Université Abou Bekr Belkaid
Tlemcen Algérie
جامعة أبي بكر بلقايد
تلمسان الجزائر
République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Laboratoire STIC

# THESE

**Présentée**

## A L'UNIVERSITE DE TLEMCEN
## FACULTE DE TECHNOLOGIE

Pour l'obtention du diplôme de

## DOCTORAT

**Spécialité :** " Systèmes et Réseaux de Télécommunications"

**Par**

*Mr SEDJELMACI Sid Ahmed Hichem*

# MISE EN ŒUVRE DE MECANISMES DE SECURITE BASES SUR LES IDS POUR LES RESEAUX DE CAPTEURS SANS FIL

**Soutenue en …….. 2013 devant le Jury:**

| | | |
|---|---|---|
| CHIKH Mohamed Amine | Professeur à l'université de Tlemcen | Président |
| SENOUCI Sidi Mohammed | Professeur à l'université de Bourgogne, France | Examinateur |
| MHAMED Abdallah | MC, HDR, à l'Institut de Télécom-Sud Paris | Examinateur |
| KECHAR Bouabdellah | MCA, à l'université d'Oran | Examinateur |
| FEHAM Mohammed | Professeur à l'Université de Tlemcen | Directeur de Thèse |
| GUYENNET Hervé | Professeur à l'Université de Franche Comté, France | Co-Directeur |

« Le génie est  fait d’un pour cent d’inspiration et de quatre vingt dix-neuf pour cent  de transpiration »

**Thomas Edison**

« On n’est jamais trop âgé pour s’instruire »

**Benjamin Franklin**

# Remerciements

Je remercie en priorité ALLAH LE TOUT PUISSANT de m'avoir donné le courage, la force et la volonté d'achever ce travail.

J'exprime mes remerciements à mon directeur de thèse Monsieur le professeur FEHAM Mohammed pour son soutien, sa bonté et sa générosité durant toutes ses années.

Mes remerciements vont à Monsieur GUYENNET Hervé co-directeur et professeur à l'université de Franche-Comté (France).

-Monsieur CHIKH Mohamed Amine président du jury et Professeur à l'université de Tlemcen.

-Monsieur MHAMED Abdallah examinateur et maître de conférence à l'institut de Télécom-sud Paris.

-Monsieur KECHAR Bouabdellah examinateur et maître de conférence à l'université d'Oran.

Mes grands remerciements et ma profonde gratitude au Professeur SENOUCI Sidi Mohammed à qui je dois beaucoup pour son aide et son soutien durant mes séjours de stage à l'université de Bourgogne (France). Ce fut un grand honneur pour moi de travailler à ses cotés.

Je remercie vivement mon oncle TABET AOUL Nasredine pour ses conseils, sa générosité, ses compétences et son expérience qui m'ont été d'une grande aide dans mon cursus universitaire.

Merci au STAFF de l'université de Tlemcen ainsi qu'au laboratoire STIC qui m'ont donnés les moyens d'achever ce travail.

Merci à tous mes amis de l'université de Tlemcen ainsi qu'à ceux de l'université de Bourgogne.

Un grand merci à mes parents et mes sœurs qui m'ont toujours encouragé à poursuivre mes études et à aboutir à ce but.

Merci à toute ma famille maternelle et paternelle qui m'a soutenu.

Je tiens à saluer et remercier toutes les personnes de prés ou de loin qui par leur contribution m'ont aidés à achever cette thèse de doctorat.

À la mémoire de mon grand père et

Ma grand mère

# Résumé

Les réseaux de capteurs sans fil (RCSF) ont attiré beaucoup d'attention en raison de leurs vastes applications dans les domaines militaires et civils. Cependant, les contraintes énergétiques et de mémoire et l'environnement hostile dont lesquels ils peuvent être déployés, rendent ce type de capteurs vulnérables aux attaques. De ce fait, la protection de ce type de réseau en utilisant des solutions de sécurité adaptées aux capteurs est un chalenge qui va être traité dans cette thèse.

L'énergie des nœuds est un point très important lors de la conception et l'implémentation de l'application. Cependant le processus de communication consomme une énergie nettement supérieure à celle causée par les opérations de calcul. Dans cette optique, plusieurs chercheurs travaillent sur cette problématique et proposent des protocoles de routage qui visent à réduire la quantité d'information échangée entre les nœuds dans le réseau. Parmi ces protocoles nous pouvons citer les algorithmes de clustering; leur objectif est d'élire un seul nœud dans chaque groupe (cluster) qui a la responsabilité de transmettre les donnés agrégées à la station de base.

Les systèmes de détection d'intrusion (IDSs) ont la capacité de détecter les attaques internes ou externes du réseau, contrairement à d'autres solutions de sécurité telle que la cryptographie qui empêche simplement les attaques externes de pénétrer dans le réseau. Les IDSs conçus pour les réseaux filaires ou ad hoc ne peuvent pas être implémentés directement dans le RCSF. De ce fait, il est impérativement important de concevoir un système de détection propre au réseau de capteurs, qui prend en considération les limites des RCSFs.

Dans cette thèse de doctorat, nous avons développé et implémenté un ensemble de modèles de détection d'intrusion pour les réseaux de capteurs sans fil à base de cluster (RCSFC) en tenant en compte les contraintes énergétiques et de mémoire des nœuds de capteurs .

**Mots-clés**: Réseaux de capteurs sans fil (RCSF), Cluster, Systèmes de détection d'intrusion (IDSs), Taux de détection, Faux positifs, L'efficacité, Consommation d'énergie

# Abstract

The wireless sensor networks (WSN) have attracted much attention due to their broad applications in military and civilian areas. However, energy and memory constraints and the hostile environment in which they can be deployed make them more vulnerable to attacks. As a result, there is a strong need to for security solutions that protect these types of network from malicious attacks. This technical challenge is the subject of this research.

It is widely known that in WSN, the energy of the nodes is a very important point in the design and implementation of the application. However the communication between the sensors nodes consumes much energy than the corresponding computational process. As a result, several researchers have investigated this problem and proposed some routing protocols that aim to reduce the amount of information exchanged between the nodes in the network. For example, the cluster-based algorithm attempts to elect a single node (called cluster-head) in each cluster that is responsible for transmitting the aggregated data to the base station.

In this context, the intrusion detection systems (IDSs) have the capability to detect both internal and external attacks; unlike other security solutions such as cryptography which simply prevents external attacks from entering the network. The IDSs designed for wired and adhoc networks, they cannot be implemented directly in the WSN. Therefore, it is absolutely important to develop specific IDSs for sensor networks that take into account the limitation of WSNs.

In this thesis, we have developed and implemented a set of models of intrusion detection for cluster-based wireless sensor networks, which do take into account the energy and memory constraints of sensor nodes.

**Key-words:** Wireless sensor networks (WSN), Clustering, Intrusion detection system (IDS), Detection rate, False positive, Efficiency, Energy consumption

# Sommaire

# Table des figures

# Liste des tableaux

# Introduction générale

Les avancées technologiques de la micromécanique, de la microélectronique et des communications sans fil ont permis le développement des capteurs minuscules, multifonctionnels et à faible coût. Cet ensemble de capteurs, qui collectent et transmettent des données environnementales vers un point centralisé, définissent un réseau de capteurs sans fil (RCSF). Parmi les caractéristiques de ce réseau, nous citons la taille réduite des capteurs permettant leur déploiement dans des environnements inaccessibles, l'auto-organisation du réseau et le fonctionnement autonome de ces capteurs.

Les RCSFs ont un énorme potentiel pour être utilisé dans des situations critiques comme les applications militaires. Cependant, ces applications sont souvent déployées dans des environnements hostiles, où les nœuds et la communication sont des cibles attrayantes pour les attaquants. De plus, vu les contraintes de miniaturisation, les nœuds de capteurs sont dotés de ressources limitées en terme de calcul, d'espace de stockage et d'énergie. Par conséquent, les ressources limitées de ces nœuds et les environnements hostiles dans lesquels ils pourraient être déployés, rendent ce type de réseaux très vulnérables à plusieurs types d'attaques similaires à celles survenant dans les réseaux adhoc.

Par conséquent, il est nécessaire d'utiliser des mécanismes efficaces pour protéger ce type de réseau. Toutefois il est bien connu, que les systèmes de détection d'intrusion (IDSs) sont des mécanismes de sécurité très efficaces pour protéger le réseau contre les attaques malveillantes ou l'accès non autorisé, contrairement à d'autres mécanismes telle que la cryptographie qui reste inefficace lorsque l'attaquant se trouve à l'intérieur du réseau. Par ailleurs, les techniques de détection d'intrusion doivent être conçues pour détecter et prévenir l'exécution des attaques les plus dangereuses. En outre, ces techniques doivent être légères pour convenir à la nature des ressources limitées du RCSF.

La consommation d'énergie est un facteur très important dans ce type de réseau. De ce fait, plusieurs chercheurs ont travaillé sur cette problématique en proposant une architecture réseau basée sur l'approche de clustering adaptée aux nœuds de capteurs. Cette architecture consiste en la construction d'un ou de plusieurs groupes (*clusetrs*) de nœuds, dont chacun d'eux dispose d'un chef de groupe élu pour la collecte des données émises par les membres de son groupe, puis l'agrégation et par la suite la transmission des données à la station de base. Cette architecture vise à minimiser la consommation d'énergie des nœuds et par conséquent le prolongement de la durée de vie du réseau. De ce fait l'idée que nous envisagions de procéder est d'intégrer les mécanismes de détection d'intrusion dans ce type de topologie.

Dans cette thèse nous allons présenter trois modèles de détection d'intrusion qui sont intégrés dans un réseau de capteurs à base de cluster. Chacun d'eux utilise une ou plusieurs combinaisons des politiques de détection à base de signatures d'attaques ou de détection d'anomalies. Dans nos premiers et troisièmes modèles, les processus de détection s'exécutent dans les membres du cluster et dans le cluster *head*; dans le second modèle les systèmes de détection d'intrusion sont intégrés dans chaque niveau: membres du cluster, cluster *head* et la station de base. Dans cette thèse nous visons à proposer de nouvelles stratégies pour sécuriser le réseau contre plusieurs types de menaces en prenant en considération les contraintes énergétiques des nœuds de capteurs. Les solutions de sécurité proposées sont implémentées dans des simulateurs et dans un réseau de capteurs réels, ce qui rend nos contributions utiles pour la communauté scientifique et industrielle. L'objectif commun de ces travaux réside dans le fait de détecter les attaques les plus dangereuses avec un taux de faux positifs faibles et une faible charge de communication et de calcul.

Cette thèse est organisée en cinq chapitres en plus d'une introduction générale et d'une conclusion générale :

- Le premier chapitre présente un aperçu sur les réseaux de capteurs, leur application ainsi que les différents types de topologies définis pour ce type de réseau. Par la suite, nous donnons quelques définitions sur plusieurs types d'attaques qui ciblent les différentes couches de la pile protocolaire. Finalement, nous fournissons un résumé de trois mécanismes de sécurité proposés par la communauté scientifique qui sont: la cryptographie, la stéganographie et le système de détection d'intrusion (IDS).
- Dans le deuxième chapitre, nous présentons tout d'abord les techniques de détection d'intrusion utilisées par les agents IDS. Par la suite, nous décrivons les exigences et les contraintes pour la conception de ce type d'agent dans les RCSF et les métriques d'évaluation des performances de ces systèmes de détection. Finalement un état de l'art sur les systèmes de détection d'intrusion dans les réseaux de capteur à base de cluster va être abordé.
- Dans le troisième chapitre, nous présentons notre première contribution. Cette dernière est un système de détection hybride intégré dans le réseau de capteurs à base de cluster. Ce système hybride est une combinaison entre les avantages de deux techniques de détection. La première concerne la détection d'anomalies à base des machines à vecteurs de support (SVM). La seconde est une détection basée sur les signatures des attaques. Cette approche a été implémentée dans un simulateur programmé en langage JAVA. Les performances de notre modèle sont comparées avec ceux d'autres modèles hybrides proposés dans la littérature, en particulier en termes du nombre de faux positifs générés par les systèmes de détection d'intrusion.

- La deuxième contribution est détaillée dans le quatrième chapitre. Dans cette partie nous exhibons un modèle hiérarchique de détection d'intrusion dans le réseau de capteurs à base de cluster. Dans ce mécanisme de sécurité le processus de détection d'intrusion s'effectue dans plusieurs niveaux. Le premier niveau consiste en un ensemble d'agents IDS qui applique les politiques de détection basées sur les règles pour la modélisation du comportement normal d'un nœud. Dans le niveau intermédiaire, un système de classification binaire à base des SVMs s'exécute dans chaque *cluster-head*. De plus ce dernier utilise un protocole de réputation afin d'évaluer le niveau de confiance de ses agents IDSs, car même ces agents peuvent être des nœuds malicieux. Le *cluster-head* est une cible attrayante pour l'attaquant en raison des données pertinentes qu'il présente. Pour cette raison, un niveau supérieur est introduit. Dans ce niveau chaque *cluster-head* surveille son *cluster-head* voisin, lorsque celui-ci présente un comportement malicieux, une alarme est envoyée à la station de base pour une meilleure confirmation du caractère malicieux du nœud soupçonné. Notre deuxième modèle de détection est implémenté dans les simulateurs TOSSIM et POWERTOSSIM est comparé avec d'autres schémas de détection en termes du nombre de faux positifs, du taux de détection et de la consommation d'énergie. De plus, une nouvelle métrique appelée « efficacité » est introduite pour calculer le temps nécessaire à un agent IDS pour détecter l'apparition du premier nœud malicieux.
- Dans le cinquième chapitre, nous abordons notre troisième contribution. Celle-ci constitue une nouvelle approche de détection basée sur le fait que lorsque la transmission de données subit au maximum deux sauts entre nœuds pour atteindre le *cluster-head*, alors tous les nœuds qui se situent dans le même cluster ont le même comportement. En tenant compte de cette démarche, confirmée par nos résultats de simulations, une politique de détection pour un certain nombre d'attaques visant les différentes couches de la pile protocolaire (réseau et physique) est proposée. De plus, dans ce chapitre nous avons proposé notre propre protocole à base de cluster qui est adapté à notre modèle de détection. Dans cette partie, les performances de notre troisième modèle de détection sont évaluées à l'aide du simulateur TOSSIM et d'une implémentation réelle dans des capteurs MICAZ dotés du système d'exploitation TINYOS. En plus du taux de détection, du taux de faux positifs et de la consommation d'énergie, une nouvelle métrique appelée « efficacité moyenne » est introduite pour déduire le temps requis aux agents IDS pour détecter toutes les attaques qui se produisent dans le réseau.

- Dans la partie conclusion nous résumons les résultats de nos contributions et nous proposons des perspectives conduisant à des mécanismes de sécurité plus robustes.

# Chapitre 1

# Réseaux de Capteurs Sans Fil & Sécurité

## 1) Introduction

Les progrès récents des communications sans fil et de la micro électronique ont permis le développement d'un nouveau genre de réseaux sans fil appelé réseau de capteurs sans fil (RCSF).
Ce dernier consiste en un très grand nombre de nœuds qui opèrent de façon autonome et qui communiquent entre eux via des transmissions radio courtes. Parmi les verrous majeurs de ces nœuds de capteurs, nous distinguons la limitation de leurs ressources en termes de capacité de calcul, l'espace de stockage des données et la faible portée radio. Les ressources limitées de ces nœuds et les environnements hostiles dans lesquels ils pourraient être déployés, rendent ce type de réseaux très vulnérable aux attaques. Dans ce contexte, une grande communauté de chercheurs tente de proposer des mécanismes de sécurité pour la prévention et la détection de tout type d'attaque, en tenant compte des contraintes de ce type de réseaux.

Dans ce chapitre, nous commençons par donner un bref aperçu sur le RCSF et leurs domaines d'application. Nous exposons ensuite les deux types de topologies existants dans les réseaux de capteurs: topologie plate et à base de cluster. La dernière section est consacrée aux problèmes de sécurité liés à ce type de réseau. Dans cette section, nous décrivons un certain nombre d'attaques pouvant cibler les différentes couches de la pile protocolaire, par la suite nous définissons quelques mécanismes de sécurité proposés dans la littérature pour protéger le réseau contre différentes menaces visant à perturber son bon fonctionnement.

## 2) Réseau de capteurs sans fil (RCSF)

### 2.1 Le nœud de capteur

Un nœud de capteur est un mini-dispositif, qui a la tâche de collecter les données, de les traiter puis les communiquer par la suite. L'intégration d'une application sur ce type de composant doit toujours prendre en compte certaines contraintes: la consommation d'énergie, l'espace mémoire, etc [**1**].

**1. Les composants hardware d'un capteur**

Le nœud de capteur est composé principalement de quatre unités: l'unité de captage, l'unité de calcul, l'unité de transmission et une source d'énergie (voir la Figure 1.1). Il peut contenir également des unités supplémentaires tel que le système de localisation (GPS) pour connaitre l'emplacement précis du nœud,...

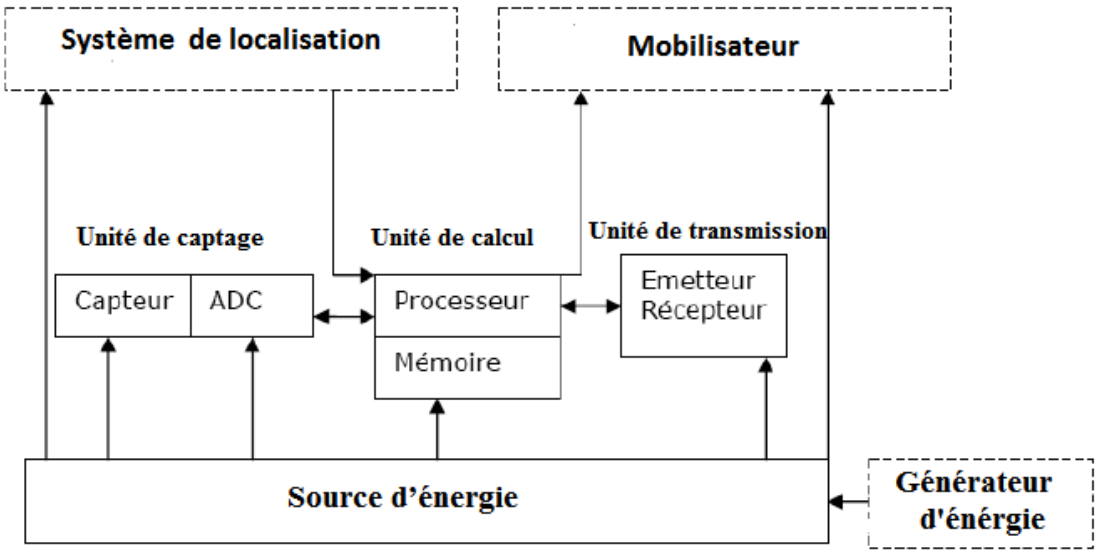



**Figure 1.1. Architecture d'un capteur [2]**

- **Unité de captage:** elle contient deux sous unités, la première permet la collecte des phénomènes physiques observés telle que la température, et la deuxième convertit le signal en signal numérique (ADC) pour être envoyé à l'unité de traitement.
- **Unité de calcul:** Elle est composée d'une mémoire de stockage et d'un processeur. Elle possède en plus deux interfaces [**3**] :

-La première, liée à l'unité de captage pour la réception des données collectées

- La seconde, liée à l'unité de transmission pour la transmission des données traitées.

- **Unité de transmission:** Elle est responsable de la transmission et la réception des données via un support de communication radio. Ce dernier peut être de type optique (comme dans les capteurs Smart Dust), où de type radio fréquence (MICA2) [**4**]. On note que la transmission consomme beaucoup d'énergie par rapport à l'unité de calcul. La Figure 1.2 résume la consommation d'énergie dans les différentes unités du capteur.
- **Source d'énergie:** elle est responsable de l'alimentation des différentes unités et elle réduit les dépenses, par exemple en mettant en veille les composants inactifs [**5**].

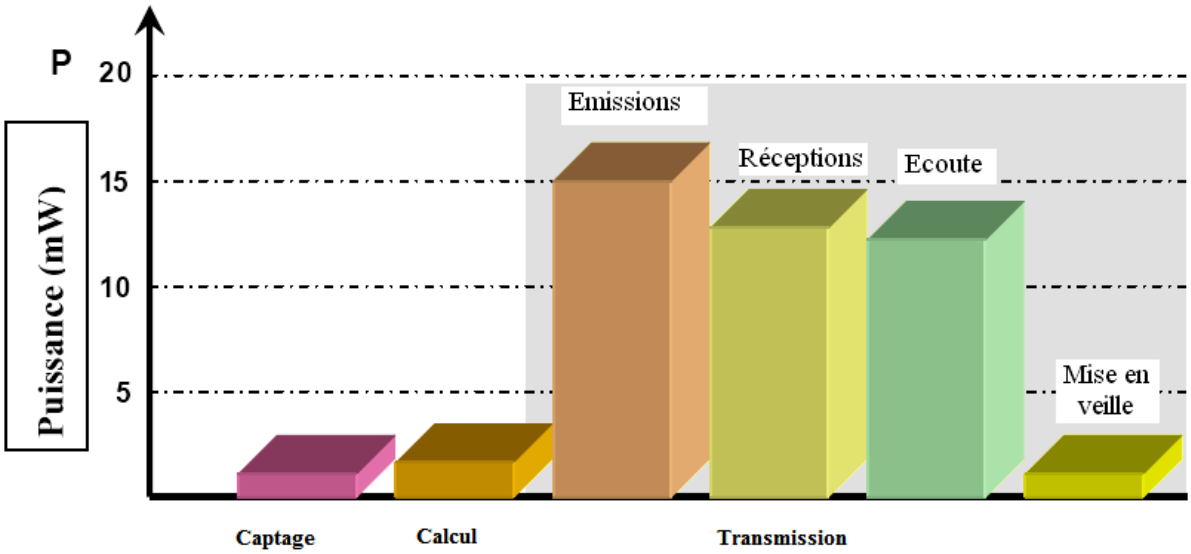



**Figure 1.2. Consommation d'énergie en captage, calcul et transmission [3]**

## 2. Les différentes Technologies des capteurs

Il existe plusieurs constructeurs de capteurs dans le monde; parmi ces fabricants les plus connus nous citons: Crossbow, Sun, EuroTherm, Dalsa et Moteiv. Le tableau 2.1 récapitule les principales caractéristiques de quelques types de capteurs.

| Constructeur | Modèle | Microcontrôleur | RAM(KB) | Radio | Système d'exploitation |
|---|---|---|---|---|---|
| Crossbow | MICA2 | Atmel Atmega 128L | 4 | Chipcon CC1000 433/915 Mhz | TinyOS |
| | MICAZ | Atmel Atmega 128L | 4 | Chipcon CC2420 2.4 Ghz IEEE 802.15.4 | TinyOS |
| | Telosb | Texas Instruments MSP430 MSP 430 | 10 | Chipcon CC2420 2.4 Ghz IEEE 802.15.4 | TinyOS |
| Moteiv | Tmote sky | Texas Instruments MSP430 | 10 | Chipcon CC2420 2.4 Ghz IEEE 802.15.4 | TinyOS |
| Sun | Sun spot | ARM920T | 512 | 2.4 Ghz IEEE 802.15.4 | Utilise la machine virtuel Squawk |

**Tableau 1.1. Caractéristiques de quelques nœuds capteurs**

## 2.2 Réseau de capteurs sans fil (RCSF)

Un RCSF est constitué d'un ensemble de nœuds qui communiquent entre eux de façon autonome via un lien radio. Dans ce type de réseau les capteurs collectent des données par exemple sur l'environnement; et sont par la suite acheminées à un point centralisé, appelé station de base. Cette dernière est généralement connectée à un ordinateur via internet. Ce type de réseau est généralement déployé dans des environnements hostiles et insécurisés.

La Figure 1.3 illustre une architecture simple d'un réseau de capteurs.

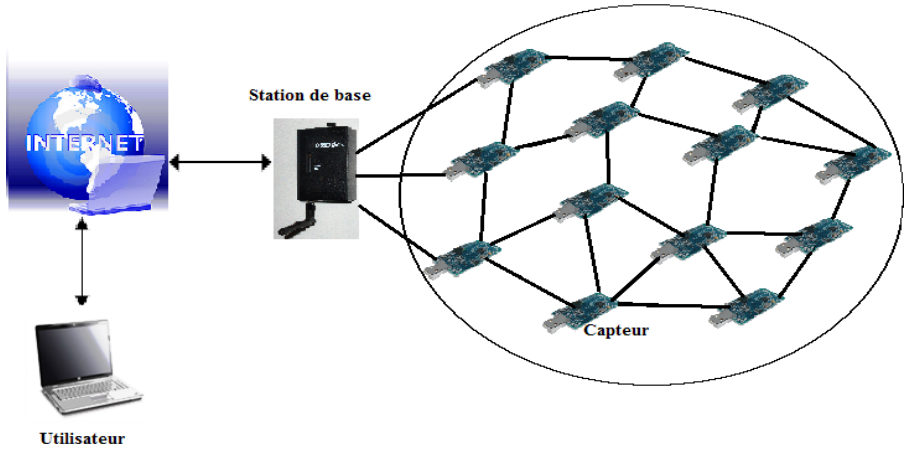



**Figure 1.3. Architecture d'un réseau de capteurs sans fil**

### 1. La pile protocolaire d'un RCSF

La pile protocolaire utilisée par les RCSFs est illustrée dans la Figure 1.4 [**6**][**7**]. Cette pile est constituée de cinq couches, une couche d'application, une couche de transport, une couche réseau, une couche de liaison de données et une couche physique. En plus de trois niveaux (plans) transverses qui sont **[8]** :

-Le niveau de gestion d'énergie: Contrôle la consommation d'énergie d'un nœud.

-Le niveau de gestion de la mobilité: Surveille la mobilité des nœuds de capteurs.

-Le niveau de gestion des tâches: Assure la distribution des tâches pour les nœuds de capteurs.

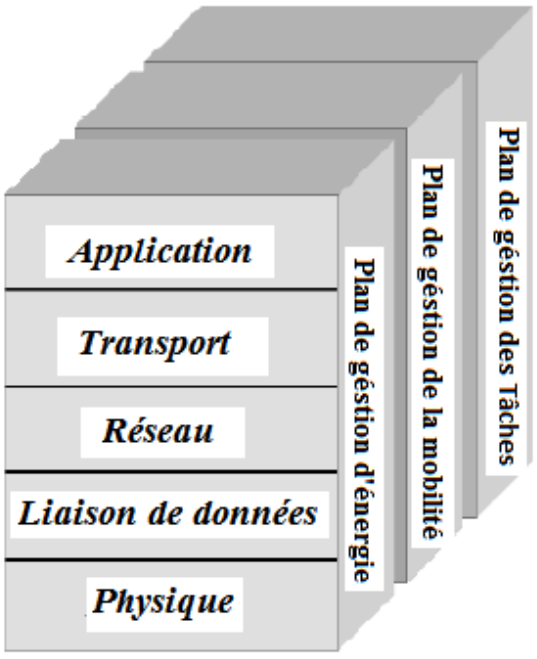



**Figure 1.4. Pile protocolaire dans un réseau de capteurs sans fil**

## 2. Domaine d'application

Les réseaux de capteurs sans fil sont utilisés dans une variété d'applications tels que la surveillance militaire, le domaine médical, des applications commerciales et des applications environnementales,...

- **Surveillance militaire.** L'utilisation des capteurs dans le domaine militaire est en pleine expansion, ces dispositifs peuvent être utilisés dans les opérations de surveillance des champs de bataille, la détection d'intrusion et reconnaissances des forces amies et ennemies. Parmi les travaux concrétisés dans ce domaine, nous pouvons citer les projets phares suivants: le projet DSN (*Distributed Sensor Network*)[**9**] développé par la DARPA (*Défence Advanced Research Projects Agency*), le projet WATS (*Wide Area Tracking System*) pour la détection des dispositifs nucléaires développés par le laboratoire *Lawrence Livermore National* **[10].**

- **Applications médicales.** Dans le domaine médical; les capteurs sont utilisés pour la surveillance des données physiologiques d'un patient. A titre d'exemple, la référence [**11**] propose une nouvelle plateforme pour la surveillance des personnes cardiaques en utilisant les capteurs pour la collecte des données ECG (la durée QRS, la durée entre deux piques R, l'amplitude du pique R) et le téléphone mobile pour la détection des pathologies cardiaques.

- **Applications environnementales.** Les capteurs sont récemment utilisés dans le domaine de l'agriculture. La fonction des capteurs dans ce domaine consiste à surveiller les taux de pesticides dans l'eau potable, le degré d'érosion, et le niveau de pollution de l'air en temps réel [**12**].

- **Applications commerciales.** Dans les entreprises, les réseaux de capteurs permettent de suivre le procédé de production à partir des matières premières jusqu'au produit final livré [**13**]. Grâce aux réseaux de capteurs, les entreprises peuvent offrir une meilleure qualité de service tout en réduisant les coûts [**14][15]**.

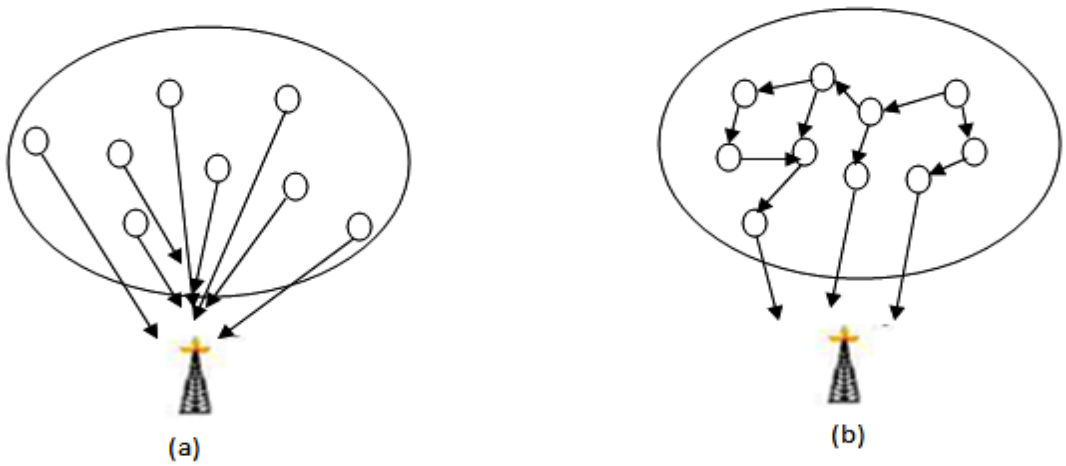
## 3. Architecture réseau

De façon générale, les architectures des réseaux de capteurs se présentent sous forme de deux topologies [**16**][**17**]:

**Topologie Plate**: Dans ce type de topologie, les capteurs communiquent entre eux afin d'acheminer l'information au nœud centralisé (station de base). Ce processus d'acheminement d'information peut prendre deux formes [**18**]: communiquer directement avec la station de base (Figure 1.5 (a)), ou via un mode multi-sauts (Figure 1.5 (b)). Parmi les protocoles de routage dans ce type de topologie nous pouvons citer: *Directed Diffusion* [**19**], SAR (*Sequential Assignment Routing*) [**20**] et SPIN (Security

*Protocols for Sensor Networks*) [**21**].Cependant, lorsque la taille du réseau augmente, sa gestion sera difficile et le protocole de routage aura du mal à bien acheminer les informations de la source à la station de base. De plus dans ce type de topologie (Figure 1.5 (a)) tous les nœuds peuvent envoyer leurs données à la station de base en utilisant une forte puissance, ceci peut conduire à la diminution de la durée de vie du réseau.

**Figure 1.5. Architecture de communication dans une topologie plate**




**Toplogie hiérarchique ou à base de cluster:** Dans cette architecture, le réseau est constitué d'un ensemble de groupe de capteurs (cluster), tel qu'il est illustré dans la Figure 1.6. Dans chaque cluster un chef de groupe appelé *cluster-head* a la responsabilité de collecter et gérer les informations à partir de ces nœuds membres, par la suite agréger ces données et les envoyer à la station de base. L'avantage majeur de ce type d'architecture est le prolongement de la durée de vie du réseau de capteurs. Ce résultat est achevé en désignant le *cluster-head* comme étant le nœud responsable de la transmission des informations (agrégées). Ce procédé est meilleur que celui où tous les nœuds envoient leurs données à un emplacement distant. Parmi le grand nombre de protocoles de routage basés sur le concept du cluster, proposés dans la littérature, nous citons: LEACH (*Low Energy Adaptive Clustering Hierarchy*) [**22**], HEED (*Hybrid Energy-Efficient Distributed Clustering*) [**23**], PEGASIS (Power-*Efficient Gathering in Sensor Information Systems*) [**24**], TEEN (*Threshold-sensitive Energy Efficient sensor Network protocol*) [**25**]. Dans notre étude nous nous sommes intéressés à cette d'architecture en raison du fait qu'elle est mieux adaptée aux nœuds de capteurs.

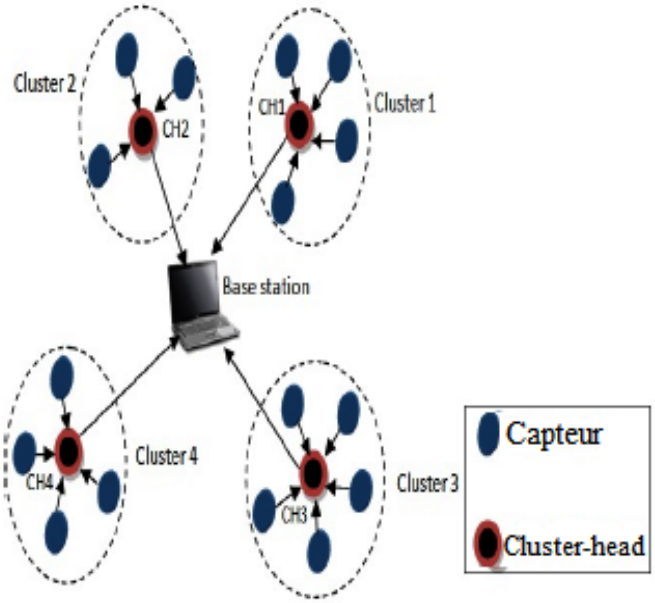



**Figure 1.6. Topologie à base de cluster**

## 3) Vulnérabilité et exigence de sécurité dans les réseaux de capteurs sans fil

Le réseau de capteurs sans fil est souvent déployé dans un environnement hostile comme le champ de bataille, ce qui peut être une cible attrayante pour les attaquants. Par conséquent, il est impérativement nécessaire d'intégrer un mécanisme de sécurité dans ces réseaux vulnérables. Dans cette section, nous allons résumer quelques types d'attaques des plus dangereuses qui ciblent les différentes couches du réseau et nous décrivons les mécanismes de sécurité proposés par la communauté scientifique et utilisés dans le secteur industriel.

### 3.1 Les attaques dans le RCSF

Les attaques dans le RCSF connaissent plusieurs classifications, mais les plus connues sont regroupées selon les catégories ci-dessous [**26**][**27**][**28**] :

**Classification selon l'origine:**

**-Attaque interne:** Elle se produit à l'intérieur du réseau. Dans ce cas, l'intrus est aperçu par les autres nœuds comme étant un nœud normal. Ce phénomène se produit lorsque le nœud malveillant connait la clé de chiffrement et peut enclencher le processus de cryptage et décryptage. Par conséquent, il peut

accéder aux messages chiffrés échangés entre les nœuds. Cette menace est la plus sévère et la plus difficile à détecter.

**-Attaque externe**: Ce type de menace se trouve à l'extérieur du réseau, en d'autres termes, il ne fait pas partie des nœuds déployés par l'administrateur du réseau. Un attaquant externe ne peut pas avoir accès aux informations pertinentes stockées par les nœuds du réseau (telles que les clés de chiffrement).

L'objectif de notre thèse est justement de détecter ces deux types d'attaques.

**Classification selon la nature:**

**-Attaque passive:** Dans cette catégorie, la technologie de communication sans fil constitue une vulnérabilité qui peut aisément être exploitée par un attaquant [**4**]. L'intrus collecte tous les paquets qui se trouvent à sa porté radio sans modifier leurs contenus. Un adversaire passif ne fait que menacer la confidentialité des données [**5**].

**-Attaque active:** Dans cette catégorie, l'attaquant vise à perturber le bon fonctionnement du réseau et à modifier le contenu des paquets envoyés par les nœuds légitimes.

Aussi, les réseaux de capteurs sont sensibles aux attaques allant de la couche physique jusqu'à à la couche transport. Wood et Stankovic [**29**] donnent la classification suivante des attaques du réseau de capteurs:

**-Les attaques ciblant la couche physique:** L'attaque *Jamming* est la plus fréquente dans la couche physique d'un RCSF. Celle-ci vise à créer des interférences pour occuper les canaux et empêcher les capteurs de communiquer normalement. Dans le chapitre 5 de cette thèse, nous décrivons quelques types d'attaques *Jamming* que nous avons l'intention de détecter.

-**Les attaques ciblant la couche liaison:** Les attaques de collisions ou d'épuisement des ressources (Resource *exhaustion*) peuvent être lancées contre la couche liaison de données d'un réseau de capteurs. L'attaque *Resource exhaustion* consiste à inonder le réseau avec un trafic indésirable afin d'épuiser les ressources des capteurs [**30**]. Ce résultat est obtenu en envoyant un nombre considérable de paquets.

-**Les attaques ciblant la couche réseau:** Parmi les attaques possibles qui ciblent la couche réseau nous citons: *black holes, selective forwarding, wormholes, spoofed, altered, et replayed packets, sinkhole et hello flood, acknowledgement spoofing* .

- ***Black holes.*** Dans cette attaque, l'intrus prétend être dans le plus court chemin vers la station de base ou le *cluster-head* en générant une puissance élevée de transmission. Le RCSF est

vulnérable à ce genre d'attaque en raison de leur paradigme de communication, où tous les nœuds acheminent les données vers un nœud centralisé. Par conséquent, tous les paquets reçus par ce nœud malveillant seront supprimés.

- ***Selective forwarding.*** Dans cette attaque, l'attaquant empêche la transmission de certains paquets. Ces derniers seront par la suite supprimés par ce nœud malveillant.
- ***Wormholes.*** Connues aussi sur le nom de *tunneling*. Dans cette attaque, un adversaire peut recevoir des messages et les rejouer dans différentes parties à l'aide d'un tunnel entre deux nœuds malicieux [**28**].
- ***Spoofed, Altered, et Replayed packets.*** L'attaquant surveille les transmissions, intercepte les paquets, puis modifie les informations de routage et les réutilise pour générer des faux messages d'erreur.
- ***Sinkhole et Hello flood.*** La caractéristique commune entre les deux attaques, est que le nœud malveillant va convaincre ses voisins que c'est le nœud le plus proche de la station de base ou du *cluster-head* en utilisant une puissance de transmission élevée. Par conséquent tous les paquets reçus seront modifiés et transmis à la station de base ou à l'utilisateur.
- ***Acknowledgement spoofing.*** Dans cette attaque, l'intrus tente de convaincre l'expéditeur que le lien faible est fort ou qu'un nœud mort est vivant [**31**]. Par conséquent, tous les paquets qui passent par ce lien ou ce nœud seront perdus.

-**Les attaques ciblant la couche transport:** Enfin, la couche de transport peut être attaquée par l'attaque d'inondation ou une attaque de désynchronisation. Le but des attaques d'inondation est d'épuiser les ressources mémoires d'un nœud en émettant un nombre considérables d'informations, tandis que l'attaque de désynchronisation modifie les numéros de séquence des paquets afin de perturber le protocole de communication [**32**].

## 3.2 Mécanismes de sécurité

De nombreux chercheurs se concentrent sur la sécurité des RCSFs car les caractéristiques de ce type de réseaux peuvent causer un risque potentiel d'attaque. Les mécanismes de sécurité contre les attaques ou les comportements malveillants proposés dans la littérature sont classés en trois catégories:

**1. Techniques cryptographiques**

Elles sont utilisées pour assurer l''authentification, l'intégrité et la confidentialité des données. Les opérations cryptographiques sont basées sur des primitives telles que les fonctions de hachage, le chiffrement symétrique et la cryptographie à clé publique [**33**]. La cryptographie protège le réseau uniquement contre les attaques externes. Cependant ce type de mécanisme ne peut pas détecter les attaques internes lorsque l'attaquant connaît les clés de chiffrement et les utilise pour effectuer les

opérations de cryptage et décryptage. La cryptographie est constituée de trois propriétés fondamentales qui sont :

- **Confidentialité:** La confidentialité vise à rendre les informations inaccessibles aux personnes non autorisées. Pour cela la solution adaptée est l'utilisation des algorithmes du chiffrement symétrique ou asymétrique. Dans le chiffrement symétrique, une même clé est utilisée entre deux nœuds communicants pour chiffrer et déchiffrer les données. Dans le chiffrement asymétrique, deux clés différentes sont générées par le destinataire (la station de base): une clé publique diffusée à tous les nœuds du réseau pour chiffrer les données qui sont par la suite émis au destinataire. Une clé privée, maintenue secrète au niveau du destinataire, sert pour le déchiffrement de ces données reçues.
- **L'intégrité:** Elle vise à assurer que les données qui circulent entre les nœuds ne puissent être falsifiées ou modifiées par les intrus. La solution qui assure cette propriété est la fonction de hachage [**3**].
- **Authentification:** C'est le service le plus important car on ne pourra pas assurer une confidentialité ou une intégrité de messages échangés si, dès le départ, nous ne sommes pas sûrs de communiquer avec le bon nœud [**3**]. Cette solution assure que les sources de données ne parviennent pas d'un nœud malveillant. L'authentification des données est assurée grâce au Code d'Authentification de Message (CAM), ou MAC en anglais (*Message Authentication Code*)[**34**].

**2. Stéganographie**

L'objectif principal de la stéganographie est de cacher ou d'intégrer un message, soit dans un autre message ou dans un ensemble de données multimédia (image, son, etc). Cependant, en comparaison avec les techniques cryptographies la stéganographie requiert plus de ressource de traitement, ce qui nécessite beaucoup d'efforts pour l'intégrer dans les RCSFs en raison de leurs contraintes.

**3. Système de détection d'intrusion (IDS :** *Intrusion Detection System***)**

Contrairement à la cryptographie, ce système a la capacité de détecter avec une grande précision les attaques internes. Ce mécanisme permet de détecter les activités anormales ou suspectes sur la cible analysée et déclenchera une alarme lorsqu'un comportement malveillant se produit. Nous croyons fermement que l'IDS est la solution la plus utile pour la détection des attaques à la fois internes et externes. Dans cette thèse nous allons focaliser notre travail sur ce type de mécanisme en proposant et concevant des nouveaux systèmes de détection d'intrusion pour l'identification et la prévention d'un certain nombre d'attaques.

**Les principaux composants d'un agent IDS.** L'agent IDS est installé dans la couche application, celui-ci est constitué de 3 composants (ou modules). Ces composants sont illustrés dans la Figure 1.7 et définis comme suit:

1) **Collecte de données.** Ce module est responsable de la capture des paquets au sein de la portée radio du nœud IDS.
2) **Détection d'intrusion.** L'agent IDS analyse les paquets capturés en ce basant sur une politique de détection. Parmi ces politiques, il y'a la détection à base de signature d'attaquant et la détection d'anomalie. Ces techniques seront détaillées dans le chapitre suivant.
3) **Prévention.** La prévention d'intrusion est un ensemble de tâches ayant pour but d'anticiper et de stopper les attaques [**35**]. Ces tâches peuvent être définies par exemple comme l'envoi d'une alarme par l'IDS à la station de base, par la suite ce dernier éjecte le nœud suspect du réseau et applique la mise à jour des clés.

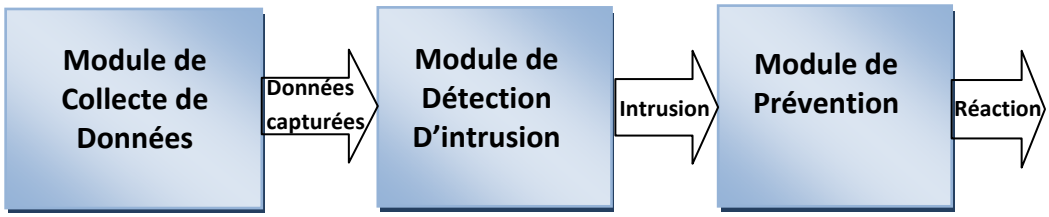



**Figure 1.7. Les composants d'un agent IDS**

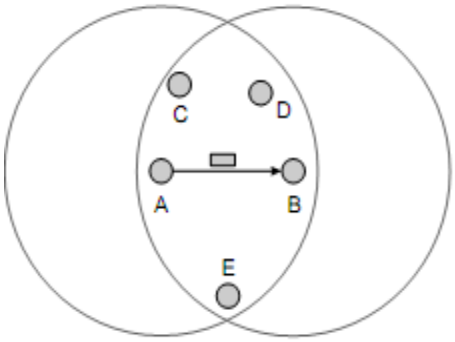
**Les différentes technologies des IDSs.** Il existe deux grandes technologies distinctes d'IDS :

- Les Systèmes de détection d'intrusion réseau (*Network Based Intrusion Detection System*). Ces systèmes visent à intercepter et analyser les paquets qui circulent dans le réseau. Toutes les communications dans le réseau sans fil sont menées sur l'air et un nœud peut entendre le trafic passant à partir d'un nœud voisin (le mode promiscuité) [**36**]. Par conséquent, les nœuds peuvent mutuellement vérifier le trafic réseau. Cette technologie applique ce concept, l'IDS écoute le trafic et examine individuellement chaque paquet.
- Systèmes de détection d'intrusion basés sur l'hôte (*Host Based Intrusion Detection System*). Analyse exclusivement les données concernant le nœud où l'IDS est installé. Toute décision prise est basée sur les informations recueillies à ce nœud. Ces IDSs utilisent deux types de source pour fournir une information sur l'activité: les fichiers logs (fichier qui enregistre toute activité sur un système en veille), et les traces d'audit (paquets entant/ sortant du nœud, etc.) [**37**]. Cette technologie permet de déterminer l'impact d'une attaque sur le nœud concerné.

**Types d'agents de détection.** Les agents IDS peuvent être classés en deux types : agent local et agent global.

- **Agent local.** La tâche des agents locaux est de découvrir toute attaque ou menace pouvant affecter le comportement normal des nœuds de capteurs en analysant uniquement les sources d'information locale (i.e. paquet reçu et émis, les mesures d'environnement captées). Cet agent peut avoir la même fonction qu'un système de détection d'intrusion basé sur l'hôte.
- **Agent global.** Chaque agent global surveille le comportement de ces voisins immédiats en analysant leurs paquets envoyés et reçus. Cet agent est basé sur le concept de diffusion (*broadcast*) de communication dans le réseau sans fil. Celui-ci peut se comporter comme un chien de garde (*watchdogs*) [**38**], le système de détection d'intrusion basé sur l'approche de chien de garde est illustré dans la Figure 1.8, d'où les nœuds C, D et E surveillent la communication du lien A et B. Dans [**39**] les auteurs s'inspirent de cette approche et appliquent des agents (nommés *spontaneous watch dog*) dans les RCSFs pour la surveillance et la détection des nœuds malicieux. Cet agent global peut avoir la même fonction qu'un système de détection d'intrusion réseau.

Ces deux agents (local et global) se trouvent dans le même nœud et l'activation de ces agents s'effectue selon le besoin. En d'autres termes l'activation simultanée ne peut pas être faite à cause des contraintes énergétiques des capteurs.




Les nœuds C, D et E peuvent être des chiens de garde du lien A⟶B

**Figure 1.8. Détection d'intrusion basée sur le concept de chien de garde [40]**

## 4) Conclusion

Nous avons présenté dans ce chapitre quelques définitions de base sur les RCSFs ainsi que leurs domaines d'applications. Nous avons défini par la suite quelques attaques proposées dans la littérature ciblant les différentes couches d'un RCSF, d'où l'objectif de cette thèse est la détection de certaines d'entre elles. Finalement un résumé a été donné concernant les différents mécanismes de sécurité proposés dans la littérature, en indiquant leurs avantages et inconvénients.

Les systèmes de détection d'intrusion restent les plus fiables pour l'identification de toutes les attaques malveillantes, cependant la conception de ce genre de système pour le réseau de capteurs doit toujours prendre en compte les caractéristiques de ce type de dispositifs (les contraintes énergétiques et de mémoire).

Dans le chapitre suivant, nous allons discuter sur les différentes techniques de détection d'intrusion utilisées par les agents IDS. La topologie à base de cluster est la mieux adaptée pour les nœuds capteurs car elle vise à prolonger la durée de vie du réseau, de ce fait un état de l'art sur les IDSs appliqués à ce type de topologie sera également présenté.

# Chapitre 2

# Etat de L'art et Problématique:

# Système de Détection D'intrusion (IDS) dans le Réseau de Capteurs

## 1) Introduction

De nombreux chercheurs se concentrent actuellement sur la sécurité des réseaux de capteurs sans fil (RCSF s) car les caractéristiques à la fois de l'infrastructure sans fil et de ces capteurs peuvent causer des risques potentiels d'attaques sur ce type de réseau. La cryptographie définie comme étant la première ligne de défense est inefficace lorsque l'attaquant se trouve à l'intérieur du réseau. La stéganographie est un mécanisme coûteux en terme de calcul; de ce fait elle est inappropriée pour les RCSFs. Le système de détection d'intrusion (IDS) défini comme étant la seconde ligne de défense, permet la détection et la prévention des attaques internes et externes.

En raison des différents types d'attaques (internes et externes) que le système de détection d'intrusion peut détecter, de nombreuses recherches dans l'application de la technologie des IDS dans les réseaux adhoc ont été effectuées. Par contre, cette recherche n'a pas progressé dans les réseaux de capteurs à cause du concept de détection d'intrusion qui n'est pas clair dans le contexte de ces réseaux. Dans [**39**], les auteurs affirment qu'il est impossible de migrer les solutions de sécurité utilisées dans le réseau adhoc directement dans le RCSF. Par conséquent la solution de détection d'intrusion et de prévention proposée dans ce type de réseau doit toujours prendre en compte les contraintes énergétiques et d'espaces mémoires des capteurs.

Dans ce chapitre, un aperçu sur les agents IDS dans le RCSF va être présenté en expliquant les politiques de détection utilisées par ces agents. Par la suite, nous nous focaliserons sur les IDSs dans les Réseaux de Capteurs Sans Fil à base de Cluster (RCSFC). Dans cet axe, nous aborderons un point très important concernant l'emplacement optimal des agents IDS dans ce type de réseau. Par la suite, nous procéderons à une étude sur les solutions existantes de détection d'intrusion qui sont appliquées à ces réseaux à base de cluster. En particulier, nous mettrons l'accent sur les caractéristiques de chacune de ces solutions de sécurité en indiquant leurs points forts et leurs faiblesses.

## 2) Les IDSs dans les RCSFs

Selon Roman et al [**39**] les solutions d'IDS développées pour les réseaux ad hoc [**41**], [**42**], [**43**], [**44**] ne peuvent pas être appliquées directement sur les réseaux de capteurs, et ceci est du à la différence de ces deux types de réseaux [**39**] :

- Dans les réseaux adhoc, chaque nœud est généralement géré par un utilisateur humain. Contrairement au RCSF où tous les nœuds sont indépendants, ces capteurs envoient leurs données captées à la station de base. Cette dernière est généralement gérée par un utilisateur humain.
- Les ressources énergétiques sont plus limitées dans les nœuds de capteurs par rapport aux nœuds adhoc.
- La tâche des réseaux de capteurs est très spécifique, par exemple la mesure de la température dans un champ agricole. Par conséquent, les modules *hardware* et les protocoles de communications doivent dépendre de l'application envisagée.
- La densité des nœuds dans les réseaux de capteurs est plus élevée que dans les réseaux adhoc.

Ainsi, il est nécessaire d'introduire un mécanisme de détection d'intrusion propre aux réseaux de capteurs. Dans cette section, nous discuterons des politiques de détection appliquées par les agents IDS et par la suite, les exigences des RCSFs que nous devons prendre en compte pour la conception des systèmes de détections d'intrusion. Finalement, nous expliquerons les différentes métriques pour l'évaluation des performances du système de détection.

### 2.1 Les politiques de détection d'intrusion

Comme le montre la Figure 2.1, les politiques de détection des intrusions dans le RCSF peuvent être classées en deux grandes techniques, détection à base de signature et détection d'anomalie [**45**].

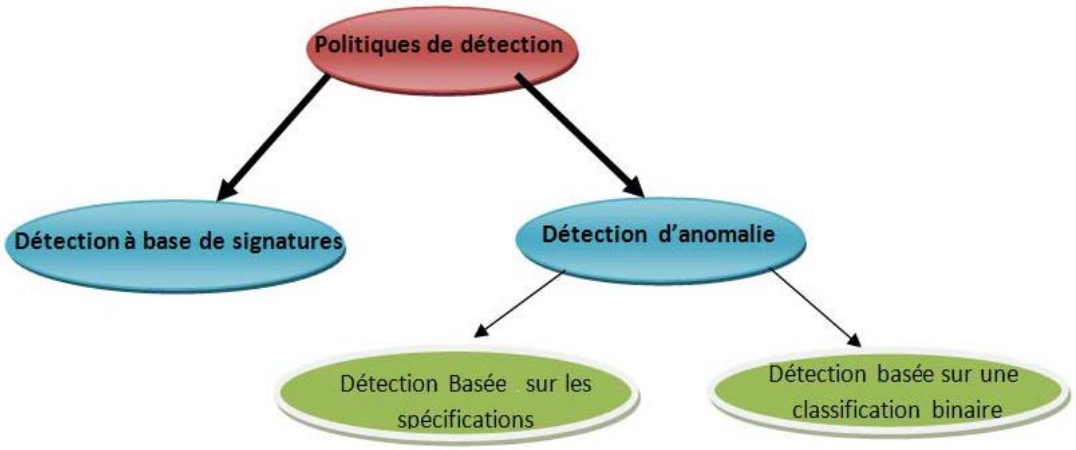



**Figure 2.1. Les techniques de détection d'intrusion**

### a. Détection à base de signatures *(Signature-based detection)*

Cette approche est basée sur la comparaison du comportement observé d'un nœud avec un ensemble de signatures d'attaques stockées dans sa mémoire. Si une correspondance est trouvée, le nœud analysé est défini comme étant un attaquant. Cette technique est précise pour la détection des attaques connues. L'inconvénient de cette technique est l'incapacité pour l'identification des attaques inconnues. La fiabilité de cette technique s'appuie sur la mise à jour continue de ces signatures, par conséquent ceci induit à une surcharge de la mémoire.

### b. Détection d'anomalie *(anomaly detection)*

Cette approche est basée d'abord sur la modélisation du comportement normal d'un nœud et puis identifier tout ce qui s'écarte de ce modèle comme étant une anomalie. Cette technique est composée de deux catégories:

- **Détection basée sur une classification binaire**

Cette catégorie utilise un algorithme d'apprentissage supervisé afin de modéliser le comportement normal. Le principal avantage de cette technique est la capacité de détection des attaques inconnues, mais elle génère un coût élevé de calcul, ce qui conduit à une diminution de la durée de vie du nœud. Parmi les techniques de détection proposées dans la littérature pour les réseaux de capteurs sans fil, nous pouvons citer: le plus proche voisin, réseaux de neurones, machines à vecteurs de support (SVM) [**30**][**46**][**47**][**48**]. L'objectif de ces algorithmes d'apprentissage est de classer les données comme étant normales ou anormales (anomalie) avec un faible taux de faux positifs. Par ailleurs, les SVMs sont les mieux adaptés pour les IDSs comparés aux autres algorithmes de classification car ils permettent une meilleure classification des données avec un temps d'apprentissage réduit, en plus ils génèrent un taux d'erreur de classification faible [**47**]. De ce fait, L'utilisation de cette technique d'apprentissage dans le RCSF doit prendre en considération les contraintes énergétiques des capteurs. Dans la sous section 2.2, quelques informations de base sur les SVMs vont être données. En particulier, nous décrivons l'avantage des SVMs utilisés dans le mode distribué par rapport au mode centralisé dans le RCSF.

- **Détection Basée sur les spécifications (*Specification-based detection*)**

Cette catégorie modélise le comportement normal en utilisant un ensemble de règles. L'avantage de cette technique est la capacité à détecter les attaques inconnues avec un faible coût de calcul. Cependant, la fiabilité de cette approche repose sur la mise à jour continue des règles au fil du temps.
Plusieurs chercheurs ont défini des règles afin de détecter certains types d'attaques. En effet, dans [**49**] les auteurs proposent un ensemble de règles afin de détecter des attaques du type: *Hello flood, Black hole, Selective forwarding, Jamming, Wormhole*, et Déni de service (DOS). Ces règles sont illustrées dans le Tableau 2.1. Une mise à jour continue de ces règles doit être appliquée pour une

détection efficace de ces attaques. Dans ce travail les auteurs ne mentionnent aucun mécanisme de mise à jour de ces règles.

| Nom de la règle | Description de la règle | Attaques détectées |
|---|---|---|
| Règle de l'intervalle | Le temps de réception entre deux paquets successifs ne doit pas être supérieur ou inferieur à un certain seuil | *Hello flood* |
| Règle de retransmission | L'agent IDS surveille si le nœud retransmet le paquet reçu à son voisin | *Black hole et Selective Forwarding* |
| règle de la répétition | Nombre de retransmissions du même message par le nœud | Déni de service (DOS) |
| Portée de transmission radio | Le message reçu par L'agent IDS doit être de provenance de l'un des ses nœuds voisins | *Wormhole, Hello flood* |
| Règle de brouillage | Le nombre de collisions associées à un message doit être inférieur au nombre prévu de collisions | *Jamming* |
| Règle de delay | Une anomalie est détectée si le message n'est pas transmis en temps demandé. | *Jamming et* DOS |

**Tableau 2.1. Règles pour la détection des attaques dans les réseaux de capteurs**

## 2.2 Détection d'anomalie à base des SVMs

Machines à vecteurs de support ou séparateurs à vastes marges sont l'objet d'une méthode d'apprentissage supervisée développée par Vapnik en 1995. Cette méthode est une alternative récente de classification binaire; elle repose sur la construction d'un hyperplan qui sépare les données en deux classes. Une multitude d'hyperplan peut être définie, le principe de la SVM est de déterminer une marge maximale entre les données d'apprentissage et l'hyperplan séparateur. On note que, la marge est la distance entre l'hyperplan et les données les plus proches. Cet hyperplan est défini comme étant la solution optimale. Les données d'apprentissage dans ce cas sont appelées **vecteurs de support**, comme le montre la Figure 2.2. La fonction des SVMs est la maximisation de la marge, de ce fait la communauté scientifique parle de séparateurs à vaste marge. Dans le cas où la SVM ne peut pas séparer les données en deux classes (séparation non linéaire), ce problème peut être résolu en utilisant les fonctions noyau (*kernel*). Le principe de ces fonctions est de permettre la transformation d'un problème de séparation non linéaire dans l'espace de représentation en un problème de séparation linéaire dans un nouvel espace de plus grande dimension [**50**].

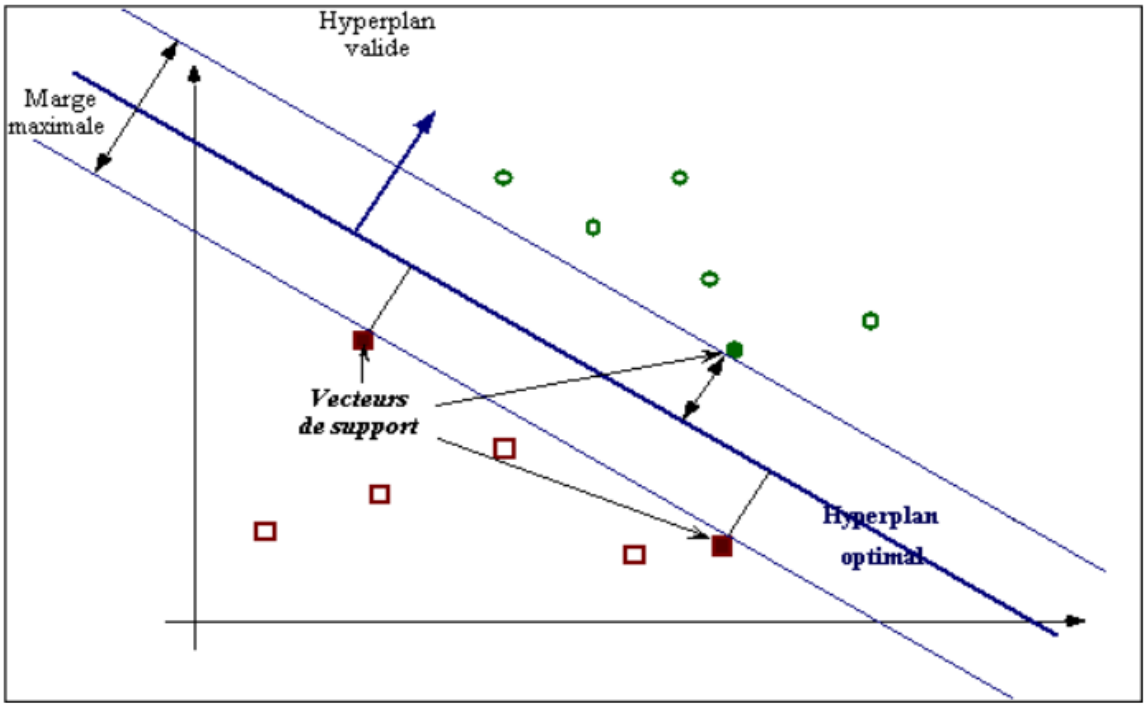



**Figure 2.2. Hyperplan optimal et vecteurs de support**

Les SVMs peuvent être utilisées de façon centralisée ou distribuée.

- Dans le premier cas, la SVM, qui est intégrée à la station de base, recueille les paquets provenant de tous les nœuds du réseau et applique le processus d'apprentissage. Cette approche oblige les nœuds d'envoyer une quantité considérable de données à un emplacement distant, ceci conduit à une charge (*overhead*) élevée de communication, par conséquent une diminution de la durée de vie des nœuds de capteurs. Cette méthode d'apprentissage centralisé permet une meilleure classification des données avec un taux d'erreur de classification proche de zéro [**51**]. Par ailleurs, elle n'est pas adaptée aux capteurs vue leurs ressources limitées.
  Kaplantzis et al. [**48**] ont travaillé sur le système de détection d'intrusion centralisée basée sur la machine à vecteurs de support pour détecter les attaques *selective forwarding* et *black hole* dans le RCSF. L'IDS qui s'exécute dans la station de base utilise une catégorie de SVM basée sur l'apprentissage à une seule classe (*one-class SVM*). Cette classe est l'activité normale du réseau. La station de base collecte les données transmises par tous les nœuds du réseau, extrait les vecteurs d'entrée d'apprentissages (*features vector*) qui sont la bande passante et le nombre de sauts et en dernier applique le processus d'apprentissage. Les auteurs notent que lors de l'apprentissage du comportement normal du réseau il n'y a aucune activité d'attaque. Lorsque ce processus est achevé, le processus de test est lancé et les attaques sont introduites dans le réseau. Dans ce cas, la SVM utilise les données d'entraînement (*training data*) pour la détection de ces attaques. Dans les simulations, les auteurs affirment que l'IDS peut détecter

avec une grande précision les attaques *selective forwarding* et *black hole*. Toutefois, ce schéma ne peut détecter que deux sortes d'attaques, et présente un faible taux de détection lorsque le nombre de *selective forwarding* n'est pas aussi important dans le réseau. Par ailleurs, la station de base ne peut pas gérer tous les paquets envoyés par les nœuds, ce qui induit qu'un grand nombre de paquets ne seront pas analysés par la SVM.

- Dans l'approche distribuée, tous les nœuds de capteurs calculent les vecteurs de support. Ces derniers sont moins nombreux que les données d'entrée utilisées pour le processus d'apprentissage. Ces vecteurs clés sont alors échangés entre les nœuds, à l'exception de l'approche centralisée où tous les paquets sont envoyés à un nœud distant (station de base). Par conséquent, l'approche distribuée induit une faible consommation d'énergie des capteurs. De nombreux auteurs ont mentionné que cette approche est adaptée à l'exigence des nœuds de capteurs en termes de coût d'énergie et présente un taux de classification proche de celle du mode centralisé ([**51**], [**52**], [**53**], [**54**]).

  Dans [**54**], deux algorithmes distribués pour la formation de la SVM dans le RCSF sont proposés. Pour les deux algorithmes le classificateur SVM est exécuté dans chaque nœud, et calcule un ensemble de données vectorielles. Pour le premier algorithme ce sont les vecteurs supports, pour le second ce sont des vecteurs situés dans l'enveloppe convexe de chacune des deux classes (normale, anormale) dont le nombre est supérieur aux nombre des vecteurs supports. Chaque nœud communique ses vecteurs avec son voisin d'un seul saut (*one-hop neighbor*), une fois ce processus terminé, l'hyperplan final est calculé, et tous les nœuds ont le même plan discriminant pour séparer les données en deux classes (normale, anomalie). Par la suite le nœud de capteur peut classifier les nouvelles mesures en utilisant cet hyperplan séparateur. Dans les simulations le deuxième algorithme présente une meilleure classification de données que le premier, mais avec une consommation énergétique supplémentaire.

## 2.3 Les exigences et les contraintes pour la mise en œuvre des IDSs dans le RCSF

De nombreuses recherches dans l'application de la solution des IDSs dans les réseaux adhoc ont été effectuées, en comparaison avec les RCSFs en raison des ressources limitées des capteurs en termes de capacités de calcul et de communication. Cependant, selon Roman et al. [**39**], les solutions d'IDS pour les réseaux adhoc ne peuvent pas être appliquées directement sur les réseaux de capteurs. Par conséquent, la conception des IDSs pour ce type de réseaux devraient tenir compte des restrictions suivantes [**32**] [**55**]:

- **Gaspillage d'énergie.** La plupart de l'énergie consommée dans les RCSFs est principalement due à l'interface de communication et non pas au processus de calcul [**55**]. Par conséquent, les

- **IDS distribués.** Dans les RCSFs, la station de base ne peut pas gérer un grand nombre de données d'audit (données de détection d'intrusion) à partir du réseau pour détecter toute intrusion. En outre, un grand nombre de paquets ne peut être transmis par les nœuds car les ressources énergétiques ne sont pas utilisées de façon optimale. Ceci est du à une transmission considérable de paquets vers une zone éloignée (station de base). Dans ce cas, une détection distribuée basée sur la coopération des agents IDS est une solution souhaitable.
- **Aucun nœud n'est digne de confiance.** Chaque agent IDS surveille ses voisins IDSs, en se basant sur le fait que même le nœud IDS peut être malicieux.
- **Le temps réel.** Afin de minimiser l'impact d'une possible attaque dans les applications critiques, il est important qu'un IDS fonctionne en temps réel.
- **Support l'ajout de nouveaux nœuds.** Dans la pratique, il est probable que de nouveaux nœuds peuvent rejoindre le réseau après le déploiement de celui-ci. L'IDS doit supporter cette opération et distinguer le nœud normal du nœud malicieux.
- **Précision.** La précision d'un IDS dans le RCSF est un autre problème majeur. La précision peut être définie comme étant l'exactitude d'un IDS à déterminer si le nœud en question est malicieux ou pas, en d'autres termes moins de faux positifs et faux négatifs (voir la section 2.4).
- **Disponibilité.** Un IDS doit fonctionner en permanence et rester transparent pour les utilisateurs.

## 2.4 Les métriques d'évaluation des IDSs dans le RCSF

Afin d'évaluer l'efficacité du modèle IDS proposé, un ensemble de métriques doit être adopté pour quantifier le niveau de sécurité et utiliser au mieux les ressources telles que la consommation d'énergie et l'espace de stockage. Ces indicateurs de performance permettront à un administrateur réseau de choisir le meilleur système de détection d'intrusion [**56**] et une optimisation de l'emplacement des agents IDS dans les nœuds de capteurs. En conséquence, les métriques suivantes sont considérées comme des caractéristiques importantes pour la conception efficaces des IDSs dans le RCSF:

- **Taux de détection.** Représente le pourcentage de détection d'attaques sur le nombre total d'attaques.
- **Taux de faux positifs (les fausses alarmes).** C'est le rapport entre le nombre des connexions normales classées comme étant une anomalie sur le nombre total des connexions normales.
- **Taux de faux négatifs**. Elle est l'inverse du taux de détection, cette métrique est définie par le rapport des fausses détections d'attaques sur le nombre total d'attaques.

Dans [**56**], les auteurs affirment que l'intrus peut lancer plusieurs types d'attaques. Par conséquent, les taux des faux positifs et négatifs doivent être calculés pour chaque type d'attaque. Ils proposent d'autres types de métriques tels que le nombre des paquets modifiés et perdus.

- **Consommation d'énergie.** Mesure de l'énergie consommée par chaque agent IDS. D'un autre coté, l'énergie totale du réseau est définie comme étant la somme de l'énergie consommée par chaque nœud.
- **L'efficacité.** Celle ci détermine le temps nécessaire pour un agent IDS de détecter l'apparition du premier nœud attaquant. Elle est calculée comme suit :

$$E = \frac{ED - ET}{fréquence\ de\ prélévement} \quad (2.1)$$

Où ET est le temps d'apparition du premier nœud malveillant et ED est le temps de détection du premier attaquant. Pour déterminer le temps requis pour les agents IDS de détecter toutes les attaques survenues dans le réseau, nous calculons l'efficacité moyenne, qui est définie comme suit:

$$EM = \frac{\sum_{i=1}^{n} E_n}{n} \quad (2.2)$$

Où n est le nombre d'attaquants.

## 3) La problématique de l'emplacement des agents IDS dans le réseau de capteurs sans fil à base de cluster (RCSFC)

Un critère important pour la réalisation des mécanismes d'IDS dans le RCSF est l'emplacement de ces agents dans ce type de réseau. De nombreux chercheurs ont travaillé sur cette problématique [**57**][**58**][**59**], d'où la stratégie de l'emplacement dépend de la topologie utilisée (plate ou hiérarchique). Dans une topologie plate, le nœud expéditeur s'appuie sur une communication multi-sauts pour atteindre la station de base. Ce processus conduit à une charge de communication élevée. Par conséquent, l'intégration des agents IDS dans ce type de topologie n'est pas une solution efficace. La topologie hiérarchique à base de cluster vise à améliorer la longévité du réseau en désignant un *cluster-head* par chaque cluster, qui a la fonction de gérer et agréger les données à partir des autres nœuds membres du cluster et de transmettre ces données agrégées à la station de base. Dans cette optique, notre étude va se focaliser sur les différentes stratégies d'emplacement des IDSs dans le réseau de capteur à base de cluster. La manière la plus simple de surveiller les comportements malicieux des nœuds dans le RCSFC est de déployer à chaque nœud un IDS ou placer un agent de contrôle (*monitoring agent*) à la station de base. Dans la première stratégie, le même paquet est

analysé à plusieurs reprises ce qui conduit à une grande quantité de données échangées entre les agents IDS. Ceci résulte à un nombre considérable de collisions et une charge élevée. Dans le second cas, l'IDS (installé à la station de base) ne peut pas gérer tous les paquets envoyés par les nœuds. En conséquence, ces deux stratégies ne sont pas adaptées pour les RCSFCs. Cela a conduit de nombreux chercheurs à fournir des nouvelles stratégies qui prennent en compte les contraintes des nœuds de capteurs dans ce type de réseau [**60**][**61**][**62**][**63**]. Dans nos investigations, nous avons classé l'emplacement des agents IDSs dans les RCSFCs en trois catégories:

### 3.1 L'emplacement des agents IDS dans les membres du cluster

Dans cette stratégie, chaque nœud est équipé d'un module de détection d'intrusion, sauf les *clusters-heads* qui sont responsables de la collecte et de l'agrégation des événements émis par les agents IDS (voir Figure 2.3). Khanum et al. [**63**] introduisent un modèle de détection d'intrusion, où les étapes de détection sont réparties en trois niveaux: membres du cluster *(cluster-members), clusters-head* et la station de base. Tous Les membres du cluster sont équipés d'un agent IDS (sauf le *clusters-head),* où chaque nœud surveille son nœud voisin. En d'autres termes, chaque IDS surveille son voisin IDS. Lorsqu'une intrusion est détectée, l'agent IDS vérifie si elle correspond à un ensemble de signatures d'attaques qui sont stockées dans chaque nœud. Si c'est le cas, une action prédéfinie est prise à l'encontre de cette intrusion. Autrement, l'agent envoie un message sous forme d'une alarme à son *cluster-head.* Ce nœud effectue un processus de vote au sein du cluster. Si la moitié des votes sont en faveur d'une attaque, le *cluster-head* envoie un message d'alerte (qui comprend le nœud suspect) à la station de base. Ce nœud prend une décision finale sur le nœud suspect et informe ce *cluster-head* à prendre des mesures supplémentaires. En particulier, ce nœud informe tous les agents au sein de son cluster et tous les *cluster-heads* à propos de la décision d'une nouvelle attaque. Lorsque le nœud IDS reçoit une confirmation que le nœud suspect est un intrus, il calcule de nouvelles règles contre cette nouvelle intrusion. En conséquence, l'avantage de ce modèle est que la mise à jour manuelle des règles est évitée. En outre, il est affirmé que le nombre de messages de contrôle est réduit, ce qui permet d'économiser les ressources de capteurs. Par contre, l'intégration d'un module de détection d'intrusion dans chaque membre du cluster, conduit à une réduction de la durée de vie du réseau. De plus aucune expérimentation n'est faite afin d'évaluer l'efficacité de ce modèle de sécurité en terme de détection d'attaque et de consommation d'énergie.

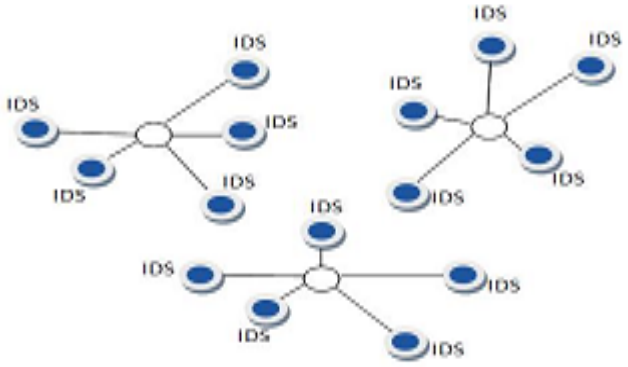



**Figure 2.3. Les agents IDS dans les membres du cluster**

### 3.2 L'emplacement de l'agent IDS dans le chef de groupe (*cluster-head*)

Dans la topologie hiérarchique, les *clusters-heads* couvrent toutes les communications dans le RCSF. Yan et al. [**61**] ont pris avantage de cette approche et installent à chaque *cluster-head* un agent IDS (*core defense*) comme illustré à la Figure 2.4. Cet agent est équipé de trois modules: un module d'apprentissage supervisé, un module de détection d'anomalie basée sur les règles et un module de prise de décision. L'agent IDS rassemble les paquets entrants et les analyse avec l'aide de la méthode fondée sur les règles (détection d'anomalie). Si les paquets analysés sont déterminés comme étant une anomalie, alors ils seront transmis au module d'apprentissage supervisé. Ce dernier utilise les réseaux à rétro-propagation (*back propagation network*) pour le processus d'apprentissage et de test. Les paquets anormaux détectés par le module de détection d'anomalie sont utilisés en tant que vecteur d'entrée au module d'apprentissage où l'algorithme apprend et classe les données en cinq classes (quatre types d'attaques et un comportement normal). Enfin, le module de prise de décision combine les sorties des deux autres modules (détection d'anomalie basée sur les règles et celle basée sur la technique d'apprentissage) afin de déterminer si l'information entrante est une intrusion ou non, et détermine la catégorie de l'attaque. Dans le cas où une intrusion se produit, ce module rapporte les résultats (concernant l'intrusion détectée) à la station de base. Les résultats des simulations montrent que ce modèle présente un taux élevé de détection et une baisse des taux de faux positifs. Mais les principaux inconvénients de ce schéma est: 1) le nœud IDS est statique (s'exécute seulement dans le *cluster-head*), dans ce cas l'intrus utilise toutes ses forces afin d'attaquer ce point chaud (*hot point*) et par la suite perturbe le réseau. 2) les auteurs ne prennent pas en considération les contraintes énergétiques des nœuds car ils implémentent un mécanisme de détection qui nécessite beaucoup de calculs dans les *cluster-heads* ce qui peut conduire à la diminution de la durée de vie du réseau.

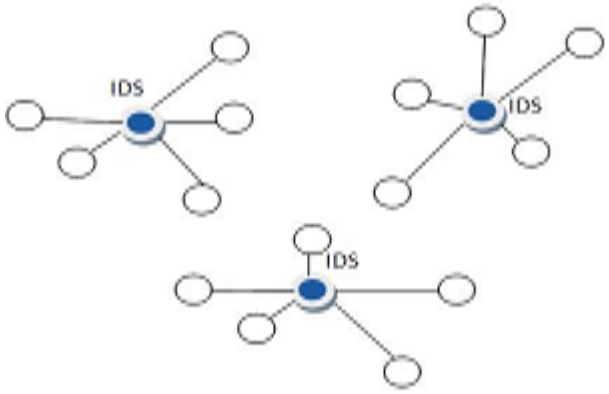



**Figure 2.4. L'agent IDSs dans le *cluster-head***

### 3.3 L'emplacement des agents IDS dans la frontière du cluster et dans le *cluster-head*

Une autre stratégie possible serait de mettre les agents IDS dans chaque *cluster-head* (*core defense*) et au niveau de la frontière limite de chaque cluster (*boundary defense*) comme illustrée dans la Figure 2.5. Huo et al. [**62**] ont adopté cette stratégie. Contrairement au schéma précédent, ils ont proposé un système dynamique de détection d'intrusion (DIDS) pour les RCSFs, où si la consommation d'énergie de l'un des nœuds IDS dans le cluster dépasse un certain seuil, le processus de reconfiguration de cluster est lancé (élection d'un nouveau *cluster-head* ). Dans ce cas, les IDSs vont être activés dans des nouveaux nœuds et dans des nouveaux clusters. Dans ce schéma, tous les nœuds ont un mécanisme d'IDS intégré, par contre l'activation de ces IDSs est lancée en cas de nécessité. Ce concept augmente la durée de vie du réseau et permet d'éviter le problème du point chaud. Par ailleurs, le principal inconvénient de ce schéma est qu'il a besoin de beaucoup de temps pour détecter toutes les intrusions, en particulier lorsque le nombre d'attaquants est très élevé.

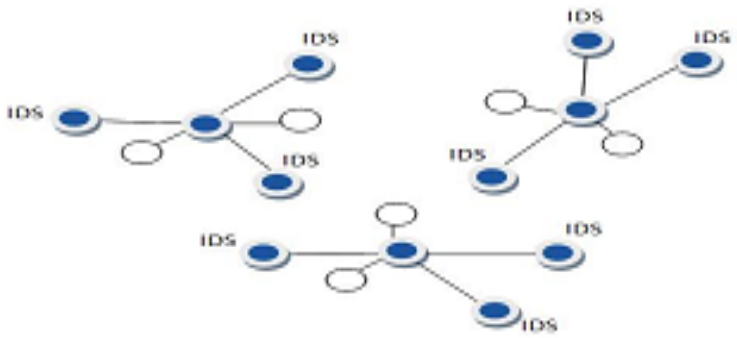



**Figure 2.5. Les agents IDS dans la frontière du cluster et dans le *cluster-head***

Toutes ces stratégies ont leurs avantages et leur inconvénients, par ailleurs l'emplacement optimal des IDSs dans le réseau doit satisfaire deux critères importants qui sont : (i) Le nombre d'IDS doit être suffisant pour couvrir tout le réseau et par conséquent analyser tous les paquets qui circulent dans le réseau. (ii) la politique de détection et le protocole de communication adoptés par les agents IDS doivent prendre en considération les contraintes énergétiques des capteurs. Cependant, l'idée que nous envisagions d'exploiter est que tous les nœuds ont la possibilité d'activer leurs agents IDS afin de surveiller les nœuds qui se situent dans le même voisinage en s'inspirant du principe du chien de garde (*watch- dog*). Par contre l'activation ne se fait pas d'une manière simultanée à cause des contraintes de ressources de ces capteurs. Dans chaque lien de communication il y'a au moins un agent qui surveille la communication entre deux nœuds. Lorsque la consommation d'un nœud IDS est supérieure à un certain seuil, un autre nœud prend la charge de surveillance et active son IDS. Cette solution d'élection d'IDS aide à éviter l'épuisement de l'énergie des nœuds et par conséquent prolonge la durée de vie du réseau.

## 4) Les différentes approches des IDSs dans le RCSFC

Les RCSFCs présentent une faible charge de communication (*communication overhead*) ce qui induit une prolongation de la durée de vie du réseau. Dans cette optique, plusieurs chercheurs ont travaillé sur l'implémentation et la mise en œuvre des mécanismes d'IDS dans ce type de réseau. Dans notre étude nous avons classé les approches des IDSs dans le RCSFC en quatre catégories. Une analyse des travaux de recherche de chaque approche va être détaillée. En particulier, nous mettrons l'accent sur les caractéristiques de chacun de ces travaux en indiquant leurs points forts et leurs faiblesses. Un résumé de ces travaux est illustré dans le Tableau 2.2.

### 4.1 Système hybride de détection d'intrusion

Il y'a quelques travaux qui visent à combiner entre la technique de détection d'anomalie et la technique de détection basée sur les signatures (modèle hybride) afin de tirer profit des avantages de ces deux politiques de détection et essayer de détecter un nombre significatif d'attaques. Nous trouvons dans la littérature quelques systèmes hybrides de détection d'intrusion dans le RCFSC telles que les références [**61**][**64**][**65**][**66**].

Hai et al. [**66**] proposent un modèle de détection d'intrusion léger (consomme moins d'énergie) pour le RCSF, basé sur le modèle d'IDS proposé par Roman et al. [**39**]. Dans leur schéma, l'agent IDS est composé de deux modules de détection, l'agent local et l'agent global (défini dans le chapitre 1, sous section 3.2.3). Les auteurs appliquent dans leur modèle un processus de coopération entre ces deux agents afin de détecter les attaques avec une meilleure précision (les deux agents se trouvent dans le même nœud). Cette coopération peut être expliquée comme suit: Lorsque l'agent local détecte une attaque, il conserve les données d'intrusion dans la mémoire du nœud afin que l'agent global puisse

les utiliser plus tard lors de son activation. L'agent global (*watch dog*) utilise la technique basée sur les règles et les données des voisins à deux sauts (*two-hop neighbors*) pour la détection d'anomalie. Lorsqu'une anomalie est détectée par cet agent; une alarme est envoyée au *cluster-head*. Les deux agents utilisent un ensemble de signatures d'attaquant, qui sont calculées et générées par le *cluster-head*. Pour une meilleure conservation d'énergie le nombre d'agent global est largement inférieur au nombre des agents locaux. Afin de réduire les collisions et éviter le gaspillage d'énergie, ils proposent un mécanisme d'écoute (*over-hearing*) qui vise à réduire la transmission des messages d'alerte. Lorsque le taux des collisions et le nombre des nœuds malicieux ne sont pas élevés, leur schéma peut détecter quelques attaques de routage tels que *Selective forwarding*, *Sinkhole*, *Hello flood* et *Wormhole*. D'après les résultats de leur simulation, leur modèle consomme moins d'énergie par rapport au modèle proposé par les auteurs dans [**39**]. Néanmoins, l'inconvénient de ce schéma est la forte augmentation des signatures qui à son tour conduisent à une surcharge de la mémoire du nœud.

### 4.2 Détection d'intrusion basée sur l'approche de la théorie de jeu

Récemment, la théorie des jeux a été largement utilisée pour la modélisation des problèmes de réseau [**67**]. Agah et al. [**67**], [**68**] et Mohi et al. [**69**] ont utilisé cette théorie dans les IDSs pour détecter et empêcher l'exécution des attaques du type Déni de service (DOS) dans le RCSF. Dans tous ces travaux, l'agent IDS et le nœud (attaquant ou nœud normal) sont formulés comme étant deux joueurs. Chaque joueur peut gagner ou perdre son gain en fonction de l'action qu'il effectue (par exemple, la réputation de l'IDS augmente lorsqu'il reconnaît correctement le nœud compromis, autrement elle diminue et la réputation du nœud attaquant va augmenter).

Plus précisément, Agah et al. [**67**] construisent un modèle de détection d'intrusion basée sur le jeu non-coopératif et à somme non-nulle entre deux joueurs (attaquant et IDS). L'idée principale de ce schéma est d'essayer de surveiller le nœud le plus attractif (*cluster head*) et de le protéger contre les attaques de DOS. Les auteurs supposent qu'à chaque intervalle de temps (*time slot*) une seule attaque peut se produire et l'IDS inspecte un seul nœud à la fois. En appliquant l'approche de la théorie des jeux, l'intrus attaque un nœud et l'IDS protège ce même nœud, conduisant ainsi à une meilleure stratégie de protection du réseau. Cette stratégie est définie comme étant l'équilibre de Nash [**70**]. Néanmoins, le modèle de sécurité proposé ne convient pas pour certaines applications car lorsque le nombre de nœuds malveillants augmente, le taux de détection de l'IDS diminue. Cela peut être expliqué si l'on considère le fait que, à chaque *time slot*, les nœuds qui lancent des attaques DOS sont importants, de ce faite l'IDS ne peut pas surveiller et gérer un grand nombre de nœuds malveillants. Par conséquent ceci peut conduire à un réseau non sécurisé.

Mohi et al. [**69**] proposent de sécuriser le protocole LEACH [**22**]. Le protocole de routage sécurisé (S-LEACH) utilise la formulation de jeu Bayesien. Ce dernier permet de représenter aisément un jeu à information incomplète [**71**], car chaque joueur dispose d'une incertitude sur l'état des autres joueurs. Cette théorie impose à chaque joueur de spécifier sa confiance concernant les autres joueurs (le nœud

est normal ou malicieux). Les auteurs affirment qu'au début, l'IDS n'a pas confiance en ces nœuds voisins (i.e. situé dans sa même couverture radio). L'affirmation du caractère malicieux ou normal par l'agent IDS à propos du nœud cible dépend de l'action effectuée par celui-ci. Dans leur approche, ils forcent les nœuds à coopérer, cette coopération est la transmission des paquets reçus. Autrement leur réputation diminue. ses adversaires sont définis comme étant des nœuds égoïstes (*selfish node*) car ils ne transmettent pas les paquets reçus dans le but d'économiser l'énergie [**42**][**72**]. Dans ce schéma l'agent IDS est divisé en deux modules: l'agent global qui réside dans la station de base et d'autres IDSs qui sont implémentés sur les *cluster-heads* nommés IDS locales. Ces derniers surveillent leurs nœuds membres et affectent une réputation négative pour ceux qui ne transmettent pas les paquets. Le *cluster-head* agrège les réputations de chaque nœud, qui seront par la suite envoyées à l'agent global (station de base). Cet IDS calcule la réputation de chaque nœud du réseau et informe les IDSs locaux (*cluster-head*) pour éjecter les nœuds qui ont une réputation négative dépassant un certain seuil. Les auteurs montrent par simulation que lorsque le nombre de nœuds n'est pas très élevé les *cluster-heads* peuvent reconnaître leurs nœuds membres avec une grande précision s'ils agissent malicieusement. Cependant, lorsque le nombre de nœuds est important leur modèle de sécurité génère un nombre considérable de faux positifs et de faux négatifs.

### 4.3 Système de détection d'intrusion basé sur les multi-agents

La détection d'intrusion basée sur les multi-agents est définie comme étant une entité logicielle (*Software*) implémentée dans certains nœuds (*cluster-head*, base station, etc.) pour effectuer des tâches spécifiques de détection d'intrusion. Bin et al. [**73**] et Ketel [**74**] proposent un IDS distribué basé sur les multi-agents dans le RCSFC. Le but de leurs travaux est de définir de multiples agents qui réalisent différentes tâches d'une unité de détection d'intrusion (collecte, analyse et réponse) et collaborent mutuellement entre eux pour la détection et l'isolement des nœuds malicieux dans le cluster. En conséquence, la séparation des tâches fonctionnelles permettra d'alléger la charge du réseau.

Bin et Al. [**73**] présentent un modèle de détection d'intrusion basé sur le concept d'agents distribués. Dans leur modèle, les auteurs tentent de mettre en place un ensemble d'agents qui permettent d'atteindre les différentes fonctionnalités d'un agent IDS. Les agents sont installés dans chaque nœud et ils sont divisés en quatre composantes: Agent sentinelle (*Sentry Agent*), Agent d'analyse (agent d'analyses), Agent de réponse (*Response Agent*) et Agent de gestion (*Management Agent*). L'agent sentinelle recueille les données pertinentes et les soumet à l'agent d'analyse. Lorsqu'une intrusion se produit, cet agent active l'agent de réponse, celui-ci génère les réponses correspondantes tel que l'envoi d'une alarme au *cluster-head* afin que celui-ci n'assigne aucun time slot au nœud malicieux, la mise à jour des clés, etc. L'agent de gestion est responsable d'informer les nœuds voisins à propos du nœud attaquant, de réélire un nouveau *cluster-head* lorsque celui-ci présente un comportement malveillant. Les auteurs proposent deux stratégies pour protéger le réseau contre les

intrus, soit le *cluster-head* défend ses nœuds membres ou les nœuds membres défendent leur *cluster-head*. Les résultats des simulations montrent que le modèle proposé est capable de détecter trois types d'attaques: *nmap, smurf* et *portsweep* (voir la référence [**75**] à-propos de ces attaques). L'agent d'analyse combine entre deux algorithmes d'apprentissage (*self-organizing map* et *K-means algorithm*). Ces deux algorithmes produisent un coût prohibitif en termes de calcul. Par conséquent l'implémentation de ce genre d'algorithmes dans les nœuds de capteurs peut causer la dégradation des performances du réseau.

### 4.4 Détection d'intrusion basée sur l'approche collaborative des agents IDS

Dans le mode distribué, les processus d'analyses sont effectués sur un certain nombre de nœuds IDS, où ces agents peuvent être soit autonomes ou collaboratifs (*stand-alone* or *collaborative*)[**76**]. Dans le premier cas, l'IDS ne partage aucune information avec d'autres systèmes ce qui implique que toutes les décisions sont basées sur l'information disponible sur le nœud individuel [**76**]. Dans ce cas, un seul nœud n'a pas une vue globale du réseau et donc il n'a pas suffisamment d'informations pour détecter un intrus. Ce genre de mécanisme ne convient pas à ce type de réseau. Dans le cas collaboratif, les nœuds IDS collaborent afin de détecter toute intrusion avec une preuve solide en utilisant une des deux approches ci-dessous (ou les deux):

- Les agents IDS collaborent entre eux pour prendre une décision finale si un nœud suspect est un intrus ou non. Cette prise de décision collaborative peut être basée sur le mécanisme de vote. Khanum et al. [**63**] utilisent ce mécanisme afin de déterminer avec une grande précision le comportement du nœud soupçonné (nœud normal ou intrus). Ce modèle de détection est détaillé dans la sous section 3.1.
- Chaque agent IDS partage ses données d'audit (données d'intrusion) avec d'autres agents afin d'obtenir une vue globale des activités d'intrusions et par conséquent avoir la capacité de détecter les intrusions les plus complexes dans chaque nœud IDS. Cette approche doit prendre en considération le compromis entre la consommation d'énergie et la quantité d'informations échangées entre les nœuds IDS.

  Besson et al. [**77**] appliquent ces deux approches de collaboration (le partage des données et la prise de décisions collaboratives). Dans chaque cluster les IDSs sont mis en place sur un sous-ensemble de nœuds, ces agents visent à propager les données d'intrusion entre eux. Lorsque l'agent IDS déclenche une alarme en ce qui concerne la présence d'une attaque dans le réseau, un mécanisme de vote est effectué entre les nœuds IDS appartenant au même cluster et le *cluster-head*. Ce dernier échange son vote avec d'autre *cluster-heads* dans le réseau. Dans ce schéma une communication sécurisée entre les IDSs coopérants est appliquée, celle-ci est basée sur le nonce (*timestamps*) pour assurer la fraicheur des données et la fonction de hachage pour assurer l'intégrité des messages partagés. Cette communication sécurisée est

inspirée du protocole OLSR sécurisé [**78**]. L'avantage de ce schéma est le niveau élevé de précision dans la détection d'un événement d'intrusion grâce à l’application de ces deux approches de collaboration. Toutefois, il en résulte une charge élevée de communication en raison du nombre important des paquets émis et reçus par chaque nœud coopérant.

| Schéma proposé | politiques de détection | Attaques détectées | l'emplacement des agents IDS | Durée de vie du réseau |
|---|---|---|---|---|
| Yan et al. [**61**] | Détection à base de signatures et détection d'anomalie | *Spoofed, Altered* et *Replayed routing information, Sinkhole, sybil, Wormholes, Acknowledgment spoofing, Selective forwarding et Hello flood* | Dans les *cluster-heads* (*core defense*) | Court |
| Huo et al. [**62**] | Détection à base de signatures et détection d'anomalie | Ne sont pas mentionnées | Dans les cluster-heads (*core defense*) et la frontière limite de chaque cluster (*boundary defense*) | Moyen |
| khanum et al. [**63**] | Détection à base de signatures | Ne sont pas mentionnées | dans les membres du cluster | Court |
| Hai et al. [**66**] | Détection à base de signatures et détection d'anomalie | *Selective forwarding, Sinkhole, Hello flood* et *Wormhole* | Sur un ensemble de nœuds dans chaque cluster | Moyen |
| Agah et al. [**67**] | Ne sont pas mentionnées | Déni de service (DOS) | N'est pas mentionné | Moyen |
| Mohi et al. [**69**] | Détection à base de signatures | Déni de service (DOS) | Dans la station de base et les cluster-heads | Moyen |
| Bin et al. [**73**] | détection d'anomalie | *Nmap, Smurf* et *Portsweep* | Dans tous les nœuds du réseau | Court |
| Ketel [**74**] | Détection à base de signatures | Ne sont pas mentionnées | Dans la frontière limite de chaque cluster, à chaque *cluster-head* et dans la station de base | Moyen |
| Besson et al. [**77**] | Détection à base de signatures et détection d'anomalie | Ne sont pas mentionnées | Sur un ensemble de nœuds dans chaque cluster | Court |

**Tableau 2.2 Résumé de quelques systèmes de détection d'intrusion dans le RCSFC**

## 5) Notre vision

Dans cette étude nous avons effectué une enquête approfondie et détaillée sur les IDSs dans le RCSF. Ces systèmes sont devenus un secteur très attractif de recherche pour la détection d'intrusion. Les systèmes de détection d'intrusion centralisée sont des systèmes à énergie efficace car ils sont implémentés dans un nœud puissant (station de base) [**58**]. Cependant, cette solution exige que tous les nœuds de capteurs doivent soumettre leurs données recueillies à la station de base, par conséquent elle introduit une charge élevée de communication. D'un autre coté, les systèmes de détection d'intrusion distribuée offrent des performances de détection légèrement inférieures à celles de l'approche précédente car ils utilisent des techniques de détection simple et légère en termes de calcul. En outre, la quantité d'informations échangée entre les nœuds n'est pas aussi importante contrairement au modèle centralisé où les nœuds envoient tous leurs paquets à un emplacement distant, par conséquent l'approche distribuée est mieux adaptée aux contraintes des ressources des capteurs.

L'architecture de clustering nécessite une faible consommation d'énergie. Appliquer une solution distribuée pour la détection d'intrusion dans une topologie basée sur les clusters entrainera un réseau sécurisé qui répond aux exigences des nœuds de capteurs.

Notre problématique de recherche des IDSs dans le réseau de capteurs à base de cluster réside sur l'utilisation des politiques de détection d'intrusion par l'agent IDS et l'emplacement de ces agents dans les nœuds de capteurs. Dans le premier point, deux grandes techniques de détection ont été proposées dans la littérature (*signature-based* et *anomaly-based detection*). Chaque technique présente des avantages et des inconvénients. L'idée c'est l'utilisation des avantages de ces techniques pour contrer un maximum d'attaques avec une limitation des charges de calcul et de communication générées par les agents IDS. Dans le second point, il est intéressant de placer les agents IDS de façon optimale dans le réseau afin de couvrir tout le réseau et avoir une vue globale sur les nœuds de capteurs. Ceci conduit à la détection de tous les paquets malicieux générés par les attaquants. Dans cette thèse nous avons proposé et conçu trois schémas de détection afin de contrer les attaques les plus menaçantes pour les RCSFCs.

## 6) Conclusion

Les RCSF sont souvent déployés dans des environnements hostiles et insécurisés. De tels capteurs sont vulnérables aux menaces internes car les attaquants connaissent les clés de chiffrement et peuvent les utiliser pour les opérations de cryptage et décryptage. Les systèmes de détection d'intrusion (IDS) étant très efficaces dans la protection du réseau contre les attaques internes. Toutefois ces systèmes peuvent produire une charge élevée de calcul et de communication, de ce fait les concepteurs de ce genre de système doivent toujours faire un compromis entre le taux de détection et la consommation d'énergie.

La consommation d'énergie est un problème majeur, plusieurs chercheurs ont proposé des solutions afin de la minimiser dans le nœud et par conséquent améliorer la durée de vie du réseau. Par conséquent, la solution la plus efficace est l'utilisation de protocoles de routage à base de cluster.

Les IDSs sont les mécanismes les plus fiables contre les attaques internes et la topologie à base de cluster est la mieux adaptée pour ce type de dispositifs car elle vise à maximiser la durée de vie du réseau. Pour cette raison dans notre première contribution nous avons développé et implémenté un système hybride de détection d'intrusion dans le RCSFC; ce système combine les avantages de deux techniques de détection (détection à base de signature et détection d'anomalie). Ce modèle de détection présente un taux de détection élevé et un nombre réduit de fausses alarmes.

# Chapitre 3

# Première Contribution :

# Modèle de Détection D’intrusion Hybride dans le Réseau de Capteurs à Base de Cluster

## Résumé

Dans ce travail, nous proposons un nouveau système de détection d'intrusion pour les réseaux de capteurs à base de cluster. Notre modèle de détection combine entre la détection d'anomalie basée sur la machine à vecteur de support (SVM) et la détection à base de signatures d'attaques. Les résultats de simulation montrent que la plupart des attaques de routage sont détectées avec un faible taux de faux positifs.

## 1) Introduction

La sécurité est l'un des problèmes les plus ardus dans les réseaux de capteurs sans fils (RCSFs) en raison de leur déploiement dans des environnements hostiles et insécurisés tels que les champs de bataille. La technique de cryptographie permet de protéger le réseau contre toutes menaces externes en appliquant l'authentification des paquets à partir de la source et d'assurer l'intégrité des données de la communication en cours. Cependant, l'inconvénient majeur de cette technique est qu'elle ne peut pas détecter les attaques internes, lorsque celles-ci connaissent les clés de chiffrement.
Dans ce contexte, le système de détection d'intrusion (IDS) permet la détection d'une activité suspecte au sein du réseau en analysant les nœuds cibles, par la suite une alarme sera déclenchée par l'agent IDS lorsque ces nœuds présentent un comportement malveillant.

Pour l'analyse des comportements des nœuds, les systèmes de détection d'intrusion (IDSs) utilisent la technique à base de signatures d'attaques ou la détection d'anomalie. Chacune de ces techniques présente des avantages et des inconvénients. Dans ce cadre, plusieurs travaux [**60**][**61**][**66**][**79**] ont combiné entre la détection d'anomalie et la détection à base de signature afin de tirer partie des avantages de ces deux techniques et d'essayer de détecter un nombre importants d'attaques. Basé sur ces modèles hybrides, notre objectif dans cette recherche est d'étudier et d'implémenter un nouveau modèle de détection d'intrusion qui combine les avantages de ces deux techniques (un taux de détection élevé avec un faible taux de faux positifs) et qui surpasse d'autres modèles hybrides proposés dans la littérature. Le modèle de détection hybride proposé utilise la machine à vecteur de support (SVM) pour la détection d'anomalies et un ensemble de signatures d'attaques représentées par des règles fixes, celles-ci visent à valider le comportement malveillant d'une cible identifiée par la technique de détection d'anomalie. L'approche de détection est intégrée dans un réseau à base de cluster, d'où l'avantage de ce type de réseau est d'augmenter sa durée de vie. Ce résultat est obtenu en désignant un seul nœud connu comme le chef de groupe (*cluster-head*) qui transmet les paquets (données agrégées) à la station de base au lieu que tous les nœuds envoient leur données collectées à un emplacement distant (station de base).

Dans cette étude, nous mettons en évidence quelques informations de base sur la machine à base de vecteurs de support (SVM), en décrivant l'utilité du choix de ce type d'algorithme d'apprentissage dans les RCSFs. Par la suite, nous expliquons le principe de fonctionnement de notre modèle hybride de détection en décrivant les différents composants qui le constituent. Finalement, les performances du modèle hybride sont évaluées.

## 2) Politique de détection basée sur la machine à vecteurs de support (SVM)

Comme il a été abordé dans le deuxième chapitre, les techniques de détection d'intrusion peuvent être classées en deux catégories: détection basée sur les signatures d'attaques et détection d'anomalie [**45**]. Plusieurs chercheurs ont travaillé sur l'hybridation de ces deux techniques afin de contrer les inconvénients que présente chacune d'elle [**60**][**61**][**66**][**79**].

La détection d'anomalie est une technique un peu coûteuse car elle nécessite un calcul considérable afin de modéliser le comportement normal d'un nœud avec un taux de faux positifs faible. Afin de concrétiser cet objectif, un algorithme d'apprentissage doit être adopté, celui-ci permet une meilleure modélisation avec un taux de faux positifs presque égal à 0%. par ailleurs la plupart de ces algorithmes ne sont pas adaptés aux contraintes énergétiques des nœuds de capteurs en raison de la charge de calcul et de communication (*Computation and communication overhead* ) qu' ils génèrent. Dans les RCSFs la communication consomme beaucoup d'énergie par rapport au processus de calcul car 1 bit transmis équivaut à 800-1000 instructions [**80**][**81**]. Par conséquent l'algorithme d'apprentissage adopté pour le RCSF doit minimiser la quantité d'informations échangées dans le réseau. De ce fait, Les SVMs sont les mieux adaptés car ils présentent une faible charge de communication [**53**][**82**]. Cette réduction de consommation est due au fait que chaque nœud échange avec son nœud voisin un ensemble de vecteurs clés appelés **vecteur de support**, contrairement au réseau de neurones où toutes les données d'entrées (utilisées lors du processus d'apprentissage) sont échangées entre les capteurs. Dans cette optique, un système d'apprentissage distribué binaire basé sur la SVM est adopté pour modéliser le comportement normal et anormal d'un nœud.

Compte tenu des ensembles de données d'apprentissage, $(x_i, y_i)\ i = 1,..n, y_i \epsilon \{-1, +1\}, x_i \epsilon R^d$, dans notre cas {1} c'est normal et {-1} c'est une anomalie. Nous voulons maximiser la marge entre l'hyperplan et les données d'apprentissage, en d'autre terme on doit déterminer l'hyperplan optimal.
L'équation de l'hyperplan séparateur est définie comme suit:

$$w.x = b \qquad (3.1)$$

Afin de trouver l'hyperplan optimal, nous devons résoudre le problème de minimisation sous contraintes suivantes:

$$\begin{cases} \min\left\{\frac{\|w\|^2}{2}\right\} + C\sum_{i=1}^{n}\varepsilon_i \\ y_i(w.x_i + b) \geq 1 - \varepsilon_i, \varepsilon_i \geq 0 \end{cases} \qquad (3.2)$$

$\varepsilon_i$ : Variables ressort (*slack variables*) permettant l'autorisation de quelques erreurs de classification lors du processus d'apprentissage. La constante de régularisation C > 0 quantifie le compromis entre le nombre d'erreurs de classement et la largeur de marge de l'hyperplan [**83**].

L'équation (3.2) peut être traitée en passant au problème dual avec l'introduction des multiplicateurs de Lagrange [**84**] :

$$\begin{cases} \max \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j k(x_j, x_i) \\ sous\ la\ contrainte \sum_{i=1}^{n} y_i \alpha_i = 0, and\ 0 \leq \alpha_i \geq C \end{cases} \quad (3.3)$$

$k(x_j, x_i)$ est la fonction noyau et $\alpha_i$ sont les multiplicateurs de Lagrange. Selon la condition de KKT (*Karush-Kuhn-Tucker*), les point $x_i$ correspondant à $\alpha_i > 0$ sont appelés les vecteurs de support.

La solution de l'équation 3.3 s'écrit sous la forme suivante [**84**]:

$$w = \sum_{i=1}^{n} \alpha_i y_i \, x_i \quad (3.4)$$

La fonction de décision associée est donc :

$$f(x, \alpha, b) = \{\pm 1\} = sgn\left( \sum_{i=1}^{n} y_i \alpha_i k(x, x_i) + b \right) \quad (3.5)$$

## 3) Le modèle hybride proposé et son fonctionnement

L'approche proposée utilise une combinaison entre la détection d'anomalie sur la base de la SVM et la technique fondée sur les signatures d'attaques. La détection d'anomalies utilise un algorithme d'apprentissage distribué pour la formation de la SVM afin de distinguer entre les activités normales et anormales. En outre, un ensemble de règles fixes associées à chaque signature d'attaque est stocké au niveau de chaque nœud. Nous utilisons une topologie à base de cluster qui divise le réseau de capteurs en un ensemble de groupes, chacun d'eux est constitué d'un *cluster-head* (CH). Comme il est mentionné dans le chapitre 1, l'objectif de cette architecture est d'économiser l'énergie et par conséquent une prolongation de la durée de vie du réseau. Enfin, chaque nœud a la possibilité d'activer

son agent IDS. Cependant, l'activation simultanée n'est pas effectuée car elle conduit à gaspiller des ressources du réseau. Dans chaque lien il doit y avoir au moins un nœud IDS, qui a la responsabilité de collecter et analyser les paquets qui circulent dans sa portée radio (*radio range*) comme il est illustré dans la Figure 3.1. On note que, les IDSs reçoivent les données à travers l'écoute de leurs entourages (*promiscuous*), en captant les paquets qui ne leurs sont pas adressés [**36**], ou en utilisant la communication multi-saut (le *cluster-head* peut agir comme étant un relai). Ce processus est illustré dans la Figure 3.1.

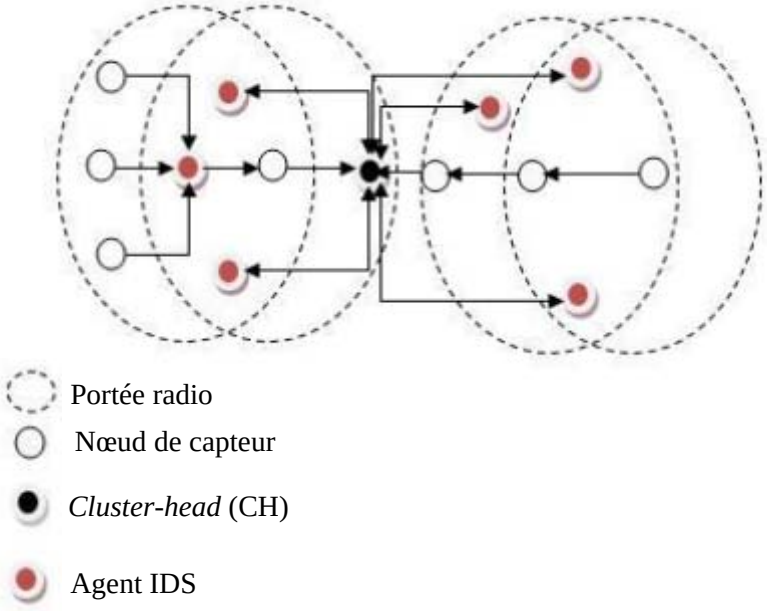



**Figure 3.1. Stratégie de l'emplacement des IDSs dans le RCSFC**

### 3.1 L'architecture des agents IDS

Dans notre modèle, nous avons divisé les agents de détection en deux catégories qui sont: agent IDS et agent CH, comme il est illustré dans la Figure 3.2. L'agent IDS est équipé de deux composants qui sont *Data Collection Framework (DCF)* et *Intrusion Detection Framework (ADF)*. L'agent CH est équipé d'un composant de coopération, *Collaborative Detection Framwork (CDF)*. L'organigramme du processus de détection est illustré dans la Figure 3.3.

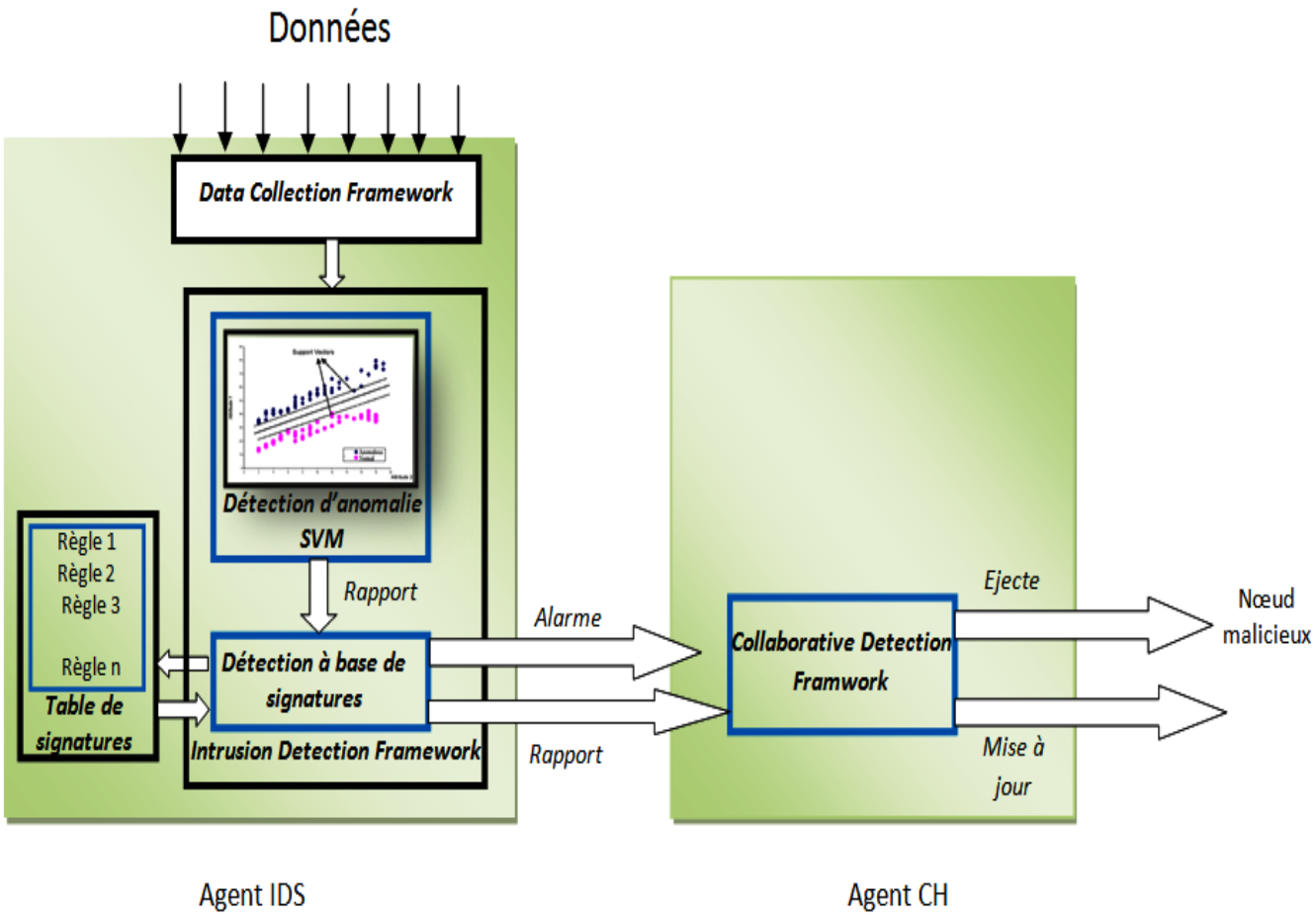



**Figure 3.2. L'architecture du modèle de détection d'intrusion**

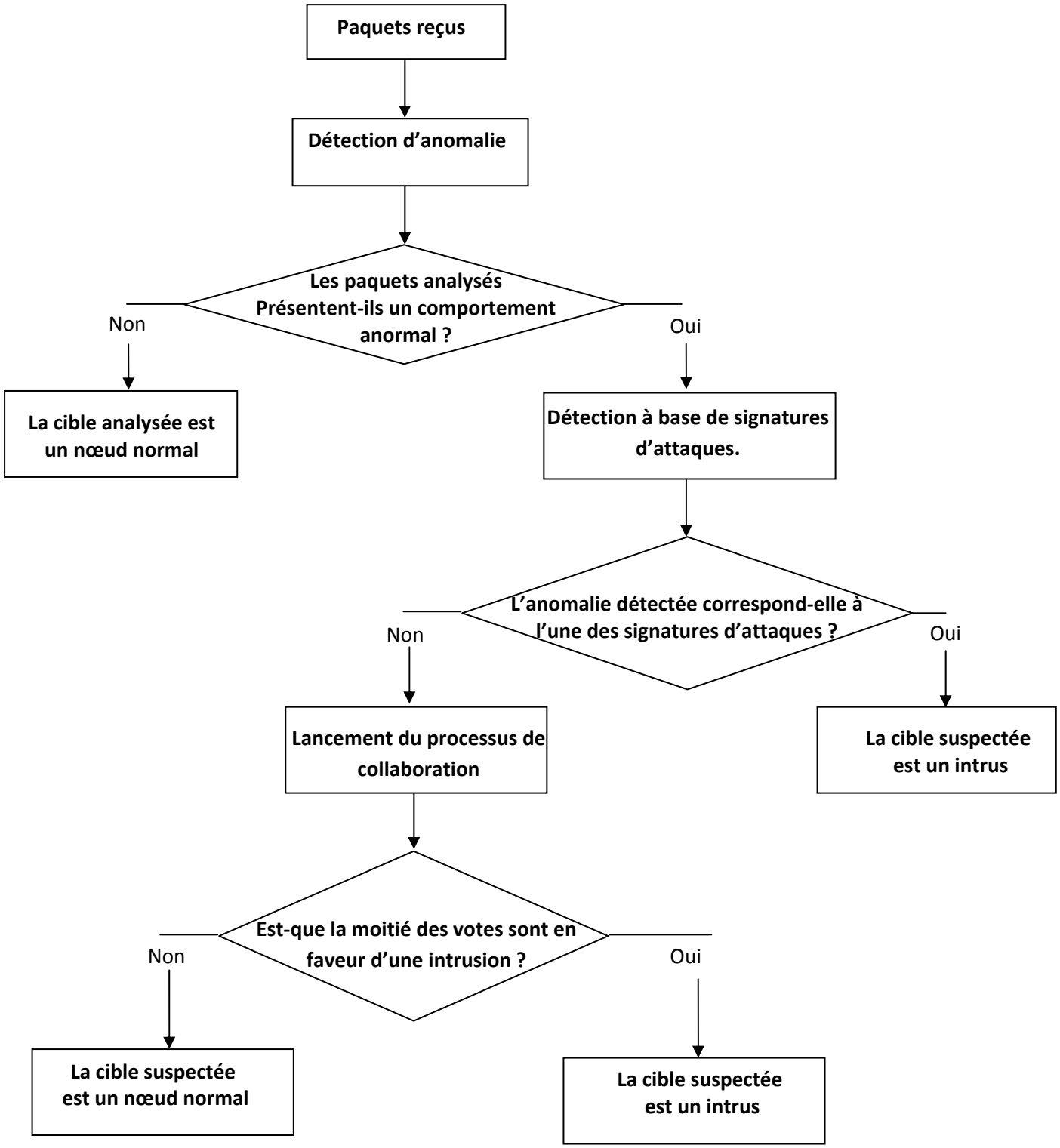



**Figure 3.3. Organigramme du modèle hybride de détection d'intrusion**

***1) DCF*** : Grâce à la nature de diffusions des réseaux sans fil, le nœud IDS collecte les paquets dans sa zone de couverture radio [**66**] et les transmet ensuite à l' *Intrusion Detection Framework (ADF)* pour une premier phase de détection.

***2) ADF*** : Comme premier phase de détection, cet organe applique la détection d'anomalie à base de la SVM. Lorsqu'une anomalie est détectée, la technique de détection à base de signatures compare les signatures de chaque attaque avec l'anomalie détectée.

**a) Détection d'anomalie :** La procédure de détection des anomalies est divisée en deux étapes:

**Etape1 : Le procédé d'apprentissage**. Chaque agent IDS applique localement le processus d'apprentissage et par la suite calcule les vecteurs de support (SVs). On note que, Ces vecteurs sont moins nombreux que les données d'entrée utilisées lors du processus d'apprentissage. Chaque IDS envoie ces SVs à son IDS voisin ou le *cluster-head* comme le montre la Figure 3.4. Lors de la réception de ces vecteurs, l'IDS combine ces derniers avec les SVs calculés, puis transmet le résultat aux nœuds voisins (IDS ou CH). Ce processus se poursuit jusqu'à ce que tous les agents IDS dans le même cluster aient les mêmes vecteurs de support, autrement dit, un passage complet à travers tous les IDSs dans le même cluster. Pour chaque cluster, l'IDS sélectionné en fonction de son énergie résiduelle, envoie ses SVs à son CH; par la suite chaque CH échange les données avec ses CHs voisins. Lorsque ce processus est achevé, chaque CH envoie les SVs à ses IDSs. Finalement, Ces derniers calculent les vecteurs de support globaux et calculent l'hyperplan séparateur, ce qui permet de classifier les nouveaux paquets comme étant des données normales ou en anomalie.

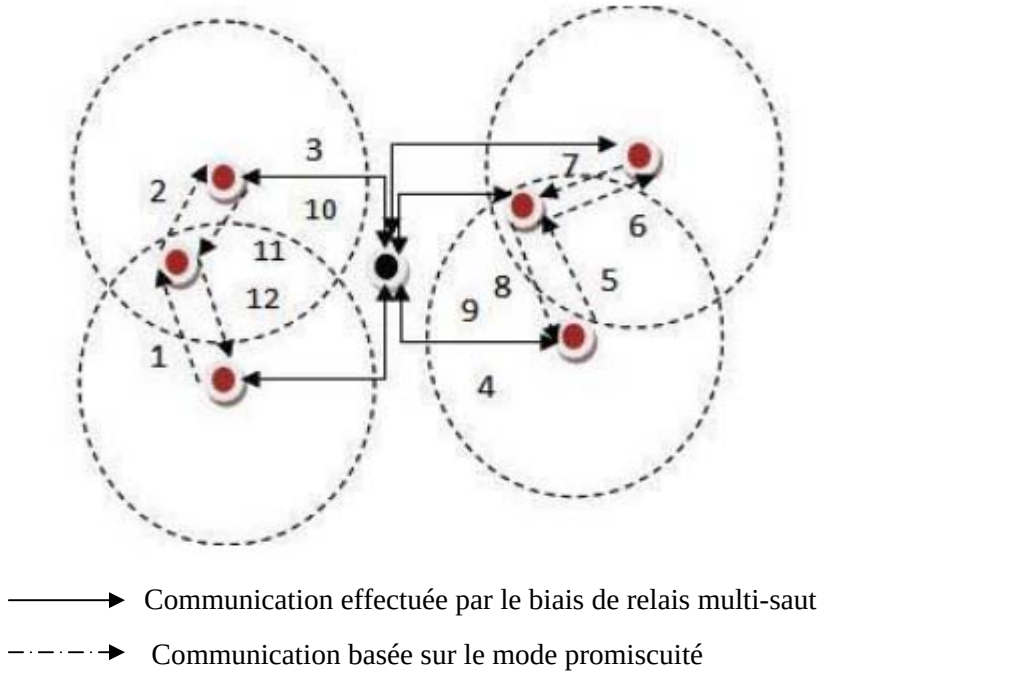



**Figure 3.4. Communication des vecteurs de support entre les nœuds IDS**

**Etape2 : Le procédé de test.** Lorsque le processus d'apprentissage est achevé, chaque IDS classifie les données conformément à la structure normale et en anomalie. Tout écart par rapport au profil normal (intrusion) est livré à la technique de détection à base de signatures d'attaques pour une détection antérieure.

**b) Détection à base de signatures d'attaques**. Avant le déploiement des nœuds, un ensemble de règles fixes associées à chaque signature d'attaque sont stockées au niveau de chaque nœud (Table de signatures), par la suite d'autre règles vont être ajoutées dans cette table, et ceci lorsqu'une nouvelle signature est détectée. Un exemple de règle associée à chaque signature est illustré dans le Tableau 3.1. Lorsqu'une anomalie se produit, la technique de détection à base de signature reçoit le *Rapport* d'intrusion à partir de la technique de détection d'anomalie comme la montre la Figure 3.2, ce rapport contient l'identifiant du nœud suspect (*id)* et un ensemble d'attribut. Ce dernier peut être défini comme étant le nombre d'octets émis et reçus, etc. Le choix et la sélection des attributs les plus pertinents vont être expliqués en détail dans la section expérimentation (voir la sous section 3.2.1). Cette technique à base de signatures consiste à comparer l'anomalie détectée avec un ensemble de signatures d'attaques, si une correspondance se produit l'IDS envoie une *Alarme* sous forme de message à son CH, disant que le nœud soupçonné est un intrus comme il est illustré dans la Figure 3.2. Celui-ci éjecte ce nœud du cluster et informe tous les CHs dans le réseau à propos du caractère malicieux du nœud. Par ailleurs, si aucune correspondance ne se produit, le processus de collaboration est lancé.

| **Signatures d'attaques** | **Règle** |
|---|---|
| Le nombre de paquets envoyés | Si NPS > Seuil 1 |
| Le nombre de paquets reçus | Si NPR > Seuil 2 |
| La force du signal reçu | Si RSSI > Seuil 3 |
| Le nombre de retransmissions du même message | Si NRM > Seuil 4 |
| Le nombre de collisions | Si NC > Seuil 5 |

**Tableau 3.1. Règle associée à chaque signature d'attaque**

***3) CDF*** : Dans le processus de collaboration, le *cluster-head* applique le mécanisme de vote. Dans le cas où il n'y a aucune correspondance entre l'intrusion détectée par la technique de détection d'anomalies et les signatures prédéfinies des attaquants, l'agent IDS envoie un message sous forme de *Rapport* (id, les attributs) au CH, comme le montre la Figure 3.2. Par la suite, le CH effectue un mécanisme de vote afin de prendre une décision finale sur le nœud suspect. Si plus de la moitié des

nœuds IDS situés dans le même cluster affirme que la cible soupçonnée est malicieuse, le CH éjecte ce nœud de son cluster et calcule la règle appropriée de cette nouvelle intrusion détectée. Le CH envoie par la suite un message de *Mise à jour* à tous les IDSs qui se situent dans le même cluster et les CHs voisins. Ce message contient l'identifiant du nœud malicieux et cette nouvelle règle (et signatures). Lorsque l'agent IDS reçoit ce message il fait une mise à jour de sa table de signatures.

## 4) Expérimentation

Dans cette section, nous évaluons les performances du modèle IDS hybride proposé. Notre approche est implémentée sous notre propre simulateur programmé en JAVA. Les paquets échangés entre les nœuds de capteurs dans ce simulateur sont des paquets de la base de données KDDcup'99 [**75**]. Le choix de la conception de notre propre simulateur réside dans le fait qu'il utilise des donnés réelles, contrairement aux autres types de simulateurs proposés dans la littérature, comme NS2 [**85**], OPNET [**86**], TOSSIM [**87**]. La Figure 3.5 illustre les différents composants de notre simulateur qui sont : nœud de capteur, nœud malicieux (intrus), paquet et agent IDS. Les paramètres clés de notre simulation sont définis dans le Tableau 3.2.

Afin d'évaluer l'efficacité de notre modèle hybride de détection, deux métriques sont utilisées à savoir le taux de détection et le nombre de faux positifs (voir la définition des ces métriques dans le second chapitre, sous section 2.4). Par la suite, notre modèle est comparé avec celui de Yan et al. [**61**] et Hai et al. [**66**] en termes de taux de faux positifs.

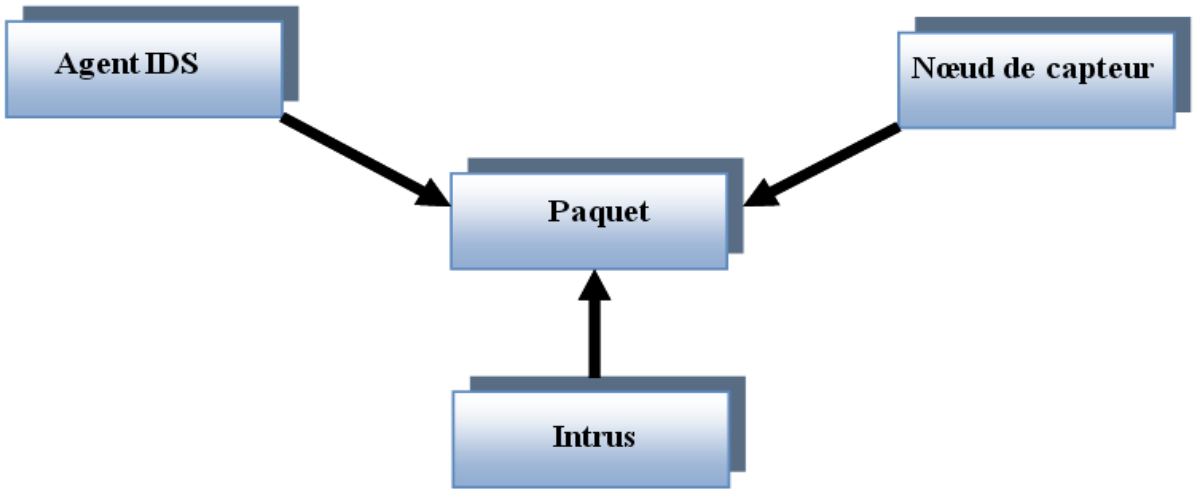



**Figure 3.5 Les principaux composants du simulateur**

| | |
|---|---|
| Temps de simulation | 320 Secondes |
| Domaine de la simulation | 60x50m$^2$ |
| Nombre de nœuds | 100 |
| Nombre de cluster | 10 |
| Nombre de nœud IDS | 3-24 |

**Tableau 3.2. Paramètres de simulation**

## 4.1 KDD Cup 1999

La base de données KDDcup'99 a été développée par MIT Lincoln Lab en 1998, cette base contient un nombre d'enregistrements de communication égal à 494021. Chaque communication dispose de 41 attributs (voir Annexe A) et elle est classifiée en cinq classes: normale et quatre comportements d'attaques (*Dos, Probe, U2r, R2l*). Ces données d'intrusion sont simulées dans un environnement militaire. *Selective forwarding* et *Black holes* utilisent les données illégitimes de retransmission pour effectuer une attaque, ces menaces sont classées comme étant des attaques de type *DoS* [**61**]. *Spoofed, altered,* et *Replayed routing information*, *Wormholes* et *Acknowledgment spoofing* ont besoin de faire une étape de test avant qu'ils ne commencent à attaquer. Ces menaces sont classées comme étant des attaques de type *Prob*[**61**]. *Sinkhole, Wormholes, et Hello floods* sont causées par des attaques internes, ces menaces sont classées comme étant des attaques de type U2R [**61**]. Spoofed, *altered et Replayed routing information, Sinkhole, Sybil, Wormholes, Hello floods et Acknowledgment Spoofing* profitent des faiblesses du système pour déclencher une attaque, ces menaces sont classées comme étant des attaques de type *R2L* [**61**]. Dans notre étude nous allons nous focaliser sur les attaques de types *Dos* et *Prob,* qui sont définies comme étant une anomalie et sont classées en tant que {-1} par rapport à l'hyperplan optimal; le comportement normal est classé comme {+1}.

## 4.2 Résultats expérimentaux et discussion

Dans cette section, nous allons étudier l'impact du choix des attributs sur les performances de notre système de détection, en déterminant les attributs les plus pertinents qui permettent au système de générer un taux de détection élevé avec une faible occurrence de faux positifs. Par la suite, nous évaluons les performances de notre modèle hybride de détection en termes de taux de détection et de nombre de faux positifs générés.

### 1. Sélection des attributs

Le choix et la sélection des attributs les plus pertinents sont un facteur important pour augmenter la précision de la classification (classifier les données comme étant normales ou anormales), réduire les faux positifs, obtenir un temps d'apprentissage rapide et une réduction de la consommation d'énergie. Dans cette recherche, nous nous sommes inspirés de la méthode de sélection des attributs proposés par Sung et al. [**47**]. Notre méthode de sélection consiste à supprimer un attribut à la fois, appliquer le processus d'apprentissage, tester ce modèle d'apprentissage et calculer le taux de détection et le nombre de faux positifs, pour finalement déterminer les attributs les plus pertinents qui correspondent à un meilleur taux de détection et un nombre de faux positifs faibles. nous intègrons par la suite le modèle d'apprentissage associé à ces attributs dans le module de détection d'intrusion afin d'obtenir un système de détection léger et efficace en détection (une faible consommation d'énergie, un taux de détection élevé et un nombre réduit de faux positifs). Ce processus de sélections des attributs est illustré dans la Figure 3.6.

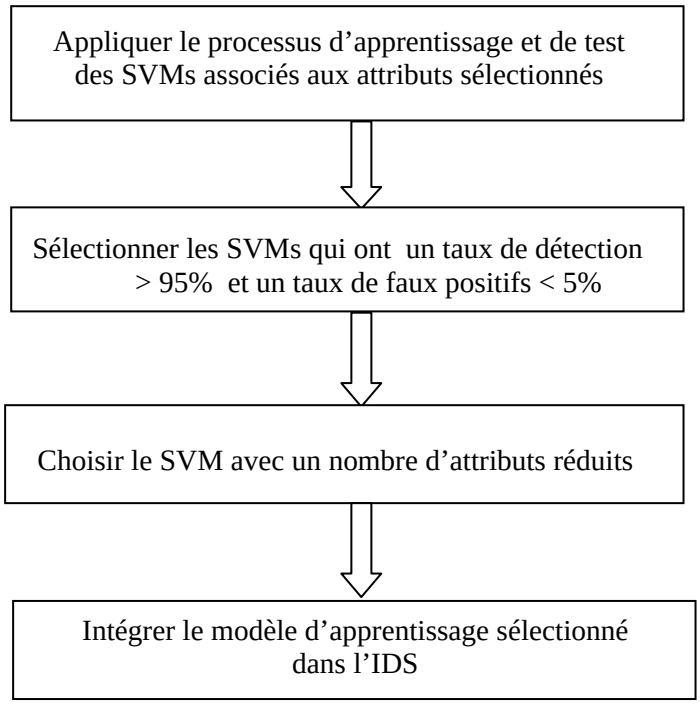



**Figure 3.6. Le processus optimal de sélection d'une SVM**

L'augmentation du nombre d'attributs conduit à un coût élevé de calcul et un débordement de la mémoire (*memory overflow*) du nœud, pour cette raison nous visons à obtenir un classificateur SVM qui s'appuie sur un nombre réduit d'attributs pour son processus d'apprentissage et qui présente un

taux de détection élevé (et un nombre réduit de faux positifs). Le résultat de notre classificateur distribué binaire lié aux attributs les plus pertinents est illustré dans le Tableau 3.3.

| **Nombre d'attributs** | **Taux de détection (%)** | **Taux de faux positifs (%)** |
|---|---|---|
| 9 | 93.66 | 4.3 |
| 7 | 95.61 | 3.7 |
| 5 | 91.21 | 4 |
| 4 | 95.37 | 3.85 |

**Tableau 3.3. Evaluation des performances des IDSs distribués à base des SVMs**

Nous constatons des résultats obtenus (Tableau 3.3) que le classificateur SVM binaire avec 7 attributs surpasse les SVMs qui utilisent les attributs (9, 5, 4), conformément aux deux métriques considérées. Par conséquent, ces 7 attributs sont les plus significatifs. Cependant, la différence du taux de détection et le nombre de faux positifs entre un SVM avec 7 attributs et un SVM avec 4 attributs est minime, et en raison des contraintes des ressources des nœuds de capteurs, il est préférable d'utiliser un SVM avec 4 attributs pour la détection d'anomalie. Les 4 attributs sélectionnés sont:

**Src_bytes:** Nombre d'octets envoyés de la source vers la destination
**Dst_bytes:** Nombre d'octets envoyés de la destination vers la source
**Count:** Nombre de connexions au même hôte de destination
**Srv_diff_host_rate:** Le pourcentage de connexions d'un nœud à différents hôtes

Dans ce qui suit nous allons étudier les performances de notre modèle de détection avec l'utilisation de ces 4 principaux attributs par le classificateur binaire SVM pour la détection d'anomalie.

### 2. Performance du modèle hybride de détection

Dans cette section, nous évaluons les performances de notre modèle de détection d'intrusion en utilisant les échantillons de la base de données KDDcup'99 [**75**]. Tout d'abord, nous évaluons notre modèle IDS en utilisant uniquement la politique de détection d'anomalie à base de la SVM, ensuite nous combinons les deux techniques (SVM et la détection basée sur les signatures). Dans les deux cas, nous étudions les variations des taux de détection et de faux positifs, lorsque le nombre d'IDS augmente dans le réseau. Finalement, nous comparons les performances de notre modèle hybride à celles des modèles cités dans les références [**61**] et [**66**].

Comme il est illustré dans la Figure 3.7.a, le taux de détection atteint presque 100% lorsque le nombre de nœuds IDS est élevé (plus de 12 agents). Cependant, nous avons remarqué une augmentation dans le nombre de faux positifs lorsque le nombre de nœuds IDS dépasse 12 agents. Par conséquent, un compromis entre le nombre de nœuds IDS et le taux de fausses alarmes doit être effectué.

La combinaison entre la détection d'anomalie basée sur la SVM et la détection à base de signatures d'attaques permet au modèle de détection d'intrusion d'atteindre un taux élevé de détection d'intrusion (presque 100%) avec un nombre très réduit de fausses alarmes (proche de 0%), lorsque le nombre d'IDS est important (i.e. dépasse 12 nœuds), comme il est illustré sur la Figure 3.7.b.

Ainsi, l'utilisation de notre approche hybride de détection d'intrusion permet de répondre à l'exigence de cette application en termes de taux de détection d'attaques et de nombre de fausses alarmes générées par les IDSs. Il est à noter que dans cette étude nous avons supposé que les agents IDS ne sont pas des nœuds malicieux.

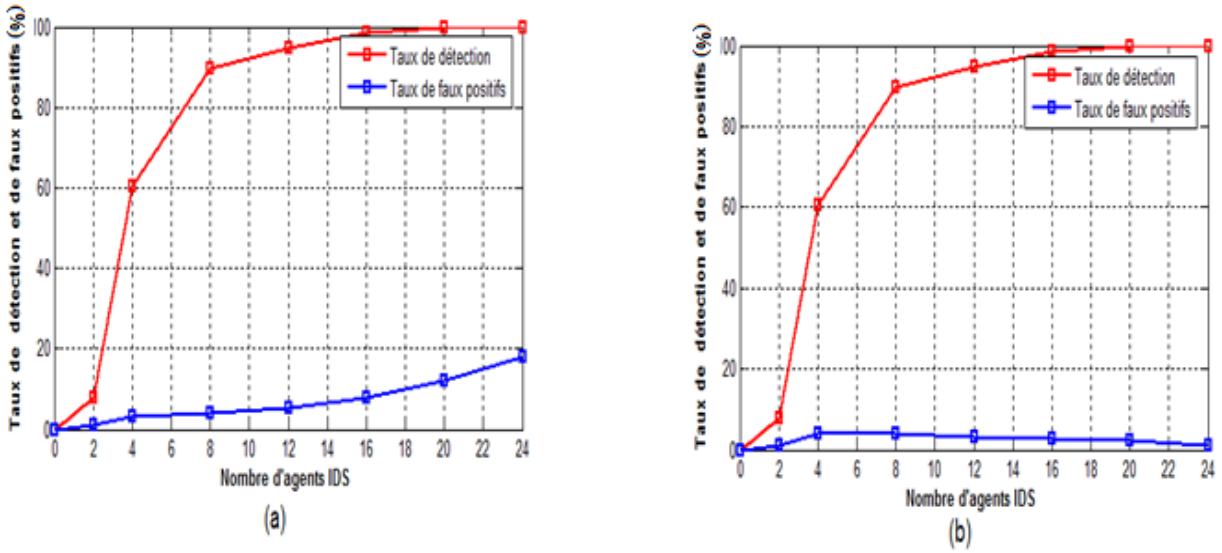



**Figure 3.7. Performance du modèle.**

**(a) Taux de détection et de faux positifs avec une détection basée sur la SVM. (b) Taux de détection et de faux positifs avec une détection basée sur la SVM & des signatures d'attaques**

D'après les résultats de simulations que nous avons menées, nous avons vérifié que notre modèle hybride a la possibilité de détecter avec une grande précision toute attaque malicieuse. Afin de déterminer l'efficacité de notre approche, nous avons comparé notre modèle avec deux modèles hybrides de détection proposés par les auteurs K.Q. Yan et al. [**61**] et T. H. Hai et al.[**66**], en analysant plus particulièrement le taux de fausses alarmes générées par les agents IDS.

Les résultats de simulation (voire Figure 3.8) montrent que lorsque le nombre d'IDS augmente, le nombre de faux positifs dans notre modèle et dans le modèle de K.Q. Yan diminuent, par contre dans le modèle hybride proposé par T. H. Hai, le taux de faux positifs augmente car le schéma de détection proposé génère un nombre important de collisions. D'après la Figure 3.8, notre approche génère un nombre réduit de fausses alarmes par rapport aux autres modèles, en particulier lorsque le nombre de nœuds IDS est supérieur à 12.

Par conséquent, le modèle hybride distribué de détection d'intrusion que nous avons proposé dispose d'une meilleure efficacité en termes de détection d'attaques et du nombre de faux positifs.

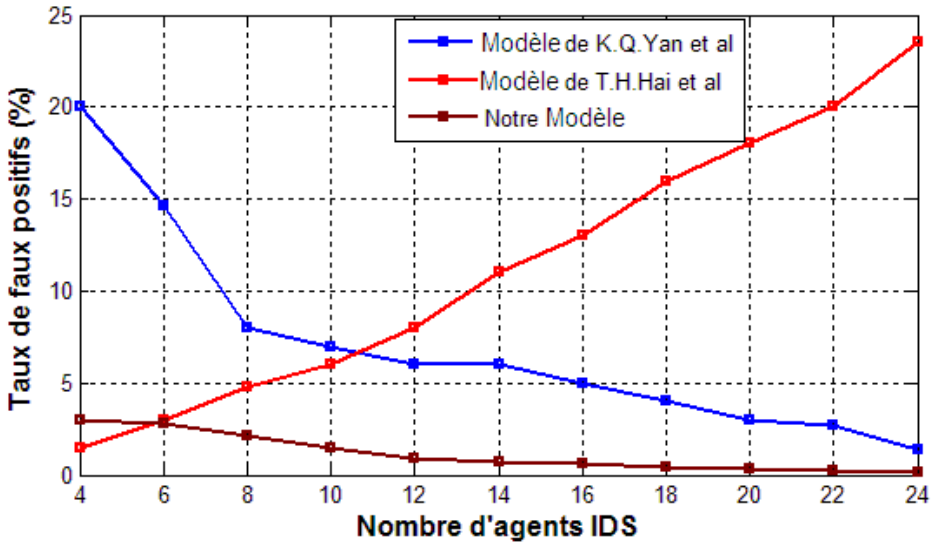



**Figure 3.8. Comparaison des taux de faux positifs dans les différents modèles**

## 5) Conclusion

Dans ce travail, nous avons proposé un modèle hybride distribué de détection d'intrusion pour les RCSFs. Notre modèle de détection utilise un algorithme d'apprentissage basé sur la SVM. Cette technique a la possibilité de détecter un nombre considérable de nœuds malicieux, par ailleurs elle génère un taux élevé de faux positifs. Afin de solutionner cette problématique, nous avons intégré dans ce modèle de détection une technique de détection basée sur les signatures d'attaques. En effet, la combinaison de ces deux techniques permet l'obtention d'un système de détection d'intrusion présentant un taux élevé de détection (presque 100%) avec un nombre réduit de faux positifs. D'après les résultats de simulation, notre modèle de détection présente un taux de fausses alarmes très faibles par rapport aux autres modèles hybrides proposés dans la littérature [**61**][**66**]. En outre, notre modèle de détection est intégré dans une topologie à base de cluster, permettant de réduire les coûts de communication, ce qui conduit à l'amélioration de la durée de vie du réseau.

Le processus de communication dans les RCSFs consomme beaucoup d'énergies par rapport au processus de calcul. Dans cette optique, le protocole utilisé doit toujours prendre en considération la quantité d'informations échangées entre les nœuds de capteurs. Dans notre processus d'apprentissage, les IDSs calculent un ensemble de vecteurs clés (appelés vecteurs de support) pour classifier les données localement, par la suite ils les transmettent aux agents voisins afin que tous les IDSs peuvent avoir une vision globale sur le réseau et peuvent détecter toute attaque malicieuse. Par contre, dans l'approche centralisée toutes les données captées par les nœuds sont transmises à la station de base, par la suite le processus d'apprentissage est appliqué par ce nœud. Par conséquent, l'utilisation d'un algorithme d'apprentissage distribué permet de réduire l'énergie consommée au niveau des nœuds et par conséquent prolonger la durée de vie du réseau.

La sélection des attributs pertinents est une étape primordiale, il nous semble intéressant comme perspective à ce travail, au lieu de supprimer un attribut à la fois, d'utiliser d'autres techniques pour la sélection des attributs pertinents comme par exemple : *Particle Swarm Optimization* (PSO) [**88**] ou colonie de Fourmies (*ant theory*) [**89**].

Dans cette étude, nous avons supposé que les agents IDSs sont des nœuds normaux, mais en réalité même les IDSs peuvent être des nœuds malicieux, ce qui conduit à la perturbation du réseau. Le *cluster-head* est un nœud attractif en raison des données cruciales qu'il contient, dans ce travail la sécurité de ce type de nœud n'est pas prise en compte. Tous ces problèmes vont être pris en considération dans les chapitres suivants en proposant d'autres modèles de sécurité qui conviennent aux contraintes énergétiques et de mémoires des nœuds de capteurs.

# Chapitre 4

# Deuxième contribution:

# Mécanisme Hiérarchique de Détection D'intrusion dans le Réseau de Capteurs à Base de Cluster

## Résumé

Dans ce chapitre nous présentons notre nouveau modèle de détection d'intrusion, basé sur une identification hiérarchique des nœuds malicieux. Il est composé de plusieurs protocoles fonctionnant à différents niveaux. Le premier est un protocole de détection basé sur les spécifications. Celui-ci est localisé au niveau des agents IDS (niveau bas**).** Le second est un protocole de classification binaire fonctionnant au niveau du *cluster-head* (niveau intermédiaire). De plus, un protocole de réputation est utilisé à chaque *cluster-head* (CH) pour évaluer le niveau de confiance de ses agents IDS. Chaque CH surveille ses voisins CHs en utilisant le protocole de détection basé sur les spécifications et un mécanisme de vote appliqué au niveau de la station de base (niveau supérieur). Nous avons évalué les performances de notre modèle en présence de quatre types d'attaques*: hello flood, selective forwarding, black hole,* et *wormholes.* Nous avons évalué spécifiquement le taux de détection, taux de faux positifs, la consommation d'énergie et l'efficacité. Notre schéma de détection surpasse d'autres schémas proposés dans la littérature en termes de détection, taux de faux positifs et la consommation d'énergie.

## 1) Introduction

Récemment un certain nombre de travaux se sont focalisés sur une nouvelle forme de détection d'intrusion [**90**][**91**][**92**], appelée détection hiérarchique pour contrer un certain nombre d'attaques qui ciblent plus particulièrement la couche réseau.

Dans [**90**], les auteurs proposent trois type d'IDS: *Misuse Intrusion Detection System* (MIDS), *Hybrid Intrusion* Detection *System* (HIDS) et *Intelligent Hybrid Intrusion Detection System* (IHIDS). Ces IDSs sont intégrés respectivement dans les membres du cluster, *cluster-head* et la station de base. Le MIDS utilise la technique de détection à base de règles pour le processus de détection. En raison des performances du *cluster-head* (défini comme étant un nœud puissant comparativement aux autres nœuds du réseau) trois modules de détection sont intégrés dans ce nœud (HIDS): (i) Détection d'anomalie (ii) Module de détection d'intrusion basée sur l'apprentissage supervisé en utilisant les réseaux de neurone. (iii) Module de prise de décision qui combine les sorties des deux modules précédents pour déterminer s'il existe une intrusion. Finalement, l'IHIDS, intégré à la station de base, utilise un algorithme d'apprentissage non supervisé (*Adaptive Resonance Theory*). Cet algorithme présente des performances de classification et de détection supérieures à celles des réseaux de neurones. Selon leurs résultats de simulation, le schéma proposé présente un taux de détection d'intrusion élevé avec un nombre réduit de faux positifs. Cependant, dans cette approche, tous les nœuds du cluster activent leurs IDSs de manière simultanée, ceci peut causer une charge élevée de communication et de calcul et par conséquent une réduction de la durée de vie du réseau.

Dans [**91**], les auteurs proposent un système de prévention et de détection d'intrusion intégré dans une topologie de clustering à un seul saut. Dans la phase de prévention, les auteurs proposent d'utiliser le

mécanisme de cryptographie pour empêcher la menace extérieure du réseau. Dans la phase de détection d'intrusion, chaque IDS surveille les nœuds qui sont situés dans sa portée radio (un seul-saut). La politique de détection appliquée par les nœuds IDS est basée uniquement sur la détection à base de règles. En utilisant uniquement cette approche pour le processus de détection, cela conduit à un taux faible de détection lorsque plusieurs types d'attaques se produisent. De plus, les auteurs n'ont pas évalué la consommation énergétique des nœuds de capteurs. Le point commun de ces systèmes de détection d'intrusion hiérarchique dans le RCSF [**90**][**91**] est que les auteurs n'ont pas pris en compte le fait que les agents IDS peuvent être aussi des nœuds malveillants.

Ce chapitre est consacré à notre deuxième contribution relative à la détection des nœuds malveillants dans le réseau de capteurs sans fil à base de cluster (RCSFC). Dans notre approche l'opération de surveillance (*monitoring*) se fait d'une manière hiérarchique, en d'autre terme le mécanisme de détection d'intrusion s'exécute dans chaque niveau (les membres du cluster, *cluster-head* et la station de base), comme il est illustré dans la Figure 4.1. L'objectif de cette approche est la détection des nœuds malicieux qui se situent dans différents niveaux. Ce processus peut conduire à une identification précise du nœud malveillant qui cible les membres du cluster et le CH (la station de base étant supposée un nœud normal). Cette détection peut être appliquée entre les membres du cluster, entre les membres du cluster et le *cluster-head* et entre les *cluster-heads*.

Dans ce qui suit, nous allons définir des informations de base liées à notre contribution. Par la suite, nous allons décrire nos différents protocoles de détections d'intrusion implémentées d'une manière hiérarchique dans les RCSFCs. Finalement, une étude approfondie basée sur des simulations et une analyse des performances de notre schéma de détection va être élaborée.

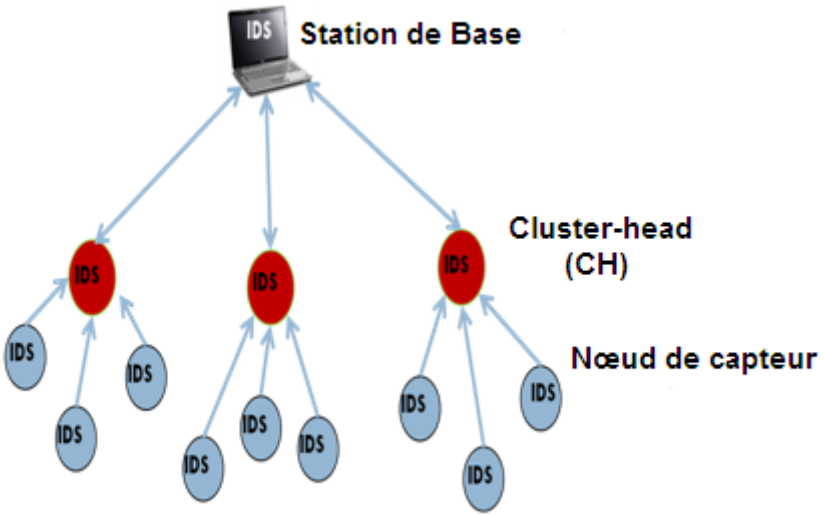



**Figure 4.1. Détection hiérarchique d'intrusion**

## 2) Contexte

Dans cette section, nous allons décrire quelques règles de détection relative aux attaques: *Selective forwarding*, *Black hole*, *Hello flood* et *Wormholes.* Par la suite, nous décrirons le principe de fonctionnement du protocole HEED (*Hybrid Energy-Efficient Distributed Clustering*) à base de cluster [**23**], sélectionné comme étant le protocole de routage utilisé par notre modèle de détection d'intrusion.

### 2.1 Attaques de routage et leurs symptômes

L'intrus pourrait réaliser l'une des quatre attaques suivantes: *Selective forwarding*, *Black hole*, *Hello flood*, et *Wormholes*. Dans cette section nous allons décrire les règles de détection liées à ces attaques. Nous notons que la politique de détection d'intrusion adoptée est basée sur des spécifications (voir chapitre 2, sous section 2.1).

**a.** ***Selective forwarding***. Dans ce type d'attaque, l'intrus arrête la transmission de certains paquets, par la suite il les supprime. Cette attaque est détectée en calculant le nombre de paquets supprimé (NPD: *Number of Packets Dropped*).

**b.** ***Black hole.*** Dans cette attaque, l'intrus prétend être dans le plus court chemin ver le *cluster-head* en utilisant une puissance de transmission élevée. Dans ce cas, l'intrus sera en mesure de recevoir tous les messages ce qui induit leur suppression. Cette attaque peut être détectée en calculant le nombre de paquets supprimés (NPD) et l'intensité du signal reçu (RSSI: *Received Signal Strength Intensity*).

**c.** ***Hello flood.*** Le nœud malicieux diffuse les paquets Hello et génère un signal assez puissant comparativement aux autres nœuds. Dans ce cas, d'autres nœuds légitimes envoient leurs paquets vers ce nœud malicieux. En conséquence, les paquets seront ensuite supprimés ou modifiés. Cette attaque peut être détectée par le calcul de RSSI.

***d.*** ***Wormholes***. Selon le travail entrepris par les auteurs dans [**93**], l'attaque *Wormhole* est classée comme étant une attaque passive ou active. Dans notre étude, nous nous focalisons sur l'attaque *Wormhole* active. Ce type d'attaque a tendance à faire semblant d'être à un seul saut d'écart *(one hop away)* du *cluster-head (CH)* en utilisant une force de signal élevée. Par conséquent, l'attaquant transmet les messages reçus à partir d'un nœud légitime à un autre attaquant, comme illustré à la Figure 4.2. Dans ce cas les deux nœuds malveillants prendront part dans le protocole de routage du réseau. Dans la Figure 4.2, nous notons que M1 et M2 sont les extrémités du tunnel de *Wormhole* et le nœud M1 génère une force de signal très importante afin de convaincre les nœuds qu'il est proche du *cluster-head (*un saut d'écart du *CH).* Le nœud A veut envoyer ses paquets au CH, soit en suivant la route valide (les nœuds B et C) ou l'itinéraire malveillant (nœuds M1, E, et M2). Dans les deux cas, le nœud A choisit la route à moindre coût via M1-M2 *Wormhole* (représenté en flèches pleines) car le nœud M1 prétend être le plus proche du CH. Par conséquent, tous les paquets reçus par M1 de A sont transmis directement au nœud malicieux M2 sans passer par le nœud E. Affin de

détecter ce type d'attaque, nous surveillons la force des signaux générés par les nœuds. De plus, les nœuds qui ce situent dans le même voisinage du nœud attaquant (qui génère un très fort signal) ne reçoivent pas les paquets transmis par ce nœud malveillant, par conséquent un taux élevé de paquets rejetés (NPD) sera produit dans le réseau. Comme il est illustré dans la Figure 4.2, l'agent IDS1 détecte que le nœud M1 émet des paquets avec un RSSI élevé. De plus cet agent constate que le nœud E ne retransmet aucun paquet au nœud M2 (NPD élevé). En se basant sur ces deux attributs (RSSI et NPD), le nœud M1 est identifié comme étant un nœud malveillant qui réalise l'attaque *Wormhole.*

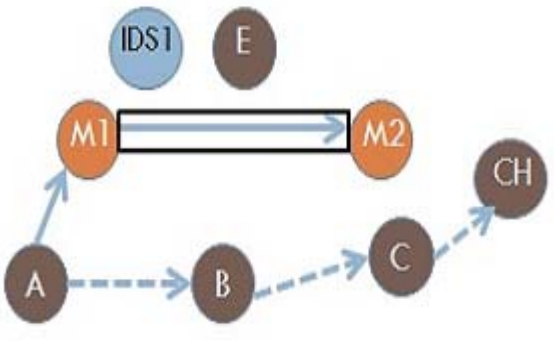



**Figure 4.2. Attaque *Wormhole* active**

## 2.2 Protocole de routage à base de cluster

L'architecture à base de cluster vise à conserver l'énergie des nœuds de capteurs, conduisant ainsi à l'amélioration de la durée de vie du réseau. Ce résultat est obtenu en affectant au nœud CH la responsabilité de la transmission des paquets (contenant les données agrégées reçues des membres du cluster) à la station de base.

Dans [**23**], les auteurs ont proposé un algorithme de clustering appelé HEED (*Hybrid Energy-Efficient Distributed Clustering*) pour les réseaux de capteurs. L'objectif de ce protocole est d'utiliser une combinaison entre l'énergie résiduelle des nœuds et le coût de communication intra-cluster pour élire le chef de groupe CH. Ce protocole vise à réaliser une distribution uniforme des *cluster-heads* (CHs) dans le réseau et à générer des clusters équilibrés en taille [**18**].

Dans notre étude, une version modifiée de ce protocole de routage est sélectionnée (en utilisant uniquement l'énergie résiduelle) pour intégrer notre modèle de détection d'intrusion.

En HEED, les auteurs ont défini deux types de nœuds: «découverts» et «couverts». Dans ces cas, le premier nœud annonce être le *cluster-head* en diffusant un message d'annonce aux autres nœuds du réseau. Ce processus se produit lorsque l'exécution de l'algorithme est terminée sans élection du *cluster-head*. Le nœud couvert est un membre du cluster qui s'attache au CH le plus proche et ceci

selon le message d'écoute (*overheard message*) émis par le *cluster-head.* A cette fin, un nœud est élu comme *cluster-head* avec une formule de probabilité égale à:

$$CH_{prob} = C_{prob} \times \frac{E_{residual}}{E_{\max}} \quad (4.1)$$

Où *Eresidual* et *Emax* sont respectivement les énergies résiduelle et maximale dans le nœud, et Cprob est le nombre optimal de clusters.

## 3) Modèle hiérarchique de détection d'intrusion : les différents composants & Principe de fonctionnement

Dans notre modèle, le processus de détection d'intrusion est effectué à trois niveaux, comme il est détaillé dans les paragraphes suivants. Dans le niveau bas, un ensemble de nœuds appelés agents IDS surveillent la communication de leurs voisins et envoient leur rapport au CH correspondant pour des détections antérieures. Pour identifier tout comportement suspect, ces agents IDS utilisent la technique de détection basée sur les spécifications. Cette technique repose sur un ensemble de règles pour détecter et prévenir les comportements malveillants (plus de détails dans le paragraphe 3.1.2). En raison des contraintes énergétiques et le fait qu'un bit transmis dans les RCSFs consomme une énergie qui équivaut à 800-1000 instruction [**80**][**81**], le nœud IDS doit limiter la quantité d'informations échangées entre les IDSs voisins et le *cluster-head*.

Dans le niveau intermédiaire, un puissant (*powerful node*) *cluster-head* utilise la machine à vecteurs de support (SVM) pour le processus d'apprentissage et du test pour la détection d'anomalie (voir chapitre 3, section 2). Cet algorithme d'apprentissage est défini comme étant un classificateur binaire car il permet de séparer les données en deux classes (normal et anomalie). Étant donné qu'aucun nœud n'est supposé être digne de confiance, un mécanisme de réputation est intégré au niveau du CH afin d'évaluer le niveau de confiance de ces membres IDSs. Le processus de détection qui se produit entre le premier niveau (agents IDS) et le second niveau (CH), il est illustré dans la Figure 4.3. Dans le niveau supérieur, chaque CH surveille ses voisins CH en se basant sur la technique des spécifications et transmet par la suite un formulaire de vote à la station de base contenant le CH suspect lorsque celui-ci produit une attaque. La station de base recueille les votes générés par les CHs, prend une décision finale sur le nœud suspect et finalement éjecte les nœuds malicieux du réseau.

Le processus de détection qui se produit entre le second niveau (CH) et le troisième niveau est illustré dans la Figure 4.4.

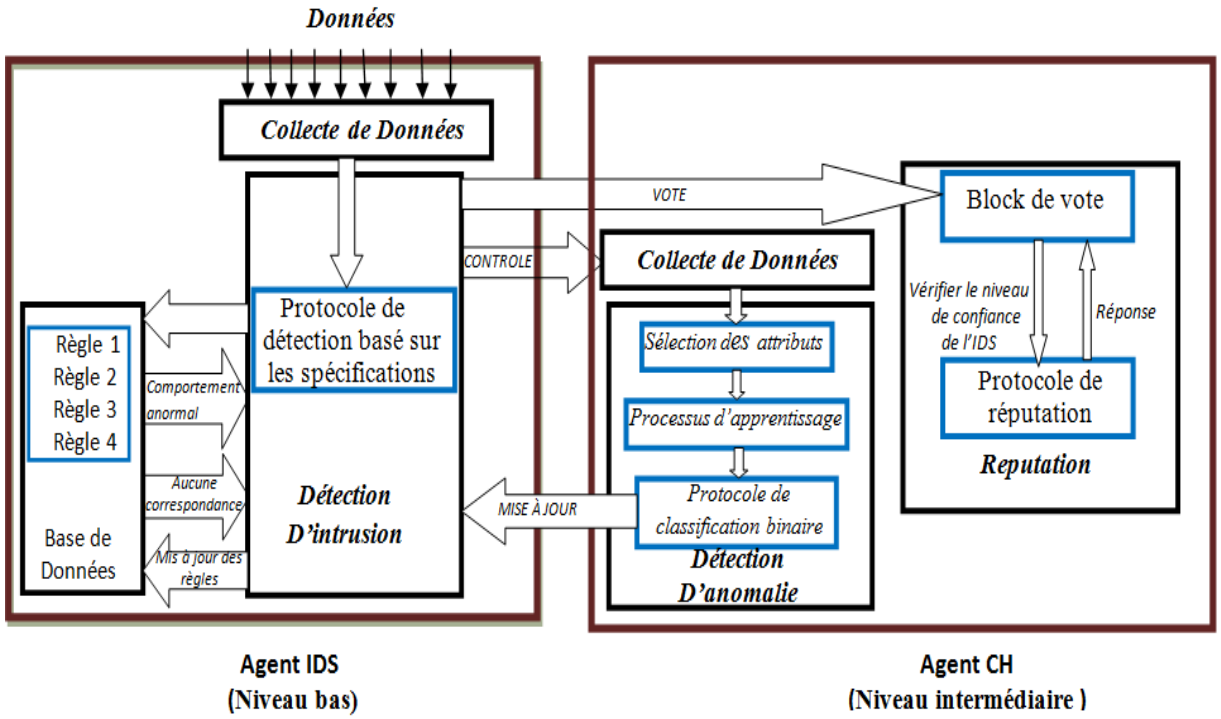



**Figure 4.3. Procédé de détection entre les agents IDS et CH**

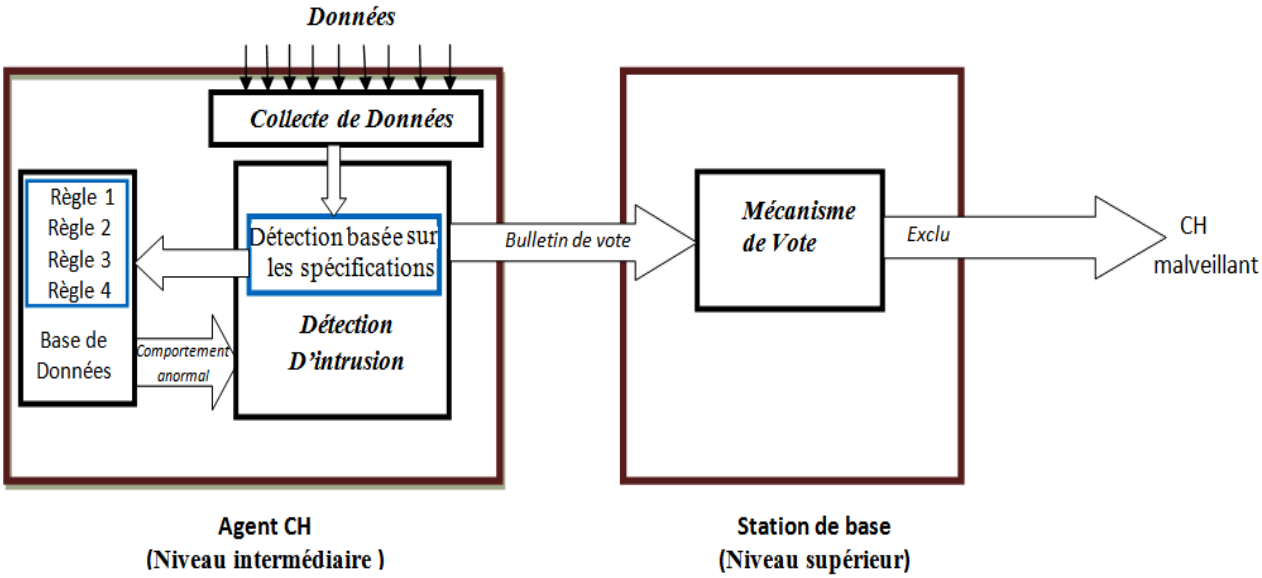



**Figure 4.4. Procédé de détection entre l'agent CH et la station de base**

Dans ce qui suit, nous donnons plus de détails sur les différents protocoles de détection utilisés par notre modèle hiérarchique.

### 3.1 Le niveau bas: Détection d'intrusion au niveau des nœuds de capteurs

Dans chaque cluster et pour chaque lien de communication, il doit y avoir au moins un agent IDS pour la collecte et l'analyse des paquets au sein de la zone de couverture radio. Tel qu'il est illustré dans la Figure 4.3 (voir agent IDS), les modules de Collecte de Données et de Détection d'intrusion sont les composants les plus importants dans ce type d'agent.

**1. Module de Collecte de Données.** En raison de la nature de diffusion des réseaux sans fil, les nœuds IDS collectent les paquets à l'intérieur de leur zone de couverture radio [**66**], par la suite les données sont transmises au module de détection d'intrusion pour le processus d'analyse comme le montre la Figure 4.3.

**2. Module de Détection D'intrusion.** Ce module utilise un protocole de détection basé sur les spécifications pour détecter les nœuds malveillants et empêcher les perturbations réseau subies par ces nœuds. Le but de ce protocole est de classer le comportement d'une cible comme étant normale ou anormale en se basant sur un ensemble de règles. Dans notre cas il y'a quatre règles relatives à chaque attaque. La règle pour détecter l'attaque *Selective forwarding* est définie par le nombre de paquets rejetés (NPD) par un nœud et qui est supérieur à un certain seuil ($\delta_{sf}$). La règle pour détecter l'attaque *Hello flood* est l'intensité du signal reçu (RSSI) au niveau de l'agent IDS, elle est supérieure à un certain seuil ($\delta_{rssih}$). La règle pour détecter l'attaque *Black hole* est définie par le nombre de NPD (supérieur au seuil$\delta_{bh}$) et l'excès de la puissance du signal (supérieur au seuil $\delta_{rssibh}$). Finalement la règle pour détecter l'attaque *Wormholes* est l'excès de la puissance du signal (supérieur au seuil $\delta_{rssiwo}$) et aucun des nœuds situés dans le voisinage du nœud malveillant ne fait la retransmission des paquets reçus de cet adversaire (le NPD dépasse le seuil $\delta_{wo}$). Toutes ces règles exploitées pour la détection des attaques sont illustrées sur la Figure 4.5.

```
// Rule for selective forwarding attack
if (NPD > δ_sf)
 // node_ID is performing a selective forwarding attack
    send_VOTE_message_CH (node_ID);
else
    send_CHECK_message_CH (node_ID,NPD);

// Rule for hello flood attack
if (RSSI> δ_rssih )
 //node_ID is performing a hello flood attack
    send_VOTE_message_CH (node_ID);
else
    send_CHECK_message-CH (node_ID,RSSI);

// Rule for Black Hole attack
if (NPD > δ_bh  && RSSI> δ_rssibh)
 //node_ID is performing a black Hole attack
    send_VOTE_message_CH(node_ID);
else
    send_CHECK_message_CH (node_ID,NPD,RSSI);

// Rule for Wormholes attack
if (RSSI> δ_rssiwo) {
    Monitor(neighbors(node_ID));
    if (NPD > δ_wo)
     // node_ID is performing a Wormholes attack
        send_VOTE_message_CH (node_ID);
    else
        send_CHECK_message_CH (node_ID, RSSI,NPD);
}
```

**Figure 4.5. Règles de détection des quatre attaques**

Comme il est illustré dans la Figure 4.3, lorsqu'un comportement anormal est détecté en fonction de la règle sélectionnée, un message de *VOTE* est soumis au bloc de vote (situé dans le CH) pour déterminer avec une grande précision si le nœud suspect est malveillant ou pas. Ce block applique un mécanisme de vote en calculant le nombre de fois où les agents IDS ont détecté un nœud comme étant une attaque. On note que le message de *VOTE* comprend le nœud suspect et le type d'attaque détecté. Lorsque le vote dépasse un certain seuil, le CH n'attribue aucun time slot au nœud malicieux et par la suite il sera éjecté du réseau. Cependant lorsque l'agent IDS n'a détecté aucun comportement anormal (aucune correspondance), un message de *CONTROLE* est envoyé par cet agent au module de détection d'anomalie (situé dans le CH) pour une détection plus avancée (un système d'apprentissage basé sur la SVM). Ce message comprend le nœud analysé avec le NPD et la RSSI.

### 3.2 Le niveau intermédiaire : Détection d'intrusion au niveau du *cluster-head*

Inspiré du travail des auteurs de la référence [**23**], notre algorithme de clustering a été implémenté sous le simulateur TOSSIM [**87**]. Nous avons choisi dans chaque cluster un CH qui a plus de ressources de puissance pour gérer et agréger les données provenant des membres du cluster. Tel qu'il est illustré dans la Figure 4.3 (voire agent CH), ce nœud puissant est composé de trois modules : Collecte de Données, Détection D'anomalie et Réputation.

**1. Module de Collecte de Données.** Ce module est responsable de collecter le message *CONTROLE* envoyé par l'agent IDS. Ce message inclut l'adresse du nœud analysé par l'agent IDS et les attributs suivantes: NPD et RSSI. Ces attributs sont ensuite transmis au module de détection d'anomalie pour le processus d'apprentissage et de classification.

**2. Module de Détection D'anomalie.** La procédure de détection des anomalies est divisée en trois étapes:

- **Etape1 :** *Sélection des attributs.* C'est un facteur important, car le choix des attributs les plus pertinents conduit à une augmentation de la précision de la classification, réduction des faux positifs et l'accélération du temps d'apprentissage. Dans cette recherche le NPD et la RSSI sont utilisés comme des données d'entrée pour le processus d'apprentissage
- **Etape 2 :** *Processus d'apprentissage*. Pour la détection d'anomalie un algorithme d'apprentissage distribué basé sur la SVM est utilisé pour classer les données comme normales ou anormales. Lors du processus d'apprentissage chaque CH calcule les vecteurs de support qui sont moins nombreux que les données d'entrées utilisées lors du processus d'apprentissage. Ces vecteurs seront envoyés au CH adjacent qui est situé dans la même zone de couverture radio. Chaque CH qui reçoit les vecteurs de support de ses voisins CHs met à jour l'information correspondante en unifiant les vecteurs reçus et ses propres vecteurs de support. Finalement tous les *cluster-heads* dans le réseau ont les mêmes vecteurs de support; ce qui conduit à la détermination de l'hyperplan séparateur global pour classer les données comme normales ou anormales.
- Etape 3 : *Protocole de classification binaire*. Lorsque le processus d'apprentissage est terminé, chaque CH classifie les nouvelles données entrantes en fonction du modèle d'apprentissage obtenu. Tout écart par rapport au comportement normal est considéré comme une anomalie. Dans ce cas, un message de *MISE À JOUR* est renvoyé aux agents IDS du même cluster pour calculer la nouvelle règle correspondante à cette attaque.

Les trames des paquets de tous les messages échangés (*VOTE*, *CONTROLE*, *MISE À JOUR*) ont les formats présentés dans la Figure 4.6.

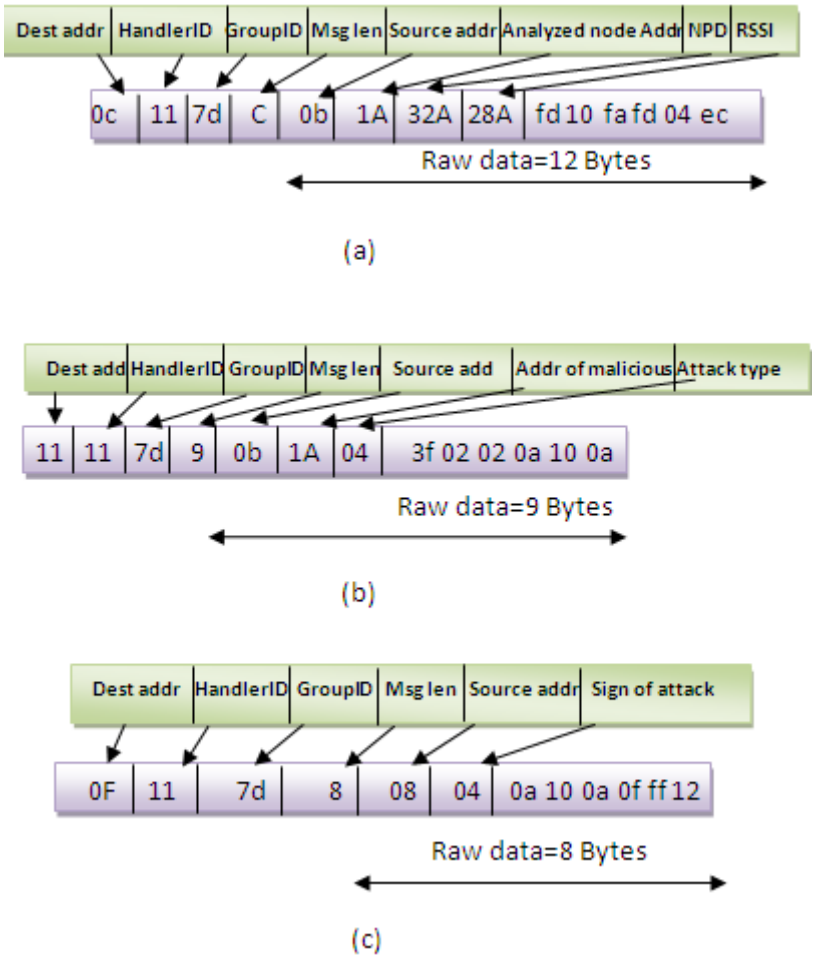



**Figure 4.6. Formats de paquet des messages de:**
**(a) CONTROLE, (b) VOTE, (c) MISE À JOUR**

**4. Module de Réputation.** Lorsque l'agent IDS détecte une attaque, il envoie un message de *VOTE* (qui contient le nœud suspect et le type d'attaque) à son CH tel qu'il a été exposé dans la Figure 4.3. Le nœud CH utilise le block de vote pour déterminer si le nœud suspect et un intrus ou pas, tandis que le protocole de réputation évalue le niveau de confiance des agents IDS. Notre protocole de réputation est inspiré du travail des auteurs dans la référence [**94**]. Si un vote est supérieur au seuil prédéfini, le nœud suspect est éjecté du réseau et la réputation des nœuds IDS ayant détecté l'attaque sera augmentée. Autrement, la réputation des IDSs sera diminuée. On note que pour chaque cluster, ce seuil est égale à n/2 où n est le nombre d'agents IDS dans chaque cluster. Les systèmes de réputation constituent une étape supérieure qui peut aider les systèmes de détection d'intrusion à mieux détecter les nœuds attaquants [**95**], et plus particulièrement à diminuer le nombre de faux positifs causés par les nœuds malicieux. Dans notre cas, nous avons pris en compte le fait que les agents IDS peuvent

être des nœuds malicieux et donner une confirmation fausse à propos du comportement d'un nœud. La réputation $R_i$ de l'agent $IDS_i$ maintenue à son CH correspondant, est définie comme suit [**94**] :

$$R_i = \beta eta(\alpha_i + 1, \beta_i + 1) \qquad (4.2)$$

*Beta* est la fonction de réputation [**94**], $\alpha_i$ et $\beta_i$ représentent respectivement le comportement normal et malveillant de l'$IDS_i$ revendiqué par le CH. La mise à jour de ces deux paramètres est décrite dans la référence [**94**].

La métrique de Confiance (*Trust metric*), définie comme le niveau de confiance du nœud IDS, est calculée comme suit:

$$T_i = E[R_i] \qquad (4.3)$$

Où *E[R]* est l'espérance mathématique de la fonction de réputation. La valeur de confiance est classée par la fonction *M (T)* suivante :

$$M(T_i) = \begin{cases} élevé\ T_i \geq TH \\ faible\ T_i < TH \end{cases} \qquad (4.4)$$

Après le calcul de la valeur de confiance, chaque CH compare cette valeur avec le niveau d'exigence de confiance (TH). Seuls les IDSs ayant une valeur de confiance élevée peuvent déclencher leur processus de détection d'intrusion. Autrement, ils seront définis comme des nœuds normaux et ne peuvent pas être en mesure de jouer le rôle d'un agent IDS. En conséquence, une communauté d'IDSs digne de confiance sera générée.

### 3.3 Le niveau supérieur : détection d'intrusion intra-cluster

Le CH est une cible attrayante pour un attaquant car il contient des données pertinentes. En conséquence, l'intrus utilise toute sa capacité afin de lancer une attaque contre ce point chaud. Afin d'éviter ce problème, chaque CH surveille ses voisins CHs. Le *cluster-head* est équipé aussi d'un module de Détection D'intrusion. La station de base est équipée d'un Mécanisme de Vote. Ces modules ont les fonctions suivantes :

**1. Module de Collecte de Données.** Chaque *cluster-head* capture les paquets provenant d'autres CHs situés dans la même zone de couverture radio, ensuite les deux attributs (NPD et RSSI) sont calculés. Par la suite, cette information sera transmise au module de détection d'intrusion pour un processus de surveillance (voire Figure 4.4).

**2. Module de Détection D'intrusion.** Chaque *cluster-head* surveille ses voisins CHs en adoptant une politique de détection basée sur les spécifications (comme celles utilisées par les agents IDS). Selon les règles relatives à chaque attaque (voir le paragraphe 3.1.2 a propos de ces règles), si un comportement anormal se produit, le CH de surveillance envoie à la station de base un bulletin de vote qui comprend le CH suspect et le type d'attaque détecté tell qu'il est illustré dans la Figure 4.4. La station de base effectue un mécanisme de vote afin d'identifier les nœuds malicieux. Dans le cas où plus de la moitié des votes sont en faveur d'une l'attaque, le CH sera exclu du réseau et un nouveau CH sera élu.

## 4) Évaluation des performances

Dans notre expérience, nous avons utilisé le simulateur TOSSIM [**87**] qui est un simulateur pour les capteurs dotés du système d'exploitation TINYOS. Le principal avantage de ce simulateur par rapport à d'autres outils, tels que NS2 [**85**], réside dans la facilité d'intégration du code source écrit en NESC [96] dans les capteurs munis du système d'exploitation TINYOS. Cependant, le simulateur TOSSIM n'a pas la capacité de modéliser l'énergie dissipée pendant l'exécution de l'application. De ce fait, une version améliorée de l'outil a été proposée par l'Université de Harvard appeler POWERTOSSIM [**97**]. Ce dernier permet la simulation de la consommation d'énergie et par conséquent la déduction de la durée de vie du réseau. Les simulateurs TOSSIM et POWERTOSSIM sont détaillés en Annexe B.

### 4.1 Hypothèses de simulation

Nous avons simulé un réseau composé de 168 nœuds (capteurs) déployés de façon aléatoire dans une zone carrée de (88* 88)$m^2$ . On note que le réseau est constitué de 8 clusters, de plus tous les nœuds sont statiques. Afin d'éviter les collisions, le protocole TDMA est utilisé. Nous utilisons le circuit intégré CC1000 [**98**] comme un émetteur-récepteur et chaque nœud transmet ses paquets à une fréquence entre 433 MHz et 868 MHz. Tous les paramètres clés de la simulation sont résumés dans le Tableau 4.1, où les valeurs des seuils de détection pour chaque attaque ont été déterminés en effectuant plusieurs simulations.

| | |
|---|---|
| Temps de simulation | 875 seconds |
| Domaine de la simulation | 88 * 88m$^2$ |
| Nombre de nœuds | 168 |
| Modèle radio | Lo*ssy* |
| Nombre de cluster | 8 |
| Nombre d'agents IDS par cluster | 1-10 |
| Protocole de routage | HEED modifier |
| MAC | TDMA |
| Portée radio | 15m |
| L'énergie initiale | 5 Joules |
| $\delta_{sf}$ | 64 % |
| $\delta_{rssih}$ | -41 (dBm) |
| $\delta_{bh}, \delta_{rssibh}$ | 94 %, -47 (dBm) |
| $\delta_{rssiwo}, \delta_{wo}$ | -44 (dBm), 99% |

**Tableau 4.1. Paramètres de simulation**

Le but de nos simulations est d'étudier l'effet de chaque attaque sur le réseau, puis l'impact de toutes ses attaques. En supposant qu'il n'y a aucune attaque au début de la simulation, nous avons varié le nombre de nœuds d'IDS par cluster de 1 à 10 afin d'évaluer les performances de notre modèle de détection pour les différentes configurations. Afin d'évaluer les performances de notre modèle; différentes métriques ont été analysées: **Taux de détection, Taux de faux positifs (fausses alarmes), Efficacité et Energie totale consommée**. Toutes ces métriques ont été détaillées dans le second chapitre de cette Thèse.

## 4.2 Analyse des résultats

1. **Scénario de l'attaque *Hello flood.*** Ce type d'attaque a été implémenté comme étant un nœud qui génère une force de signal élevée par rapport aux autres nœuds du réseau. Comme le montre la Figure 4.7 (a), lorsque le nombre d'IDSs augmente, le taux de détection augmente en même temps avec le nombre de faux positifs. Lorsque le nombre moyen d'IDSs dans chaque cluster est égal à 4, les taux de détection et de faux positifs sont proches respectivement de 98% et de 2%. En outre, comme le montre la Figure 4.7 (b) lorsque le nombre moyen d'IDSs dans chaque cluster est égal à 4, notre modèle de détection nécessite moins de temps pour détecter l'attaque *Hello flood* (l'efficacité est proche de 2 secondes). Enfin, nous concluons que lorsqu'un nombre optimal d'agents IDS est déterminé (4 agents par

clutser) notre modèle présente un taux de détection élevé, un faible nombre de fausses alarmes et nécessite moins de temps pour détecter ce type d'attaque.

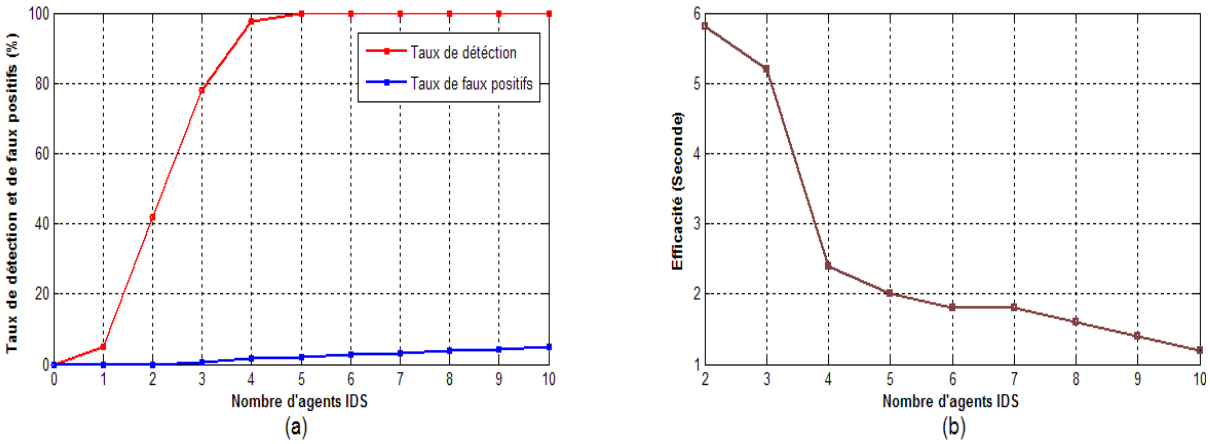



**Figure 4.7. Scénario de l'attaque *Hello flood* :**
**(a)Taux de détection et de faux positifs, (b) Efficacité**

2. **Scénario de l'attaque *Selective forwarding.*** Cette attaque empêche la retransmission d'un nombre considérable de paquets en comparaison aux nœuds légitimes. Le taux de détection et le nombre de fausses alarmes sont liés aux nombre d'agents IDS dans chaque cluster. Comme le montre la Figure 4.8 (a), les valeurs de ces deux paramètres augmentent avec le nombre d'agents. Par conséquent, le nombre optimal d'IDSs qui permet la détection de l'attaque *Selective forwarding* avec une faible occurrence de faux positifs est égal à 6. En outre, selon ce nombre optimal d'agent IDS, notre modèle nécessite un temps de détection presque égal à 2 secondes pour détecter cette attaque comme le montre la Figure 4.8 (b). Par conséquent, un compromis entre le nombre d'IDSs et les faux positifs doit être effectué.

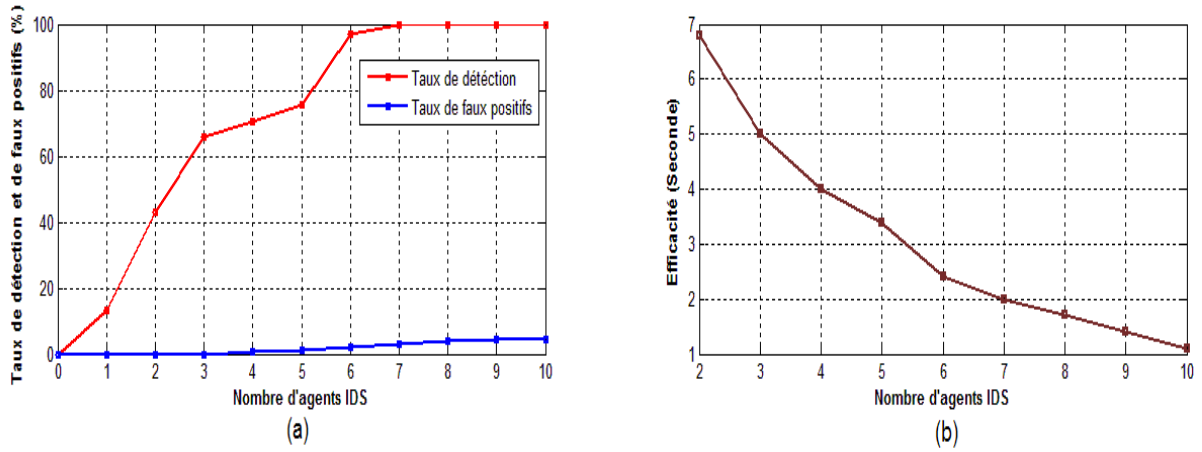



**Figure 4.8. Scénario de l'attaque *Selective forwarding :***
**(a)Taux de détection et de faux positifs, (b) Efficacité**

3. **Scénario de l'attaque *Black hole.*** Dans ce type d'attaque le nœud malveillant génère une très forte force du signal et supprime tous les paquets reçus. La performance de détection de notre modèle sous les attaques de type *black hole* est illustré dans la Figure 4.9 (a). Lorsque le nombre moyen d'IDSs dans chaque cluster est égal à 5, notre modèle a la possibilité de détecter ce type d'attaque avec un taux de détections supérieur à 96%. De plus, selon ce nombre optimal d'agents IDS (égale à 5) nous remarquons d'après la Figure 4.9 (a) qu'un nombre réduit de faux positifs est généré par les agents lorsque l'attaque *black hole* se produit. D'après la Figure 4.9 (b), lorsque le nombre d'IDS par chaque cluster est égal à 10, le temps nécessaire pour la détection de ce type d'attaque de l'ordre de 1,5 secondes. Par ailleurs, un nombre élevé de fausses alarmes se produit lorsque nous choisissons 10 agents dans chaque cluster. En conséquence, le nombre optimal des nœuds IDS par chaque cluster, répondant aux exigences de l'application (le temps de détection, le taux de détection et le nombre de fausses alarmes), est égal à 5.

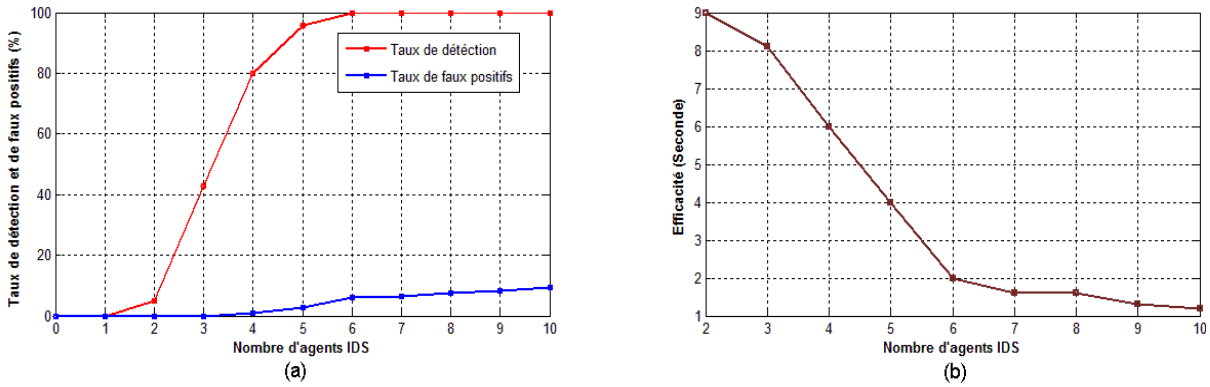



**Figure 4.9. Scénario de l'attaque *Black hole :***
**(a)Taux de détection et de faux positifs, (b) Efficacité**

4. **Scénario de l'attaque *Wormholes.*** Cette attaque a été implémentée comme suit: le nœud malveillant génère un très fort signal. De plus, les nœuds qui se situent dans le même voisinage de cet attaquant ne reçoivent aucun paquet transmis par celui-ci. Le taux de détection atteint presque 100% lorsque le nombre d'agents augmentent, comme le montre la Figure 4.10 (a). Dans ce cas, le nombre optimal d'agents IDS par cluster, fournissant un compromis entre le taux de détection et le nombre de faux positifs sous l'attaque *Wormhole*, est égal à 5. La détection de l'attaque *Wormhole* nécessite un temps considérable par rapport aux autres types d'attaques, tel qu'il est illustré dans la Figure 4.10 (b). L'utilisation de 6 agents dans chaque cluster conduit à un temps de détection égal à 4,5 secondes. En conclusion, un nombre optimal de 6 agents IDS permet de contrer les attaques *Wormholes*, avec un faible nombre de faux positifs, un taux de détection élevé et un temps de détection court.

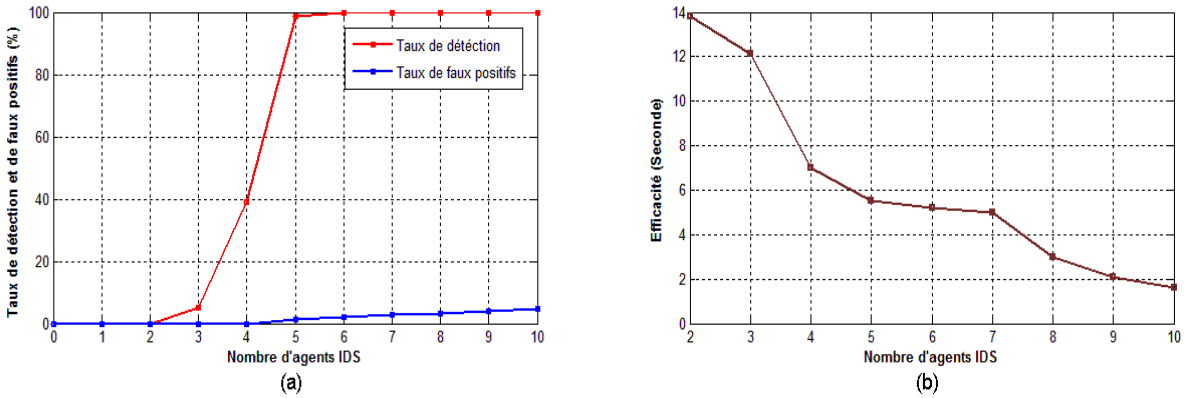



**Figure 4.10. Scénario de l'attaque *Wormholes* :**
**(a)Taux de détection et de faux positifs, (b) Efficacité**

5. **Scénario de plusieurs attaques.** Dans cette section, nous évaluons les performances de notre modèle de détection lorsque différentes attaques apparaissent dans le RCSFs. Tout d'abord, nous évaluons le taux de détection de notre modèle sous les attaques *Selective forwarding* et *Black hole*, et nous le comparons à celui proposé par les auteurs dans [**99**]. Deuxièmement, nous étudions les performances de notre modèle de détection lorsque toutes les attaques citées précédemment (i.e. *hello flood, selective forwarding, black hole* et *wormholes* ) apparaissent. Ici, nous comparons les performances de notre modèle avec un schéma proposé dans la référence [**66**] en termes du taux de détection, du taux de faux positifs et de l'efficacité. De plus, afin de déterminer l'efficacité énergétique de notre modèle, nous comparons les résultats obtenus à ceux de la référence [**79**]. Ainsi, comme le montre la Figure 4.11, notre modèle effectue une meilleure détection contre les attaques *Black hole* et *Selective forwarding* comparé au schéma proposé par les auteurs de la référence [**99**], en particulier lorsque le nombre d'IDSs est important. Dans ce cas, le nombre de fausses alarmes est lié aux nombres d'IDSs. En conséquence, l'augmentation du nombre d'agents IDS par cluster engendre une augmentation du taux de faux positifs. Nous devons donc envisager un équilibre entre le nombre de faux positifs et le taux de détection. Par conséquent, le nombre optimal d'IDSs par cluster répondant aux exigences de l'application est égal à 6.

   En présence de toutes les attaques, notre modèle de détection d'intrusion est efficace lorsque le nombre d'agents IDS est important (Figure 4.12 (a)). Cependant, le nombre de faux positifs affectera la performance de notre modèle de détection lorsque le nombre d'agents est élevé (dépasse 6 agents pour chaque cluster). De ce fait, nous devons envisager un compromis entre le nombre d'IDSs et le taux de faux positifs. Ainsi, le nombre optimal d'agents IDS par chaque cluster, répondant aux exigences de l'application, est égal à 5.

Les taux de détection et de faux positifs sont respectivement de l'ordre de 98% et 2%. Comme le montre la Figure 4.12 (a), les deux schémas présentent un taux de détection élevé avec un faible taux de fausses alarmes. Par ailleurs, lorsqu'un nombre optimal d'agents IDS est sélectionné (5 agents dans chaque cluster), notre modèle effectue une meilleure détection avec un nombre faible de fausses alarmes par rapport au schéma proposé dans la référence [**66**]. En utilisant ce nombre optimal d'agents pour chaque cluster, le temps requis d'IDS pour détecter le premier nœud malveillant dans le réseau est proche de 4 secondes (voir Figure 4.12(b)). Enfin, nous concluons que lorsque nous utilisons ce nombre optimal d'agents IDS dans chaque cluster, notre modèle de détection d'intrusion présente un faible nombre de faux positifs, un taux de détection élevé et un temps de détection court.

Nous pouvons observer dans la Figure 4.12 (c), que notre modèle de détection nécessite moins d'énergie pour détecter toutes ces attaques, comparativement à l'approche de détection utilisée par les auteurs dans [**79**]. Cette amélioration a été obtenue grâce à deux principales raisons: la première est que nous utilisons une topologie à base de cluster qui vise à sélectionner un seul nœud par cluster (*cluster-head*) pour transmettre les données agrégées à la station de base. La deuxième raison est le fait que chaque agent IDS s'appuie sur une politique qui minimise la transmission des paquets, qui à son tour permettra d'économiser l'énergie. En conclusion, nous pouvons affirmer que notre approche améliore la durée de vie du réseau.

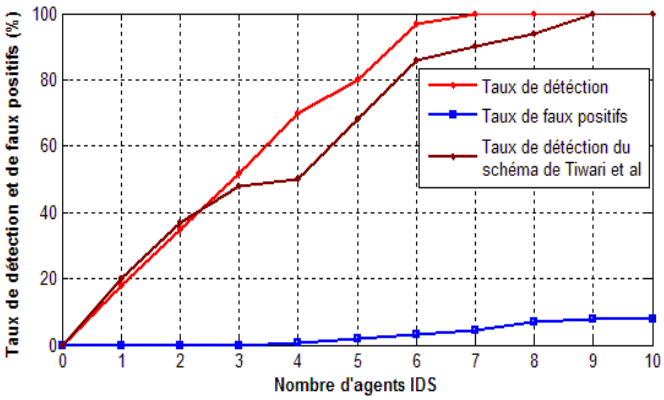



**Figure 4.11.Comparaison de notre modèle sous les attaques *Black hole* et *Selective forwarding***

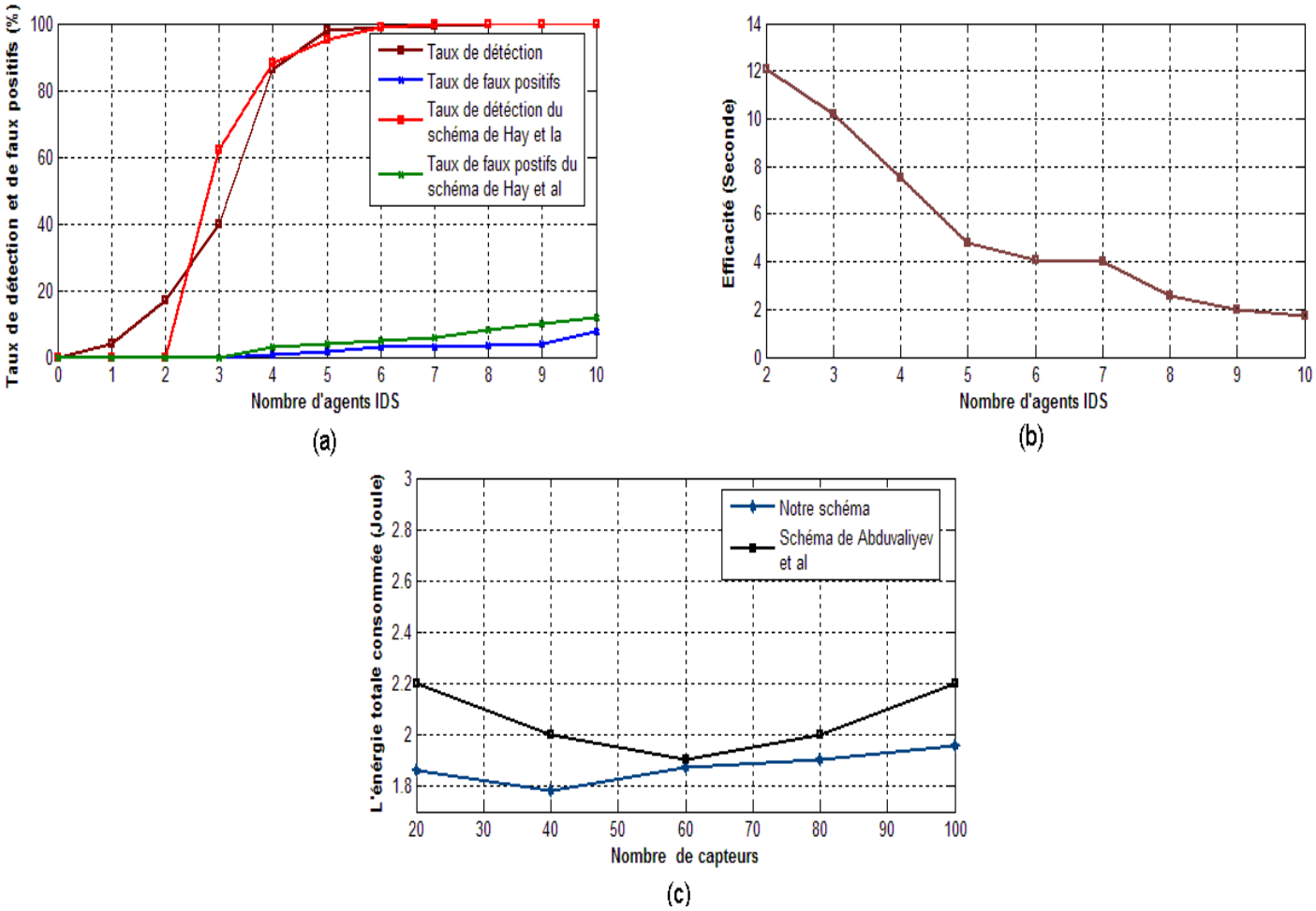



**Figure 4.12. Scénario de plusieurs attaques :**

**(a)Taux de détection et de faux positifs, (b) Efficacité**

**(c) L'énergie totale consommée**

## 5) Conclusion

Dans ce chapitre, nous proposons un schéma de détection d'intrusion, efficace contre certains types d'attaques de routage pouvant entrainer un mauvais fonctionnement du réseau. De plus, ce schéma consomme moins d'énergie pour détecter ce type d'attaques. Le but de notre modèle est d'appliquer un ensemble de protocoles de détection d'intrusions sur les réseaux de capteurs à base de cluster qui s'exécutent à différents niveaux (i.e., dans les membres du cluster, *cluster-head* et la station de base) afin d'identifier et empêcher toutes les attaques malicieuses qui visent à perturber le réseau. Au niveau des membres (nœuds) du cluster, la technique de détection à base de règles est implémentée sur les agents IDS pour identifier toute attaque entrante. En même temps, au niveau du *cluster-head*, la détection d'anomalie basée sur la classification binaire est intégrée dans chaque CH, celle-ci vise à actualiser les règles stockées dans les IDSs. De plus, un protocole de réputation est utilisé par le CH pour évaluer le niveau de confiance de ces IDSs. À un niveau supérieur, l'agent CH envoie un rapport d'intrusion sur le CH suspect à la station de base qui à son tour effectue un mécanisme de vote.

Les résultats des simulations montrent que notre schéma présente des performances supérieures de détection des attaques (telles que *hello flood, selective forwarding, black hole et wormholes)* en comparaison avec les autres schémas proposés dans la littérature [**66**][**99**]. Ceci est principalement spécifique pour le RCSFC avec un nombre optimal d'agents IDS par cluster. Dans ce cas, l'agent IDS va générer un temps de détection court avec un faible nombre de fausses alarmes. De plus, les résultats de simulations ont confirmé la légère consommation énergétique par notre modèle de détection par rapport au modèle proposé dans la référence [**79**].

Les systèmes de détection d'intrusion sont encore au stade théorique et de simulation. En effet, la plupart des travaux proposés dans la littérature se sont limités aux résultats de simulation. Par contre, le déploiement des capteurs dans un environnement hostile réel, nécessite l'intégration des mécanismes de sécurité dans ces capteurs. Dans le chapitre suivant, une nouvelle approche de détection d'intrusion est implémentée dans des capteurs MICAZ et testée dans un environnement réel.

# Chapitre 5

# Troisième Contribution :

# Modèle de Détection D’intrusion Basé Sur le Comportement Des nœuds au Sein du Même Cluster

## Résumé

L'approche de sécurité proposée applique la politique de détection basée sur le fait que tous les nœuds situés dans le même cluster doivent avoir un comportement similaire. Ce fait est démontré par des simulations lorsque le nombre de sauts dans chaque cluster ne dépasse pas deux sauts. Nous montrons les performances de notre modèle de détection par simulation sous TOSSIM et ensuite par une étude expérimentale. Nous évaluons ses performances contre plusieurs types d'attaques telles que : *selective forwarding, black hole, jamming, sinkhole, hello flood* et *resource exhaustion*. Plus précisément, nous calculons, le taux de détection, le taux de faux positifs, la consommation d'énergie et le temps nécessaire pour les agents IDS de détecter les attaques (l'efficacité moyenne). Selon les résultats de simulation et expérimentaux, notre modèle présente une grande précision de détection (taux de détection égal à 100% et taux de faux positifs proche de 0%), une faible consommation d'énergie et un temps court de détection.

## 1) Introduction

Une nouvelle approche de détection d'intrusion a été proposée récemment pour l'identification des nœuds malicieux dans les réseaux de capteurs sans fil (RCSFs), celle-ci est basée sur le fait que les nœuds qui ce trouvent dans le même voisinage ont tendance à avoir le même comportement (le même nombre de paquets transmis, reçus et rejetés, la même force du signal généré). Les auteurs dans [**81**][**100**] utilisent ce concept pour détecter un certain nombre d'attaques dans le RCSFs. Dans tous ces travaux les agents IDS surveillent leurs voisins afin de détecter les attaques internes. La surveillance consiste à collecter des données d'intrusion à partir des messages transmis dans leur portée radio, puis analyser ces paquets selon les règles sélectionnées (le nombre de paquets rejetés (*packet-dropping rate*), le nombre de paquets transmis et la force du signal reçu, etc.). Néanmoins, les inconvénients communs de ces schémas [**81**] [**100**] sont: (i) le caractère statique de l'IDS (s'exécute d'une manière permanente dans un nœud fixe), ce qui conduit à une consommation énergétique excessive par le nœud, (ii) La stratégie de l'emplacement des IDSs dans le RCSF est un aspect important et n'a pas été prise en compte dans ces travaux de recherche.

Dans ce chapitre, nous proposons un nouveau concept de détection pour identifier et prévenir différents types d'attaques dans les réseaux de capteurs. Cette approche de détection est basée sur la technique des spécifications (décrite dans le chapitre 2, sous section 2.1), mais sans la nécessité d'une mise à jour continue des règles pour maintenir la fiabilité du système de détection d'intrusion. Nous avons utilisé le concept de détection appliquée par ces deux travaux [**81**][**100**] dans une topologie à base de cluster, nous avons d'abord démontré par simulation que lorsque la taille maximale d'un cluster est de deux sauts, tous les nœuds situés au sein du même cluster ont des comportements similaires. Basés sur ce résultat, nous avons développé un nouveau modèle de détection qui repose sur ce concept afin de détecter les attaques les plus dangereuses pour les RCSFCs. L'approche de sécurité

proposée est implémentée dans des capteurs réels de type MICAZ [**101**] et elle est évaluée, en présence de plusieurs types d'attaques, à l'aide de quatre métriques: le taux de détection, le nombre de faux positifs, l'efficacité moyenne et l'énergie totale consommée.

Dans ce qui suit, nous décrivons notre politique de détection basée sur le concept de la distribution normale et les règles de détection relatives à chaque attaque. Par la suite, nous présenterons la conception du modèle de détection proposé et son principe de fonctionnement.

## 2) Détection d'intrusion dans le réseau de capteurs à base de cluster

Dans cette section, nous proposons de nouvelles politiques de détection basées sur le concept de la distribution normale afin de détecter un ensemble d'attaques et permettre un fonctionnement normal du RCSF. Dans un concept de distribution normale, la moyenne et l'écart-type (ET) des données sont calculées. Ces données sont correctement distribuées si elles se situent dans trois écarts-types autour de la moyenne comme l'illustre la Figure 5.1. Dans notre approche, nous affirmons que tous les nœuds qui sont situés dans le même cluster doivent avoir les mêmes comportements (démontré dans nos simulations). Par conséquent, un nœud est considéré comme un attaquant si son comportement diffère des membres de son cluster.

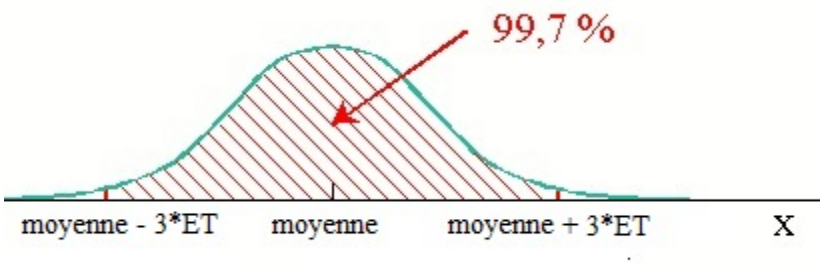



**Figure 5.1. La distribution normale**

Nous organisons cette section en deux sous-sections: Dans la première, nous présentons quelques résultats de simulation sur la distribution des comportements des nœuds d'un même cluster et la description d'une politique de détection des nœuds malveillants. Dans la seconde sous-section, nous donnons un ensemble de règles de détection relatives à chaque attaque que nous avons l'intention de détecter.

### 2.1 Distribution normale dans le RCSF à base de cluster

Dans notre travail de recherche, chaque nœud a été modélisé avec un ensemble de comportements qui sont définis comme suit:

- Nombre de paquets supprimés-*Number of Packets Dropped* (NPD)
- La force du signal reçu-*Received Signal Strength Intensity* (RSSI)
- Nombre des paquets envoyés-*Number of Packets Sent* (NPS)
- Nombre des messages retransmis- *Number of Retransmitted Message* (NRM)
- Le temps entre l'émission de deux paquets consécutifs- *Time between Transmission of two Consecutive Packets* (JITTER)

La surveillance des comportements d'un nœud $a_i$ par l'agent IDS est modélisée par la fonction suivante :

$$f(a_i) = \{f_1(a_i), f_2(a_i), \dots, f_q(a_i)\} \qquad (5.1)$$

Où q est le nombre de comportements surveillés, définis par: $f_1(a_i) = NPD$, $f_2(a_i) = RSSI$, $f_3(a_i) = NPS$, $f_4(a_i) = NRM$, $f_5(a_i) = JITTER$

Nos résultats de simulation ont révélé que lorsque la transmission des données d'un nœud au *cluster-head* subit au maximum deux sauts, tous ces comportements suivent une distribution normale au sein du cluster**.** Ainsi, comme l'illustre la Figure 5.2, toutes les valeurs liées à NPS, NPD, RSSI, JITTER et NRM se trouvent dans l'intervalle de 3 écarts-types autour de leurs valeurs moyennes. La fonction décrivant la distribution normale est la suivante :

$$F(x) = \frac{1}{ET\sqrt{2\,\pi}} e^{-\frac{1}{2}\left(\frac{x-moyenne}{ET}\right)^2} \qquad (5.2)$$

Où x== NPS, NPD, RSSI, JITTER ou NRM.

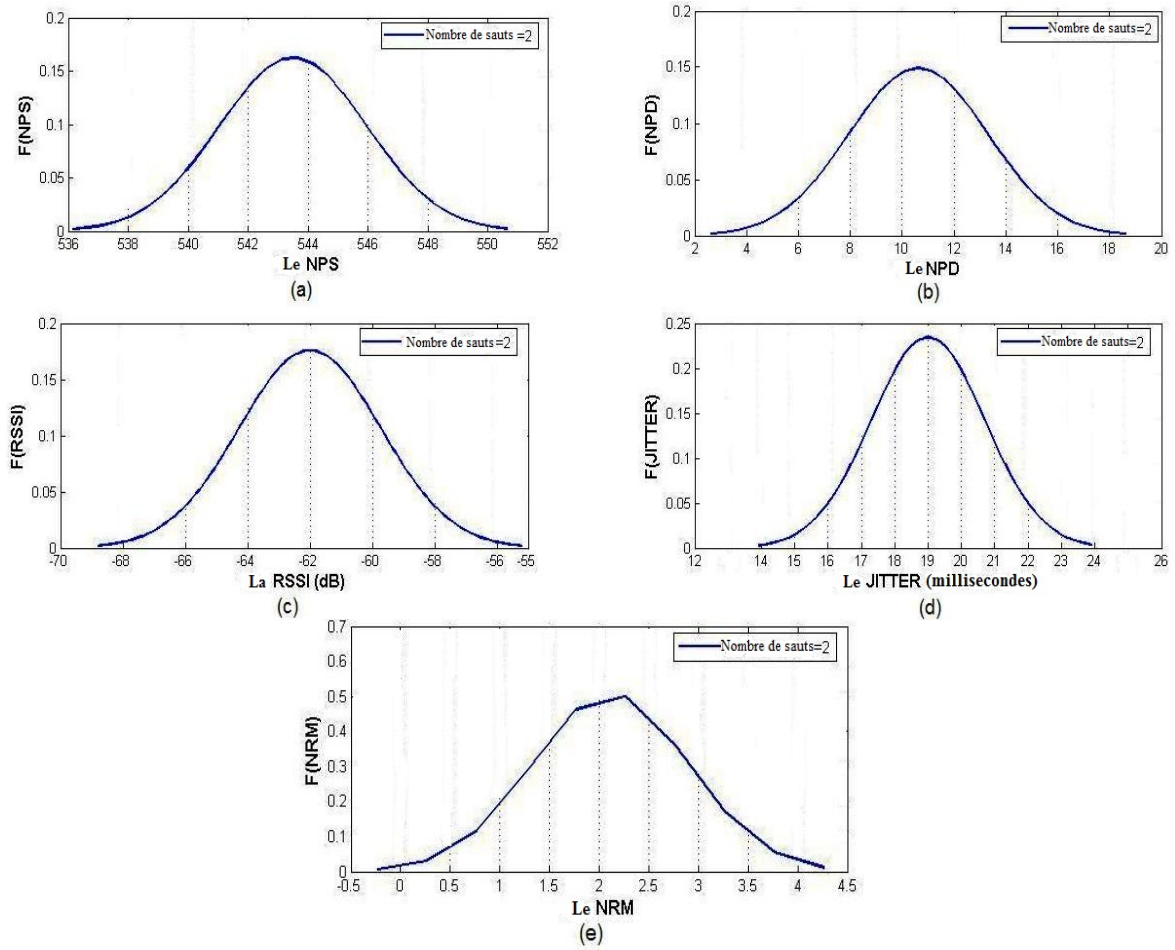



**Figure 5.2. La distribution normale des comportements d'un nœud**

Afin de déterminer si les nœuds situés à l'intérieur du même cluster ont les mêmes comportements, l'écart-type (ET) et la distance euclidienne (DE) de NPD, RSSI, NPS, NRM et JITTER ont été

calculés. Dans notre modèle de détection, chaque agent IDS calcule l'écart-type de l'ensemble $\{f_m(a_i)$ ,..., $f_m(a_n)$, $i = 1, \ldots, n\}$ (voir équation 5.4), où n est le nombre de nœuds surveillés par cet agent et m est le comportement sélectionné. Lorsque ET est supérieur à un certain seuil (σ), l'IDS conclut qu'un nœud ou plus, parmi ces nœuds surveillés, pourrait être un attaquant. Pour déterminer le nœud qui présente un comportement malveillant, l'agent IDS calcule la DE de $f_m(a_i)$ au centre de l'ensemble $\{f_m(a_1)$ ,..., $f_m(a_n)\}$ (voir équation 5.5), donné par le calcul de la moyenne arithmétique (MA) de ses éléments. Lorsque la DE est supérieure à un certain seuil (γ), le nœud $a_i$ est considéré comme un attaquant.

$$MA(f_m(a)) = \sum_{i=1}^{n} \frac{f_m(a_i)}{n} \quad (5.3)$$

$$ET(f_m(a)) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f_m(a_i) - MA(f_m(a)))^2} \quad (5.4)$$

$$DE(f_m(a_i)) = f_m(a_i) - MA(f_m(a)) \quad (5.5)$$

## 2.2 Politique de détection des attaques

Dans notre travail de recherche, nous tentons de détecter quelques attaques des plus dangereuses qui peuvent causer des dommages importants dans le réseau RCSFC, telles que *selective forwarding, black hole, jamming, resource exhaustion, sinkhole* et *hello flood*. Pour détecter ces attaques, nous appliquons des politiques de détection basée sur le concept considérant que tous les nœuds d'un cluster doivent avoir un comportement similaire.

a. *Jamming*

Dans [**102**], les auteurs proposent quatre modèles de *jamming* : (1) *constant jammer*, (2) *deceptive jammer, (3) random jammer,* et *(4) reactive jammer.* Dans notre travail de recherche, nous nous sommes concentrés uniquement aux attaques *deceptive jammer* et *random jammer.* La première attaque injecte constamment les paquets sans aucun écart entre les transmissions des paquets ultérieurs [**103**]. Au lieu de l'envoi continu des paquets, l'attaque *random jammer* alterne entre la phase de sommeil et la phase de brouillage. Pendant la phase de brouillage, il peut jouer le rôle de *deceptive jammer*. Le but de ces deux attaques est de conduire les nœuds légitimes, à gaspiller leurs ressources énergétiques. Lorsque le nœud effectue une attaque de brouillage, il envoie une quantité considérable de paquets d'où son NPS diffère de celui de ses nœuds voisins. Le JITTER est très bas ou très élevé selon les attaques respectives *deceptive jammer* ou *random jammer*. En outre, dans la référence [**103**], les auteurs affirment que la distribution de la RSSI est affectée par la présence de *deceptive jammer.* En conséquence, nous pouvons conclure que dans ce cas JITTER, NPS et RSSI suivent une distribution normale dans chaque cluster. La règle de détection de l'attaque *jamming* est illustrée dans la Figure 5.3 (a).

*b. Selective forwarding* et *Black hole*

Comme il est expliqué dans le premier chapitre, l'attaque *selective forwarding* supprime un certain nombre de paquets reçus, tandis que l'attaque *black hole* supprime tous les paquets interceptés. Lorsqu'un nœud effectue l'une de ces deux attaques, son NPD ne suit pas une distribution normale. En conséquence, nous pouvons conclure que durant l'absence de ces attaques, tous les nœuds d'un cluster ont presque la même valeur du nombre de paquets-supprimés (*paquets-dropping)*. Mais en présence de ces attaques, un nœud au moins présente un haut NPD par rapport à ses voisins; il est alors considéré comme un attaquant. La règle de détection des attaques *selective forwarding* et *black hole* est illustrée dans la Figure 5.3 (b).

*c. Sinkhole* et *Hello flood*

Ces attaques sont détectées en calculant la force du signal généré. En l'absence de ces attaques, les RSSIs des nœuds suivent une distribution normale au sein du cluster. Cependant, lors d'une attaque *sinkhole* ou *hello flood* au niveau d'un nœud, sa RSSI mesurée par l'agent IDS (son plus proche voisin) devient élevée. La règle de détection des attaques *sinkhole* et *hello flood* est illustrée dans la Figure 5.3 (c).

d. *Resource exhaustion*

Comme il est mentionné dans le premier chapitre, ce type d'attaque consiste à inonder le réseau avec un nombre considérable de paquets afin d'épuiser les ressources énergétiques des nœuds légitimes. Ce type de déni de service peut être détecté en calculant le NPS et le JITTER d'un nœud cible, qui seront respectivement haut et bas lorsque cette attaque se produit. De plus, l'attaquant peut retransmettre le même message plusieurs fois, ce qui conduit les nœuds victimes à faire un calcul supplémentaire. Par conséquent, il en résulte une consommation importante d'énergie. Comme résultat, nous pouvons conclure que le NPS, le JITTER et le NRM doivent suivre une distribution normale dans chaque cluster. La règle de détection de l'attaque *resource exhaustion* est illustrée dans la Figure 5.3 (d).


**Figure 5.3. Règles de détection des attaques:**
**(a) *Jammer,* (b) *Selective forwarding* et *Black hole,***
**(c) *Sinkhole* et *Hello flood,* (d) *Resource exhaustion***

## 3) Le modèle proposé de détection d'intrusion

Notre objectif dans ce travail de recherche est de proposer un mécanisme de détection d'intrusion fiable en termes de détection des attaques et léger en termes de processus de calcul et de communication (*low overhead*). De ce fait, notre mécanisme de détection est basé principalement sur le concept que tous les nœuds d'un même cluster, devraient avoir des comportements analogue. Ces comportements sont représentés par les attributs notés NPD, NPS, RSSI, NRM et JITTER, décrits précédemment. Dans notre travail de recherche, nous avons utilisé une topologie à base de cluster car elle permet une prolongation de la durée de vie du réseau par rapport à une topologie plate. Dans cette section, nous décrivons d'abord notre protocole de routage (à base de cluster) avant de détailler les composantes de notre modèle de détection.

## 3.1 Protocole de routage à base de cluster

Notre algorithme de clustering divise le réseau en un ensemble de cluster, dans chaque cluster il ya un *cluster-head* (CH) responsable de l'agrégation des données transmises par ses nœuds membre. L'élection du CH est principalement basée sur l'énergie résiduelle: le nœud qui a une plus grande énergie restante est choisi comme étant le CH. les clusters membres écoutent les messages émis par les CHs et rejoignent le plus proche. Comme il est indiqué précédemment, le nombre de sauts à chaque cluster ne doivent pas dépasser deux sauts afin de satisfaire le concept stipulant que dans le même cluster tous les nœuds doivent avoir les mêmes comportements. Notre algorithme de clustering utilise un autre type de nœud appelé passerelle, qui est situé entre chaque deux CH tel qu'il est illustrée dans la Figure 5.4. Nous supposons que la passerelle est un nœud de confiance et il n y'a aucune présence d'attaque durant la phase de création des clusters.

L'élection d'un nouveau CH est lancée lorsque ce nœud a soit consommé une énergie supérieure à un certain seuil *Ec,* ou présente un comportement malveillant. Dans le premier cas (consommation d'énergie du CH), le CH envoie les données recueillies à la passerelle et informe ses membres qu'un nouveau *cluster-head* sera élu. Le processus d'élection du CH est basé sur trois paramètres:

(i) L'énergie consommée: sélection des nœuds qui ont consommé une énergie inferieure à un certain seuil *Ee.*

(ii) Vote des IDSs local (LIDSs): sélection des nœuds identifiés comme étant les moins malicieux par les LIDSs (voir la sous-section 3.2.1, composant de prévention).

(iii) Proximité du nœud: parmi ces nœuds sélectionnés, le nœud qui est à proximité de l'ancien CH est désigné comme étant le nouveau CH. Lorsque le nouveau CH est désigné, la passerelle envoie les données de l'ancien CH à ce nouveau nœud.

Dans le second cas (le comportement malveillant du CH), lorsque le nœud passerelle reçoit une confirmation (i.e *message de suppression*) que le CH est un nœud malveillant par plus de deux LIDSs dans le même cluster (voir la sous-section 3.2.1, composant de prévention), il informe la station de base que ce nœud est un attaquant et il sera éjecté du réseau. L'élection du nouveau CH est basée sur l'énergie consommée, la proximité du nœud et le vote de LIDS.
Pour les deux cas, quand un nouveau CH est élu les membres du cluster seront attachés aux plus proche CH.

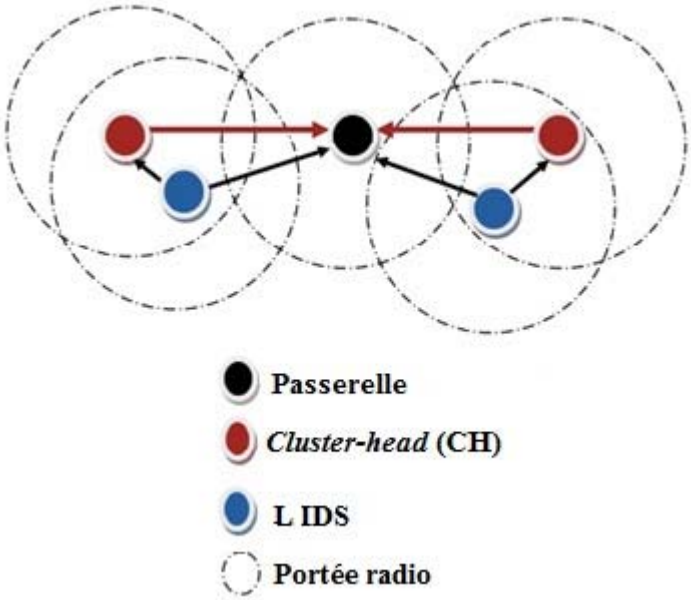



**Figure 5.4. Topologie à base de cluster**

## 3.2 Agents de détection d'intrusion

Dans notre schéma, chaque nœud a la possibilité d'activer son agent de détection d'intrusion. Cependant, l'activation simultanée de tous les nœuds n'est pas effectuée car elle conduit à un gaspillage de l'énergie des nœuds. Pour le processus d'analyse et de détection nous proposons deux agents de détection: IDS local (LIDS) et IDS global (GIDS), situés respectivement au niveau du membre du cluster (nœud) et du *cluster-head*. Le premier applique une détection basée sur le comportement des voisins pour identifier les nœuds malveillants. Le second vise à atténuer le nombre de faux positifs qui ont eu lieu lorsque l'agent LIDS soupçonne le nœud normal comme étant un attaquant.

**1. IDS local (LIDS)**

La stratégie de l'emplacement des agents LIDS dans le réseau est un point très important, puisque l'augmentation du nombre d'agents dans le réseau conduit à une surcharge (*overhead*) de communication et de calcul, et par conséquent une diminution de la durée de vie du réseau. De ce fait, la stratégie proposée doit tenir compte de la contrainte énergétique des nœuds. Notre solution utilise un agent LIDS pour surveiller chaque deux liens. Comme il est illustré dans la Figure 5.5, les deux liens (B, C) et (D, E) sont surveillés par le LIDS1 et les autres deux liens (C, CH) et (E, CH) sont surveillés par le LIDS2. Cette stratégie permet d'obtenir une vue d'ensemble sur tous les paquets qui circulent dans le réseau par un faible nombre d'agents LIDS. Par conséquent, cette stratégie conduit à détecter tous les nœuds malveillants avec une faible charge (*low overhead*). Pour une meilleure

économie d'énergie du réseau, lorsque le nœud joue le rôle d'un LIDS pour un processus de surveillance, sa fonction de routage n'est pas activée.

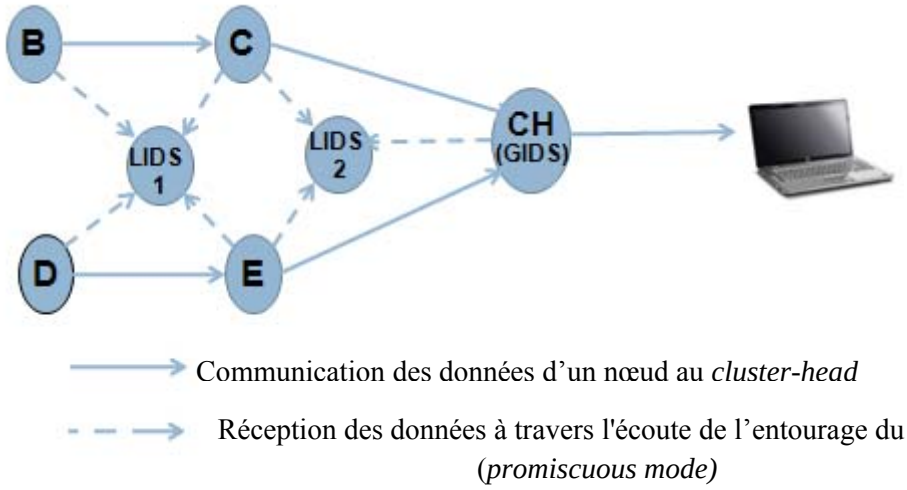



**Figure 5.5. Stratégie de l'emplacement des agents LIDS**

Le LIDS est équipé des composants suivants (voir la Figure 5.6):

- **Composant de collecte des données:** Il est responsable de la collecte des paquets dans la couverture radio du LIDS, du stockage de l'identifiant (*id)* du nœud analysé et du calcul des comportements NPD, NPS, RSSI, NRM et JITTER, se rapportant à chaque nœud. Afin d'éviter les collisions au niveau des LIDSs, le protocole CDMA / CA est utilisé au niveau de la couche MAC.
- **Composant de détection:** Il vise à appliquer la politique de détection basée sur le fait qu'à chaque cluster, les comportements NPD, NPS, RSSI, NRM et JITTER devraient suivre des distributions normales (résultat prouvé par nos simulation, voir la sous-section 2.1). L'agent LIDS surveille les nœuds situés à l'intérieur de sa portée radio en calculant l'écart-type et la distance euclidienne de leurs comportements (voir la sous-section 2.2 pour les règles de détection des attaques). Le CH est un nœud attractif et par conséquent l'intrus peut attaquer ce nœud, prendre sa place et lancer une attaque au sein de son cluster. Afin d'éviter ce problème, LIDS surveille le comportement du CH.
- **Composant de prévention:** Lorsqu'un comportement anormal se produit, le nœud LIDS déclenche une alarme sous forme d'un message à son CH, afin que celui-ci puisse confirmer le caractère malveillant du nœud soupçonné. Ce *Message d'alarme* comprend le nœud suspect (son *id*) et le type d'attaque détectée. Lorsque LIDS identifie le CH comme étant un attaquant, il diffuse un message *CH_ alarme* (contenant l'id du CH suspect et le type d'attaque détecté) au sein de son cluster. Dans ce cas, l'agent recevant un tel message déclenchera un compteur d'alarme. Lorsque ce compteur atteint un certain seuil *Tch,* l'agent envoie un *Message de*

*suppression* (contenant l'id du CH et le seuil *Tch*) au nœud passerelle afin de prendre une décision finale. On note que dans chaque cluster il y'a quelques agents LIDS situés dans la couverture radio de la passerelle. Dans le cas où, plus de deux LIDSs au sein du même cluster envoient un *message de suppression* à la passerelle, le mécanisme de l'élection du nouveau *cluster-head* sera lancé (voir la sous-section 3.1). Cette détection coopérative contribue à atténuer le nombre de faux positifs et à augmenter le taux de détection.

- **Composant de gestion:** Si l'agent LIDS a consommé plus d'un certain seuil *Eids* de son énergie ou a été identifié par GIDS comme étant un nœud malveillant (voir la sous-section 3.2.2), il sera désigné comme un nœud ordinaire et un nouveau LIDS sera élu. Lorsque l'ancien LIDS n'est pas malveillant, le nouveau LIDS récolte l'ensemble des comportements cités ci-dessus à partir de cet ancien agent. L'élection d'un nouveau LIDS est basée sur deux paramètres: (i) Stratégie de placement : sélection des nœuds se trouvant dans la même zone de couverture radio de l'ancien LIDS, et (ii) l'énergie résiduelle suffisante: parmi ces nœuds sélectionnés, élire le nœud qui présente une énergie résiduelle élevée. L'élection des nouveaux LIDSs doit assurer la condition stipulant que chaque deux liens de communication doivent être surveillés par un agent LIDS. Ce modèle dynamique d'élection de LIDS permet d'éviter l'épuisement de l'énergie des nœuds et donc prolonger la durée de vie du réseau et atténue le nombre de faux positifs lorsque cet agent est malveillant (génère de fausses alarmes).

**2. IDS global (GIDS)**

À chaque CH, il est associé un agent GIDS, qui est équipé des composants suivants (voir la Figure 5.6):

- **Composant de collecte des données :** Il reçoit un *Message d'alarme* auprès des agents LIDS au sein du cluster. Ce message contient le nœud suspect et le type d'attaque détectée.
- **Composant de décision:** Le CH stocke l'*id* du nœud suspect dans une base de données (liste noire) et augmente un compteur spécifique aux nœuds malveillants. Ce dernier est calculé comme le nombre de fois où les LIDSs au sein du même cluster identifient un nœud comme étant malveillant. Lorsque ce compteur dépasse un certain seuil *TH*, le nœud correspondant sera éjecté du cluster. Lorsque le CH identifie un nœud comme étant normal et l'agent LIDS le détecte comme étant un nœud malveillant, le CH stocke l'id de cette LIDS dans une liste noire et un compteur lié à cet agent est augmenté. Lorsque ce compteur dépasse le seuil *TH* le LIDS sera désigné comme étant un nœud ordinaire, en d'autres termes il ne peut pas jouer le rôle d'un IDS et un nouveau LIDS sera élu (voir la sous-section 3.2.1, Composant de gestion pour l'élection de LIDS). L'ancien LIDS (nœud ordinaire) sera éjecté, lorsque les autres LIDSs l'identifient comme un nœud malveillant et le CH confirme cette décision. Nous notons que, le LIDS suspect n'est pas éjecté directement du réseau en raison du fait que le CH pourrait être un attaquant et pourrait accuser à tort que le LIDS est un intrus.

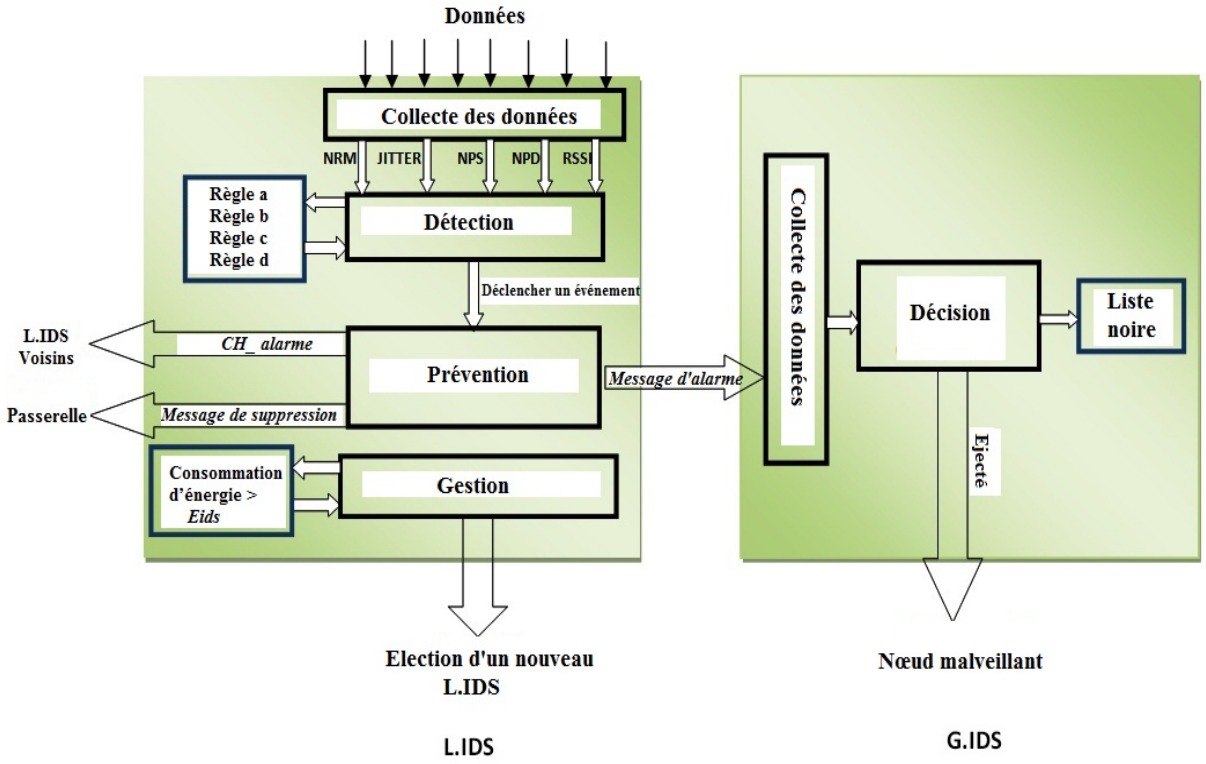



**Figure 5.6. Architecture du système de détection d'intrusion par les agents IDS**

## 3.3 Les activités de communication entre les agents IDS

Dans le RCSF, le processus de communication nécessite une grande quantité d'énergie par rapport au processus de calcul. Par conséquent, notre approche de détection vise à atténuer le coût de communication entre les agents de détection d'intrusion afin d'augmenter la durée de vie du réseau. Ce résultat est obtenu en minimisant la quantité d'informations échangées entre les LIDSs, entre le LIDS et le GIDS et entre le LIDS et la passerelle. Comme mentionné ci-dessus, le LIDS envoie trois types de messages: le premier est destiné au CH, le second à tous les LIDSs qui sont situés sur sa couverture radio et le dernier à la passerelle. Le premier et le second message contiennent l'*id* du nœud malveillant et le type d'attaque détectée et le troisième comprend l'*id* du CH et le seuil *Tch*. Comme il est expliqué dans le second chapitre, les mécanismes de coopération entre les agents IDS peuvent être classés en deux approches: (i) chaque agent IDS échange les données d'intrusion avec les autres IDSs. Cette approche génère une charge élevée de communication. (ii) Chaque agent IDS collabore avec ses voisins IDSs pour prendre une décision finale à propos du nœud suspect (intrus ou pas). Dans cette approche, l'agent IDS envoie uniquement un message d'alarme à ses voisins IDS ou CH, où la longueur de ce message est beaucoup plus petite par rapport à l'approche précédente, ce qui induit une

faible charge de communication. En conséquence, notre modèle de détection est basé sur cette approche de coopération pour détecter les nœuds malveillants avec une grande précision et une faible consommation d'énergie.

## 4) Résultats de simulation et résultats expérimentaux

Dans notre étude, nous utilisons le simulateur TOSSIM [**87**], pour évaluer les performances de notre modèle de détection en termes de détection et le taux de faux positifs. Selon ces deux métriques, nous avons déterminé les seuils optimaux pour chaque détection d'attaque (relative à l'écart-type et la distance euclidienne) pour satisfaire les exigences de notre objectif, à savoir un taux de détection élevé et une faible occurrence de faux positifs. Par la suite, nous avons intégré notre modèle de détection d'intrusion dans les capteurs MICAZ afin d'évaluer expérimentalement l'efficacité moyenne traduite par le temps nécessaire aux agents IDS pour détecter toutes les attaques survenues dans le réseau, le taux de détection (DR) et le nombre de faux positifs (FPR). De plus, nous avons évalué l'énergie totale consommée lors de l'exécution de notre modèle. Toutes ces métriques ont été définies dans le second chapitre, sous section 2.4.

Dans ce qui suit, nous présentons les résultats de simulation de notre modèle de détection. Ensuite, nous exposons dans la deuxième sous-section les résultats expérimentaux effectués sur les capteurs MICAZ.

### 4.1 Résultats de simulation

Nous avons considéré dans nos simulations un réseau de capteurs de 200 nœuds statiques déployés d'une manière aléatoire dans une zone carrée ($88 * 88m^2$). Nous avons utilisé des capteurs MICAZ équipés d'un émetteur-récepteur radio CC2420 [**104**]. Afin d'éviter les collisions, le protocole CDMA/CA est utilisé. Tous les paramètres de simulation sont résumés dans le Tableau 5.1.

| | |
|---|---|
| Temps de simulation | 680 seconds |
| Domaine de la simulation | 80×80 $m^2$ |
| Nombre de nœuds | 200 |
| Modèle radio | Lossy |
| Nombre de cluster | 8 |
| Nombre de passerelles | 4 |
| Période de détection | 24 seconds |
| Protocole de routage | À base de cluster |
| MAC | CDMA/CA |
| Portée radio | 15 m |
| L'énergie initiale | 5 Joules |
| *Tch* | arrondi (Nombre de LIDSs par cluster/3) |
| *TH* | arrondi (Nombre de LIDSs par cluster/2) |
| *Ec* | Consomme 50% de l'énergie initiale |
| *Ee* | Consomme 40% de l'énergie initiale |
| *Eids* | Consomme 60% de l'énergie initiale |

**Tableau 5.1. Paramètres de simulation**

Dans la phase de simulation, nous avons étudié la variation des seuils relatifs à l'écart-type et à la distance euclidienne et son impact sur la détection des attaques et le taux de faux positifs. Selon les résultats de simulation, les valeurs optimales des comportements qui satisfont les exigences de notre objectif (un taux de détection élevé et un nombre de faux positifs faible) sont sélectionnées. Les seuils optimaux de NPS, NPD, RSSI, JITTER et NRM relatifs à l'écart-type et la distance euclidienne sont définis dans le Tableau 5.2. Une anomalie se produit lorsque l'écart-type et la distance euclidienne sont supérieurs aux seuils optimaux correspondants.

**1. Seuils de l'écart-type**

Afin de déterminer les seuils optimaux de l'écart-type relatif à chaque comportement, le taux de détection et le taux de faux positifs sont calculés. Comme l'illustre la Figure 5.7, lorsque l'écart-type augmente ces deux métriques diminuent. Par conséquent, une solution optimale correspond à un taux de détection élevée et un faible taux de faux positifs. Les seuils optimaux ( $\sigma_{nps}$, $\sigma_{npd}$, $\sigma_{rssi}$, $\sigma_{jitter}$ et $\sigma_{nrm}$ ) relatifs respectivement à NPS, NPD, RSSI, JITTER et NRM sont présentés dans le tableau 5.3. Par conséquent, une anomalie peut se produire lorsque l'écart-type est supérieur au seuil optimal.

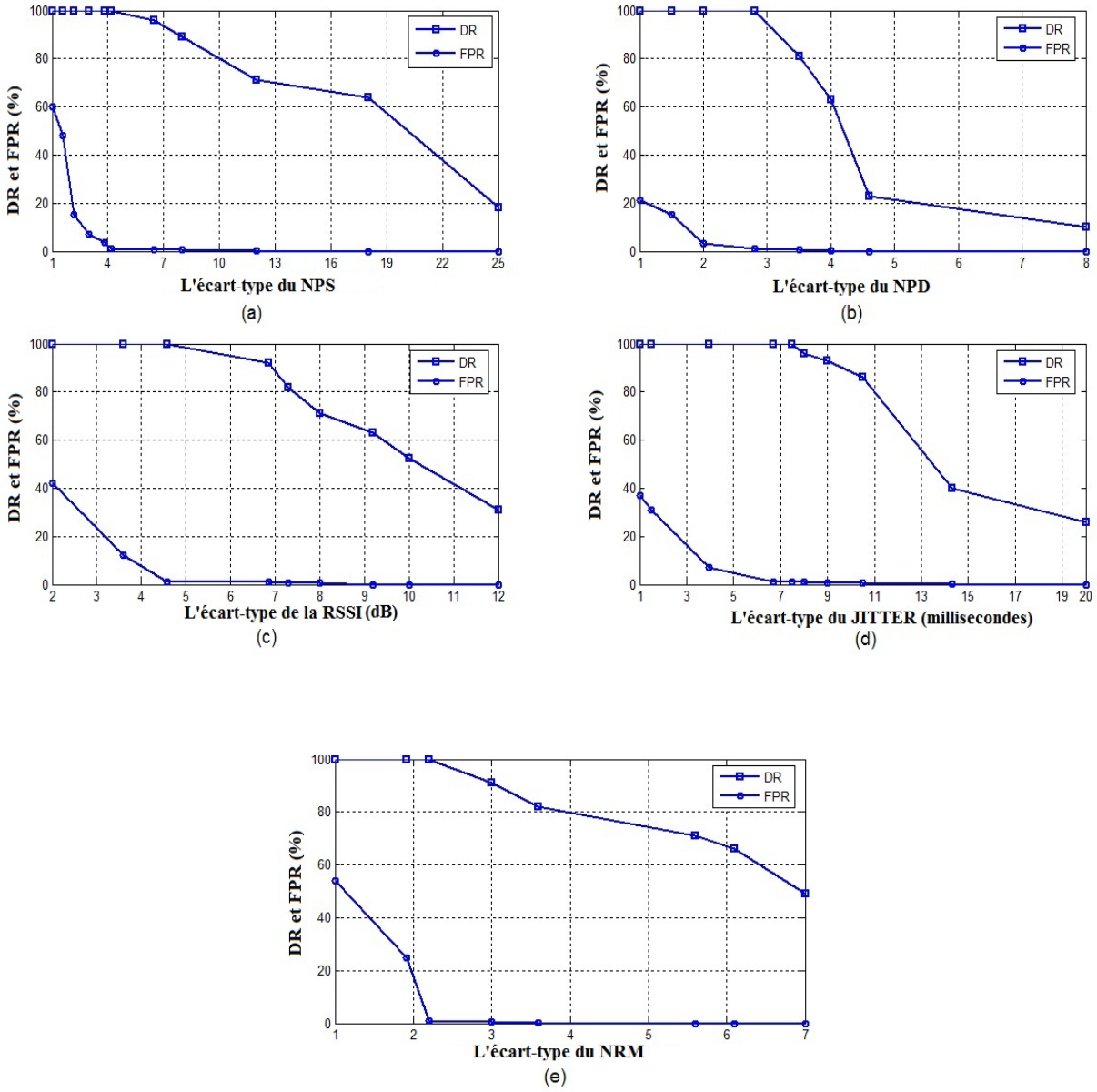



**Figure 5.7. Sélection des seuils optimaux de l'écart-type pour:**
**(a) NPS, (b) NPD, (c) RSSI, (d) JITTER, (e) NRM**

Dans ce qui suit, nous étudions la variation des seuils relatifs à la distance euclidienne pour chaque attaque. Nous examinons son impact sur la détection et le taux de faux positifs en utilisant les seuils optimaux de l'écart-type trouvé.

**2. Seuils de la distance euclidienne**

**a. Détection des attaques *Selective forwarding* et *Black hole***

Comme il est expliqué dans la section 2.2, les attaques *selective forwarding* ou *black hole* se produisent au niveau d'un nœud quand leurs NPD sont plus élevés que ceux des autres nœuds du même cluster. Lorsque l'agent LIDS détermine au sein de sa couverture radio que l'écart type du NPD des nœuds est plus grand que le seuil optimal ( $\sigma_{npd}$), la distance euclidienne relative à ce comportement est calculée pour détecter les nœuds qui exécutent les attaques *selective forwarding* et/ou *black hole*. Selon la Figure 5.8 (a), lorsque $\gamma_{sf}$ est fixée à 9,2 la détection de l'attaque *selective forwarding* est égale à 100% avec un nombre réduit de faux positifs rapporté à 1%. Pour l'attaque *black hole,* lorsque $\gamma_{bh}$ est fixée à 52,4 le taux de détection et le taux de faux positifs sont égaux respectivement à 100% et 0,8% (voir Figure 5.8 (a)). En conséquence, l'application de ces seuils ( $\sigma_{npd}$,$\gamma_{sf}$ et $\gamma_{bh}$), conduit à une détection efficace de ces deux attaques.

**b. Détection des attaques *Sinkhole* et *Hello flood***

Comme il est mentionné dans la sous-section 2.2, les attaques *sinkhole* et *hello flood* génère un RSSI élevé par rapport aux nœuds normaux. Par conséquent, lorsque l'agent LIDS constate au sein de sa couverture radio que le RSSI ne suit pas une distribution normale, il calcule la distance euclidienne pour identifier les nœuds qui exécutent l'attaque *sinkhole* et/ou *hello flood.* Comme il est illustré dans la Figure 5.8 (b), lorsque $\gamma_{sh}$ et $\gamma_{hf}$ augment, le taux de détection et le nombre de faux positifs diminuent. Par conséquent, un compromis entre ces deux métriques doit être considéré afin de rendre notre modèle efficace contre ces attaques. Lorsque $\gamma_{sh}$= 9 dBm et $\gamma_{hf}$ = 7,25 dBm, notre modèle donne un taux de détection élevé (100%) et génère un faible nombre de faux positifs (moins de 1%). En conclusion, notre modèle est capable de détecter *sinkhole* et *hello flood* avec une faible occurrence de faux positifs.

**c. Détection des attaques *Random jammer* et *Deceptive jammer***

Notre modèle de détection vise à détecter deux types d'attaques *jamming*: *random jammer* et *deceptive jammer.* Les caractéristiques de ces deux attaques sont expliquées dans la sous-section 2.2. La première attaque se produit lorsque le NPS et le JITTER ne suivent pas une distribution normale. Dans ce cas, l'agent LIDS vérifie si les distances euclidiennes du NPS et du JITTER sont supérieures à certains seuils. Comme il est illustré dans les Figures 5.8 (c) et 5.8 (d), les seuils optimaux relatifs à ces comportements ( $\gamma_{rj}'$,$\gamma_{rj}$), qui satisfont nos objectifs (i.e taux de

détection élevé et faible occurrence de faux positifs), sont égaux respectivement à 13,4 et 13,5 millisecondes. Dans la seconde attaque, le RSSI, NPS et JITTER ne suivent pas une distribution normale et la distance euclidienne pour chacun de ces comportements est supérieure à un certain seuil. Selon les Figures 5.8 (b), 5.8 (c) et 5.8 (d), les seuils optimaux du NPS, JITTER et RSSI ($\gamma_{dj}{}'', \gamma'_{dj}, \gamma_{dj}$) qui permettent un taux de détection élevé avec un nombre de faux positifs proche de 0 sont égaux respectivement à 28; 6,8 millisecondes et 5 dBm. En conclusion, l'utilisation de ces seuils permet à notre modèle une meilleure précision de détection contre ces deux types d'attaques avec un taux de détection égal à 100% et un taux de faux positifs inférieur à 1%.

**d. Détection de l'attaque *Resource exhaustion***

Comme il est mentionné dans la sous-section 2.2, pour détecter l'attaque *resource exhaustion* les comportements NPS, JITTER et NRM sont calculés. Lorsqu'une telle attaque se produit, chacun de ces comportements ne suit pas une distribution normale, ce qui conduit l'agent LIDS à déterminer si les nœuds surveillés sont malveillants ou pas, en calculant la distance euclidienne de chaque comportement. Comme il est illustré dans les Figures 5.8 (c), 5.8 (d) et 5.8 (e), lorsque les distances euclidiennes de NPS, JITTER et NRM sont supérieures respectivement à 35; 5,2 millisecondes et 2,6, notre modèle présente un taux de détection élevé (100%) et un faible taux de faux positifs (moins de 1%). Comme résultat, les attaques de type *resource exhaustion* sont détectées avec une grande précision lorsque les seuils optimaux de NPS, JITTER et NRM valent respectivement: $\gamma_{re}{}''$ =35, $\gamma_{re}{}'$ = 5,2 millisecondes et $\gamma_{re} = 2{,}6$.

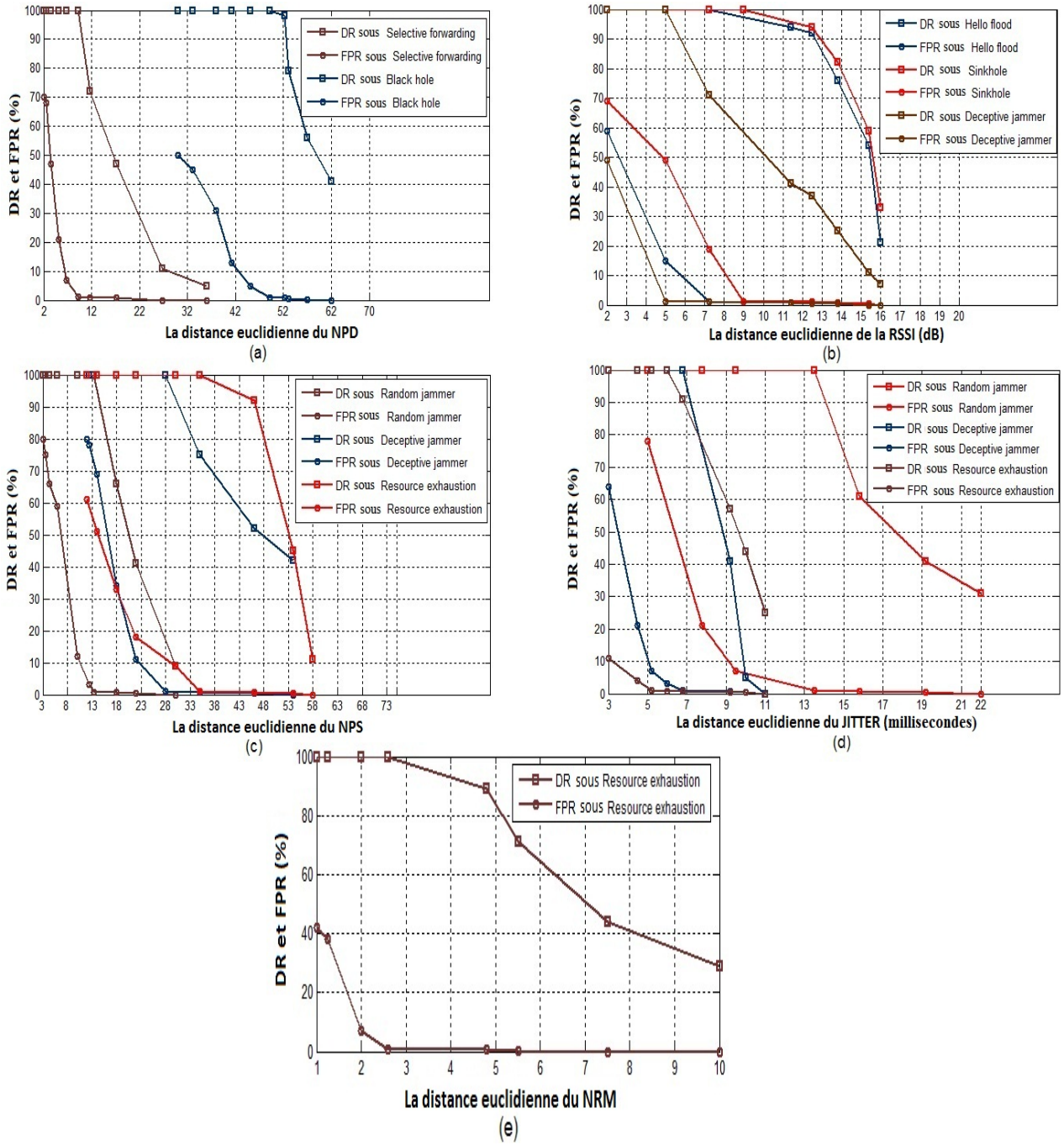



**Figure 5.8. Sélection des seuils optimaux de la distance euclidienne pour:**

**(a) NPD, (b) RSSI, (c) NPD, (d) JITTER, (e) NRM**

Le Tableau 5.2 résume l'ensemble des résultats des seuils optimaux des paramètres NPS, NPD, RSSI, JITTER et NRM relatifs à l'écart-type et à la distance euclidienne.

| | |
|---|---|
| $\sigma_{nps}$ : Seuil de l'écart-type du NPS | 2.8 |
| $\sigma_{npd}$ : Seuil de l'écart-type du NPD | 4.2 |
| $\sigma_{rssi}$ : Seuil de l'écart-type de la RSSI | 4.57 dBm |
| $\sigma_{jitter}$ : Seuil de l'écart-type du JITTER | 7.5 millisecondes |
| $\sigma_{nrm}$ : Seuil de l'écart-type du NRM | 2.2 |
| $\gamma_{sf}$ : Seuil de la distance euclidienne du PDR sous *Selective forwarding* | 9.2 |
| $\gamma_{bh}$ : Seuil de la distance euclidien du PDR sous *Black hole* | 52.4 |
| $\gamma_{sh}$ : Seuil de la distance euclidien de la RSSI sous *Sinkhole* | 9 dBm |
| $\gamma_{hf}$ : Seuil de la distance euclidien de la RSSI sous *Hello flood* | 7.25 dBm |
| $\gamma_{rj}'$ : Seuil de la distance euclidien du NPS sous *Random jammer* | 13.4 |
| $\gamma_{rj}$ : Seuil de la distance euclidien du JITTER sous *Random jammer* | 13.5 millisecondes |
| $\gamma_{dj}''$ : Seuil de la distance euclidien du NPS sous *Deceptive jammer* | 28 |
| $\gamma_{dj}'$ : Seuil de la distance euclidien du JITTER sous *Deceptive* jammer | 6.8 millisecondes |
| $\gamma_{dj}$ : Seuil de la distance euclidien de la RSSI sous *Deceptive jammer* | 5 dBm |
| $\gamma_{re}''$ : Seuil de la distance euclidien du NPS sous *Resource exhaustion* | 35 |
| $\gamma_{re}'$ : Seuil de la distance euclidien du JITTER sous *Resource exhaustion* | 5.2 millisecondes |
| $\gamma_{re}$ : Seuil de la distance euclidien du NRM sous *Resource exhaustion* | 2.6 |

**Table 5.2. Les seuils optimaux**

## 4.2 Résultats expérimentaux

Comme il a été souligné précédemment, nous avons également intégré notre modèle de détection dans des capteurs MICAZ et évalué ses performances en termes du taux de détection, du taux de faux positifs, d'efficacité moyenne et de consommation d'énergie. Nous notons que, nous avons utilisé les seuils optimaux, trouvés par simulation (Tableau 5.2), de l'écart-type et de la distance euclidienne dans notre modèle de détection.

Nous avons déployé dans un champ aléatoire 12 nœuds MICAZ, où le nombre de passerelle, de *cluster-heads*, de membres du cluster et d'agents LIDS sont égaux respectivement à 1, 2, 6 et 3. Lorsque les clusters ont été formés, nous avons injecté séparément les différentes attaques: *Selective forwarding, Black hole, Jamming, Sinkhole, Hello flood* et *Resource* exhaustion. Dans cette partie, nous exposons l'effet de chaque attaque sur le réseau en faisant varier le nombre des nœuds malveillants de 1 à 3. La Figure 5.9 illustre le processus d'élection du CH et la détection d'intrusion.

(a) (b) (c)

**Figure 5.9. Élection du CH et détection d'intrusion. (a) Messages envoyés par le membre du cluster (rouge-clignotant), (b) Élection du CH (jaune-clignotant), (c) Intrus détecté par l'agent LIDS (vert-clignotant)**

**1. *Sinkhole et Hello flood***

Nous remarquons dans la Figure 5.10 que le taux de détection et le taux de faux positifs restent constants, même lorsque le nombre de *sinkhole* ou *hello flood* augmente. En outre, comme le montre la Figure 5.11 (a), les agents LIDS nécessitent moins de temps pour détecter ces attaques. Le temps nécessaire pour les agents IDS afin de détecter toutes les attaques *sinkhole* est proche de 2 secondes. Pour *hello flood*, il est égal à 2 secondes. En conséquence, notre modèle de détection a la capacité de détecter ces deux attaques dans un temps court et avec une grande précision.

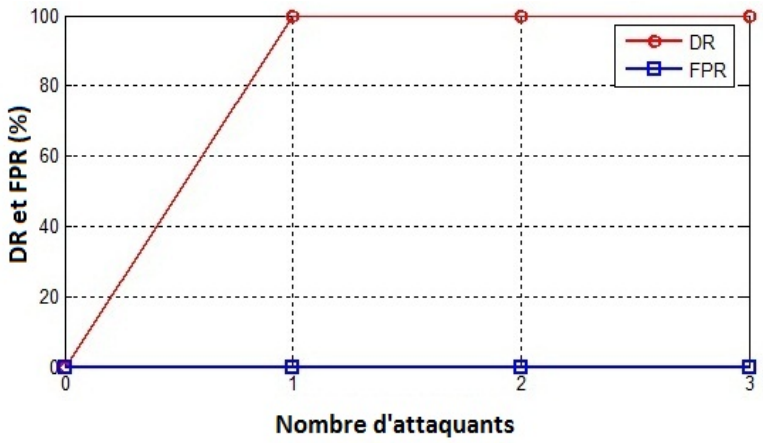
***2. Selective forwarding* et *Black hole***

Lorsque les attaques *selective forwarding* et *black hole* se produisent, l'efficacité moyenne de notre modèle de détection pour chacune d'elle est proche de 3 secondes pour la première et égale à 2 secondes pour la deuxième, comme le montre la Figure 5.11 (b). La détection de ces attaques atteint une grande précision (i.e. Taux de détection = 100% et Taux de faux positifs = 0%) tel qu'il est illustré dans la Figure 5.10. Selon ces résultats, nous avons constaté que notre modèle de détection est très fiable, même lorsque le nombre de ces attaques augmente.

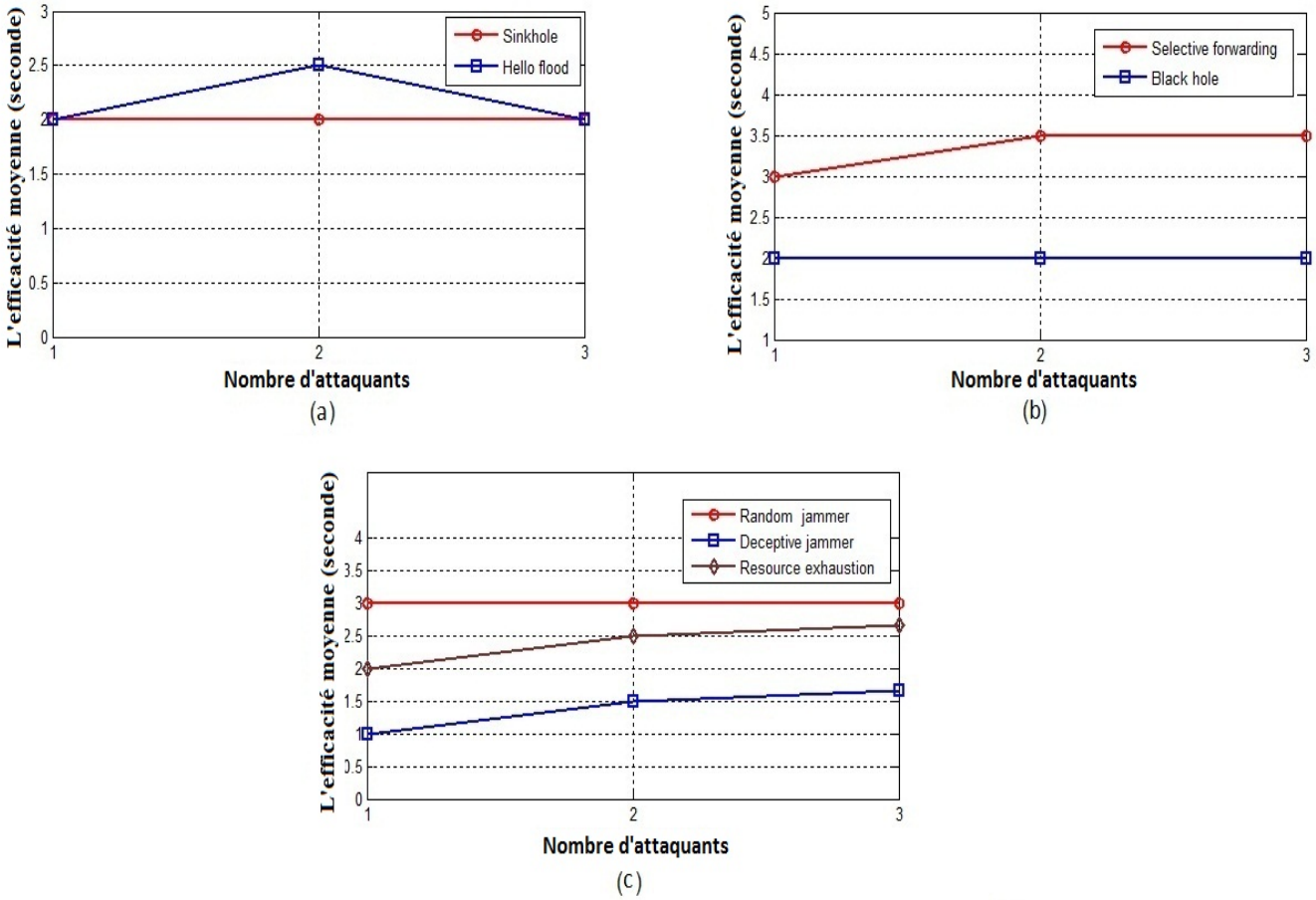
***3. Random jammers, Deceptive jammers* et *Resource exhaustion***

Comme il est mentionné ci-dessus, notre modèle de détection permet de détecter deux types d'attaques *jammer*. Comme la montre la Figure 5.11 (c), les agents IDS nécessitent moins de temps pour détecter l'attaque *deceptive jammer* comparée à *random jammer.* L'efficacité moyenne de détection sous les attaques *deceptive jammer* et *random jammer* est respectivement proche de 1seconde et égale à 3 secondes. En conséquence, le modèle proposé est capable de détecter ces deux types d'attaques *jammer* avec moins de temps et une grande précision (voir Figures 5.10 et 5.11 (c)).

Pour détecter les attaques de type *resource exhaustion* avec une grande précision, notre modèle de détection nécessite un temps égal à 2 secondes (voir la Figure 5.11 (c)). Comme il est illustré dans la Figure 5.10, lorsque le nombre d'attaque *resource exhaustion* augmente, les agents IDS peuvent les détecter avec un nombre de faux positifs égal à 0%. Par conséquent, nous affirmons que notre modèle de détection a la capacité de détecter ce type d'attaque avec une grande précision et un temps de détection court.




**Figures 5.10. Performances expérimentales de détection d'intrusion: taux de détection et taux de faux positifs pour chaque attaque**

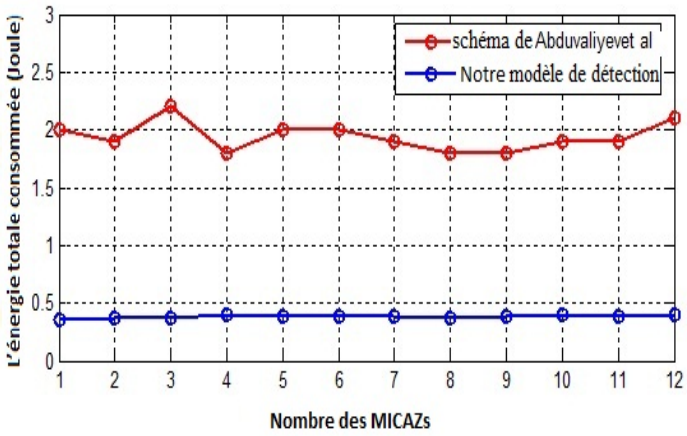



**Figures 5.11. Performances expérimentales de détection d'intrusion : (a) Efficacité moyenne sous les attaques *Sinkhole* et *Hello flood*, (b) Efficacité moyenne sous les attaques *Selective forwarding* et *Black hole*, (c) Efficacité moyenne sous les attaques *Random jammer*, *Deceptive jammer* et *Resource exhaustion***

## 4. L'énergie totale consommée

L'amélioration de la durée de vie du réseau est un facteur important dans les RCSFs. Dans cette section, nous étudions la consommation d'énergie causée pat les processus de communication et de calcul. Afin de mesurer l'énergie utilisée par chaque nœud, nous avons exploité l'approche utilisant une résistance en parallèle avec le MICAZ. Dans cette approche, nous mesurons deux tensions: la première est aux bornes du MICAZ, qui est constante et égale à 2,7 V; la seconde est aux bornes de la résistance, qui est variable dans le temps. Nous notons qu'ici nous calculons la consommation d'énergie relative à l'apparition de chaque attaque séparément, par la suite nous calculons l'énergie moyenne.

Comme le montre la Figure 5.12, lorsque le nombre de nœuds augmente, l'énergie consommée par tous ces capteurs reste constante et égale en moyenne à 0,4 joules. Nous comparons l'efficacité énergétique de notre modèle de détection avec celle proposée par A. Abduvaliyev et al. [**79**]. D'après la Figure 5.12, Il est clair que notre modèle de détection présente une faible consommation d'énergie. Cette amélioration est obtenue grâce au fait que les agents IDSs génèrent une charge faible de communication et de calcul (*low communication and computation overhead*). En conséquence, nous pouvons affirmer que notre modèle de détection améliore la durée de vie du réseau.




**Figures 5.12. L'énergie totale consommée**

## 5) Conclusion

La sécurité est devenue un aspect important à relever dans l'exploitation des réseaux de capteurs sans fil. Cela est particulièrement vrai pour les environnements hostiles et insécurisés. Dans ce chapitre, nous avons proposé une nouvelle approche de détection basée sur la notion comportementale des nœuds voisins. Celle-ci est basée sur le fait que les nœuds qui se situent dans le même cluster doivent avoir le même comportement. En outre, nous avons étendu notre recherche et appliqué ce concept pour la détection de quelques attaques, qui peuvent causer des dommages importants dans les RCSFs tels que: *Jamming, Selective forwarding, Black hole, Sinkhole, Hello flood* et *Resource exhaustion.* Le processus de détection d'intrusion s'exécute dans les membres du cluster, le *cluster-head* et la passerelle pour éliminer toute menace de sécurité qui tente de perturber le réseau.

Selon les résultats de simulation et expérimentaux obtenus, lorsque les seuils optimaux relatifs à l'écart-type et à la distance euclidienne sont sélectionnés, notre modèle de détection nécessite un temps de détection court avec un taux de faux positifs très faible et un taux de détection presque égal à 100%. Ces résultats sont obtenus avec une faible consommation d'énergie.

Malgré les progrès de recherche pour sécuriser les réseaux de capteurs, la plupart des solutions de détection d'intrusion proposées dans la littérature se limitent seulement à des niveaux théoriques ou de simulation. Dans ce travail de recherche, nous avons intégré notre modèle de détection d'intrusion dans les capteurs réel MICAZ, afin d'évaluer de façon pratique le niveau de détection par notre modèle, des attaques décrites précédemment.

# Conclusion générale et perspectives

L'utilisation des réseaux de capteurs sans fil (RCSF) dans des applications critiques nécessite un certain degré de sécurité afin de les protéger contre des menaces qui profitent de la vulnérabilité des nœuds pour attaquer ces réseaux. La sécurité des RCSFs présente des défis liés aux contraintes énergétiques des nœuds et leurs capacités physiques. De ce fait les chercheurs travaillent sur cette problématique et proposent des protocoles de sécurité adaptés aux nœuds de capteurs.

Dans cette thèse trois modèles de détection d'intrusion pour les réseaux de capteurs sans fil à base de cluster (RCSFC) sont proposés:

Le premier modèle consiste à combiner les avantages de la technique de détection basée sur les signatures d'attaques et de la détection d'anomalie à base de la machine à vecteur de support. Les résultats de nos simulations, exploitant un certain nombre d'attaques définies dans la base de données KDDcups' 99, ont démontré une nette précision de détection, traduite par un taux de détection proche de 100% avec une faible occurrence des faux positifs. Cependant, dans cette approche nous n'avons pas évalué l'énergie totale consommée par les nœuds dans le réseau. De plus deux points importants n'ont pas été pris en considération: (i) nous avons supposé que l'agent IDS ne peut pas être un nœud malicieux. Mais en réalité ce type d'agent peut être attaqué et l'attaquant peut prendre la place de cet agent. (ii) En raison des données pertinentes que le *cluster-head* peut avoir, il est une cible très attractive pour les attaquants. Dans ce modèle, la protection de ce type de nœud n'est pas prise en compte. Par conséquent, nous avons proposé, développé et implémenté d'autres modèles de sécurité pour le RCSFC afin de contrer ces problèmes.

Dans le second modèle de détection un certain nombre de protocoles de détection sont intégrés de façon hiérarchique dans le réseau ( i.e dans les membres du cluster, le *cluster-head* et la station de base) pour la détection de quelques attaques qui ciblent la couche réseau tels que: *hello flood, selective forwarding, black hole,* et *wormholes.* Les résultats de nos simulations ont montré que lorsque ces attaques se produisent dans le réseau, notre modèle nécessite un temps de détection court et présente un taux de détection presque égal à 100% avec un nombre de faux positifs proche de 0%. Ce résultat est obtenu lorsqu'un nombre optimal d'agent IDSs dans chaque cluster est déterminé. Avec ce nombre optimal d'IDS dans chaque cluster, notre schéma présente des performances supérieures en termes de détection à celles proposées par d'autres auteurs dans la littérature [**66**] et [**99**]. La consommation d'énergies dans le RCSF est un facteur très important, pour cela l'approche de détection proposée doit prendre en compte les contraintes énergétiques des capteurs. D'après le résultat de notre étude, notre schéma nécessite une énergie réduite par rapport au modèle de sécurité dans [**79**] pour la détection de ce type d'attaques.

Finalement dans notre troisième contribution, nous avons élaboré une étude basée sur des simulations et de l'expérimentation pour l'évaluation des performances d'une nouvelle approche de détection basée sur le fait que les nœuds qui se trouvent dans le même cluster doivent avoir un comportement similaire. Dans nos résultats de nos simulations, nous démontrons que lorsque le nombre de sauts dans chaque cluster et inférieur ou égal à deux, tous les comportements (i.e nombre de paquets envoyés et supprimés, la force de signal reçu, le nombre de messages retransmis et le temps entre l'émission de deux paquets consécutifs) suivent une distribution normale au sein du même cluster. En se basant sur ce résultat fort intéressant, nous proposons des règles de détection pour détecter et prévenir l'exécution des attaques les plus dangereuses tels que *selective forwarding, Black hole, Jamming, Sinkhole, hello flood* et *Resource exhausion,* qui ciblent les couches réseau et physique. D'après les résultats de nos simulations et expérimentations, notre modèle a la capacité de détecter ces attaques dans un temps court avec un nombre de faux positifs proche de 0%. De plus, dans nos résultats expérimentaux, nous avons obtenu une consommation énergétique nettement inférieure à celle du modèle de détection proposé par les auteurs dans la référence [**79**].

Les travaux présentés tous au long de cette thèse traitent les problèmes de sécurité contre des attaques externes et internes en proposant et en concevant des mécanismes de sécurité adaptés à ce type de réseau. Par ailleurs, plusieurs perspectives peuvent être envisagées pour améliorer davantage ces travaux.

Tout d'abord, il est intéressant d'élargir le champ de détection et d'essayer de détecter d'autres types d'attaques. De ce fait diverses techniques de détection qui répondent aux exigences des capteurs (i.e les contraintes énergétiques) doivent être proposées et implémentées dans les réseaux de capteurs.

Très peux de chercheurs travaillent sur la détection d'intrusion dans un réseau de capteurs à grande échelle, car il est difficile d'implémenter des systèmes de détection d'intrusion dans ce type de réseau en raison du délai important qui peut être engendré. De ce fait, il est intéressant d'implémenter nos modèles de détection dans un réseau à grande échelle et étudier le délai généré et le temps requis pour les agents IDS de détecter toutes les attaques survenues dans le réseau (i.e l'efficacité moyenne).

Un nombre limité des travaux propose des solutions de sécurité pour les nœuds mobiles dans le RCSF. Il serait important de faire d'autres simulations en prenant en compte le caractère mobile des nœuds et observer les performances de notre schéma de détection dans ce contexte.

Finalement, nous suggérons l'implémentation de nos trois modèles de détection d'intrusion dans un réseau de capteurs réels à grande échelle, en prenant en considération le caractère mobile des nœuds. L'évaluation des performances de ces modèles est déterminée en calculant les quatre principales métriques: le taux de détection, le taux de faux positifs, l'énergie totale consommée et l'efficacité moyenne.

## Annexe A : La base de données KDDcups' 99

Dans cette annexe, nous allons décrire les 41 attributs proposés par le laboratoire Lincoln du MIT en 1998 (voir Table A .1). Ces attributs sont les données d'entrée utilisées dans les systèmes d'apprentissages pour la détection des attaques : Dos, Probe, U2r, R2l (Décrit dans le chapitre 3, sous section 4.1).

| Nombre | Nom d'attributs | Description |
|---|---|---|
| 1 | *Duration* | Longueur (nombre de secondes) de la connexion |
| 2 | *Protocol type* | Type de protocole, ex. tcp, udp, etc. |
| 3 | *Service* | Service réseau au niveau de la destination, ex. http, telnet, etc. |
| 4 | *Flag* | Etat de la connexion. Les états possibles sont : SF, S0, S1, S2, S3, OTH, REJ, RSTO, RSTOS0, SH, RSTRH, SHR |
| 5 | *Src_bytes* | Nombre d'octets envoyés à partir de source à la destination |
| 6 | *Dst_bytes* | Nombre d'octets envoyés de destination vers la source |
| 7 | *Land* | 1 si la connexion est à partir de/ vers le même hôte / port ; 0 autrement |
| 8 | *Wrong_fragment* | Nombre de fragments erronés |
| 9 | *Urgent* | Nombre de paquets urgents |
| 10 | *Hot* | Nombre de *hot* indicateurs |
| 11 | *Num_failed_logins* | Nombre raté de tentatives de connexion |
| 12 | *Logged in* | 1 si l'utilisateur se connecte avec succès ; 0 autrement |
| 13 | *Num_compromised* | Nombre des conditions compromises |
| 14 | *Root_shell* | 1 si la racine *shell* est obtenue ; 0 autrement |
| 15 | *Su_attempted* | 1 Si la commande « *su root* » est lancée par l'utilisateur ; 0 autrement |
| 16 | *Num_root* | Nombre d'accès à la racine |
| 17 | *Num_file_creations* | Nombre d'opérations de création des fichiers |
| 18 | *Num_shells* | Nombre de *shells* sollicités |
| 19 | *Num_access_files* | Nombre d'opérations sur les fichiers de contrôle d'accès |
| 20 | *Num_outbound_cmds* | Nombre de commandes sortantes (*outbound commands*) dans une session ftp |
| 21 | *Is_host_login* | 1 si la connexion appartient à la liste *hot* ; 0 autrement |

| | | |
|---|---|---|
| 22 | *Is_guest_login* | Si la connexion est en mode invité de connexion (*guest login*) ; 0 autrement |
| 23 | *Count* | Nombre de connexions au même hôte de destination. |
| 24 | *Srv_count* | Nombre de connexions au même service dans les dernières deux secondes |
| 25 | *Serror_rate* | Nombre de connexions qui ont des erreurs de " SYN " |
| 26 | *Srv_serror_rate* | Nombre de connexions qui ont des erreurs de " SYN " |
| 27 | *Rerror_rate* | Nombre de connexions qui ont des erreurs de "REJ" |
| 28 | *Srv_rerror_rate* | Nombre de connexions qui ont des erreurs de "REJ" |
| 29 | *Same_srv_rate* | Nombre de connexions au même service |
| 30 | *Diff_srv_rate* | Nombre de connexions au différent service |
| 31 | *Srv_diff_host_rate* | Nombre de connexions au différent hôte |
| 32 | *Dst_host_count* | nombre de connexions à partir du même hôte jusqu' au nœud destinateur pendant un laps de temps spécifié |
| 33 | *Dst_host_srv_count* | Nombre de connexions ayant le même service et la même destination |
| 34 | *Dst_host_same_srv_rate* | Pourcentage des connexions vers le même destinataire et faisant appel aux même services |
| 35 | *Dst_host_diff_srv_rate* | Pourcentage des connexions vers le même destinataire et faisant appel à des services différents |
| 36 | *Dst_host_same_src_port_rate* | Pourcentage des connexions vers la même destination en utilisant le même port |
| 37 | *Dst_host_srv_diff_host_rate* | Pourcentage des connexions vers des destinations différentes en utilisant le même port |
| 38 | *Dst_host_serror_rate* | Pourcentage des connexions vers la même destination avec l'activation du drapeau : SYN |
| 39 | *Dst_host_srv_serror_rate* | Pourcentage des connexions en utilisant le même port avec l'activation du drapeau : SYN |
| | | |

| | | |
|---|---|---|
| 40 | *Dst_host_rerror_rate* | Pourcentage des connexions vers la même destination avec l'activation du drapeau : REJ |
| 41 | *Dst_host_srv_rerror_rate* | Pourcentage des connexions en utilisant le même port avec l'activation du drapeau : REJ |

**Table A.1. Les différents attributs de la base de données KDDcup'99**

# Annexe B : Les outils logiciels et Matériels utilisés par nos modèles de détection d'intrusion

Les capteurs dotés du système d'exploitation TinyOS sont largement utilisés par la communauté scientifique et industrielle. De ce fait, nous avons évalué les performances de nos deux modèles de détection (décrits dans les chapitres 4 et 5) par les simulateurs propres à ce type de capteurs, tels que TOSSIM et POWERTOSSIM, et l'implémentation du notre troisième modèle de détection (décrit dans le chapitre 5) dans les capteurs MICAZ. Dans cette annexe, nous décrivons les logiciels et matériels utilisés dans les deux laboratoires de recherche: laboratoire STIC (université de Tlemcen) et le laboratoire de recherche Drive (Université de Bourgogne).

## 1) Le système d'exploitation TINYOS

TINYOS est un système d'exploitation développé par l'université de BERKELEY. Ce système est basé sur un fonctionnement évènementiel; c'est-à-dire, il devient actif lorsque un événement se produit (réception d'un message), dans le cas contraire les nœuds de capteurs sans en état de veille. Ce processus permet de mieux économiser les ressources énergétiques des capteurs. Le langage de programmation pour la conception de système est le NesC [**96**], celui-ci s'approche du langage C. TINYOS a une bibliothèque de composants, celle-ci est constituée d'un ensemble de protocoles et programmes écrits en langage NesC, des pilotes de capteurs et des outils d'acquisition de données.
Une application s'exécutant sur TINYOS est constituée d'une sélection de composants systèmes et de composants développés spécifiquement pour l'application [**18**]. L'ordonnanceur TINYOS est considéré parmi les principaux composants, celui-ci est composé de : (i) Deux niveaux de priorités (bas pour les tâches, haut pour les évènements). Les tâches réalisent un nombre considérable de traitements. Par ailleurs TINYOS ne gère pas l'interruption entre les tâches mais donne la priorité aux interruptions matérielles. Les événements consistent à réaliser des processus urgents et courts. (ii) Une file d'attente FIFO.

## 2) Le langage de programmation NesC

Le langage NesC utilise une architecture basée sur des composants, celle-ci vise à réduire la taille mémoire du système et ces applications. Cette dernière est un ensemble de composants ayant un but précis. Chaque composant correspond à un élément matériel (LEDs, timer, ADC, etc) et peut être réutilisé dans différentes applications [**105**]. Un composant est constitué de trois éléments essentiels :

**Les interfaces :** Spécifient un ensemble de fonctions à mettre en application par le fournisseur de l'interface (commandes) et par l'utilisateur de l'interface (événements) [**18**]. Ces fonctions sont

précédées par des mots-clés respectifs *command* ou *event*. L'utilisation des mots clés *use* et *provide* au début d'un composant permet de savoir respectivement si celui-ci fait appel à une fonction de l'interface ou redéfini son code [**3**]. De plus, tous les composants possèdent l'interface *StdControl* car sa tâche est l'initialisation, le démarrage et l'arrêt des composants.

**Les modules :** Définissent les éléments de base de la programmation, ils utilisent une ou plusieurs interfaces. Par ailleurs, il est à noter que le processus d'exécution repose sur les tâches et les mécanismes d'interruption. De ce fait, les modules permettent aussi d'implémenter ces tâches.

**Les configurations :** Constituent un ensemble de modules et d'interfaces ainsi que des liaisons entre les composants de l'application déployée dans les capteurs.

## 3) Les simulateurs TOSSIM & POWERTOSSIM

Pour la simulation des comportements des nœuds au sein d'un réseau de capteurs, un outil très puissant a été développé  pour les capteurs doté d'un système d'exploitation TINYOS, sous le nom de TOSSIM [**87**]. Le principal but de ce simulateur est de créer une simulation très proche à celui d'un réseau de capteurs réels.

TOSSIM simule le comportement des applications de TINYOS au niveau des bits et chaque interruption dans le système est capturée. L'avantage majeur de ce type de simulateur est la simulation exacte du code qui tourne sur les capteurs.

TOSSIM peut être utilisé avec une interface graphique, sous le nom de TinyViz (voir Figure B.1). Cette dernière est équipée de plusieurs API plugins qui permettent d'ajouter plusieurs fonctions à notre simulateur comme par exemple l'illustration de l'envoi des messages en modes *broadcast* et *unicast*.

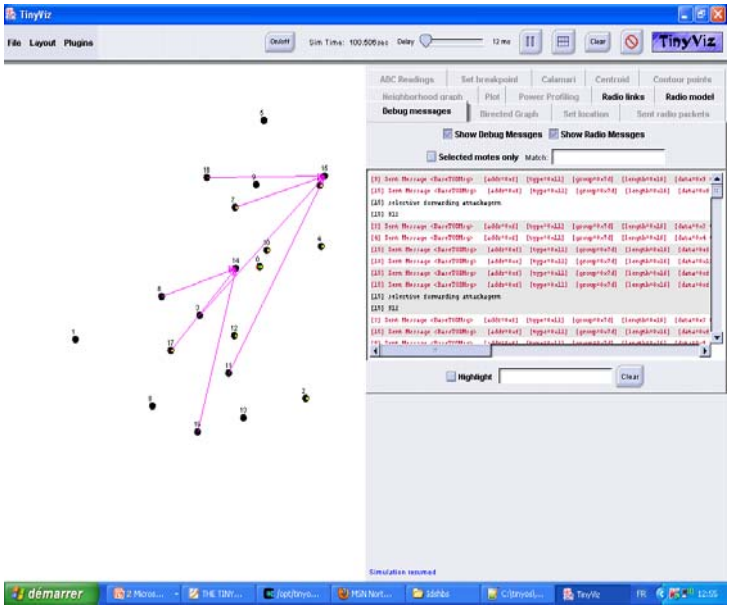



**Figure B.1. Fenêtre graphique de TinyViz**

Le simulateur TOSSIM n'a pas la capacité de suivre la dépense d'énergie des nœuds de capteurs pendant l'exécution de l'application. De ce fait, une version améliorée de cet outil, appelée POWERTOSSIM [**97**], a été proposée par l'Université de Harvard. Cette version permet la simulation de la consommation d'énergie et par conséquent la déduction de la durée de vie du réseau de capteurs. Un fichier de l'extension .trace est généré par le simulateur qui enregistre les détails de l'énergie consommée dans le réseau [**106**]. Ce fichier .trace est illustré dans la Figure B.2.

```
Mote 0, cpu total: 695.695771
Mote 0, radio total: 1161.897077
Mote 0, adc total: 0.000000
Mote 0, leds total: 0.000000
Mote 0, sensor total: 0.000000
Mote 0, eeprom total: 0.000000
Mote 0, cpu_cycle total: 0.000000
Mote 0, Total energy: 1857.592848

Mote 1, cpu total: 695.695771
Mote 1, radio total: 1029.464046
Mote 1, adc total: 0.000000
Mote 1, leds total: 0.000000
Mote 1, sensor total: 0.000000
Mote 1, eeprom total: 0.000000
Mote 1, cpu_cycle total: 0.000000
Mote 1, Total energy: 1725.159817

Mote 2, cpu total: 695.695771
Mote 2, radio total: 1118.733555
Mote 2, adc total: 0.000000
Mote 2, leds total: 0.000000
Mote 2, sensor total: 0.000000
Mote 2, eeprom total: 0.000000
Mote 2, cpu_cycle total: 0.000000
Mote 2, Total energy: 1814.429326
```

**Figure B.2. Fichier trace de l'énergie consommé de chaque nœud**

## 4) Détection d'intrusion dans un environnent réel

Dans cette section, nous décrivons le logiciel et matériel utilisés pour l'intégration de notre troisième approche de détection (décrit dans le chapitre 5) dans un réseau de capteurs réels. L'implémentation de cette approche a été effectuée au sein du laboratoire DRIVE (Université de Bourgogne).

**1-MoteConfig[107].** Ce logiciel (illustré dans la Figure B.3) fournit aux programmeurs une interface très simplifiée pour l'intégration de leur code écrit en Nesc dans les capteurs dotés d'un système d'exploitation TINYOS. De plus, MoteConfig permet aux utilisateurs de configurer l'identifiant du nœud (*Mote ID*), l'identifiant du group (*Group ID*), le canal RF (RF *channel*), et la puissance RF (RF *power*).


**Figure B.3. Fenêtre graphique de MoteConfig**

**2- Matériel utilisé.** Dans notre expérience nous avons utilisé 12 capteurs MICAZ [**101**], et une station de base (MIB520).

- Capteurs MICAZ. Ce type de capteur est composé de deux cartes (voir Figure B.4) : MPR2600 et MTS400, la première carte contient un microcontrôleur et un *transceiver* radio (émetteur-récepteur). La deuxième carte est constituée d'un certain nombre de capteurs pour la mesure de la température, accélération, luminosité et la pression. Les donnés captées sont transformées par la suite en valeurs numériques et transmises à la carte MPR2600.
- Station de Base. Comme le montre la Figure B.5, elle se compose d'une carte MPR2600 et une carte MIB520. Cette dernière, reçoit les donnés communiquées par la MPR2600 et les communique à l'ordinateur via un câble USB.

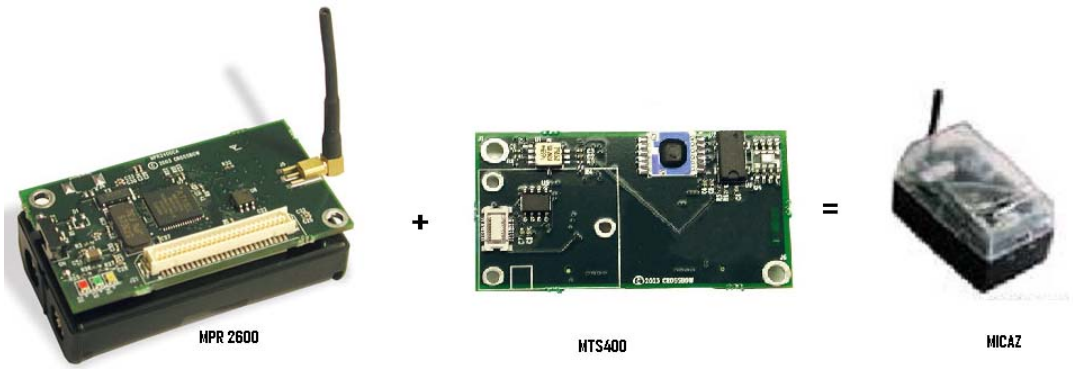



**Figure B.4. Capteur MicaZ**

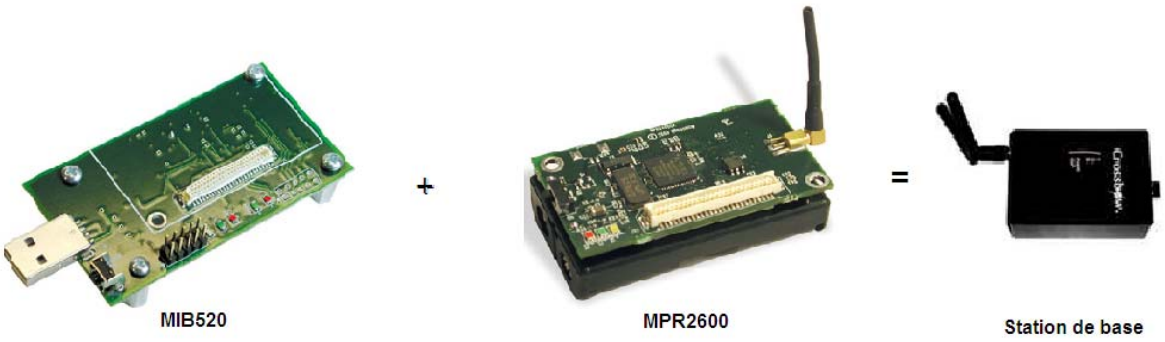



**Figure B.5. Station de base**

# Bibliographies

**1:** S. Sentilles, « Architecture logicielle pour capteurs sans-fil en réseau », Master Technologies de l'internet, Université de Pau et des Pays de l'Adour, France, Janvier - juin 2006.

**2**: I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. l. Cayirci, « A survey on sensor networks », IEEE Communications Magazine, 40(8): 102-116, August 2002.

**3**: A. Berrachedi, et A. Diarbakirli, « Sécurisation du protocole de routage hiérarchique LEACH dans les réseaux de capteurs sans fil», Ingénieur d'état en informatique, Ecole nationale Supérieure d'Informatique (E.S.I), Algérie, Juin 2009.

**4**: K. Benahmed, « Surveillance distribuer pour la sécurité d'un réseau de capteurs sans fil », Thèse de Doctorat en informatique, Université d'Oran, Algérie, 2011.

**5:** B. Djawhara, « Sécurité de la dissémination de données dans un réseau de capteurs sans fil : cas du protocole Tiny Diffusion » Ingénieur d'état en informatique, Ecole nationale Supérieure d'Informatique (E.S.I), Algérie, juin 2009.

**6**: R. Kacimi, « Techniques de conservation d'énergie pour les réseaux de capteurs sans fil », Thèse de Doctorat spécialité : Réseaux et Télécommunications, Université de Toulouse, France, Septembre 2009.

**7**: K. Bouabdellah, « Problématique de la consommation de l'énergie dans les réseaux de capteurs sans fil », Séminaire LIUPPA, Université de Pau et des Pays de l'Adour, 14 Octobre 2007.

**8**: I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. « Wireless sensor networks: a survey», Computer Networks, 38(4): 393-422, March 2002.

**9**: C.Y. Chong and S.P. Kumar, « Sensor network: evolution, opportunities, and challenges », In proceedings of the IEEE, 91(8), pp. 1247-156, 2003.

**10:** T.B. Gosnell, J.M. Hall, C.L. Hall, C.L. Ham, D.A. Knapp, Z.M. Koening, S.J. Luke, B.A. Pohl, A.Schan von Wittenau, and J.K. Wolford, «Gamma-ray identification of nuclear weapon materials», Technical Report DE97053424, Lawrence Livermore National Lab, CA, USA, 1997.

**11**: R. Merzougui, M. Feham and H. Sedjelmaci, « Design and implementation of an algorithm for cardiac pathologies detection on mobile phone », International Journal of Wireless Information Networks, 18 (1):11-23, 2011.

**12**: M. Mana, « Adaptation et intégration de la sécurité biométrique aux réseaux de capteurs corporels sans fil », Thèse de Doctorat en télécommunication, Université de Tlemcen, Algérie, Janvier 2011.

**13**: K. Beydoun, « Conception d'un protocole de routage hiérarchique pour les réseaux de capteurs », Thèse de Doctorat en informatique, Université de Franche-Comté, France, Décembre 2009.

**14**: M. Fitzgerald. Technnology Review: Tracking a Shopper's Habits, August 2008. http://www.technologyreview.com/computing/21161/

**15:** V. Tsetsos, G. Alyfantis, T. Hasiotis, O. Sekkas, and S. Hadjiefthymiades, «Commercial wireless sensor networks: technical and business issues», Second Annual Conference on Wireless On-demand Network Systems and Services, St. Moritz, Switzerland, pp.166-173, 2005.

**16**:A. Kamal, and J. Al-Karaki, « Routing techniques in wireless sensor networks: a survey », IEEE Wireless communications, 11(6): 6-28, 2004.

**17**: M. Ilyas, and I. Mahgoub, «Architecture and modeling of dynamic wireless sensor networks», Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems, chapitre 15, CRC Press LLC, 2005.

**18**: M. Lehsaini, « Diffusion et couverture basées sur le clustering dans les réseaux de capteurs : application à la domotique», Thèse de Doctorat en informatique, Université de Tlemcen et Université de Franche-Comté, 2009

**19**: L. Khelladi, N. Badache, « Improving directed diffusion with power-aware topology control for adaptation to high density », LOCALGOS'08 workshop, in conjunction with The 4th IEEE/ACM International Conference on Distributed Computing In Sensor System, Greece, 2008.

**20**: K. Sohrabi, J. Gao, V. Ailawadhi, and G.J. Pottie, « Protocols for self-organization of a wireless sensor network », IEEE Journal of Personal Communications, 7(5):16-27, 2000.

**21**: A. Perrig, R. Szewczyk, J.D. Tygar, V.E. Wen, and DE. Culler, « SPINS: security protocols for sensor networks ». Wireless Networks Journal,8(5):521-534, September 2002.

**22**:W. R. Heinzelman, A. Chandrakasan , and H. Balakrishnan, "Energy efficient communication protocol for wireless microsensor networks", Proceeding of the 33rd Hawaii International Conference on System Sciences, IEEE, pp.1-10, 2000.

**23**: O. Younis, and S. Fahmy, "Heed: A hybrid energy-efficient distributed clustering approach for ad hoc sensor networks", IEEE Transactions on Mobile Computing, 3(4): 366-379, 2004.

**24**: S. Lindsey, and C. Raghavendra, "PEGASIS: Power efficient gathering in sensor information system", In Proc.IEEE Aerospace Conference, vol.3, pp.1125-1130, 2002.

**25**:A. Manjeshwar, and D.P. Agarwal, TEEN: a routing protocol for enhanced efficiency in wireless sensor networks, 15th International Proceedings on Parallel and Distributed Processing, IEEE, pp. 2009-2015, 2000.

**26**: C. Bidan, « Sécurité des systèmes distribuée: apport des architecture logiciel », Thèse de Doctorat en informatique, Université de Rennes I, 1998.

**27**: G. Athanasios, «Security threats in wireless sensor networks: implementation of attacks &defense mechanisms», PhD in Wireless Communications, Aalborg University, Denmark, 2011.

**28**: M.L. Messai, « Sécurité dans les réseaux de capteurs sans fil», Magister en informatique, Université Abderrahmane Mira de Bejaia, Algérie, 2008.

**29**: A. D.Wood, and J. A. Stankovic, «Denial of service in sensor networks », IEEE Computer, 35(10): 54–62, 2002.

**30**: S.Rajasegarar, C.Leckie, and M.Palaniswami, « Detecting data anomalies in wireless sensor networks », in R. Beyah, J.McNair, and C. Corbett, editors, Security in Ad-hoc and Sensor Networks, World Scientific Publishing, Inc, ISBN 978-981-4271-08-0, pp 231-260, July 2009.

**31**: S. Kaplantzis, « Security models for wireless sensor networks », PhD Conversion Report, Centre of Telecommunications and Information Engineering, Monash University, Australia, 2006.

**32**: A. Mitrokotsa , and A. Karygiannis, « Intrusion detection techniques in sensor networks », Wireless Sensor Network Security, 1(1):251-272, 2008.

**33**: R. Roman, C. Alcaraz, and J. Lopez, "A Survey of Cryptographic Primitives and Implementations for Hardware-Constrained Sensor Network Nodes", Mobile Networks and Applications, 12 ( 4): 231-244, 2007.

**34**: M. Saraogi, «Security in wireless sensor networks », research report, University of Tennessee, Knoxville.

**35**: N. Dagorn1, «Détection et prévention d'intrusion : présentation et limites», Rapport de recherche, Université de Nancy, France,

**36**: V.S. Bhuse, «Lightweight intrusion detection: a second line of defense for unguarded wireless sensor networks», Ph. D. thesis, Western Michigan University, USA, 2007.

**37**: A.stetsko, « Intrusion detection for wireless sensor networks, PHD dissertation thesis topic», Faculty of informatics, Masaryk University, Czech Republic, 2008.

**38**: S. Marti, T. Giuli, K. Lai, and M. Baker. «Mitigating routing misbehavior in mobile ad hoc networks». 6th ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'00), pp. 255-265, August 2000.

**39**: R. Roman, J. Zhou, J. Lopez, «Applying intrusion detection systems to wireless sensor networks », In: 3rd IEEE Consumer Communications and Networking Conference, pp.640-644, 2006.

**40**: I. Krontiris, « Intrusion prevention and detection in wireless sensor networks », PHD thesis, Mannheim, Germany, 2008.

**41**: Y. Zhang, and W. Lee, « Intrusion detection techniques for mobile wireless networks», ACM/Kluwer Wireless Networks Journal, 9(5):545-556, September 2003.

**42**: J. Kong, H. Luo, K. Xu, D. L. Gu, M. Gerla, and S. L, « Adaptive security for multi-layer ad-hoc networks », Special Issue of Wireless Communications and Mobile Computing, 2002.

**43**: P. Albers, O. Camp, J. Percher, B. Jouga, L. Me, and R. Puttini, «Security in ad hoc networks: a general intrusion detection architecture enhancing trust based approaches », 1st International Workshop on Wireless Information Systems (WIS'02), April 2002.

**44**: I. Stamouli, P.G. Argyroudis, and H. Tewari, «Real-time intrusion detection for adhoc networks», Proceedings of the Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM'05), 2005.

**45**: S. Kumar, « Classification and detection of computer intrusions », PhD Thesis, Department of Computer Sciences, Purdue University, USA, 1995.

**46**: J. Gama and R. Pedersen. Predictive learning in sensor networks, « In learning from data streams», editors João Gama and Mohamed Gaber, Springer, Chapter 10, pp.143-164, 2007.

**47**: A. H. Sung, and S. Mukkamala, «Identifying Important Features for Intrusion Detection using Support Vector Machines and Neural Networks», Symposium on Applications and the Internet, IEEE, Orlando, USA, pp.209-216, 2003.

**48**: S. Kaplantzis, A. Shilton, N. Mani, and Y. A. Sekercioglu, «Detecting selective forwarding attacks in wireless sensor networks using support vector machines », In 3rd International Conference on Intelligent Sensors, Sensor Networks and Information, IEEE, Melbourne, Australia, pp.335-340, 2007.

**49**: A. P. Silva, M. H. Martins, B. P. Rocha, A. A. Loureiro, L. B. Ruiz, H. C. Wong, «Decentralized intrusion detection in wireless sensor networks ». In Proceedings of the 1st ACM International Workshop on Quality of Service & Security in Wireless and Mobile Networks , Montreal, Quebec, Canada, October 13, pp.16-23, 2005.

**50**: H. Mohamadally, «SVM : machines à vecteurs de support ou séparateurs à vastes marges», Rapport de recherche, Versailles St Quentin, France, janvier 2006.

**51**: K. Flouri, B. B.Lozano, and P. Tsakalides, « Optimal Gossip Algorithm for Distributed Consensus SVM Training in Wireless Sensor Networks », In Proc.16th International Conference on Digital Signal Processing, IEEE, Santorini, Greece, pp.1-6, 2009.

**52**: K. Flouri, B. B.Lozano, and P. Tsakalides, «Training a SVM-based classifier in distributed sensor networks», In Proc.14nd European Signal Processing Conference, Florence, Italy, 2006.

**53**: S. Rajasegarar, C.Leckie, M.Palaniswami, and J. C Bezdek, « Quarter sphere based distributed anomaly detection in wireless sensor networks », In IEEE International Conference on Communications, Glasgow, Scotland, pp.3864-3869, 2007.

**54**: K. Flouri, B. B. Lozano, and P. Tsakalides, « Distributed consensus algorithms for SVM training in wireless sensor networks », In Proc.16th European Signal Processing Conference, Lausanne, Switzerland, 2008.

**55:** I. Krontiris, T. Dimitriou, F.C. Freiling, « Towards intrusion detection in wireless sensor networks », Proceedings of the 13th European Wireless Conference, Paris, France, 2007.

**56**: A. Stetsko, V. Matay, « Effectiveness metrics for intrusion detection in wireless sensor networks», European Conference on Computer Network Defense, IEEE, Milan, Italy, 2009; 21-28.

**57**: F. Anjum, D. Subhadrabandhu, S. Sarkar and R. Shetty, « On optimal placement of intrusion detection modules in sensor networks», In Proceedings of the 1st International Conference on Broadband Networks, San Jose, California, USA, pp. 690-699, October 2004.

**58**: A. H. Farooqi, and F.A. Khan, « Intrusion detection systems for wireless sensor networks: a survey», Proc. Communications in Computer and Information Science, Springer, Volume 56, Jeju, South Korea, pp.234-241, 2009.

**59**: T. Techateerawat, A. Jennings, «Energy efficiency of intrusion detection systems in wireless sensor networks », In Proceedings of the 2006 IEEE/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Hong Kong, China, pp. 227-230, 2006.

**60**: T. H. Hai, F. khan, E. N. Huh, «Hybrid intrusion detection system for wireless sensor networks», In Proceeding of the ICCSA, Springer, Kuala Lumpur, Malaysia, pp. 383-396, 2007.

**61**: K. Q. Yan, S. C. Wang, S. S. Wang, C. W. Liu, «Hybrid intrusion detection system for enhancing the security of a cluster-based wireless sensor network », Proceedings of 3rd IEEE International Conference on Computer Science and Information Technology , Chengdu, China, pp. 114-118, 2010.

**62**: G. Huo, X. Wang , «A dynamic model of intrusion detection system in wireless sensor networks», International Conference on Information and Automation, IEEE, Zhangjiajie, China, pp. 374-378, 2008.

**63**: S. Khanum, M. Usman, K. Hussain. «Energy-efficient intrusion detection system for wireless sensor network based on MUSK architecture», Second International Conference on High Performance Computing and Applications, Springer, Shanghai, China, pp.212-217, 2009.

**64**: H. Sedjelmaci, and M. Feham, «Novel hybrid intrusion detection system for clustered wireless sensor network», International Journal of Network Security & its Applications (IJNSA), 3(4): 1-14, 2011.

**65**: H. Sedjelmaci, S.M. Senouci, and M. Feham, « Intrusion detection framework of cluster-based wireless sensor network », IEEE Symposium on Computers and Communications, Cappadocia, Turkey, pp. 893-897, July 2012.

**66**: T. H. Hai, E. N. Huh, and M . Jo, « A lightweight intrusion detection framework for wireless sensor networks », Wireless Communications and Mobile Computing, 10(4): 559-572, 2010.

**67**: A. Agah, S.K. Das, and K. Basu , « A non-cooperative game approach for intrusion detection in sensor networks », IEEE Vehicular Technology Conference, Los Angeles, USA, pp. 2902 - 2906, 2004.

**68**: A. Agah, and S.K. Das, « Preventing DoS attacks in wireless sensor networks: a repeated game theory approach », International Journal of Network Security, 5(2):145-153, 2007.

**69**: M. Mohi, and A. Movaghar, P. M. Zadeh, « A bayesian game approach for preventing dos attacks in wireless sensor networks » , International Conference on Communications and Mobile Computing, IEEE, Yunnan, China, pp. 507-511, 2009.

**70**: E.V. Damme, « Stability and perfection of Nash equilibria », Springer; 2nd edition, October 2002.

**71**: P. Gonzales, et J. Crête, « Jeux de société: une initiation à la théorie des jeux en sciences sociales», Les presses de l'Université Laval, mai 2006.

**72**: L. Blazevic, L. Buttyán, S. Capkun, S. Giordano, J.P. Hubaux, J.Y. Le Boudec, «Self-organization in mobile ad-hoc networks: the approach of terminodes», IEEE Communication Magazine, 39(6) :166-174, 2001.

**73**: W. H. Bin, Y. Zheng, W. C. Dong. **«** Intrusion detection for wireless sensor networks based on multi-agent and refined clustering **»**, International Conference on Communications and Mobile Computing, IEEE, Yunnan, China, pp. 450-454,2009.

**74**: M. Ketel. **«** Applying the mobile agent paradigm to distributed intrusion detection in wireless sensor networks **»**, the 40th Southeastern Symposium on System Theory, IEEE, New Orleans, USA, pp. 74-78, 2008.

**75:** KDD Cup 1999 Data, http://kdd.ics.uci.edu/databases/kddcup99/task.html; 1999.

**76**: P. Brutch, and C. Ko, **«** Challenges in intrusion detection for wireless ad-hoc networks **»**, Symposium on Applications and the Internet Workshops, IEEE, Orlando, USA, pp. 368-373, 2003.

**77**: L. Besson, P. Leleu, **«**A distributed intrusion detection system for ad-hoc wireless sensor networks: the AWISSENET distributed intrusion detection system **»**, 16th International Conference on Systems, Signals and Image Processing, IEEE, Chalkida, Greece, pp. 1-3, 2009.

**78**: A. Hafslund, A. Tonnesen, R. B. Rotvik, J. Anderson, O. Kure, **«** Secure extension to the OLSR protocol **»**, Paper Presented at OLSR Interop and Workshop, San Diego, USA, 2004.

**79**: A. Abduvaliyev, S. Lee, Y. K. Lee, **«** Energy efficient hybrid intrusion detection system for wireless sensor networks **»**, International Conference on Electronics and Information Engineering, IEEE, Kyoto, Japan, p. 25-29, 2010.

**80**: H. C. Le, H. Guyennet, N. Zerhouni, **«** Overhearing for energy efficient in event-driven wireless sensor networks **»**, IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS), Vancouver, Canada, p. 633-638, 2006.

**81**: A. Stetsko, L. Folkman , V. Matay, **«** Neighbor-based intrusion detection for wireless sensor network », The 6th International Conference on Wireless and Mobile Communications, IEEE, Valencia, Spain, p. 420-425, 2010.

**82**: R. U. Pedersen, **«**Using support vector machines for distributed machine learning, Doctor of Philosophy, University of Copenhagen, Denmark, August 2004.

**83**: P. Mahé, **«** Noyaux pour graphes et support vector machines pour le criblage virtuel de molécules **»**, Rapport de stage, Septembre 2003.

**84**: B. Scholkopf, and A. J. Smola, **«** Learning with kernels **»**, The MIT Press, pp.204-205, 2006.

**85**: The network simulator NS-2, http://www.isi.edu/nsnam/ns/.

**86**: OPNET Modeler, http://www.opnet.com/solutions/network_rd/modeler.html, 2012.

**87**: Simulating TinyOS networks, http:// www.cs.berkeley.edu/ pal/research/tossim.html, November 2003.

**88**: S.B. Akat, V. Gazi, « Particle swarm optimization with dynamic neighborhood topology: three neighborhood strategies and preliminary results », In: IEEE Swarm Intelligence Symposium, St. louis, Missouri, USA, 2008.

**89**: M. He, « Feature Selection based on ant colony optimization and rough set theory », International Symposium on Computer Science and Computational Technology, pp. 247 - 250, Shanghai, China, 2008.

**90**: S. S. Wang, K. Q. Yan, S. C. Wang, and C. W. Liu, **«** An integrated intrusion detection system for cluster-based wireless sensor networks **»**, Expert Systems with Applications 38 (12): 15234–15243, 2011.

**91**: S. Shin, T. Kwon, G.Y. Jo, Y. Park, H. Rhy, **«** An experimental study of hierarchical intrusion detection for wireless industrial sensor networks **»,** IEEE Transactions on Industrial Informatics 6(4): 744-757, 2010.

**92**: M. S. Mamun, and A.F.M. Sultanul Kabir, **«** Hierarchical design based intrusion detection system for wireless ad hoc sensor network **»**, International Journal of Network Security & Its Applications (IJNSA), 2(3), July 2010.

**93**: R. DeGraaf, I. Hegazy, J. Horton, and R. Safavi-Naini, **«** Distributed detection of wormhole attacks in wireless sensor networks **»**, Proceedings of 1rst International Conference on Ad hoc Networks, Springer, Niagara Falls, Canada, pp. 208-223, 2009.

**94**: S. Ganeriwal, and M. B. Srivastava, **«** Reputation based framework for high integrity sensor networks **»,** Proceeding of the Second ACM Workshop on Security of Ad Hoc and Sensor Networks, New York, USA, pp. 66-77, 2004.

**95**: H. Alzaid, E. Foo, J.G. Nieto, E. Ahmed, « Mitigating On-Off attacks in reputation-based secure data aggregation for wireless sensor networks », Security and Communication Networks, 5(2):125-144, 2012.

**96**: NesC 1.1 Language Reference Manual, http://nescc.sourceforge.net/papers/nesc-ref.pdf, 2003

**97**: Efficient power simulation for TinyOS applications, http://www.eecs.harvard.edu/ shnayder/ptossim/, 2004.

**98**: CC1000 chip, very low power RF transceiver, http://www.ti.com/lit/ds/symlink/cc1000.pdf, 2009.

**99**: M. Tiwari, K.V. Arya, R. Choudhari, K.S. Choudhary, «Designing intrusion detection to detect black hole and selective forwarding attack in WSN based on local information », Fourth International Conference on Computer Sciences and Convergence Information Technology, IEEE, Seoul, Korea, pp. 824-828, 2009.

**100**: G. Li, J. He, Y. Fu, « A group-based intrusion detection scheme in wireless sensor networks **»**, Computer Communications 31 (18): 4324–4332, 2008.

**101**: MICAz, Wireless measurement system, http://www.openautomation.net/uploadsproductos/ micaz_datasheet.pdf.

**102**: W. Xu, W. Trappe, Y. Zhang, T. Wood, « The feasibility of launching and detecting jamming attacks in wireless networks **»,** In Proc. 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Illinois, USA, pp. 46-57, 2005.

**103**: W. Xu, K. Ma, W. Trappe, Y. Zhang, «Jamming sensor networks: attack and defense strategies **»**, IEEE Network Magazine, 20 (3): 41-47, 2006.

**104**: CC2420 Datasheet, 2.4 GHz IEEE 802.15.4/ZigBee-ready RF Tranceiver, http://inst.eecs.berkeley.edu/cs150/Documents/CC2420.pdf, 2006.

**105**: M. Damou, L. Mounier, « Simulation d'un réseau de capteurs avec TinyOS », Rapport de recherche, Laboratoire VERIMAG, Grenoble, France.

**106**: B. Chen, G.W. Allen, M. Hempstead, M. Welsh, V. Shnayder, « Simulating the power consumption of large scale sensor network applications», Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, Harvard University, USA, pp. 188 – 200, 2004.

**107**: XMesh MoteConfig User Manual, http:// http://www.memsic.com/, 2010

# Liste des publications

## Revues Internationale

1- **Hichem Sedjelmaci**, and Mohamed Feham, "Novel Hybrid Intrusion Detection System For Clustered Wireless Sensor Network", *International Journal of Network Security & Its Applications (IJNSA)*, Vol.3, No.4, July 2011.
**ISSN: 0974-9330**
**DOI:** 10.5121/ijnsa.2011.3401

2-Rachid Merzougui, Mohammed Feham, **Hichem Sedjelmaci**, "Design and implementation of an algorithm for cardiac pathologies detection on mobile phone", *International Journal of Wireless Information Networks (IJWIN), Springer*, Vol. 18, Number 1, pp. 11–23, March 2011.
**ISSN:** 1068 – 9605.
**DOI:** 10.1007/s10776-011-0129-1.

3-**Hichem Sedjelmaci**, Sidi Mohammed Senouci, and Mohammed Feham, "An Efficient Intrusion Detection Framework in Cluster-Based Wireless Sensor Networks", *to appear in Security and communication networks, John Wiley & Sons*, 2013.

4- **Hichem Sedjelmaci**, Sidi Mohammed Senouci, and Mohammed Feham,"Efficient and Lightweight Intrusion Detection for Clustered Wireless Sensor Networks Based on Neighbors' Behaviors", *Transactions on Emerging Telecommunications Technologies, John Wiley & Sons (*Soumis*)*

## Conférence Internationale

5- **Hichem Sedjelmaci**, Sidi Mohammed Senouci, and Mohammed Feham, "Intrusion Detection Framework of Cluster-based Wireless Sensor Network", IEEE ISCC'2012, Cappadocia, Turkey, pp. 893 - 897 ,July 1 - 4, 2012.
**ISSN :** 1530-1346
**DOI :** 10.1109/ISCC.2012.6249415