé des solutions basées sur l'authentification des paquets de contrôle en introduisons une version améliorée de la fonction de hachage appelée «Enhanced HMAC» (EHMAC).

Les travaux de recherche sur la détection d'attaque par brouillage virtuel sont très récent, on peut citer les travaux de Xiaocheng Zou and Jing Deng [42] pour la détection de faux paquet CTS, les auteurs ont proposé une détection basé sur le schéma d'inspection d'adresse (AIS : address inspection schema), dans cette solution chaque nœud envoi un message Hello a ses voisins a deux saut, afin d'enregistrer leurs adresses.

Quand un nœud reçoit un paquet CTS, il compare l'adresse récepteur du paquet avec les adresses enregistrées préalablement. Puis, demande le nœud destinataire s'il a effectivement envoyer un RTS qui précède l'envoi d'un CTS.

Ainsi que les travaux de Geethapriya Thamilarasu et al [43], publié en avril 2011, les auteurs se sont focalisés sur la détection d'attaque par brouillage au niveau des couches physiques et MAC. Leur analyse a montré que les attaques basées sur le brouillage génèrent un nombre important de collisions dans le réseau. Pour améliorer la précision de détection, ils ont développé un algorithme pour classer les collisions dues aux conditions du trafic réseau par rapport aux collisions causées par le brouillage.

3.2.3.1 Brouillage virtuel basé sur le faux RTS

Les faux paquets RTS peuvent être générés par un attaquant dans le but de créer une situation de blocage (brouillage virtuel). La figure 3 montre le cas classique d'une attaque via le faux RTS. Lorsque les nœuds (A, B) dans le rayon de transmission de l'attaquant (M) reçoivent le faux paquet RTS, ils sont inutilement bloqués pendant la durée de transmission présumée NAV(RTS), même sans la réception du paquet RTS (ces nœuds se considèrent comme des nœuds cachés). L'attaque de type faux RTS peut bloquer les nœuds qui se trouvent dans la portée de transmission de l'attaquant, mais aussi les nœuds qui se trouvent en dehors du rayon de transmission de l'attaquant. Les nœuds qui se trouvent dans le rayon d'interférence seront bloqués pendant EIFS (Extended Inter-Frame Space)[3] [1]. En effet, les nœuds qui sont en dehors du rayon de transmission ne sont pas capables de décoder ou de recevoir correctement les paquets. L'impact de cette attaque ne se limite pas seulement au rayon de transmission de l'attaquant mais aussi au rayon d'interférence.

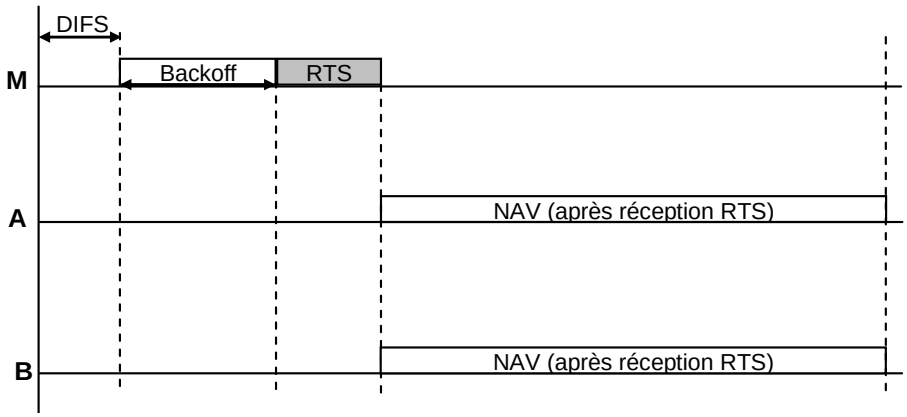

Figure 3.3 Attaque de type faux RTS.

[3] The EIFS est estimé à 364*ms* dans le cas d'une vitesse de transmission de 1 Mbps

Dans la figure 3.4, nous présentons sous forme d'un organigramme l'algorithme de l'attaque de type faut RTS. Cette attaque n'est prise en compte par le mécanisme RTS/CTS, car selon les caractéristiques des réseaux mobiles Ad hoc, même si le nœud bloqué par le NAV(RTS) ne reçoit pas le paquet CTS qui succède au faux paquet RTS, il ne sera pas en mesure d'en conclure si le paquet RTS est vrai ou faux. Le nœud bloqué peut être un voisin de l'émetteur du paquet RTS mais pas forcément de l'émetteur du paquet CTS.

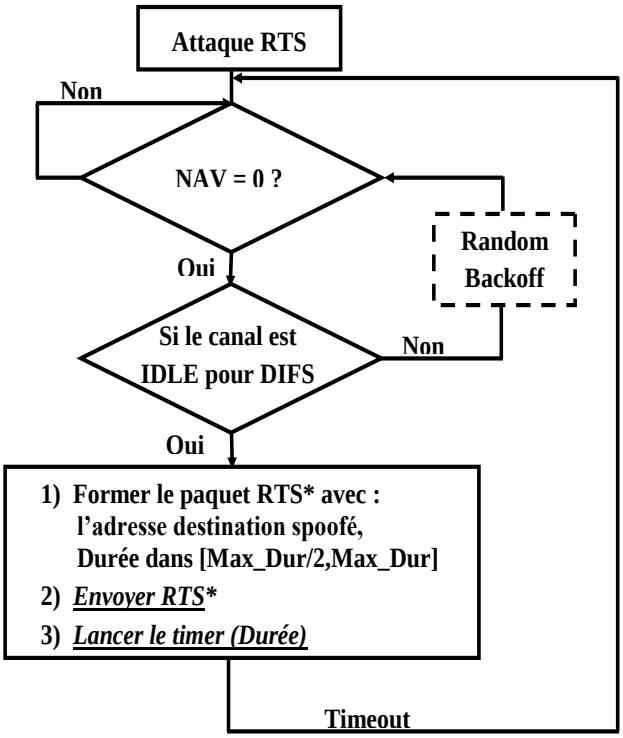

Figure 3.4 Organigramme d'une attaque de type faux RTS

3.2.3.2. Impact des attaques sur le mécanisme de surveillance

Le mécanisme de surveillance est développé pour contrôler l'activité des nœuds surveillés. L'attaquant avec le faux RTS peut facilement perturber le mécanisme de surveillance en se focalisant sur le nœud surveillé. Il peut ensuite le bloquer pendant une longue période pour l'empêcher de transmettre les paquets. Les paquets qui font l'objet du contrôle, et qui coopèrent dans le réseau, doivent participer à l'opération de routage des paquets, mais avec l'attaque répétitive de type faux RTS, le nœud sera bloqué et ne pourra pas transférer les paquets qu'il reçoit. Lorsque le nœud surveillant n'observe aucune transmission de paquet de la part du nœud surveillé, il va le classer comme un nœud égoïste qui ne veut pas coopérer dans le réseau [39].

Ainsi, il va réduire le niveau de confiance du nœud surveillé et, bien sûr, sa réputation va diminuer aussi. Le nœud surveillant peut détecter que le nœud surveillé est bloqué si le nœud attaquant est un voisin des deux nœuds (surveillé et surveillant).

3.3. Proposition d'Architecture Distribué et Coopérative

Nous proposons d'équiper chaque nœud du réseau d'un agent local de détection d'intrusions (LIDSA), qui sera responsable de la détection d'intrusions localement via l'audit des paquets RTS entrants et l'état du canal de communication après la réception de ces derniers. Les ressources réseaux ne sont utilisées que pour informer les autres nœuds d'une attaque détectée localement et, si nécessaire, pour collecter des informations complémentaires disponibles uniquement sur les autres nœuds du réseau. En outre, cette coopération entre les détecteurs d'intrusions locaux devrait être tenue par le biais de canaux sécurisés. L'ensemble de tous les agents de détection d'intrusions (LIDSA) forment le système de détection d'intrusion IDS pour le réseau sans fil ad hoc.

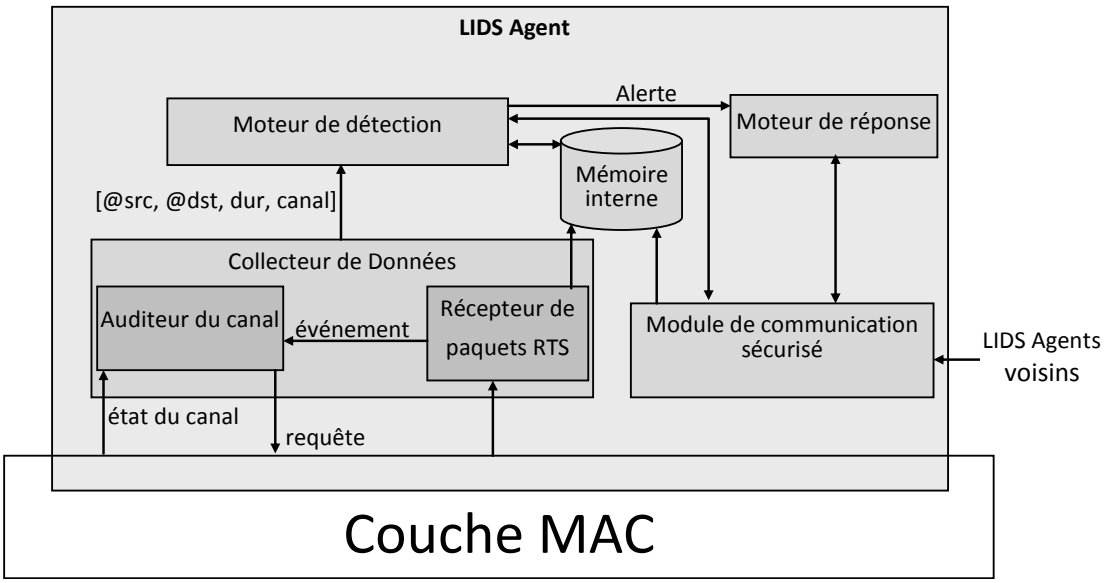

Figure 3.5 Architecture modulaire des LIDSA.

Les agents LIDSA sont responsables des fonctions suivantes :

- L'audit et l'analyse des paquets RTS entrants: l'agent est responsable de la détection locale d'anomalies en analysant les données d'audit sur les paquets RTS reçus et l'état du canal de communication après la réception de ces derniers.

- Journalisation des données: différentes informations sont enregistrées afin de faciliter la détection d'intrusions, comme la journalisation des cinq derniers paquets émis par le nœud, les informations concernent le dernier paquet RTS suspect reçu de chaque nœud voisin, et d'autres données relatives aux nœuds voisins (statut, degré de confiance …).
- Coopération avec les agents: une coopération avec les agents voisins est effectuée lorsque les preuves locales d'intrusion sont insuffisantes ou quand il s'agit d'isolé le nœud attaquant.

Notre agent LIDSA est constitué des cinq modules suivants :

1) **Collecteur de données:** ce module est chargé d'accueillir une copie des paquets RTS reçus en temps réel, filtre les donnée du paquet RTS, lance son timer pour une durée égale a $2 \times SIFS + T_{CTS}$, puis écoute le canal de communication et vérifie son statut (canal libre ou occuper). Enfin, transmet les informations traitées au moteur de détection.

2) **Moteur de détection:** analyse les données locales recueillies par le module de collecte de données, il réalise les procédures suivantes :

- Si le canal est libre, alors il vérifie si la durée de transmission indiquée dans le paquet RTS est supérieure à MAX_DUR/3.
    - Si la vérification est positive, il vérifie si le dernier RTS émis par ce nœud est suspect, si c'est le cas, il envoie un message d'alerte au moteur de Réponse. Sinon il met à jour le statut du nœud comme suspect, puis envoi un message de confirmation au nœud suspect.
    - Sinon si la vérification est négative et les signes d'intrusions nécessitent d'êtres confirmés, une procédure de coopération est lancée afin que l'agent LIDSA concerné confirme le paquet RTS supposé envoyer par son nœud, et les nœuds voisins envoient des informations complémentaires sur le nœud suspect. Si l'agent ne donne pas de réponse et les informations collectées sur le nœud sont négatives ; il est considéré compromis. Un message d'alerte est alors envoyé au moteur de Réponse.

- En fin, si le canal est libre, les informations du paquet RTS reçu sont journalisés.

3) **Mémoire interne:** est une mémoire dynamique pour contenir les adresses MAC des nœuds voisins et leurs statuts, la journalisation des paquets RTS envoyés et reçus.

4) **Module de réponse***:* après la détection d'intrusion et l'identification des nœuds compromis, les paquets reçus par ces nœuds seront ignorés et une notification est envoyée à l'utilisateur lui indiquant qu'une attaque a été détectée.

5) **Module de communication sécurisé**: assure une communication sécurisée entre les agents LIDSA.

3 .4 Implémentation de l'Agent LIDSA dans le noyau de NS

NS[44] est un simulateur à événements discrets orienté objet. Il est écrit en C++ avec une interface textuelle (ou shell) qui utilise le langage OTcl (Object Tool Command Langage). L'OTcl est une extension objet au langage de commande Tcl. Le langage C++ sert à décrire le fonctionnement interne des composants de la simulation. En effet, NS bénéficie de toutes les possibilités qu'offrent les techniques objets comme l'héritage, le polymorphisme, la surcharge, etc. La figure 3.6 représente l'arborescence de classes utilisée par le simulateur ainsi que l'implémentation de notre classe LIDSA qui hérite directement de la classe NsObject

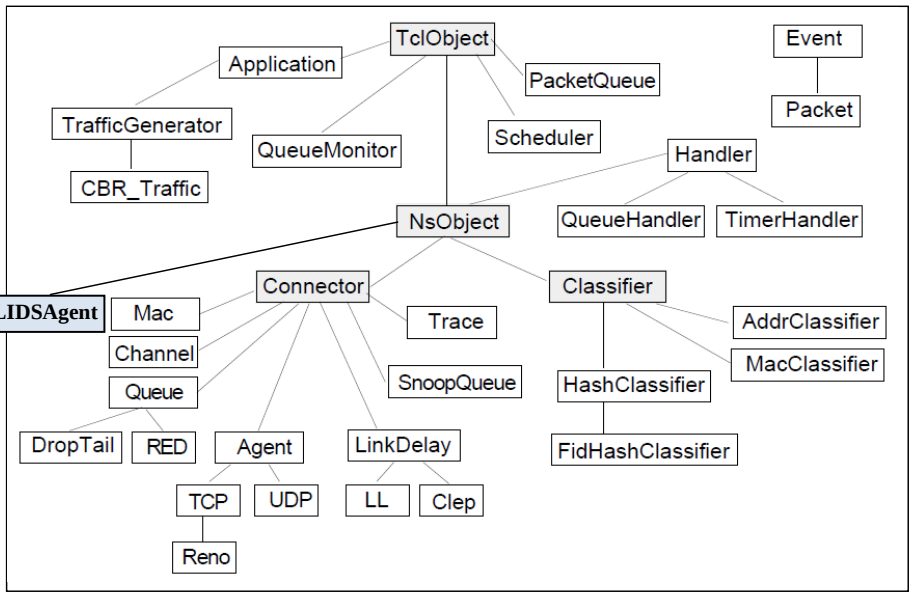

Figure 3.6 Positionnement de la classe LIDSA dans l'arborescence de classes de NS

La classe LIDSAgent est définie comme une sous-classe de NsObject

```
class  LIDSAgent : public NsObject {
DEFCLASS(LIDSAgent) // Macro liee au nommage des LIDS agents
friend ostream& operator<<(ostream& os, const LIDSAgent& anLIDSAgent);
public:
//=================== Allocateurs/Desallocateurs ==================//
       LIDSAgent(void);
       LIDSAgent(const LIDSAgent& anLIDSAgent);
       LIDSAgent& operator=(const LIDSAgent& anLIDSAgent);
       virtual ~LIDSAgent(void);
.
.
.
//======================== HANDLER ==============================//
       void RunLIDSAHandler(void);
       void LIDSAHandler(void);
       void BRDLIDSAHandler(void);
//==================== Collecteur de données ====================//
       void    CollecteurDonnee(void);
       void    RecepteurRTS(Packet *p);
       void    AuditeurCanal(void);
//===================== Mémoire interne =========================//
       void    AjoutNoeudList(int num);
       void    SuppressionNoeudList(int num);
       Void    MiseaJourMemoire(int num, int Nouveau_status);
       int     ExisteNoeud(int num);
       int     StatutNoeud(int num);
       int     NoeudMalicieux(int num);
 //========================= Journal RTS ========================//
       void    MiseaJourJournal(int dst, int duration);
       void    MiseaJourSignatures(int num, double Nouveau_Temps, double Duration, int
       Dst);
       double  TempsRTS(int num);
       double  DurationRTS(int num);
       int     DestinationRTS(int num);
//====================== Moteur de détection ====================//
       void    MoteurDetection(MacState EtatCanale, int Src, int Dst, double Duration);

//======================= Module de réponse =====================//

       void    ModuleRepense(int adresse);

//=============== Module de communication sécurisé ==============//
       struct          Message_Infos {
       int             Noeud_id;
       u_int16_t       MsgSeq;
       };
 struct Message_Infos *AuthfMessage_list;
 std::list<struct Message_Infos> AuthfMessage_list1;
//================ opération sur la liste des messages ==========//
       void    AjoutAuthfMessageList(int num, u_int16_t NumSeq);
       int     ExisteAuthfMessage(int num, u_int16_t NumSeq);
//================= Procédures d'envoi et de réception ==========//
       Int     AuthfSeqno();
       void    EnvoiAutentif();
       void    ReceptionAutentif(Packet *p);
       void    EnvoiConfirmation(int dst);
       void    ReceptionConfirmation(int src);
       void    ReponseConfirmation(int dst);
       void    ConfirmationNegative();
       void    ConfirmationPositive();
       void    EnvoiAlerte(int adresse);
       void    ReceptionAlerte(Packet *p);
       void    sendRLIDSA(int dst);
       void    sendCLIDSA(int dst);
       int     check_pktCLIDSA();
       int     check_pktRLIDSA();
       void    recvRLIDSA(Packet *p);
       void    recvCLIDSA(Packet *p);
 //=============================================================//
```

```
private:
        BRDLIDSATimer   BRDLIDSA_;
        RunLIDSATimer   RunLIDSA_;
        LIDSATimer      IdsTimer;
       Packet          *pktCLIDSA_;
       Packet          *pktRLIDSA_;
      u_int16_t         AuthfMsgSeq;
//======================= Mémoire interne ========================//
       struct          Noeud_Infos {
       int             Noeud_id;
       int             Noeud_Status;
       };
       struct Noeud_Infos *Noeud_list;
       std::list<struct Noeud_Infos> Noeud_list1;
//================== Signatures en attente ======================//
       struct  Signature_Infos {
       int     Noeud_id;
       double  Precedent_RTS;
       double Duration_RTS;
       int     Destination_RTS;
       };
       struct Signature_Infos *Signature_list;
       std::list<struct Signature_Infos> Signature_list1;

//===============================================================//
       int     DestAdrRTS;
       int     SrcAdrRTS;
       double Duration_RTS;
       double DurRcpRTS;
       int     ActiveLIDSA;
       int     ActiveEcoute;
       MacState EtatCanaleRTS;
       Int     PktConfirmation;
...
protected:
// Methodes a appeler par une classe derivee
// Methode qui doit etre appelee dans le consvoid
newLIDSAAgent(void); // tructeur d'une classe derivee
…
}
```

La classe LIDSAgent définit RecepteurRTS(Packet *p). Cette fonction est appelée quand un paquet RTS est reçu par le nœud, qu'il soit destinataire de ce paquet ou non.

```
void LIDSAgent::RecepteurRTS(Packet *p)
{
       if (IdsTimer.busy() != 0) return;
       double timeout;
       u_int16_t us;
.
.
.
struct rts_frame *rf = (struct rts_frame*)p->access(hdr_mac::offset_);
        EtatCanaleRTS = MAC_IDLE;
        DestAdrRTS = ETHER_ADDR(rf->rf_ra);
        SrcAdrRTS = ETHER_ADDR(rf->rf_ta);
          us = rf->rf_duration;
          Duration_RTS = us * 1e-6;
        timeout = (2 * phymib_.getSIFS())+ txtime(phymib_.getCTSlen(), basicRate_);
        IdsTimer.start(timeout);
}
```

IdsTimer est un temporisateur, a la fin du timeout, la fonction AuditeurCanal va être appelée.

```
LIDSAgent::LIDSAengtHandler()
{
 AuditeurCanale();
}
```

Plusieurs procédures sont utilisées pour la mise à jour, la consultation ou la recherche d'information dans la mémoire interne.

```
//======================= Mémoire interne ========================//
void LIDSAgent::AjoutNoeudList(int num)
{
  Noeud_list = (struct Noeud_Infos*)malloc(sizeof(struct Noeud_Infos));
  Noeud_list->Noeud_id=num;
  Noeud_list->Noeud_Status=0;
   Noeud_list1.push_front(*Noeud_list);
  free(Noeud_list);
}
void LIDSAgent::MiseaJourMemoire(int num, int Nouveau_status)
{
 std::list<Noeud_Infos>::iterator it;
 for (it=Noeud_list1.begin(); it != Noeud_list1.end(); it++) {
  if ((*it).Noeud_id == num) {
   (*it).Noeud_Status = Nouveau_Status;
   break;
  }
 }
  if (it == Noeud_list1.end()) {
  Noeud_list = (struct Noeud_Infos*)malloc(sizeof(struct Noeud_Infos));
  Noeud_list->Noeud_id=num;
  Noeud_list->Noeud_Status=Nouveau_Status;
  Noeud_list1.push_front(*Noeud_list);
  free(Noeud_list);
 }
}
int LIDSAgent::StatutNoeud(int num)
{
 std::list<Noeud_Infos>::iterator it;
 for (it=Noeud_list1.begin(); it != Noeud_list1.end(); it++) {
  if ((*it).Noeud_id == num) {
   return (*it).Noeud_Status;
  }
 }
 return -1;
}
int LIDSAgent::ExisteNoeud(int num)
{
 std::list<Noeud_Infos>::iterator it;
 for (it=Noeud_list1.begin(); it != Noeud_list1.end(); it++) {
  if ((*it).Noeud_id == num) {
   return 1;
  }
 }
 return 0;
}
int LIDSAgent::NoeudMalicieux(int num)
{
 std::list<Noeud_Infos>::iterator it;
 for (it=Noeud_list1.begin(); it != Noeud_list1.end(); it++) {
  if ((*it).Noeud_id == num) {
                  if ((*it).Noeud_Status == 1) {
   return 1;
                        } else return 0;
  }
 }
 return 0;
}
```

Le module de réponse est appelé par le moteur de détection dans le cas de découverte de nœud malicieux localement ou à travers des messages d'alertes des LIDSA voisins

```
void LIDSAgent::ModuleRepense(int adresse)
{
NoeudDetecte++;
printf ("Temps = %2.9f ||------------------ >Le noeud: %d à détecté une attaque par faux
paquet RTS en provenance du noeud  %d\n",NOW(),idNoeud, adresse);
           MiseaJourMemoire(adresse, 1);
         if (DetectionLocal)
           EnvoiAlerte(adresse);
}
```

3.5 Résultats de simulations

3 .5 .1 Evaluation de l'impact des attaques

Nous avons implémenté l'attaque de brouillage virtuel par faut paquet RTS sous NS2 [44] et nous avons simulé plusieurs scénarios dans différentes situations.

Tout d'abord, nous simulons un scénario classique sur la topologie réseau illustrée dans la figure 3.7. Dans ce scénario, nous avons deux ensembles de nœuds, *S*1 = {0, 1, 2, 3} et S2 = {4, 5, 6, 7, 8}. Dans chaque ensemble, les nœuds peuvent communiquer directement avec les nœuds du même ensemble, mais les nœuds de l'ensemble *S*2 ne peuvent pas atteindre les nœuds de l'ensemble *S*1. Dans les deux ensembles, nous avons deux connexions de type CBR (la taille des paquets est de 1000 bytes et le débit est de 50 paquets/seconde).

Dans l'ensemble *S*1, il n'y a aucun nœud malicieux, mais dans l'ensemble *S*2 le nœud 8 est un nœud malicieux capable d'attaquer par de faux *RTS*.

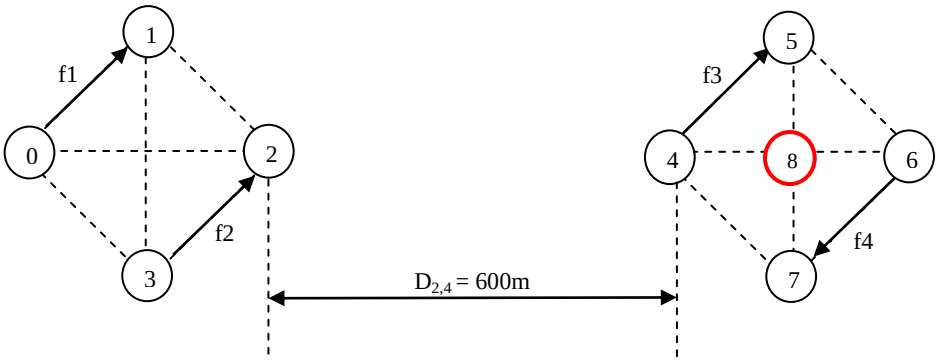

Figure 3.7 Topologie réseau

Nous choisissons le débit comme métrique pour montrer l'impact de ces attaques sur le réseau. Dans la figure 3.8, nous traçons le débit moyen du réseau en fonction du temps de simulation. Dans le cas de l'attaque de type faux RTS, nous remarquons une différence significative entre le débit obtenu sans et avec un attaquant dans le réseau.

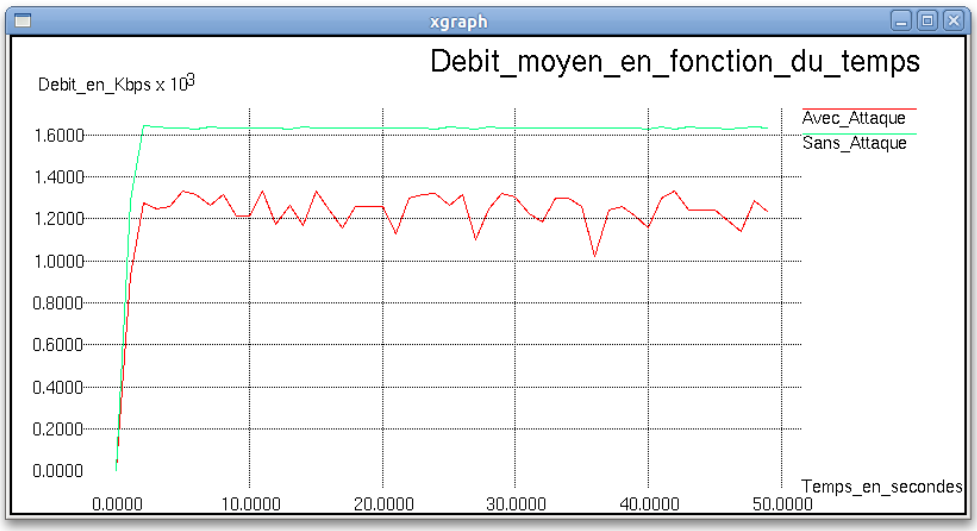

Figure 3.8 Débit moyen du réseau en fonction du temps

Dans le but de montrer l'impact des faux RTS au niveau de chaque récepteur et de prouver les résultats d'analyses précédents, nous traçons les résultats obtenus dans la figure 3.9. Nous remarquons que les nœuds récepteurs 1 et 3 de l'ensemble *S*1 ont un débit plus ou moins stable, contrairement aux nœuds 5 et 7 de l'ensemble *S*2 : leur débit est faible et instable dans le cas d'une attaque avec le faux RTS, car les nœuds 5 et 7 sont situés dans la zone de communication du nœud malicieux 8, ce qui n'est pas le cas des nœuds 1 et 3.

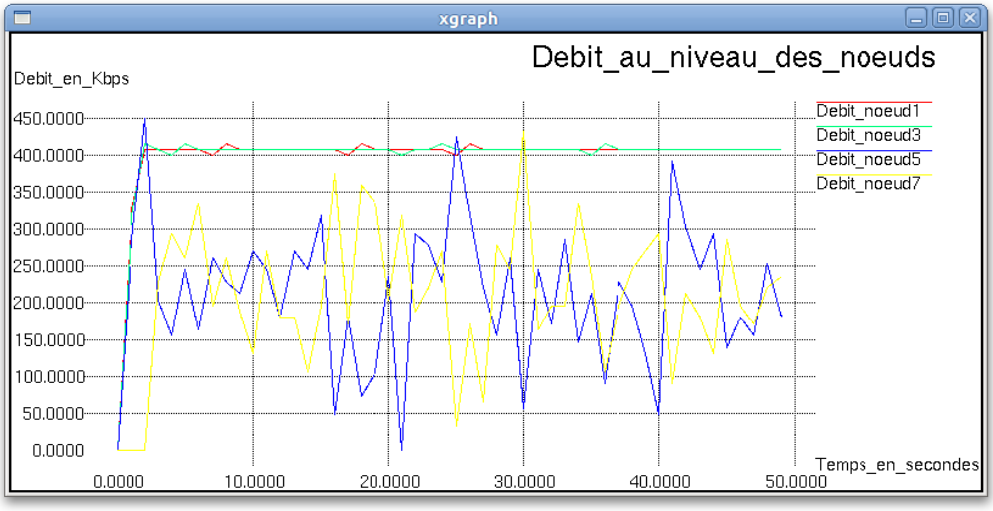

Figure 3.9 Débit moyen chez les nœuds récepteurs en fonction du temps

3.5.2 Evaluation de la performance de notre solution

Dans le but d'évaluer l'impact de la solution proposée sur le réseau, nous avons implémenté l'agent LIDSA dans le noyau du simulateur réseau NS2 [44] et dans différentes situations. Nous utilisons le mécanisme classique RTS/CTS comme référence pour le comparer avec la solution proposée.

Sur la base du même scénario précédent, nous avons intégré l'agent LIDSA dans les nœuds 0 à 7, mais en activant la détection qu'a partir de la 20ème secondes. Le nœud 8 (l'attaquant) commence à envoyer les faux paquets RTS dé la 10ème secondes. Nous avons ensuite tracé les résultats dans la figure 3.10.

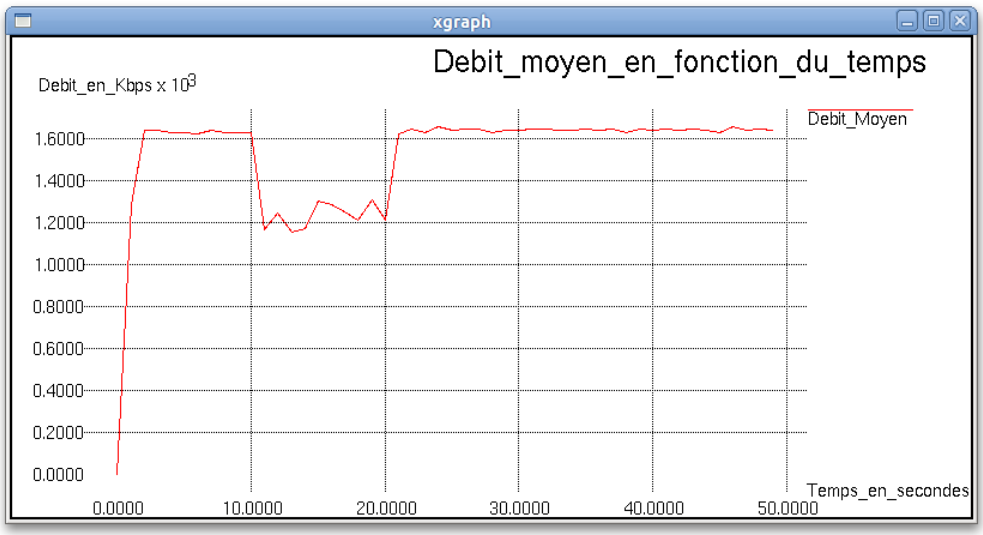

Figure 3.10 Débit moyen en fonction du temps (attaque faux RTS + Agent LIDSA)

Nous constatons que le débit moyen revient à son niveau normal après la 20ème secondes, ce qui correspond au moment d'activation du mécanisme de détection et d'isolement de l'attaquant.

Afin de mesurer le temps de réponse de notre solution sur chaque nœud, nous avons refait la même expérience en intégrant l'agent LIDSA avec l'option « actif » (L'agent est activé dans les nœuds par défaut). Le nœud malicieux 8 déclenche l'attaque à la 10ème secondes. Les résultats du débit au niveau des nœuds récepteur sont représentés dans la figure 3.11. Nous remarquons que le débit est stable tout au long de la simulation avec une petite perturbation insignifiante d'environ une seconde après le lancement de l'attaque.

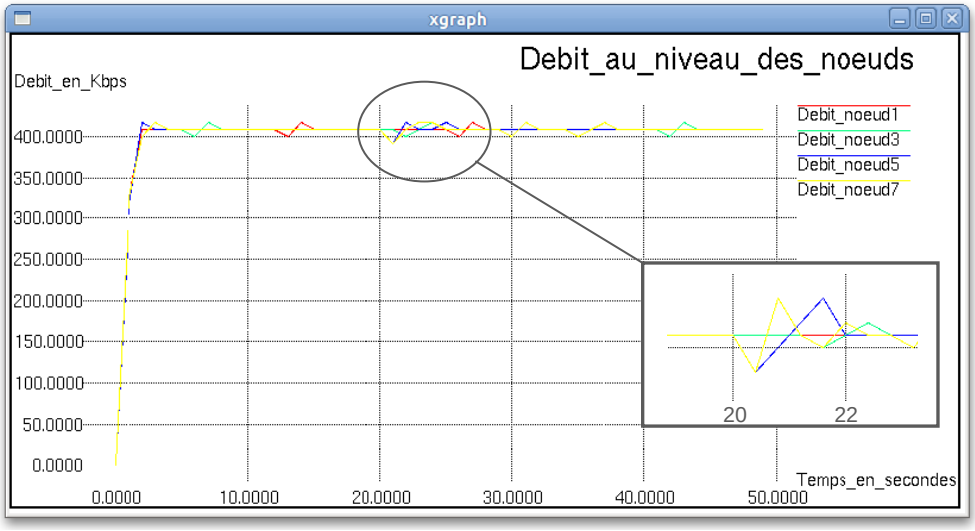

Figure 3.11 Débit moyen au niveau des nœuds récepteur (Faux RTS+ LIDSA)

3.5.3 Etude comparative

Dans le but de positionné notre solution avec les travaux existants dans la littérature, nous avons choisie de comparer notre solution avec la solution de Ashikur Rahman et Pawel Gburzynski [40] nommé « random RTS validation » proposée dans IEEE, 23rd Biennial Symposium on Communications. Les auteurs ont mis en place sous NS2 [44] une topologie typique représentée dans la figure 3.12(a) que nous avons repris dans la figure 3.12(b). Dans ce scénario, les nœuds intérieurs 1à 4 forment un ensemble. Chacun des nœuds extérieurs 8 à 5 est accessible uniquement à partir de sont voisin interne. Le trafic se compose de trois flux CBR (paquets de 1024 octets): 2→6, 3→7 et 4→8. Le taux de transmission est 1Mbps pour les trois sources.

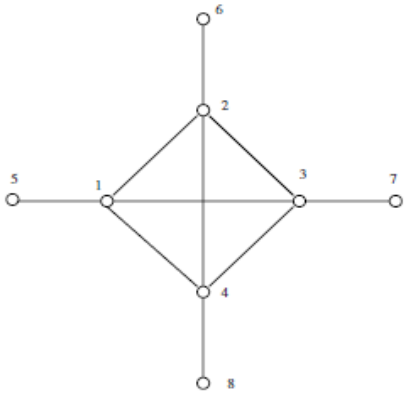

Figure 3.12(a) Scénario de A.Rahman

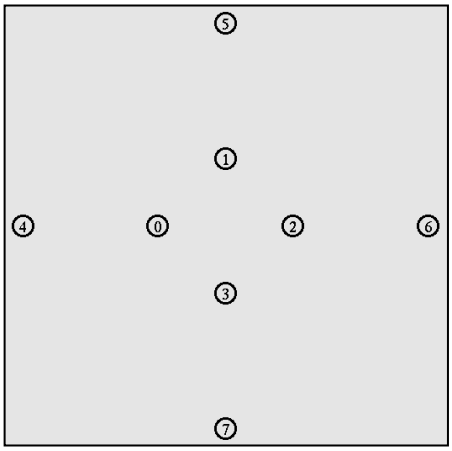

Figure 3.12(b) Notre scénario

Pour la raison de la limite de notre matérielle, nous ne pouvons pas utiliser le même temps de simulation que dans [40], qui est de 1000s, le temps de notre expérimentation est fixé à 100s.

La figure 3.13(a) et 3.13(b) montre respectivement le débit moyen obtenu dans [40] et notre débit moyen obtenue pour la même topologie.

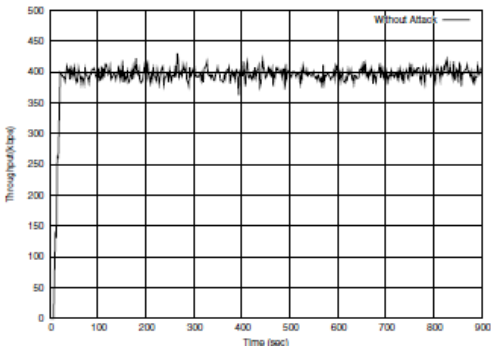

Figure 3.13(a) Modèle de A.Rahman : Débit moyen / Sans attaque

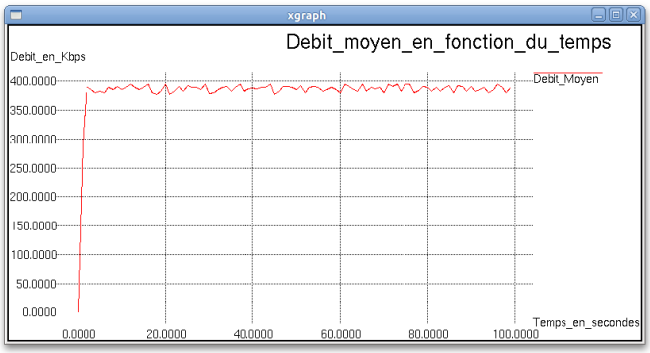

Figure 3.13(b) Notre modèle : Débit moyen / Sans attaque

Dans le second scénario élaboré par A.Rahman (Figure 3.14), le nœud 1 (l'attaquant) envoie de faux paquets RTS au nœud 5. Ces paquets RTS sont reçus par les nœuds 2, 3 et 4. L'attaque commence à 150 secondes et se poursuit jusqu'à 600 secondes dans l'expérience effectuée par [40]. Dans notre expérience l'attaque commence a 15 secondes et ce poursuit jusqu'à la fin de la simulation.

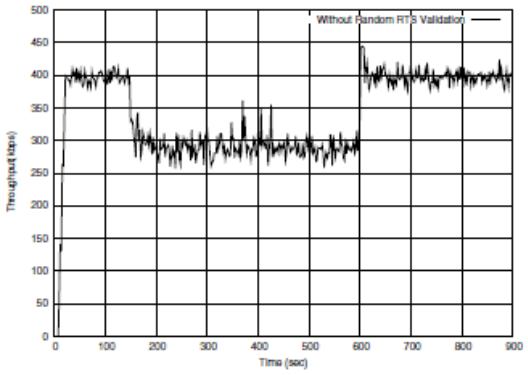

Figure 3.14(a) Modèle de A.Rahman :Débit moyen + Attaque

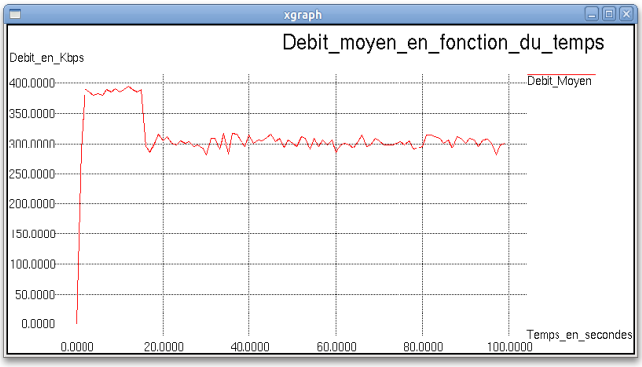

Figure 3.14(b) Notre modèle : Débit moyen + Attaque

Les figures 3.13 et 3.14, montre que le débit et l'impact de l'attaque dans notre scénario sont très similaire au du débit et de l'impact de l'attaque du scénario utiliser dans [40]. Nous avons donc un environnement ou les deux solutions peuvent être comparées.

La figure 3.15(a) se réfère au troisième scénario réalisé par [40]. Le nœud 1 lance l'attaque par faux paquet RTS (entre 150 et 600 secondes), avec l'implémentation de la solution « random RTS validation ». Dans la figure 3.15(b) le même scénario est repris avec l'implémentation de notre solution (Agent LIDSA).

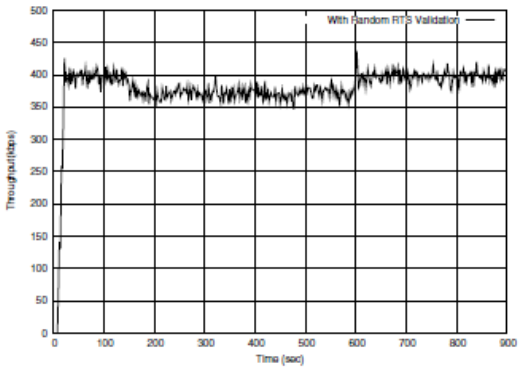

Figure 3.15(a)Faux RTS avec Rondom RTS Validation

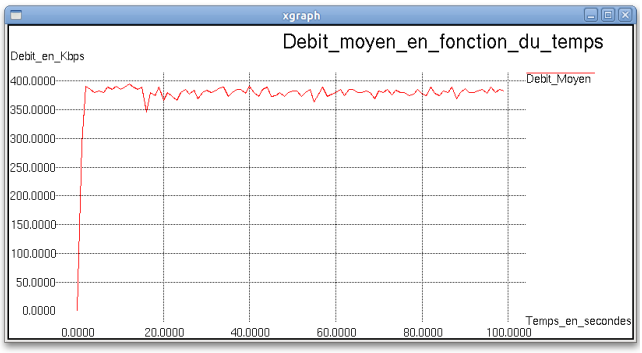

Figure 3.15(b) Faux RTS et solution avec agent LIDSA

Bien que l'impacte de l'attaque soit presque insignifiant après l'application de « random RTS validation », en remarque qu'il existe une légère perte du débit tout au long de l'attaque. Dans les cas de notre approche, la perte du débit est perceptible seulement entre le lancement de l'attaque et la détection de l'attaque (environ une seconde dans le cas présent), ensuite le débit reprend sont taux normale. Cela s'explique par le faite que la solution « random RTS validation », traite l'attaque, mais pas l'attaquant, alors que dans notre solution nous traitons l'attaque et nous agissons pour contrer l'attaquant (tous les paquets en provenance de l'attaquant sont ignorés par les nœuds du réseau).

3.5.4 Evaluation de performance dans le cas général

Dans le but d'étudier l'impact de ces attaques dans le cas général, nous avons simulé un réseau de 60 nœuds dans une topologie en deux Grid superposé, le premier de type 6 x *6*, la distance entre les nœuds est fixée à 225 mètres, le deuxième Grid de type 5 x 5 pour contenir uniquement les nœuds malicieux, la même distance entre les nœuds et respecter. Chaque nœud malicieux ce trouve exactement au milieu de quatre nœud du premier Grid. Le rayon de la portée de transmission est fixé à 250 mètres. Pour le trafic réseau, nous utilisons Cinque connexions de type CBR (0→35, 30→5, 11→6, 12→17, et 23→18) avec des paquets de taille égale à 512 bytes pendant une durée de simulation de 10 secondes.

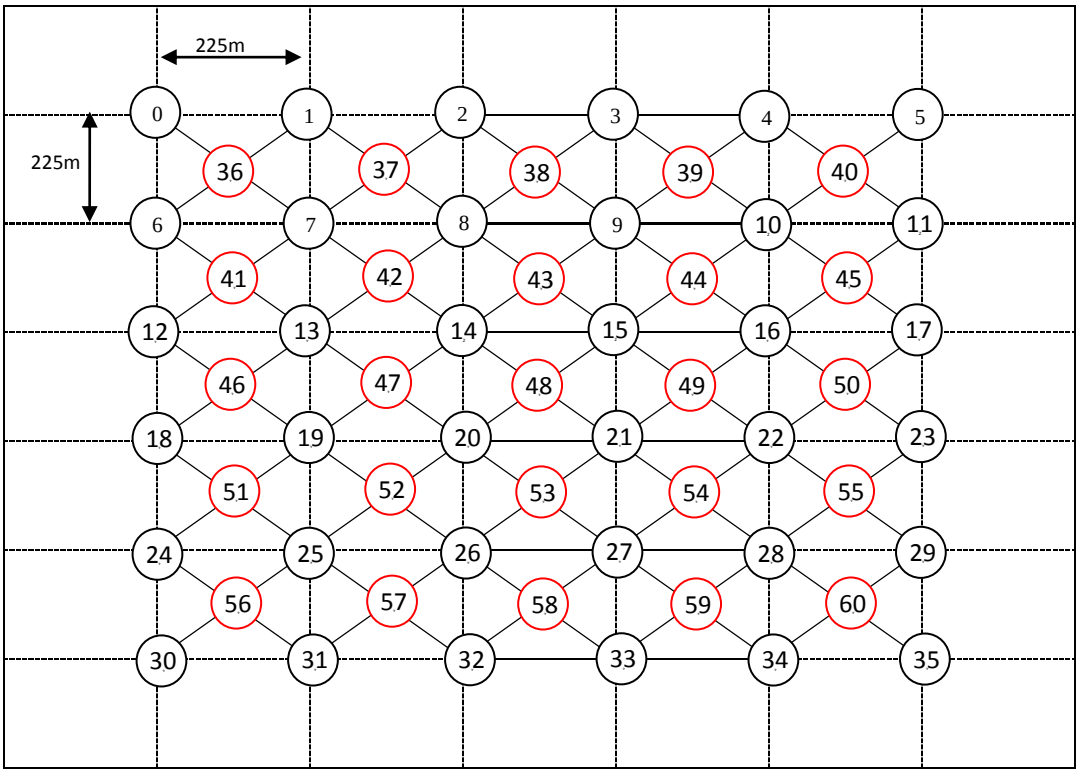

Figure 3.16 Topologie de type Grid

Pour évaluer les performances de nos solutions, nous intéressons aux métriques suivantes :

- Le débit moyen.
- Le nombre de collisions des paquets.

La figure 3.17 illustre l'impact du nombre des nœuds attaquants dans le réseau (de 0 à 25, soit 0 à 41%) avec 10 secondes comme durée de simulation.

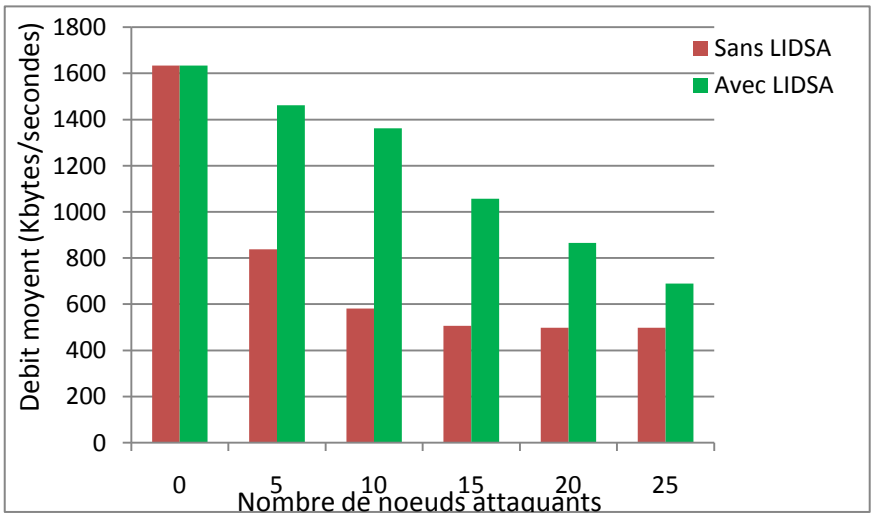

Figure 3.17 Impact du nombre d'attaquants sur le débit du réseau

Nous remarquons que lorsque le nombre de nœuds malicieux augmente, le débit du réseau diminue rapidement. Par ailleurs, en absences de nœuds malicieux, le débit sans et avec notre solution reste inchangé, cela implique que notre solution n'est pas coûteuse en termes de débit. De plus, le débit avec l'implémentation des agents LIDSA est toujours supérieur comparer au débit sans notre solution en cas d'attaque.

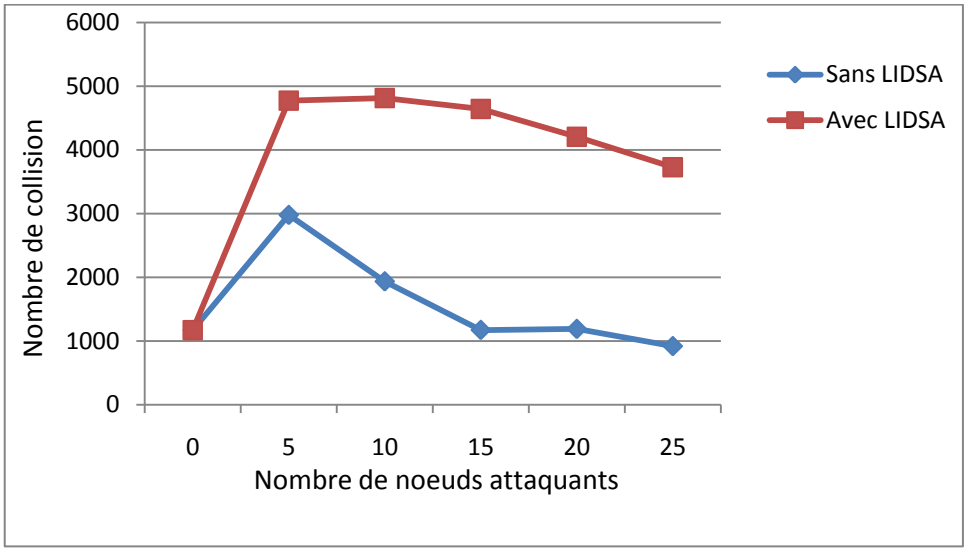

Figure 3.18 Nombre de collisions générées

La figure 3.18 montre le nombre de collisions dans le réseau avec le protocole RTS/CTS (sans implémentation de l'agent LIDSA), et le nombre de collisions obtenue en intégrant l'agent LIDSA.

Nous remarquons que le nombre de collisions augmente dans le cas de 5 attaquant pour le protocole RTS/CTS sans LIDSA, ensuite il commence à diminuer avec l'augmentation des nœuds attaquant. Cela est causé par l'envoi répétitif des faux paquets RTS de la part des attaquants. En effet, avec 5 attaquants dans le réseau, le canal n'est pas considérer totalement occupé par les nœuds émetteurs, les paquets envoyés par les nœuds peuvent entrer en collision avec les faux paquets RTS envoyés par les nœuds malicieux.

Avec 15 nœuds malicieux occupant les deux diagonales et la colonne verticale de notre topologie, le canal de communication est considéré presque tout le temps occupé par les nœuds émetteurs, d'où la baisse d'émission ce qui diminue le nombre des collisions. Dans le cas ou on implémente l'agent LIDSA, les nœuds ignorent toute demande de réservation du canal de communication par les attaquants en continuant l'échanger des paquets.

Parallèlement, les attaquants envoient indéfiniment les faux paquets RTS, ce qui engendre un nombre important de collisions.

3.6 Conclusion

Dans ce chapitre, nous avons montré les vulnérabilités au niveau de la couche MAC de la technologie IEEE 802.11 et les attaques susceptibles d'exploiter ces faiblesses. Nous avons illustré l'impact négatif de cette attaque via des simulations. Un autre impact négatif sur les mécanismes de surveillance est étudié. Un attaquant peut facilement réduire la réputation ou le niveau de confiance d'un nœud qui se comporte bien. De plus, l'opération de routage peut être affectée par cette attaque. Pour contrer ces attaques, nous avons proposé un système de détection d'intrusion distribué et coopératif, pour le traitement de l'attaque de type faux RTS, la détection et l'isolement de l'attaquant afin d'éviter le phénomène du faux blocage qui peut être engendré.

Conclusions Générale et Perspectives

Notre mémoire a pour objectif d'apporter des solutions de détection d'intrusions dans les réseaux mobiles Ad hoc. Les caractéristiques de ces réseaux ne permettent pas l'utilisation des solutions de détection déjà existantes.

Dans un premier temps, nous nous sommes intéressés au problème lié à la couche MAC (Medium Access Control), de la technologie IEEE 802.11. Nous avons montré les vulnérabilités basées sur les paquets de contrôle *RTS* (Request to Send)) qu'un nœud malveillant peut exploiter dans le but de perturber les mécanismes de surveillance et de routage. Notre approche basée sur la coopération entre agents, utilise les traces des paquets RTS comme donnée d'audit, si ces derniers répondent au critère d'une signature de l'attaque par faux paquet RTS, une procédure de coopération est envoyée aux nœuds voisins pour détecter la source de l'attaque et y remédier.

Nous avons étudié l'impact de ces attaques sur des topologies de réseau Ad hoc avec différents scénarios. Après simulation, les résultats montrent l'efficacité de notre approche en matière de détection. De plus, le temps de réponse de la solution proposée est étudié. En comparaison avec l'impact négatif des attaques, ce temps est insignifiant.

Toutefois, dans la continuité du travail présenté, nous pouvons étendre notre architecture pour traiter les différentes attaques basées sur le brouillage virtuel au niveau de la couche mac. A cet effet, nous avons l'intention d'augmenter notre solution pour supporter la détection des attaques de type faux paquet CTS et faux paquet ACK.

Enfin, l'approche de détection d'intrusions proposée dans ce présent mémoire peut être étendue et adaptée aux réseaux de capteurs.

Bibliographie

Bibliographie

[1] I.EEE "***Institute of Electrical and Electronics Engineers***" http://standards.ieee.org/getieee802/802.11.htm.

[2] OSI, "***International Standards Organization. Information processing systems***"- osi reference model - part 2: Security architecture. Technical Report7498-2, 1989.

[3] J.M.Percher " ***Un modèle de détection d'intrusion distribuée pour les réseaux sans fil ad Hoc***", thèse de doctorat université de Versailles, 2004

[4] T.seng, P.Balasubramanyan, C.Ko, R.Limprasittiporn, J.Rowe, and K.Levitt,"***A specification-based Intrusion Detection system for AODV***", In Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks, pp.125-134, October 2003.

[5] R.Abdellaoui, " ***SU-OLSR une nouvelle solution pour la sécurité du protocole OLSR***". Maîtrise en génie concentration réseaux de télécommunications à l'école de technologie supérieure, Montréal, 05-05-2009

[6] L.Zhou and Z. Haas. "***Securing ad hoc networks***", IEEE Network Magazine, 13(6), November-December 1999

[7] Y. C.Hu, A. Perrig et D.B.Johnson. 2003. "***Packet leashes: a defense against wormhole attacks in wireless networks***". In Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies, NFOCOM 2003. Vol.3,p. 1976- 1986.

[8] S.R.Snapp, J.Brentano, G.V.Dias, T.L.Goan, L.T.Heberlein, and D. Mansur. "***Dids distributed intrusion detection system***". Technical report, Computer Security Laboratory, Department of Computer Science, University of California, Davis, June 1992.

[9] Y.Huang, and W.Lee, "***A Cooperative Intrusion Detection System for Ad Hoc Networks***", in Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks, pp. 135-147, October 2003.

[10] C.E.Perkins, et P.Bhagwat.. "***Highly dynamic Destination-Sequenced Distance Vector routing (DSDV) for mobile computers***". ACM SIGCOMM Computer Communication Review1994, vol. 24, no 4, p. 234-244.

[11] K.Sanzgiri, B.Dahill, B.N.Levine, C.Shields et E.M.Belding-Royer. "***A secure routing protocol for ad hoc networks***". In Proceedings of the 10 th IEEE International Conference on Network Protocols 2002. (ICNP'02). p. 78-87. .

[12] M. G Zapata, "***Secure ad hoc on-demand distance vector routing***". ACM Mobile Comuting and Communications Review SIGMOBILE 2002, vol. 6, no 3, p. 106-107.

[13] S.Wang, J.Wang, X.Zhang et J.Wei. "***Performance of anti-jamming ad hoc networks using directional beams with group mobility***". In IFIP International Conference on Wireless and Optical Communications Networks, 2006, p. 4 pp.

[14] V.Bharghavan, A.Demers, S.Shenker and L.Zhang, "***MACAW: A Media Access Protocol for Wireless LANs***", In Proceedings. of ACM SIGCOMM' Computer Communication Review 1994, pp. 212-225.

[15] S.Ray, D.Starobinski, "***On False Blocking in RTS/CTS Based Multihop Wireless Networks****"* Vehicular Technology, IEEE Transactions 2007, Volume 56, pp.849-862,

[16] S.Ramaswamy, H.Fu, M.Sreekantaradhya, J.Dixon et K.Nygard. "***Prevention of cooperative black hole attack in wireless ad hoc networks***". In Proceedings of International Conference on Wireless Networks 2003 (ICWN'03). p.570-575.

[17] B.Awerbuch, R.Curtmola, D.Holmer et C.Nita-Rotaru. "***Mitigating Byzantine Attacks in Ad Hoc Wireless Networks***". Archipelago project2004,Technical Report v1.

[18] Y.C.Hu, A.Perrig. et D.B.Johnson, "***Secure efficient distance vector routing in mobile wireless ad hoc networks***". In Proceedings of the 4th IEEE Workshop on Mobile Systems and applications - WMCSA, June 2002.

[19] P.Papadimitratos, et Z.J.Haas."***Secure Routing for Mobile Ad Hoc Networks***". In CS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS). San Antonio, TX, 2002.

[20] J.PAnderson. "***Computer Security threat monitoring and surveillance***". Technical report, Fort Washington, Pennsylvania, April 1980.

[21] D.Dorothy E. Denning. "***An intrusion detection model***". IEEE Transactions on software engineering 1987, ss-13(NO.2):222-232.

[22] M.Wood, M.Erlinge, "***Intrusion detection message exchange requirement***" , http://www.ietf.org/internet-drafts/draft-ietfs/draft-ietf-idwg-requirements-10.txt; 2002

[23] M.A.Mitrokotsa, R.Mavropodi, C.Douligeris, "***Intrusion Detection of Packet Dropping Attacks inMobile Ad Hoc Networks***", in proceedings of the International Conference on Intelligent Systems And Computing: Theory And Applications, Ayia Napa, Cyprus, July, 2006, pp. 111-118,

[24] K.Boudaoud, Z.Guessoum. "***A Multi-agents System for Network Security Management***". In Proceedings of the 6th IFIP Conference on Intelligence in Networks (SmartNet), Vienna, (Austria), 18-22 September 2000, pages 407-418.

[25] H.Debar ,M.Dacier, A.Wespi "***A Revised Taxonomy for Intrusion Detection Systems***" Computer science, 1999

[26] J.A.Freebersyser et B.Leiner. "***A DoD perspective on mobile Ad hoc networks***".In Ad hoc networking; C.E Perkins EdAddison-Wesley LongmanPublishing,2001, pp. 29-51.

[27] M.S.Mamun, A.F.M.Sultanul Kabir, " ***Hiarchical Design Based Intrusion Detection System For Wireless Ad Hoc Sensor Network***", International Journal of network Security & its Application (IJNSA),jul 2010, Vol.2,N°3,

[28] Y.Zhang, W.Lee, Y.Huang, "***Intrusion Detection Techniques for Mobile Wireless Networks***", Wireless Networks, 2003.Vol. 9, pp. 545-556,

[29] D.E.Denning and P.G.Neumann. "***Requirements and model for ides a real-cime intrusion-detection expert system***". Technical report: Document a005, Computer Science Laboratory, SRI International, Menlo Park, California, 1985.

[30] A.Ejaz, S.Kashan,M.Waqar, "***Cluster-based Intrusion Detection (CBID) Architecture for Mobile Ad Hoc Network***," AusCERT 2006 R&D Stream

[31] Y. Zhang et W. Lee, "***Intrusion detection in wireless ad-hoc networks***". Dans les actes de ACM MobiCom'2000, PP.275–283

[32] H.Deng, Q.Zng, and D.P.Agrawal, "***SVM-based Intrusion Detection System for Wireless Ad Hoc Networks***", In Proceedings of the IEE Vehicular Technology Conference (VTC'03), October 2003, Vol.3, pp 2147-2151,

[33] O.Kachirski, and R.Guha, "***Intrusion Detection Using Mobile agents in wireless Ad hoc Networks***", in Proceedings of the IEEE workshop on Knowledge Media Networking, July 2002, pp.153-158.

[34] Y.Liu, Y.Li, H.Man, "MAC ***Layer Anomaly Detection in Ad Hoc Networks***", In Proceedings of 6th IEEE Information Assurance Woskshop, June 17, 2005.

[35] F. Anjum, D.Subhadrabandhu, S.Sarkar, "***Signature based Intrusion Detection for Wireless Ad-Hoc Networks***", In Proceedings of Vehicular Technology Conference, Wireless Security Symposium, Orlando, Florida, Oct. 2003.

[36] D. Sterne1, P.Balasubramanyam, "***A General Cooperative Intrusion Detection Architecture for MANETs***".

[37] R.Ranjana; M.Rajaram "***Detecting Intrusion attacks in ADHOC Networks***"; Asian Journal of Information Technology 2007, 6 (7) : 758-761,ISSN: 1682-3915

[38] P. Kiran sree, I.Ramesh Babu, "***Power-Aware Hybrid Intrusion Detection System (PHIDS) using Cellular Automata in Wireless Ad Hoc Networks***"

[39] A.Rachedi and A.Benslimane, ***" Impacts and Solutions of Control Packets Vulnerabilities with IEEE 802.11 MAC*** " , *Journal of Wireless Communications and Mobile Computing, JohnWiley InterScience,* 2008."

[40] A.Rahman, P.Gburzynski,., "***Hidden Problems with the Hidden Node Problem****"*, in 23rd Biennial Symposium on Communications, 2006, pp. 270- 273.

[41] S.Ray, J.B.Carruthers and D.Starobinski, "***RTS/CTS-induced congestion in ad hoc wireless LANs***" *in IEEE WCNC*, 2003

[42] X. Zou , J. Deng, "***Detection of Fabricated CTS Packet Attacks in Wireless LANs***" in Proc of 7th International ICST Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QSHINE '10), Houston, TX, USA, Nov17-19, 2010.

[43] G. Thamilarasu, S. Mishra, R. Sridhar "***Improving Reliability of Jamming Attack Detection in Ad hoc Networks****",* International Journal of Communication Networks and Information Security (IJCNIS), Vol. 3, No. 1, April 2011

[44] UC Berkeley and USC ISI, "***The network simulator ns-2****"*, Part of the VINT project. Available from http://www.isi.edu/nsnam/ns, 1998.