Détection d'intrusions dans les résaux *ad hoc*

Jean-Marc Percher[1], Bernard Jouga[2]

[1] Ecole Supérieure d´Electronique de l'Ouest (ESEO),
4, rue Merlet de la Boulaye,
49000 Angers, France
jean-marc.percher@eseo.fr
[2] Supélec, BP 81127,
35511 Cesson Sévigné cedex, France
bernard.jouga@supelec.fr

Résumé Les réseaux sans fil ad hoc ou MANET, *Mobile Ad hoc NETwork*, sont des réseaux dont la topologie ne bénéficie d'aucune infrastructure préexistante. Elle se forme au gré de l'apparition et du mouvement des nœuds. Les participants d'une réunion, les intervenants des opérations de secours menées sur un site en cours d'exploration, les éléments engagés sur un champ de bataille peuvent tirer profit des caractéristiques de tels réseaux pour échanger de l'information. L'évolution rapide des performances des réseaux locaux sans fil et leur utilisation de plus en plus importante par les utilisateurs mobiles devraient bénéficier au développement des MANET. 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. Dans cet article nous proposons une architecture de sécurité pour les réseaux ad hoc. Les mécanismes de sécurité sont renforcés par un système de détection d'intrusions distribué et coopératif. Chaque nœud est équipé d'un IDS local et des agents mobiles autonomes sont mis en œuvre, si nécessaire, pour collecter les informations stockées sur les autres nœuds. Nous validons cette architecture d'IDS, *Intrusion Detection System*, construite autour d'une plate-forme à agents mobiles à l'aide d'un prototype et de tests en laboratoire.

1 Introduction

Nous proposons pour les réseaux *ad hoc* la définition suivante : "Un réseau ad hoc est un réseau sans fil capable de rendre transparentes aux utilisateurs mobiles les modifications de topologie qu'il subit". Le groupe MANET de l'IETF fournit une définition plus précise en introduction de la RFC 2501 [1] :

> *Un réseau ad hoc comprend des plates-formes mobiles (par exemple, un routeur interconnectant différents hôtes et équipements sans fil) appelées nœuds qui sont libres de se déplacer sans contrainte. Un réseau ad hoc est donc un système autonome de nœuds mobiles. Ce système peut fonctionner d'une manière isolée ou s'interfacer à des réseaux fixes au travers de passerelles. Dans ce dernier cas, un réseau ad hoc est un réseau d'extrémité.*

Il n'en reste pas moins que la terminologie "réseau ad hoc" est relativement peu explicite. C'est sans doute la raison pour laquelle la communauté scientifique la remplace parfois par celle de "réseau spontané", traduction de *spontaneous network*.

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

- *Mobilité.*- La mobilité des noeuds constitue à l'évidence une caractéristique très spécifique des réseaux ad hoc. Cette mobilité est intrinsèque au fonctionnement du réseau. Elle se distingue de la nomadicité (mobilité des seuls noeuds terminaux) ou de l'itinérance (équipements statiques mais pouvant être déplacés). Dans un réseau ad hoc, la topologie du réseau peut changer rapidement, de façon aléatoire et non prédictible et les techniques de routage des réseaux classiques, basées sur des routes préétablies, ne peuvent plus fonctionner correctement.
- *Equivalence des nœuds du réseau.*- Dans un réseau classique, il existe une distinction nette entre les nœuds terminaux (stations, hôtes) qui supportent les applications et les nœuds internes (routeurs par exemple) du réseau, en charge de l'acheminement des données. Cette différence n'existe pas dans les réseaux ad hoc car tous les noeuds peuvent être amenés à assurer des fonctions de routage.
- *Liaisons sans fil.*- Les technologies de communication sans fil sont indispensables à la mise en place d'un réseau ad hoc. Malgré des progrès très importants, leurs performances restent et resteront en deçà de celles des technologies des réseaux filaires. La bande passante est moins importante, alors que le routage et la gestion de la mobilité génèrent davantage de flux de contrôle et de signalisation que dans une architecture de réseau filaire. Ces flux doivent être traités de façon prioritaire pour prendre en compte rapidement les modifications de topologie.
- *Autonomie des nœuds.*- La consommation d'énergie constitue un problème important pour des équipements fonctionnant grâce à une alimentation électrique autonome. 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 ce problème.
- *Vulnérabilité.*- Les réseaux sans fil sont par nature plus sensibles aux problèmes de sécurité. Pour les réseaux ad hoc, le principal problème ne se situe pas tant au niveau du support physique mais principalement dans le fait que tous les 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 plus délicate et l'absence de centralisation pose un problème de remontée de l'information de détection d'intrusions.

L'article est organisé comme suit : la section 2 présente les caractéristiques des protocoles de routage ad hoc. La section 3 présente les caractéristiques des systèmes de détection des intrusions et les principaux IDS distribués. La section

4 décrit les caractéristiques de l'architecture proposée. La section 5 présente une partie des tests fonctionnels réalisés. Enfin nous concluons et proposons des suites possibles aux travaux réalisés.

2 Le routage dans les réseaux ad hoc

2.1 Taxonomie des protocoles de routage pour les réseaux Ad Hoc

Les protocoles de routage des réseaux ad hoc s'appuient sur deux modèles de fonctionnement : les protocoles proactifs et les protocoles réactifs. On peut les différencier par la méthode utilisée pour découvrir le chemin entre le nœud source et le nœud destination. Pour maintenir leur table de routage, les protocoles proactifs recherchent à intervalle régulier les différentes routes disponibles dans le réseau. Quand un paquet doit être transmis, sa route est alors connue à l'avance et peut ainsi être immédiatement utilisée. Les protocoles réactifs entreprennent la recherche d'une route uniquement avant de transmettre un paquet. La figure

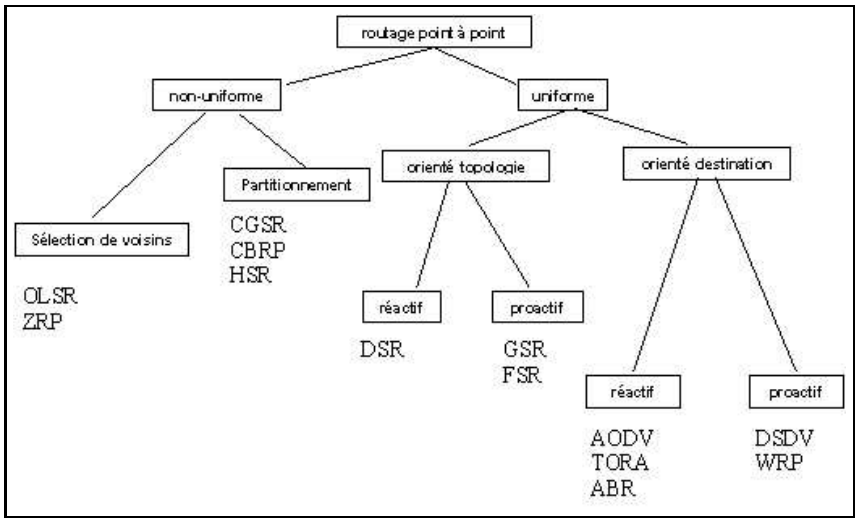

Fig. 1. Taxonomie des protocoles de routage pour les réseaux ad hoc.

1 présente une taxonomie des protocoles de routage pour les réseaux ad hoc. Ces protocoles se différencient d'abord par le niveau d'implication des nœuds dans le routage. Ils sont dits *uniformes* si tous les nœuds du réseau jouent le même rôle pour la fonction de routage. Ils peuvent à l'inverse être *non-uniformes* si une structure hiérarchique est donnée au réseau et que seuls certains nœuds assurent le routage. Ainsi, dans les protocoles à *sélection de voisins*, chaque nœud sous-traite la fonction de routage à un sous ensemble de ses voisins directs. Pour les

protocoles à *partitionnement*, le réseau est découpé en zones dans lesquelles le routage est assuré par un unique nœud maître.

Les protocoles de routage uniformes peuvent également être regroupés selon les données qu'ils utilisent pour effectuer leur tâche. Dans les protocoles *orientés topologie*, plus connus sous le nom de *link-state protocols*, chaque nœud utilise comme données l'état de ses connexions avec ses noeuds voisins ; cette information est ensuite transmise aux autres nœuds pour leur donner une connaissance plus précise de la topologie du réseau. Les protocoles *orientés destinations*, plus connus sous le nom de *distance vector protocols*, maintiennent pour chaque nœud destination une information sur le nombre de nœuds qui les en séparent (la distance) et éventuellement sur la première direction à emprunter pour y arriver (le vecteur).

Avec un protocole proactif, les routes sont disponibles immédiatement. Cependant, le trafic induit par les messages de contrôle et de mise à jour des tables de routage peut être important et partiellement inutile. De plus, la taille des tables de routage croit linéairement en fonction du nombre de nœuds. A l'opposé, dans le cas d'un protocole réactif, aucun message de contrôle ne charge le réseau pour des routes inutilisées. Mais, pour ces derniers, la mise en place d'une route par inondation peut être coûteuse et provoquer des délais importants avant l'ouverture de la route.

En terme de performances, les protocoles orientés topologie (*link state*) convergent plus rapidement que les protocoles orientés destination (*distance vector*). Cependant, dans le cas de réseaux à forte mobilité, le trafic induit par les fréquents messages de contrôle est souvent pénalisant.

D'une façon générale, les protocoles de routage proactifs "plats", qu'ils soient orientés destination ou topologie, ne se sont pas adaptés aux réseaux de taille importante (nombre de nœuds supérieur à 100) et à forte mobilité. Une première solution pour ce type de réseau est l'utilisation de protocoles dits hiérarchiques (tels que HSR, FSR, etc.). Une seconde solution peut être d'utiliser un protocole réactif. Ce type de routage permet de gérer de très gros réseaux si la mobilité des nœuds reste faible ; le trafic reste faible s'il est dirigé vers un nombre réduit de destinations. D'autre part, le calcul d'une route sur demande est très pénalisant pour du trafic multimédia demandant des garanties en matière de qualité de service.

Des études comparatives montrent que certains protocoles sont plus performants que d'autres selon les caractéristiques du réseau. Par exemple, AODV, DSR ou GSR sembleraient convenir à des réseaux à forte mobilité, tandis que TORA est mieux adapté à des réseaux plus statiques. Par ailleurs, TORA et OLSR gèrent efficacement les grands réseaux à forte densité, AODV semble quant à lui performant dans les réseaux de faible densité, et DSR est bien adapté aux petits réseaux.

A l'heure actuelle, aucun protocole de routage dans les réseaux ad hoc n'a été adopté au sein de l'IETF. Les différents protocoles présentés sur la figure 1 ne sont encore que des *drafts* et restent en cours de développement et/ou de spécification. Certaines propositions ont été abandonnées et les plus abouties

sont AODV, DSR et OLSR. Les deux premiers protocoles sont parmi les plus anciens et ils ont fait l'objet de simulations comparatives détaillées [2].

2.2 Exemple du protocole de routage OLSR

OLSR, *Optimized Link State Routing Protocol*, est un protocole proactif, non uniforme et basé sur la sélection de voisins [3]. OLSR s'appuie sur le concept de Relai Multi Point (*Multi Point Relay*, MPR). Les MPR d'un nœud correspondent à l'ensemble des voisins qui permettent d'atteindre tous les nœuds situés à deux sauts. La diffusion des différents messages de contrôle ne se fait que vers les MPR (voir la figure 2), réduisant ainsi les répétitions inutiles. D'autre part OLSR distingue les liens unidirectionnels des liens bidirectionnels, seuls utilisés pour le routage. Chaque nœud maintient de l'information sur les nœuds qui l'ont élu en

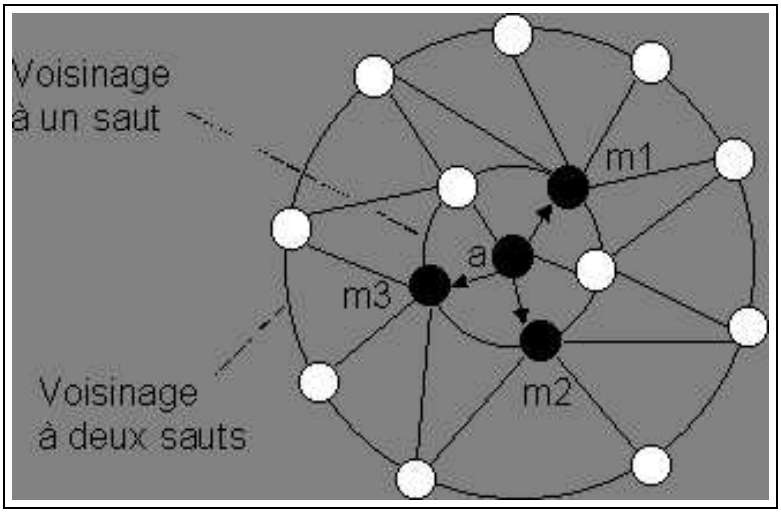

Fig. 2. Relais Multi Point. La station A a choisi m_1, m_2 et m_3 comme relais multi point. Quand A émet un message TC (*Topology Control*), il est seulement retransmis par m_1, m_2 et m_3, qui le retransmettent à leur tour vers leur MPR.

tant que MPR. Ceci est fait grâce à des message de présence (*Hello messages*) envoyés par chaque nœud à ses voisins. Ces messages contiennent :

- la liste des nœuds avec lesquels l'émetteur a des liens bidirectionnels,
- la liste des nœuds que l'émetteur peut entendre (ils entretiennent un lien unidirectionnel vers lui)
- la liste des nœuds que l'émetteur a choisi comme MPR.

La diffusion de ces messages de présence permet aux nœuds du réseau de stocker, dans leur table des voisins, une vision à deux sauts de leur voisinage et de calculer

l'ensemble de leurs MPR. Cet ensemble est recalculé dès qu'un changement est détecté dans le voisinage à deux sauts.

La diffusion sur la totalité du réseau (via les MPR) de messages de contrôle de topologie (*Topology Control messages, TC messages*) donne l'information topologique nécessaire au routage. Ces messages contiennent, pour chaque MPR, la liste des nœuds qui l'ont choisi. Grâce à ces messages, les nœuds peuvent maintenir une table de topologie (*Topology Table*), indiquant le dernier saut pour chaque destination.

Un algorithme de plus court chemin, appliqué à la table des voisins et à la table de topologie, permet de construire la table de routage de chaque nœud. Cette table mémorise, pour tous les nœuds du réseau, le nombre de sauts et le premier saut pour l'atteindre. Elle doit être recalculée dès que l'une des deux tables sources est modifiée.

OLSR fournit des routes optimales en nombre de sauts. Il convient pour des grands réseaux grâce à son mécanisme de MPR, mais est sans doute moins efficace pour de petits réseaux.

3 Détection d'intrusions dans les réseaux ad hoc

3.1 Le constat

Les réseaux sans fil sont par nature plus sensibles aux problèmes de sécurité. L'intrusion sur le support de transmission est plus facile que pour les réseaux filaires et il est possible de mener des attaques par déni de service en brouillant les bandes fréquences utilisées. Le contexte ad hoc augmente le nombre de failles de sécurité potentielles. Etant par définition sans infrastructure, les réseaux ad hoc ne peuvent bénéficier des services de sécurité offerts par des équipements dédiés : pare feux, serveurs d'authentification, etc. Les services de sécurité doivent être distribués, coopératifs et compatibles avec la bande passante disponible. Le routage pose aussi des problèmes spécifiques : chaque station du réseau peut servir de relai et a donc la possibilité de capturer ou bien de détourner le trafic en transit. Des attaques en déni de service sont également possibles. Un routeur malicieux peut aussi générer des messages de routage dans le but de submerger les vrais routeurs. Nous avons mené dans [4,7] une étude sur les vulnérabilités du protocole de routage OLSR. Les quatre types d'attaques recensées reposent essentiellement sur l'émission de messages fallacieux par un nœud malveillant inséré dans le réseau ad hoc. Cette possibilité étant offerte par presque tous les autres protocoles de routage ad hoc, des attaques de même nature peuvent être développées. Les attaques décrites ont été implémentées et une plate- forme expérimentale a permis de jouer les attaques et de constater les effets attendus. Les signatures de deux de ces attaques ont été déterminées afin qu'elles puissent ensuite être détectées par un système de détection d'intrusions réparti.

Aujourd'hui la sécurisation des MANET s'appuie sur les mécanismes de sécurité du lien radio (WEP) et des échanges IP (Ipsec-VPN). Dans l'attente de solutions de sécurité adaptées aux caractéristiques des MANET comme, par

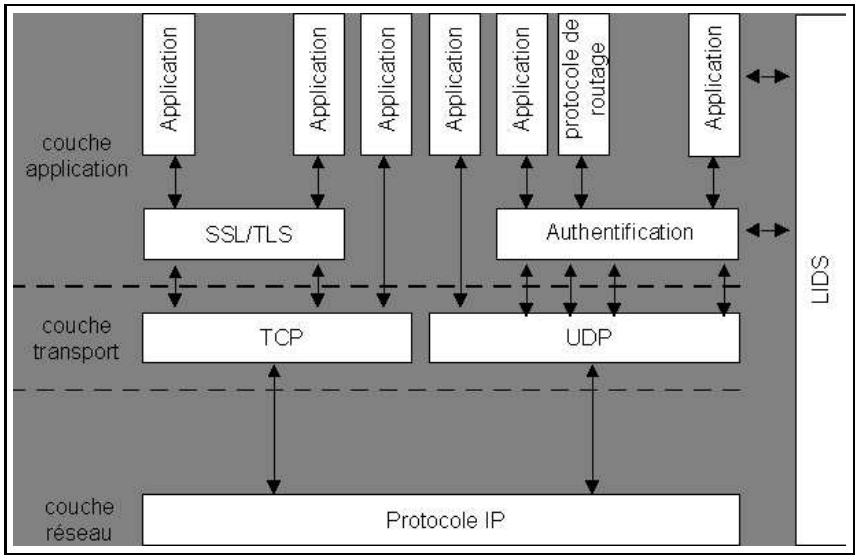

Fig. 3. Modèle de sécurité pour les réseaux ad hoc.

exemple, les infrastructures d'authentification distribuées [5,6], la gestion des clés de codage doit être adaptée aux différents cas d'usage.

Nous avons proposé dans [4] une architecture de sécurité pour les réseaux ad hoc, présentée figure 3, dans laquelle les mécanismes de sécurité des MANET sont renforcés par un système local de détection des intrusions (*LIDS, Local Intrusion Detection System*).

3.2 Rappels sur les systèmes de détection d'intrusions

Nous considérons ici comme intrusion toute action visant à compromettre l'intégrité, la confidentialité ou la disponibilité d'une ressource. L'observation des attaques dirigées vers les systèmes d'information nous montre que, quelles que soient les techniques de prévention mises en place contre les intrusions, il existe toujours des failles exploitables pour celui qui les traque. L'attaque d'un système peut même être réalisée simplement à partir de l'enchaînement d'opérations élémentairement autorisées ; elle ne nécessite donc pas toujours de contourner les mécanismes de sécurité. La détection d'intrusions peut donc être considérée comme une action complémentaire à la mise en place des mécanismes de sécurité. En effet, si une tentative d'intrusion est détectée suffisamment tôt, les réponses du système peuvent permettre de limiter les conséquences d'une attaque. Dans le cas d'une attaque de type déni de service, la réponse d'un système de détection d'intrusions (souvent abrégé IDS, traduction de *Intrusion Detection System*) pourrait être de refuser les nouvelles demandes de connexions au-delà d'une certaine fréquence estimée normale.

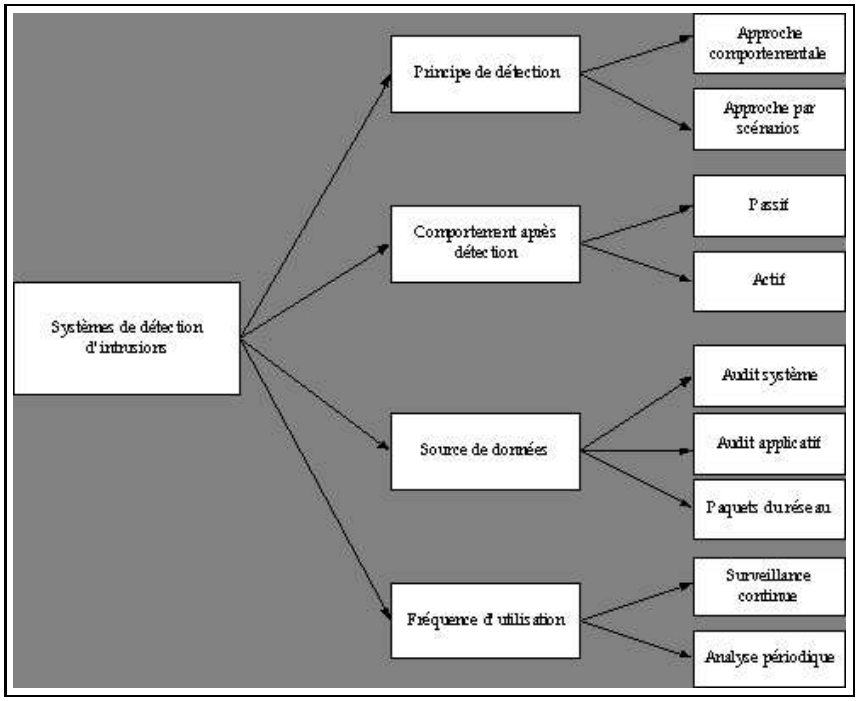

Fig. 4. Taxonomie des IDS.

Pour présenter les caractéristiques des systèmes de détection d'intrusions nous retenons la classification proposée par L. Mé et C.Michel [8]. Celle-ci, représentée sur la figure 4, utilise les critères de classification suivants :

- Le principe de détection des intrusions,
- Le comportement après détection,
- La source des données,
- La fréquence d'utilisation.

Le principe de détection des intrusions. Les deux approches utilisées à ce jour sont l'approche comportementale et l'approche par scénarios. La première se base sur l'observation du comportement de l'utilisateur et sur la détection d'un comportement déviant par rapport à ses habitudes. Cette modification du comportement peut traduire une tentative d'intrusion, due soit à une usurpation de l'identité de l'utilisateur, soit à l'exécution par celui-ci de commandes non autorisées. La seconde approche consiste à identifier chaque attaque par une signature propre et ensuite à rechercher dans les fichiers d'audits du système les traces de ces signatures. On peut noter que cette approche par scénarios nécessite de connaître la signature d'une attaque avant de pouvoir la détecter. L'approche comportementale permet de détecter des attaques inconnues, mais la définition du comportement normal de utilisateur demeure la principale difficulté.

Le comportement après détection La nature de la réponse apportée par les systèmes après détection d'une intrusion peut aussi être utilisée pour classifier les IDS. Deux types de réponses sont implémentées à ce jour, soit une réponse passive, par exemple l'émission d'une alarme à l'administrateur, soit une réponse active, comme la mise en place de nouvelles règles de filtrage sur un pare-feu.

La source des données Les IDS peuvent être classés en fonction de la provenance de leurs données d'audit, selon qu'elles viennent du système, des applications, des paquets du réseau ou encore d'un autre IDS.

La fréquence d'utilisation La fréquence d'analyse des données d'audit est aussi un élément distinctif des systèmes de détection d'intrusions. Certains IDS peuvent surveiller en permanence le système d'information tandis que d'autres se limitent à une analyse périodique.

3.3 Motivation des travaux

Les techniques de détection d'intrusions utilisées dans les réseaux traditionnels ne peuvent être exploitées telles quelles dans les réseaux ad hoc. Comparées aux architectures des réseaux filaires où la supervision du trafic est généralement réalisée au niveau des commutateurs, des routeurs ou des passerelles, celles des réseaux ad hoc ne possèdent pas de point de concentration. En l'absence de ce point de concentration, la détection d'intrusions doit être distribuée sur l'ensemble des nœuds actifs à l'intérieur du réseau. De plus, dans un réseau sans fil ad hoc chaque nœud possède une vision limitée de l'activité du réseau. Cette limite dépend essentiellement des caractéristiques des stations en terme d'émission-/réception et constitue une autre contrainte importante pour les algorithmes de détection des intrusions. Sans possibilité d'analyse globale des activités du réseau, la détection des intrusions est plus difficile.

Il est donc nécessaire de définir un modèle d'architecture d'IDS distribué adapté aux spécificités des réseaux ad hoc. Dans ce type d'architecture, basée sur un modèle distribué et coopératif, chaque nœud est indépendant et responsable localement de la détection d'éventuels signes d'intrusions. La prise en compte des informations collectées, ou des intrusions identifiées par les nœuds voisins situés dans sa zone d'émission/réception, permet à un nœud d'obtenir une vision plus large de l'état du réseau. Nous présentons dans la partie suivante une synthèse des architectures des principaux IDS distribués et coopératifs.

4 Les IDS distribués et coopératifs

Les architectures d'IDS présentées dans la table 1 ne sont pas spécifiques aux environnements des MANET. Leur point commun est un pré-traitement des informations à analyser dès leur collecte mais, à l'exception de CSM, MICAEL

Nom	Type de détection	Prétaitement des données	Analyse données (détection)	Type de réponse	Origine
AAFID [9]	scénarios	distribué	centralisée	passive	système
CSM [10]	comportementale	distribué	distribuée	active	système
DIDS [11]	hybride	distribué	centralisée	passive	système/réseau
DPEM [12]	comportementale	distribué	centralisée	passive	système
GrIDS [13]	hybride	distribué	centralisée	passive	système/réseau
IDA [14]	scénarios	agents mobiles	centralisée	passive	système
JiNao [15]	hybride	distribué	centralisée	passive	MIB/réseau
MICAEL [16]	scénarios	agents mobiles	distribuée	passive	MIB
SPARDA [17]	scénarios	agents mobiles	distribuée	passive	système/réseau

Tab. 1. Synthèse des caractéristiques des IDS distribués.

et SPARTA, le processus de détection nécessite des organisations hiérarchiques autour d'un nœud central permanent.

L'architecture de CSM (*Cooperating Security Managers*) est totalement distribuée ; l'IDS local installé sur chaque nœud coopère avec les autres IDS locaux pour identifier l'origine des connexions dans un réseau. Cet IDS a été étudié spécifiquement pour suivre les connexions à travers les réseaux.

SPARTA (*Security Policy Adaptation Reinforced Through Agents*) utilise des agents mobiles pour collecter les informations à analyser. L'architecture nécessite la présence d'un nœud central dont le rôle est de maintenir une base de connaissance des nœuds présents dans le réseau. Cette dernière caractéristique ne permet donc pas à SPARTA d'être déployé dans les réseaux sans fil ad hoc.

MICAEL utilise une plate-forme pour agents mobiles sur chaque nœud. La distribution de ces agents mobiles sur chacun des nœuds est dynamique et gérée par un nœud central.

Nécessitant la présence d'un nœud central permanent ces différents modèles d'architectures d'IDS distribués ne sont pas adaptés aux réseaux sans fil ad hoc.

5 Proposition d'architecture d'IDS distribué pour réseau sans fil ad hoc

5.1 Les spécifications

Les spécifications s'appuient sur les caractéristiques générales des IDS et sur les contraintes spécifiques imposées par les réseaux ad hoc. Les caractéristiques résumées ici ont été présentées dans les chapitres précédents.

- *Principes de détection* : l'architecture de l'IDS doit être indépendante de la méthode de détection.
- *Sources de données* : l'architecture de l'IDS doit être indépendante de la source des données.
- *Fréquence d'utilisation* : la détection des tentatives d'intrusions doit être réalisée en temps réel pour permettre aux utilisateurs de réagir immédiatement.

- *Comportement 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 systèmes membres du même réseau.
- *Distribution 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.
- *Débits 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.
- *Mobilité des nœuds* : l'IDS doit posséder les mécanismes lui permettant de prendre en compte la mobilité des nœuds. Toutefois, différentes optimisations pourront être proposées selon les caractéristiques de mobilité et de densité des nœuds.
- *Caractéristiques des nœuds et portabilité de l'IDS* : la surcharge induite par la détection des intrusions ne doit pas altérer de plus de 10% les performances des systèmes et être indépendante des systèmes d'exploitation.
- *Normalisation* : l'architecture de l'IDS doit adopter les normes et standards afin de pouvoir coopérer avec d'autres IDS.

5.2 L'architecture globale de l'IDS distribué

Nous proposons d'équiper chaque nœud du réseau d'un système de détection local (*LIDS, Local Intrusion Detection System*). 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 d'autres nœuds du réseau. L'architecture globale est représentée figure 5. Ce modèle d'architecture est basé sur une plate-forme pour agents mobiles. Un agent mobile est défini comme une entité logicielle qui fonctionne de manière autonome et continue dans un environnement particulier, capable de se déplacer et de s'adapter aux changements de l'environnement, de communiquer et de coopérer avec d'autres agents.

Dans notre modèle, les entités logicielles hébergées sur chaque nœud (LIDS) fonctionnent de manière indépendante et observent les activités locales. Le LIDS détecte les intrusions à partir de cette surveillance locale et peut amorcer des réponses en conséquence. Si une anomalie est détectée ou si les signes d'intrusion nécessitent d'être confirmés, des agents mobiles peuvent être créés et envoyés vers d'autres nœuds pour y effectuer des tâches spécifiques et ensuite retourner les informations recherchées.

La mise en œuvre des agents mobiles, qui permet de déplacer le code vers les données à analyser, est une alternative aux architectures clients/serveurs. Cette solution de communication entre les nœuds est efficace si le code de l'agent est moins volumineux que celui des données à analyser. Notre architecture nécessite uniquement une mobilité faible (codes et données) des agents. Les travaux présentés dans [16,17,18,19] donnent les avantages apportés par les agents mobiles dans les réseaux sans fil ad hoc :

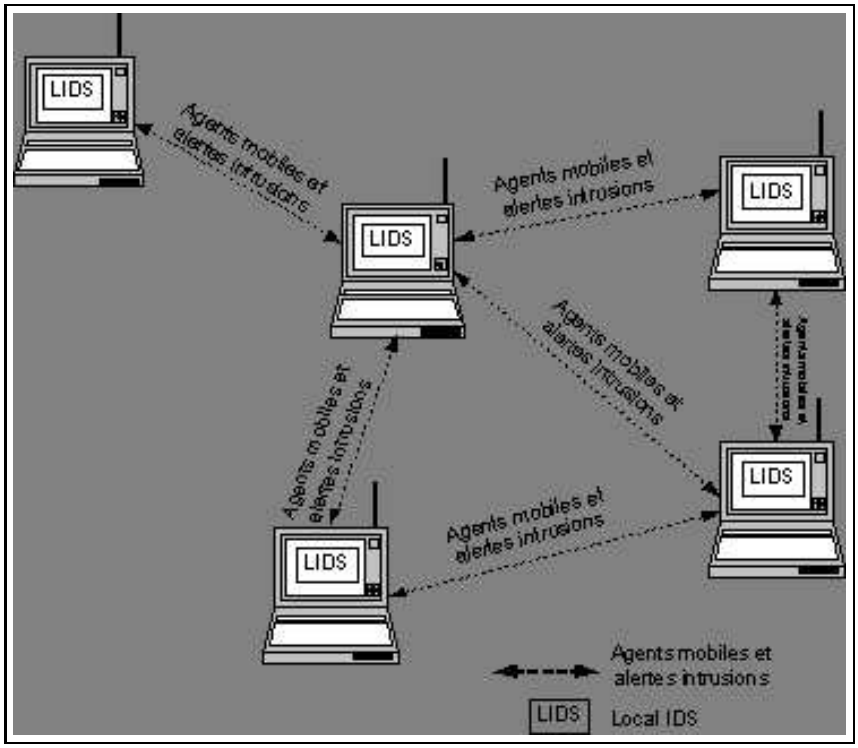

Fig. 5. Architecture globale de l'IDS.

- L'*asynchronisme* se traduit par la possibilité de déléguer une tâche à un agent mobile sans rester en attente du résultat.
- L'*autonomie* et l'*intelligence* permettent aux agents d'adapter leur comportement selon les informations collectées localement et éventuellement de poursuivre leurs déplacements au sein du réseau pour terminer leur mission.
- Le *temps de réponse des applications* et la *charge du réseau* peuvent être réduits par le déplacement du code vers les données plutôt que de déplacer les données vers les applications.

6 Architecture interne d'un LIDS

La figure 6 représente l'architecture interne d'un LIDS. La détection des intrusions est effectuée localement à partir des données collectées dans la MIB (*Management Information Base*).

- L'*agent IDS local* forme le cœur du système de détection d'intrusion. Il analyse les données collectées localement. Par défaut la détection s'appuie sur l'état des variables de la MIB II. Ces informations peuvent être complétées

par des MIB spéciales, par exemple une extension de MIB spécifique au protocole de routage ad hoc, ou la MIB RMon pour prendre en compte les activités du réseau.
- L'*agent MIB local* a pour fonction de gérer les extensions de MIB spécifiques à la détection d'intrusions. Par exemple, pour utiliser une extension de MIB spécifique au protocole de routage, un agent local sera chargé de la mise à jour des variables.
- La *plate-forme à Agents Mobiles* gère à la demande de l'agent IDS local l'ensemble des activités des agents mobiles (création, mobilité, sécurité, exécution, communication).
- L'*agent SNMP* est un agent SNMP standard dont la fonction est de répondre aux requêtes locales des LIDS et des agents mobiles.

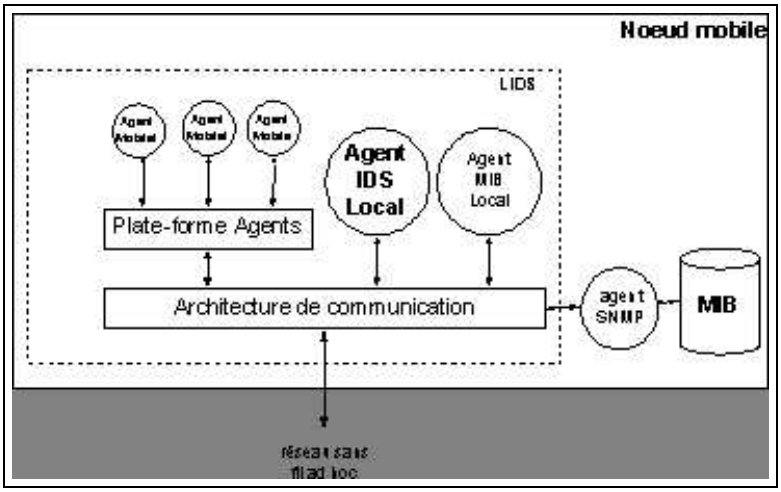

Fig. 6. Architecture d'un LIDS.

Architecture de l'agent IDS Local L'architecture interne retenue pour la conception des LIDS s'appuie sur la description des modules internes du modèle d'IDS proposé par l'IDWG (*Intrusion Detection Working Group* de l'IETF[3]). Celui-ci définit les trois composants de base d'un IDS, représentés sur la figure 7 :
- le *senseur* dont la fonction est de collecter et de mettre en forme les données brutes collectées dans le système ou sur le réseau,
- l'*analyseur* qui traite les informations générées par le senseur afin d'identifier les intrusions,

[3] Document officiel de l'IDWG : *http ://www.ietf.org/html.charters/idwg-charter.html*

– le *manager* qui a pour fonction de gérer les alarmes après détection.

Notre modèle ajoute un nouveau type de message entre l'analyseur et le senseur. Ce message appelé *requête* permet à l'analyseur de demander et de collecter, s'il le souhaite, des informations complémentaires.

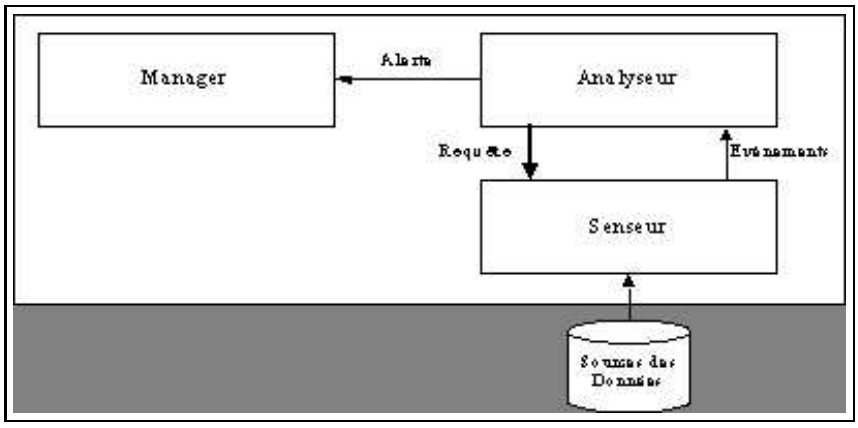

Fig. 7. Modèle IDS de l'IDWG, complété par le message requête.

Architecture détaillée Le LIDS possède une architecture construite autour de quatre modules de base :

- *IDS Framework* : ce module constitue le cœur du système de détection d'intrusion. Dans la première version, il utilise une détection par analyse de signatures. Chaque attaque est représentée par un diagramme d'états dont les transitions sont réalisées selon des évènements prédéfinis représentés par des valeurs de variables MIB.
- *Event Abstraction Framework* : le module d'abstraction des évènements a pour fonction de convertir un événement représentatif d'une signature en requête sur des variables de la MIB locale.
- *Mobile Agent Framework* : le module d'agents mobiles est constitué de la plate-forme AGLET (sources IBM, *http ://www.trl.ibm.co.jp/agletts/*).
- *IDS communication Framework* : le module de communication avec des IDS externes a pour fonction d'assurer l'interopérabilité avec d'autres systèmes de détection d'intrusions.

Les modules externes *Attack Signatures, Event and Alert specification et Event Abstractor Database* sont des sources de données dont l'objectif est de faciliter les paramétrages du LIDS.

La MIB, *Management Information Base*, regroupe l'ensemble des variables de la MIB II et des variables complémentaires implémentées dans la MIB expérimentale.

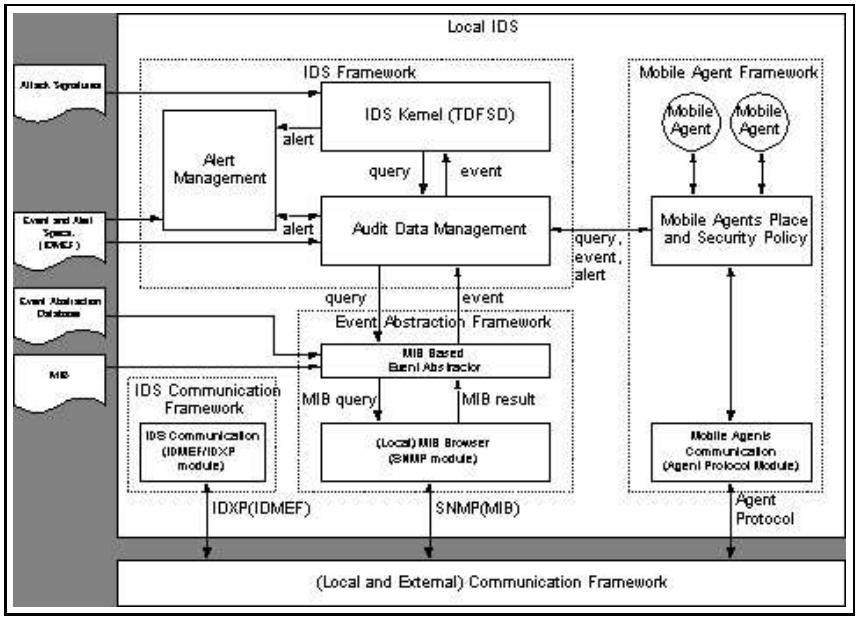

Fig. 8. Architecture modulaire des LIDS.

Les différents composants du LIDS communiquent par les flots d'informations internes suivants :

- *La signature d'une attaque* est représentée par une machine d'états dans le module *IDS Kernel*. Chaque transition d'un état à un autre est la conséquence d'un événement.
- *La gestion des évènements* est assurée par le module *Audit Data Management*. Celui-ci reçoit du module *IDS Kernel* les caractéristiques des événements à détecter.
- *Le module Audit Data Management* demande soit au module *local Event Abstraction Framework* soit au *module Mobile Agent Framework* de détecter l'événement. Des alertes issues d'autres LIDS peuvent aussi être prises en compte par le module *Alert Management*. La recherche d'événement peut être réalisée à la demande ou de façon périodique selon les besoins de l'*IDS Kernel*.

Le module MIB *Base Event Abstractor* a pour fonction d'associer à chacun des évènements les variables MIB correspondantes. Il s'appuie sur le module *Local MIB Browser* pour réaliser les requêtes SNMP (*GET et GETNext*).

Selon la valeur des variables MIB reçues, le module MIB *Base Event Abstractor* peut indiquer au module *Audit Data Management* l'occurrence d'un événement recherché. Celui-ci transmettra l'information au module *IDS Kernel*.

7 Tests de validation fonctionnelle

7.1 Description des tests réalisés

Seuls les modules du LIDS (figure 8) nécessaires aux tests de validation fonctionnelle ont été implémentés. Dans cette première phase de tests, dont l'objectif est de valider les choix techniques réalisés, nous avons sélectionné deux familles d'attaques représentatives des vulnérabilités inhérentes aux réseaux sans fil ad hoc :

- Attaques sur le protocole de routage OLSR.
- Déni de service N sauts : L'attaquant génère pour chaque message HELLO transmis par un nœud voisin un nouveau message HELLO dans lequel toutes les liaisons, contenues dans le message HELLO initial, sont annoncées asymétriques. Cette modification de l'état des liens a pour conséquence d'isoler le nœud émetteur du message HELLO initial et ainsi de perturber le routage.
- Déni de Service $1 + N$ Sauts, détournement de MPR : L'attaquant se fait élire MPR par ses nœuds voisins en annonçant dans de faux messages TC sa capacité à atteindre des nœuds inexistants. Ses proches voisins vont ainsi lui transmettre l'ensemble des messages destinés à des nœuds distants.
- Attaques réseaux sur le protocole applicatif Telnet.

Ce type d'attaque, connu sous le nom de *Stepstone* [20], a pour objectif de masquer l'identité d'un attaquant en prenant le contrôle d'un nœud et en générant ensuite depuis celui-ci des attaques à destination d'autres nœuds. Une suite en cascade de connexions Telnet sur différents nœuds représente un exemple classique de ce type d'attaque.

Les nœuds d'un MANET, dépourvus des protections liées à l'infrastructure du réseau (Firewall, Proxy, Serveurs d'authentification, etc.) sont particulièrement exposés à ce type d'attaque.

Pour détecter les attaques sur le protocole de routage OLSR, les LIDS utilisent uniquement les variables spécifiques au protocole OLSR, mémorisées dans l'extension de la MIB locale. L'attaque de type *Stepstone* nécessite la mise en œuvre de la plate-forme à agents mobiles pour identifier l'origine des connexions Telnet.

Ces différentes attaques ont été réalisées et détectées sur la plate-forme de tests du projet RAHMS. Nous décrivons dans la section suivante la détection de l'attaque par rebonds Telnet.

7.2 Description et détection de l'attaque Stepstone

Principe de la détection Pour expliquer le mécanisme de la détection des attaques par rebonds Telnet, nous représentons sur la figure 9 les seuls modules des LIDS impliqués dans cette détection.

- Le *manager global* est chargé de détecter l'origine les connexions. Pour cela il s'appuie sur les informations collectées par les agents mobiles. Une interface homme machine (IHM) a été développée pour visualiser le suivi

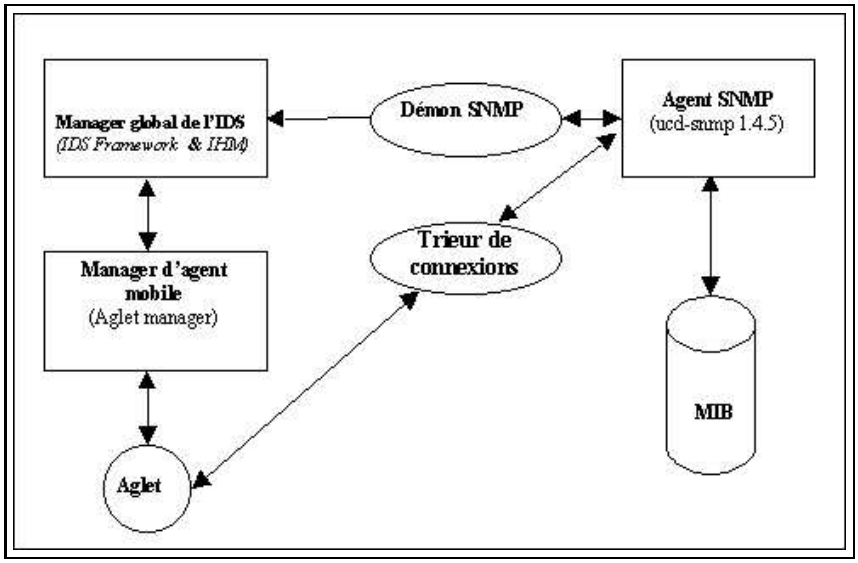

Fig. 9. Modules LIDS impliqués dans la détection de l'attaque Stepstone.

des connexions. La réponse à la détection d'une attaque *Stepstone*, i.e. la fermeture de la connexion, peut-être réalisée à l'initiative de l'opérateur.
- L'*agent SNMP* est utilisé pour collecter les informations dans la MIB.
- Le *démon SNMP* a pour fonction de détecter toute nouvelle connexion entrante ou sortante sur le nœud protégé. Il scrute la MIB à intervalles réguliers.
- Le *gestionnaire d'agents mobiles* contrôle l'activité des agents mobiles (création, accueil, sécurité, arrêt).
- Le *module trieur de connexions* a pour fonction de communiquer à un agent mobile la source locale de la connexion détectée sur le nœud suivant.

Détection de l'attaque Nous considérons ici une connexion Telnet réalisée en cascade sur trois nœuds équipés d'un LIDS et d'un agent SNMP. Cette chaîne de connexions est représentée sur la figure 10. Nous nous intéressons au LIDS du nœud C. Le comportement des LIDS situés sur les autres nœuds est identique.
- *Première étape* : sur le nœud C, le démon SNMP informe le manager global de l'arrivée d'une connexion TCP. Il transmet l'adresse IP et le port à l'origine de cette connexion sur le nœud distant. Dans notre exemple @IPB et PsB.
- *Deuxième étape* : sur le nœud C, le manager global ne peut détecter une chaîne de connexion sans informations complémentaires, non disponibles dans la MIB locale. Il transmet les informations collectées (@IPB et PsB) au manager d'agents mobiles. Ce dernier va alors créer une Aglet et l'envoyer sur le nœud B pour collecter des informations complémentaires.

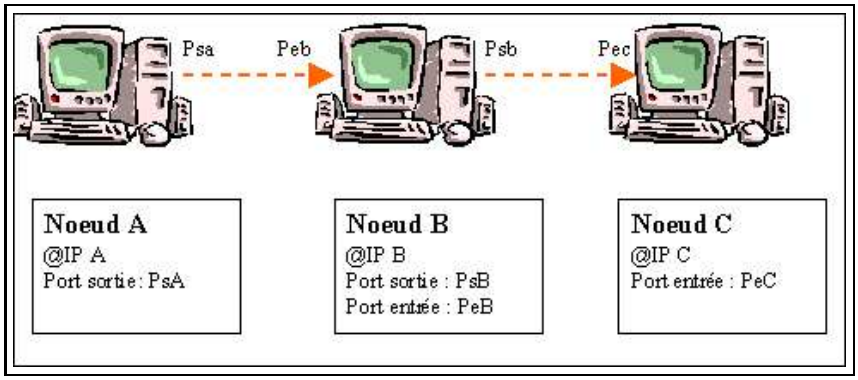

Fig. 10. Chaîne de connexions Telnet.

- *Troisième étape* : sur le nœud B, l'Aglet, est authentifiée et accueillie par la plate-forme à agents locale. Elle s'adresse alors au module trieur de connexions pour lui demander de lui fournir l'origine de la connexion dont le port de sortie est PsB, suite à la requête effectuée dans la MIB par l'agent SNMP pour identifier l'origine de la connexion. Dans notre exemple, le trieur de connexions transmet @IPA et PsA à l'Aglet.
- *Quatrième étape* : l'Aglet se déplace alors du nœud B vers le nœud A. Sur ce dernier le trieur de connexions indique à l'Aglet que le nœud est l'origine de la connexion.
- *Cinquième étape* : l'Aglet se déplace vers le nœud C avec les données collectées sur les différents nœuds impliqués dans la connexion. Le manager d'agents mobiles transmet les données au manager global du LIDS qui, après analyse, les transmet au module IHM, représenté sur la figure 11. La chaîne de connexions Telnet peut ainsi être visualisée par l'utilisateur du nœud C qui pourra décider de l'interrompre ou non.

8 Conclusions et travaux futurs

Nous avons présenté dans cet article l'architecture d'un IDS distribué pour les réseaux sans fil ad hoc. Les tests fonctionnels réalisés en laboratoire sont positifs : les attaques implémentées sont détectées en temps réel. L'utilisation des agents mobiles dans un environnement sans fil a été validé par le biais d'un prototype, mais des mesures de performances restent toutefois à réaliser, notamment dans un réseau constitué d'un grand nombre de nœuds et avec différentes conditions de trafic. Cette prochaine étape nécessitera peut-être la définition d'une nouvelle plate-forme à agents mobiles optimisée pour la détection d'intrusions et les réseaux sans fil. Dans notre prototype, la sécurité des LIDS s'appuie sur les mécanismes de sécurité du langage Java et de la plate-forme Aglet. L'ajout d'un agent mobile chargé d'évaluer l'intégrité globale du système constitué par

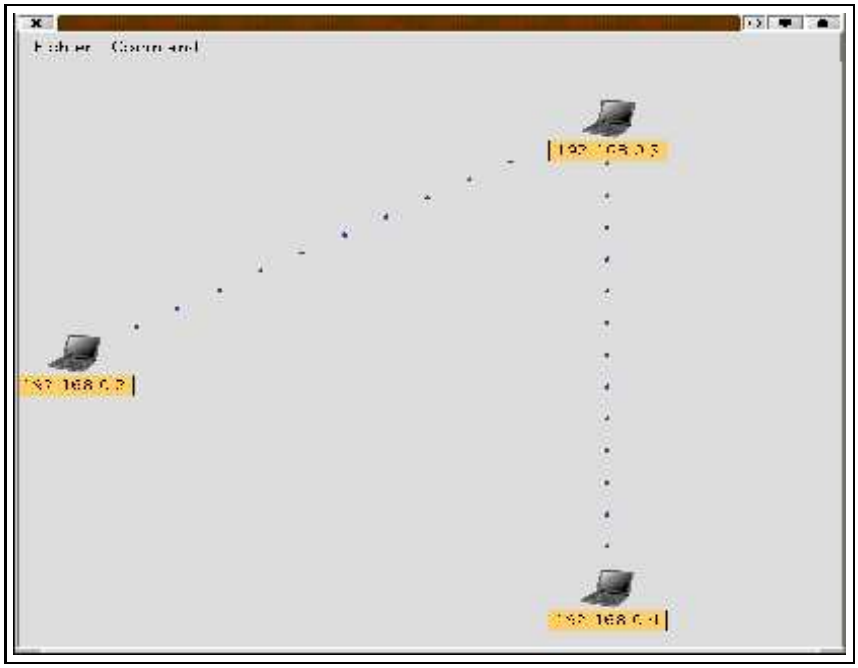

Fig. 11. Interface LIDS.

l'ensemble des LIDS est actuellement en cours d'étude. La conception modulaire des LIDS nous permet aussi d'évaluer les performances d'une détection comportementale ou mixte et de différentes sources d'information. La détection de l'attaque par rebonds Telnet est adaptée à la détection de tous types de connexions TCP, néanmoins les possibilités de détection des LIDS devront être étendues à un plus grand nombre d'attaques.

9 Remerciements

L'architecture distribuée du système de détection d'intrusions pour les réseaux sans fil ad hoc, présentée dans cet article, a été réalisée dans le cadre du projet RNRT pré-compétitif RAHMS (Réseau Ad Hoc Multiservice Sécurisé). Les auteurs tiennent ici à remercier le RNRT et les ministére ministères de l'industrie et de la recherche pour leur soutien. En outre, les auteurs remercient l'ensemble des participants au projet et tout particulièrement Patrick Albers et Olivier Camp (ESEO), Ludovic Mé et Ricardo Puttini (Supélec).

Références

1. Scott Corson, Joseph Macker. *RFC 2501, Mobile Ad hoc Networking (MANET).* IETF (1999)

2. Samir R. Das, Charles E. Perkins, et Elizabeth M. Royer. *Performance comparison of two on-demand routing protocols for ad hoc networks*. In Proceedings of the IEEE Conference on Computer Communications, INFOCOM (2000).
3. Cedric Adjih, Thomas Clausen, Philippe Jacquet, Anis Laouiti, Pascale Minet, Paul Muhlethaler, Amir Qayyum, Laurent Viennot. draft-ietf-manet-olsr-09.txt. *Optimized Link State Routing Protocol*. IETF (2003).
4. *RAHMS, Réseau Ad Hoc Multiservices Sécurisé*. Projet RNRT 1999 numéro 65. Dossier d'étude de l'architecture de sécurité. Rapport interne (2002).
5. L.Zhou and Z. Haas. *Securing Ad Hoc Networks*. IEEE Network, Vol 13, Nov.-Dec. 1999, pp.24-30 (1999).
6. H. Luo, P. Zerfos,J. Kong, S. Lu, and L.Zhang. *Self Securing Ad Hoc Wireless Networks*. Proceedings of the Seventh International Symposium on Computers and Communications, ISCC'02 (2002)
7. Patrick Albers, Olivier Camp, Jean-Marc Percher, Bernard Jouga, Ludovic Mé, and Riccardo Puttini. *Security in Ad hoc Networks : a General Intrusion Detection Architecture Enhancing Trust Based Approaches*. WIS 2002, Ciudad Real, Spain (2002).
8. Ludovic Mé et Cédric Michel. *La détection d'intrusions : bref aperçu et derniers développements*. Actes du congrès EUROSEC'99 (1999).
9. Jai Sundar Balasubraniyan, Jose Omar Garcia-Fernandez, David Isacoff, Eugene Spafford, Diego Zamboni. AAFID, *Autonomous Agents For Intrusion Detection*. Technical report 98/05, COAST Laboratory, Purdue University (1998).
10. Gregory B White, Eric A. Fish and Udo Pooch. *CSM - Cooperating Security Managers : a Peer Based Intrusion Detection System*. IEEE Networks, pp.20-23, January/February (1996).
11. Steven R. Snapp, James Brentano, Gihan V. Dias, Terrance L. Goan, L. Todd Heberlein, Che-Lin Ho, Karl N. Levitt, Biswanath Mukkherjee, Stephen E. Smaha, Tim Grance, Daniel M. Teal, and Doug Mansur. *DIDS, Distributed Intrusion Detection System*. Computer Security Laboratory, Department of Computer Science, University of California, Davis. (1992).
12. C. Ko, M. Ruschitzka, and K. Levitt. *Execution Monitoring of Security-Critical Programs in Distributed Systems : A Specification-based Approach*. Proceedings of the 1997 IEEE Symposium on Security and Privacy (1997)
13. Stuart Staniford-Chen, S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K. Levitt, C. Wee, R. Yip, D. Zerkle. *GrIDS, A Graph Based Intrusion Detection System for Large Networks*. Computer Security Laboratory, Department of Computer Science, University of California, Davis (1996).
14. Midori Asaka, Atsushi Taguchi, Shigeki Goto. *The Implementation of IDA : An Intrusion Detection Agent System*. IPA, Waseda University (1999).
15. Y.F Fou, F. Gong, C. Sargor, X. Wu, S. F. Wu, H. C. Chang, F. Wang. *JINAO, Design and Implementation of a Scalable Intrusion Detection System for the OSPF Routing Protocol*. Advanced Networking Research, MCNC Computer Science Dept, NC State University (1999).
16. J. Duarte de Queiroz, L. Fernando Rust da Costa Carmo and L. Pirmez. *Micael : An Autonomous Mobile Agent System to Protect New Generation Networked Applications*. Nucleo de Computacao Eletronica UFRJ, Brazil. In the Proceedings of RAID'99 (1999).

17. C. Krugel and T.Toth. *Flexible, Mobile Agent Based intrusion Detection for Dynamic Networks*. European Wireless 2002, Florence, Italy (2002).
18. G. Helmer, J. S. K. Wong, V. Honava,. L. Miller and Y. Wang. *Lightweight Agents For Intrusion Detection*. Journal of Systems and Software, *http ://citeseer.nj.nec.com/helmer00lightweight.html*
19. Nadia Boukhatem. *Les agents mobiles et leurs applications*. DNAC'99, Paris (1999).
20. Stuart Staniford-Chen and Todd Heberlien. *Holding Intruders Accountable on the Internet*. Proceeedings of the 1995 IEEE Symposium on Security and Privacy, pp.39-49, Oakland (1995).