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 Février 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

ملخص

شبكات الملتقطات اللاسلكية جذبت الكثير من الأاهتمام نظرا لتطبيقاتها في ميادين مختلفة منها العسكرية و المدنية بالإضافة إلى هذا هناك عوامل طاقية وأخرى مرتبطة بالبيئة العدائية للملتقطات ممّا يجعله عرضة للهجوم القرصنة.للحماية هذا النوع من الشبكة تستعمل الحلول الأمنية للملتقطات حيث هو بحد ذاته تحدى تناولتها بدقة هذه الاطروحة.

إن طاقة العقد هي نقطة هامة جدًا في تصميم وتنفيذ التطبيق. فعملية الاتصال تستهلك طاقة أكبر بكثير من تلك التي تسببها العمليات الحسابية. وفي هذا السياق، العديد من الباحثين يعملون على هذه المشكلة باقتراح بروتوكولات التوجيه التي تهدف إلى تقليل كمية المعلومات المتبادلة بين العقد في الشبكة. من بين هذه البروتوكولات يمكننا الإستشهاد بخوارزمية المجموعات هدفهم هو تنصيب عقدة واحدة في كل مجموعة التي تعد العقدة المسؤولة عن تجميع و نقل المعلومات المتاحة إلى المحطة.

أنظمة كشف الإختراقات لها القدرة على اكتشاف الهجمات الداخلية و الخارجية للشبكة على عكس غيرها من الحلول الأمنية مثل التشفير المعروف بمحدودية قدرته حيث لا يمنع الإ الهجمات الخارجية من الدخول إلى الشبكة.

في هذه الأطروحة قمنا بتطوير و تنفيذ مجموعة من النماذج لكشف الاختراقات التي تحدث للملتقطات اللاسلكية مع أخذ بعين الاعتبار العوامل الطاقية.

كلمات البحث: شبكات الملتقطات اللاسلكية، أنظمة كشف الإختراقات، معدل اكتشاف، إيجابية كاذبة، فعالية، طاقة.

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,...

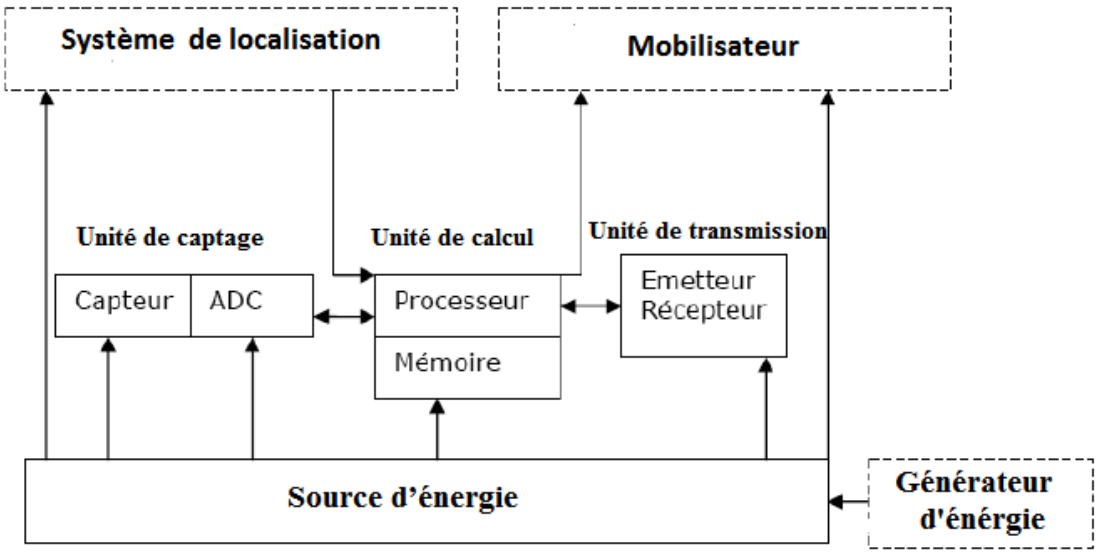


Figure 1.1. Architecture d'un capteur [2]

- **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**].

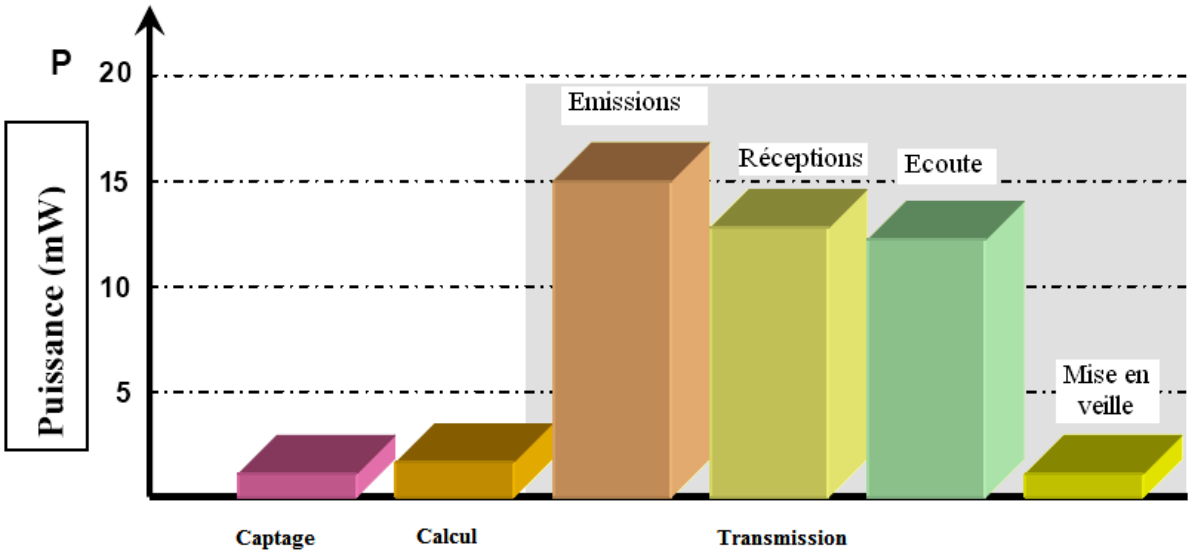


Figure 1.2. Consommation d'énergie en captage, calcul et transmission [3]

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.

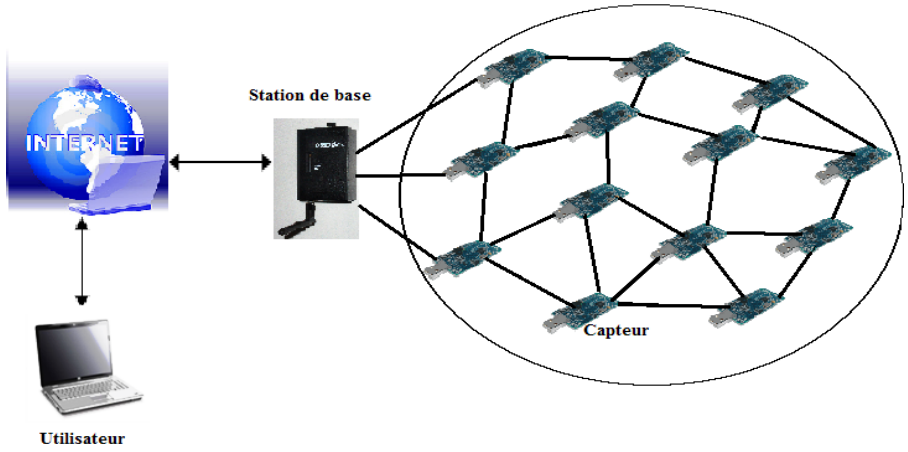


Figure 1.3. Architecture d'un réseau de capteurs sans fil

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.

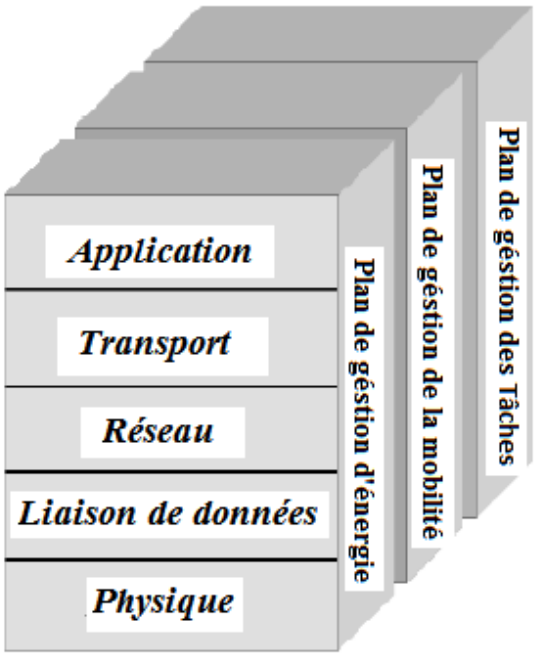


Figure 1.4. Pile protocolaire dans un réseau de capteurs sans fil

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]**.

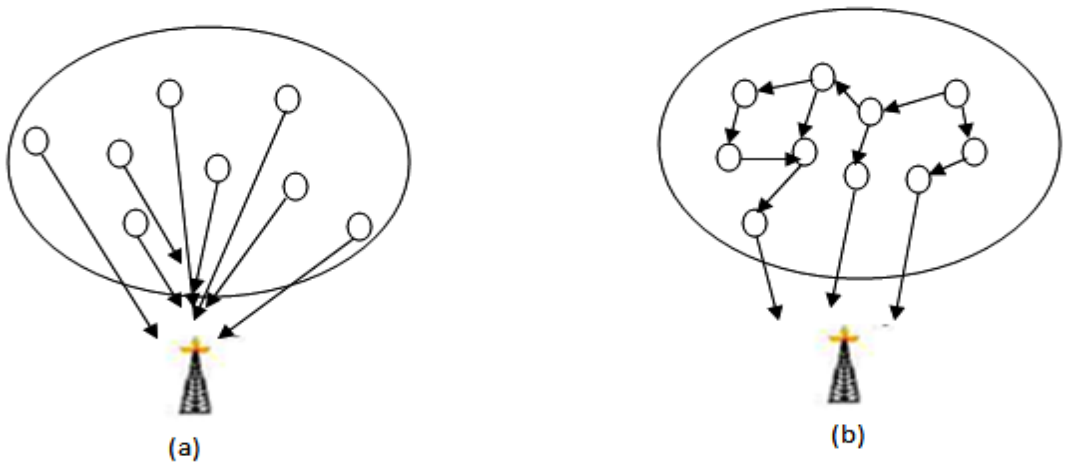
3. Architecture réseau

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.

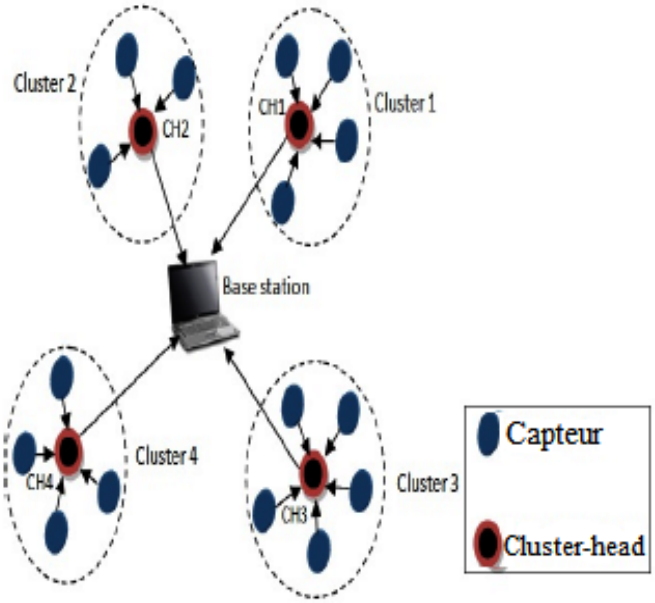


Figure 1.6. Topologie à base de cluster

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.

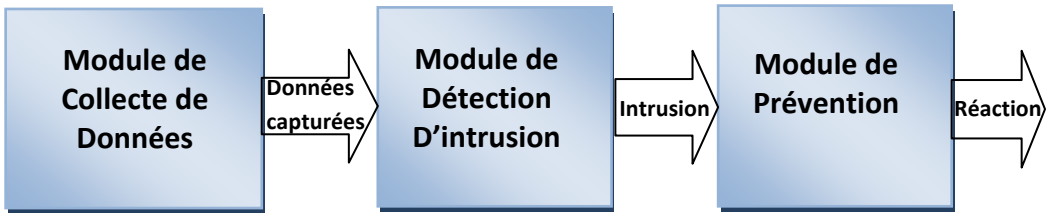


Figure 1.7. Les composants d'un agent IDS

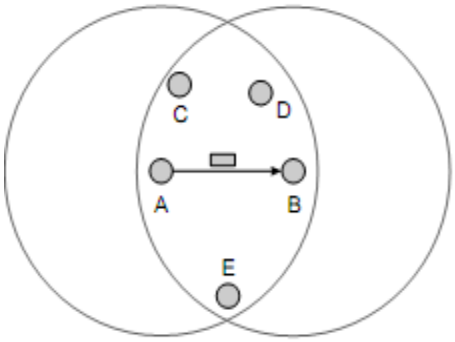
Les différentes technologies des IDSs. Il existe deux grandes technologies distinctes d'IDS :

- 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**].

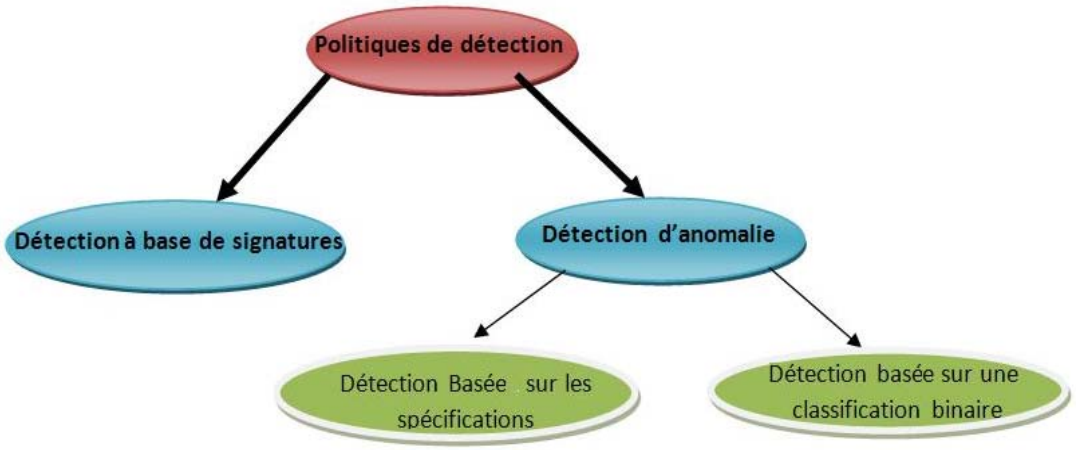


Figure 2.1. Les techniques de détection d'intrusion

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**].

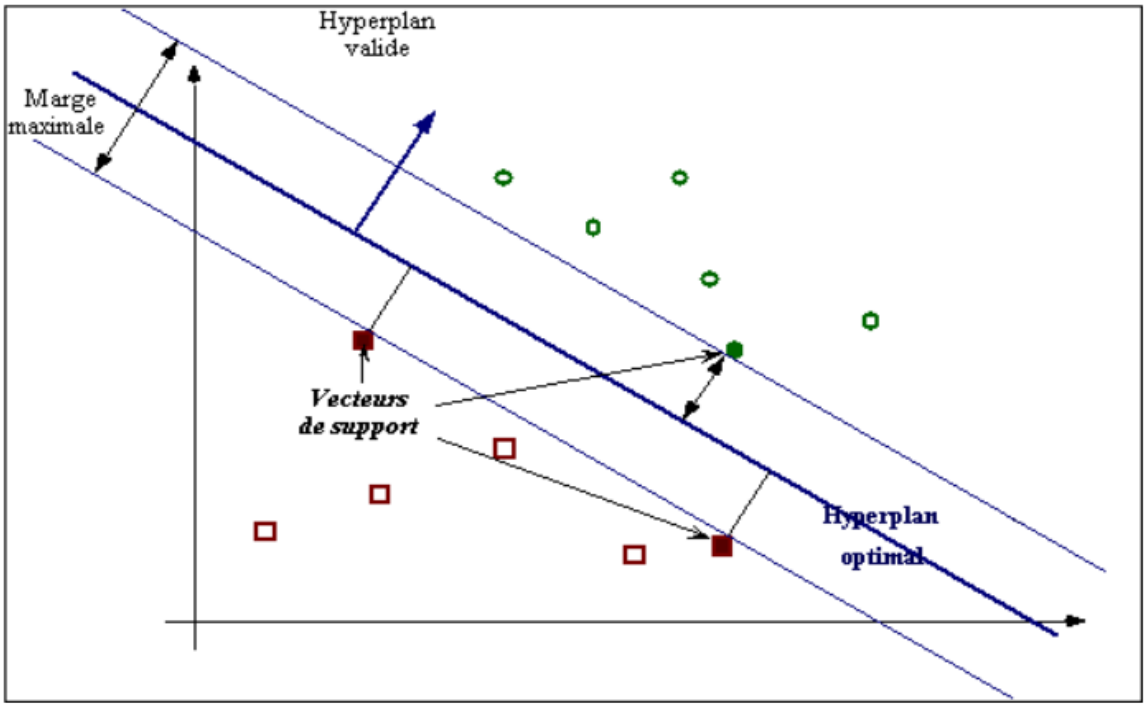


Figure 2.2. Hyperplan optimal et vecteurs de support

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

IDSs devront préserver leur puissance d'émission et minimiser l'échange des données entre eux ou avec un autre nœud (par exemple le *cluster-head*).

- **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.

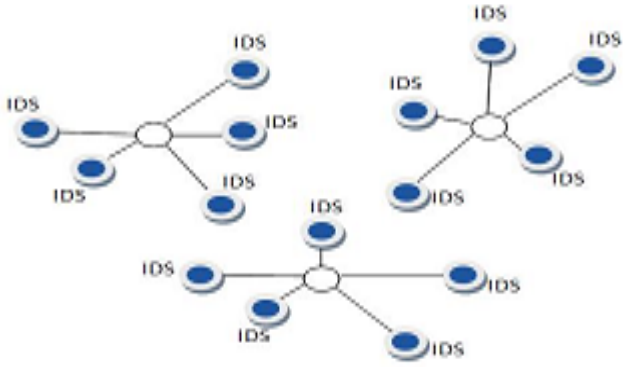


Figure 2.3. Les agents IDS dans les membres du cluster

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.

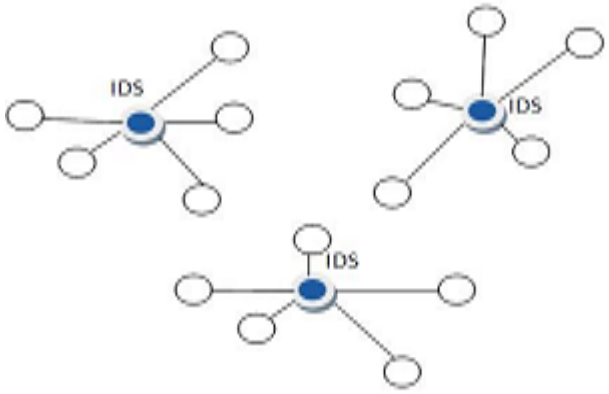


Figure 2.4. L'agent IDSs dans le *cluster-head*

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.

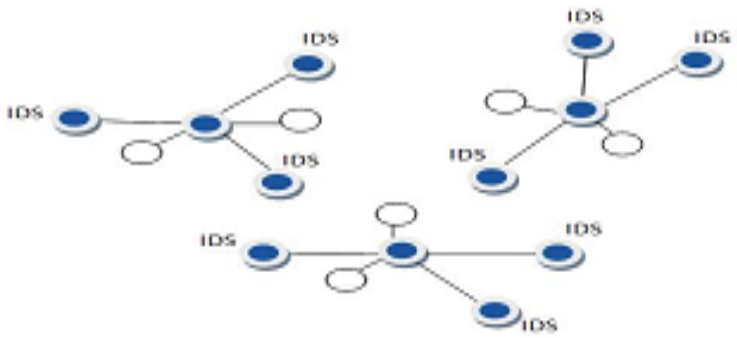


Figure 2.5. Les agents IDS dans la frontière du cluster et dans le *cluster-head*

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:

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

ε_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 α_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.

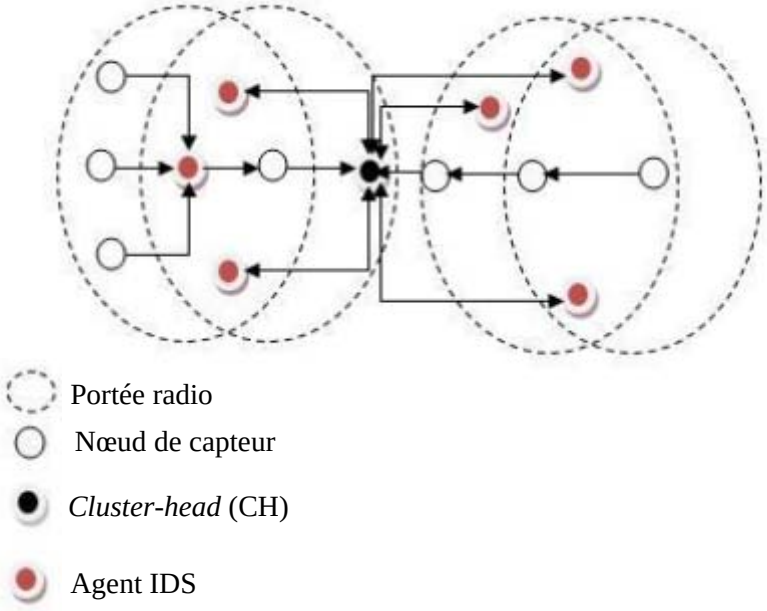


Figure 3.1. Stratégie de l'emplacement des IDSs dans le RCSFC

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.

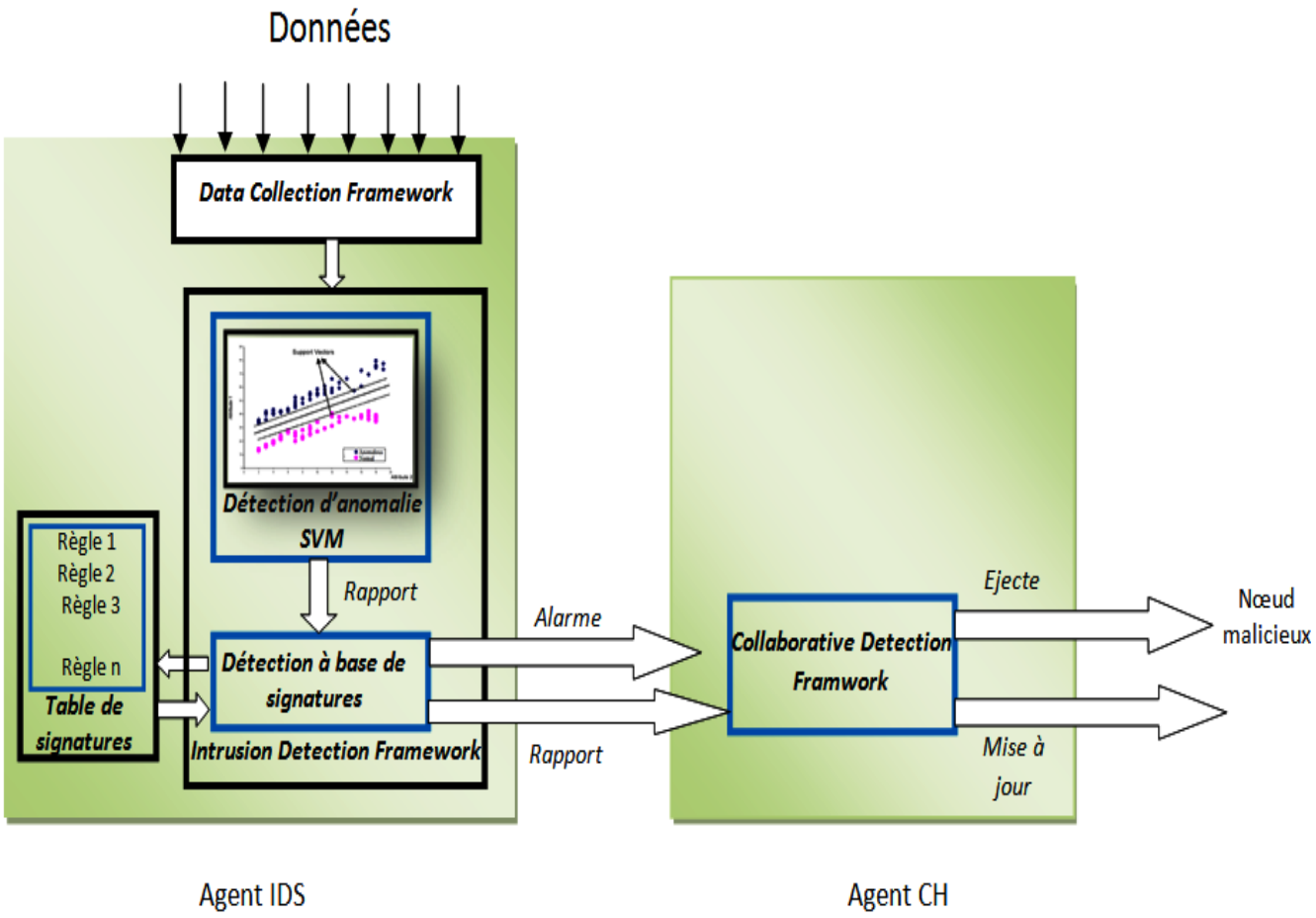


Figure 3.2. L'architecture du modèle de détection d'intrusion

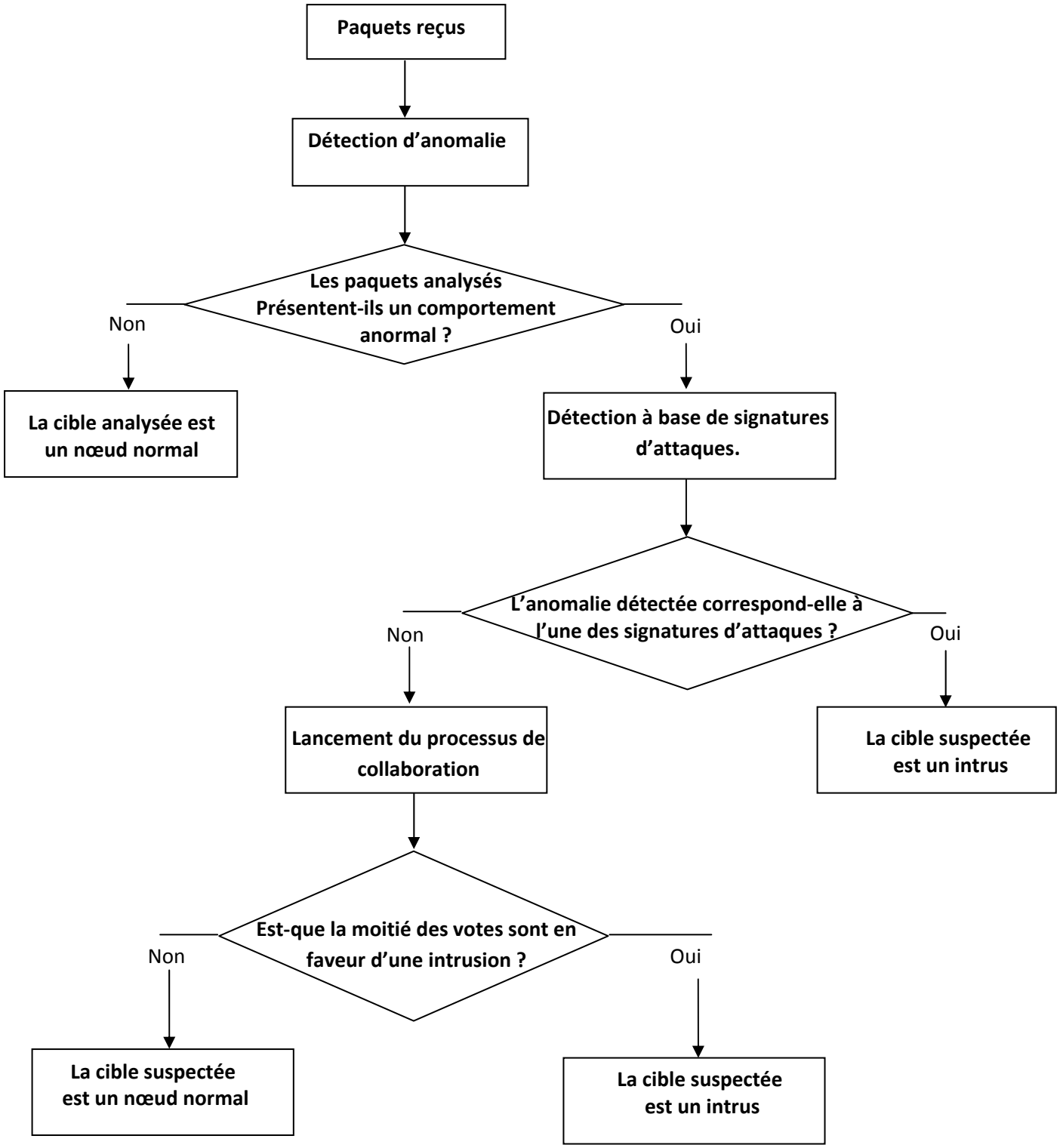


Figure 3.3. Organigramme du modèle hybride de détection d'intrusion

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.

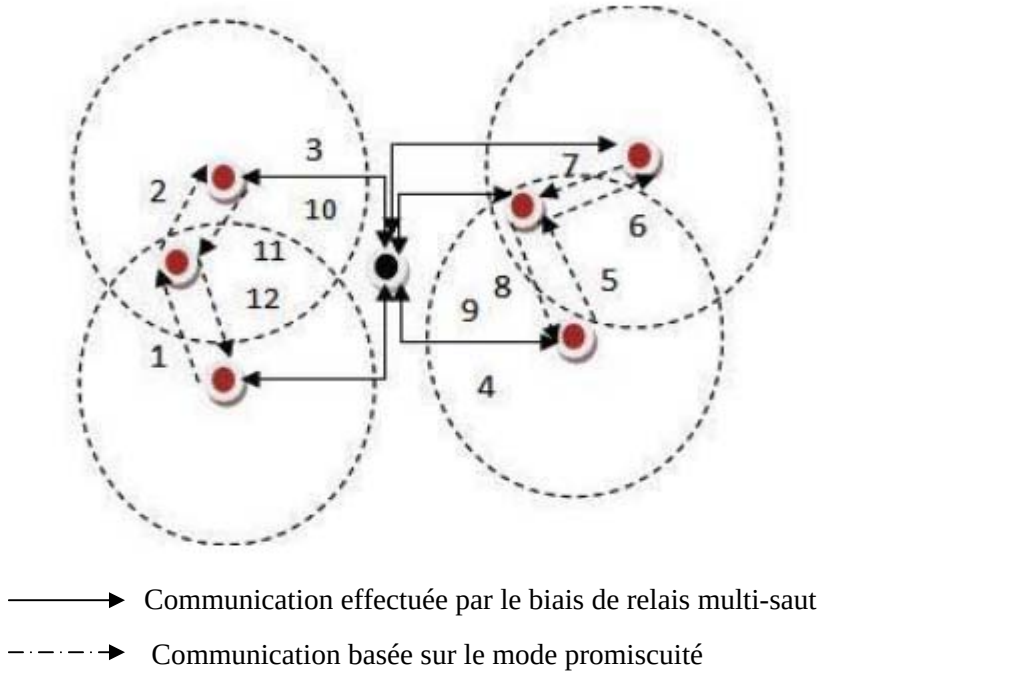


Figure 3.4. Communication des vecteurs de support entre les nœuds IDS

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.

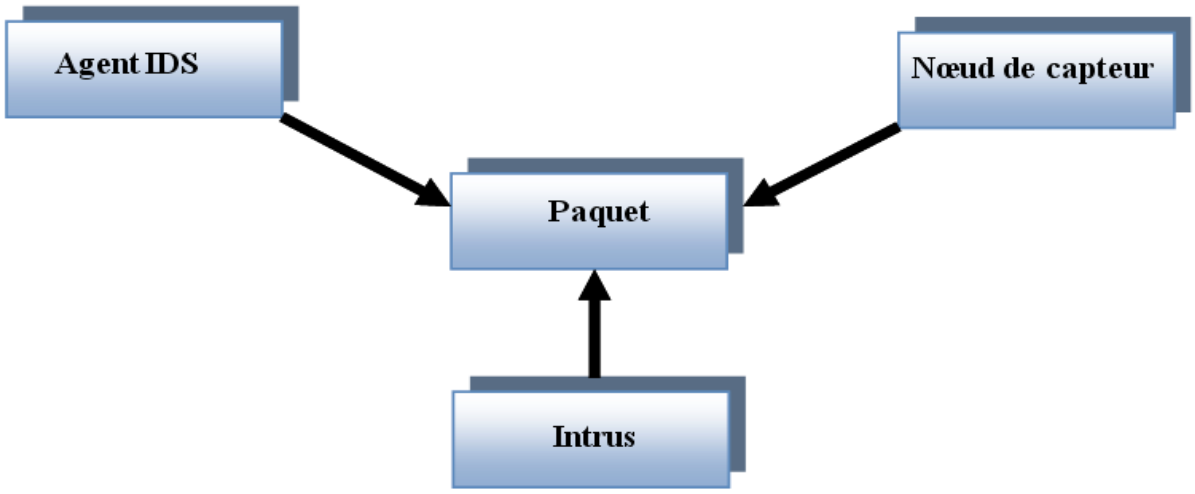


Figure 3.5 Les principaux composants du simulateur

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.

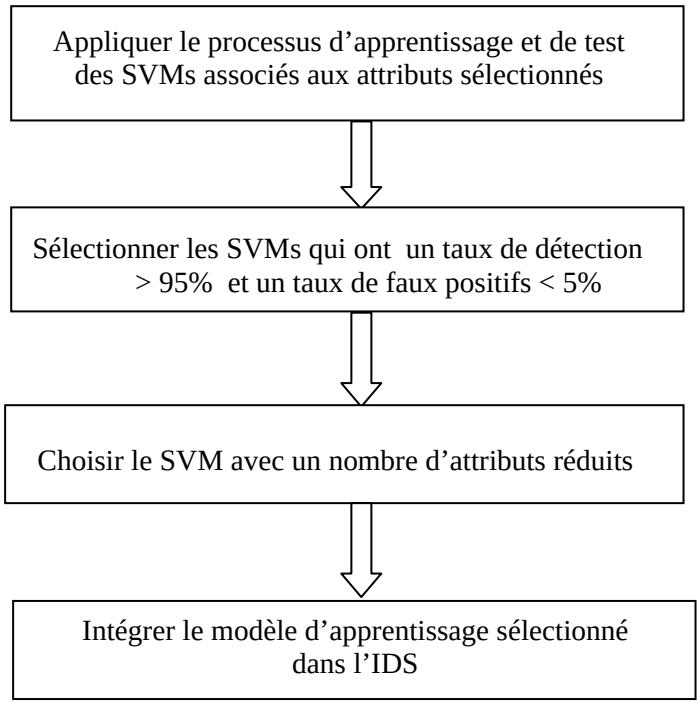


Figure 3.6. Le processus optimal de sélection d'une SVM

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.

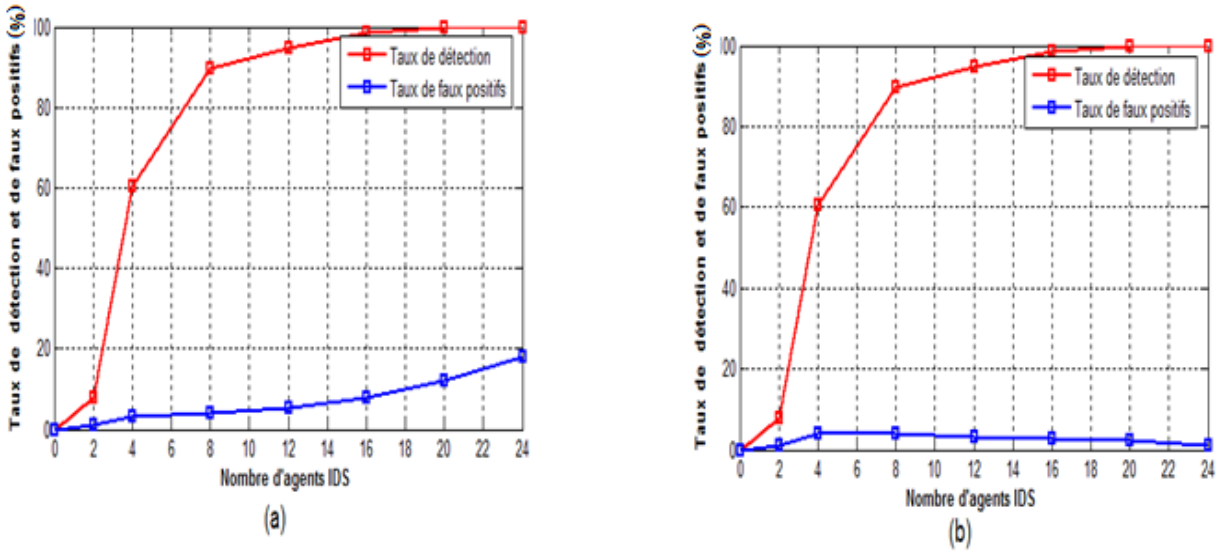


Figure 3.7. Performance du modèle.

(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.

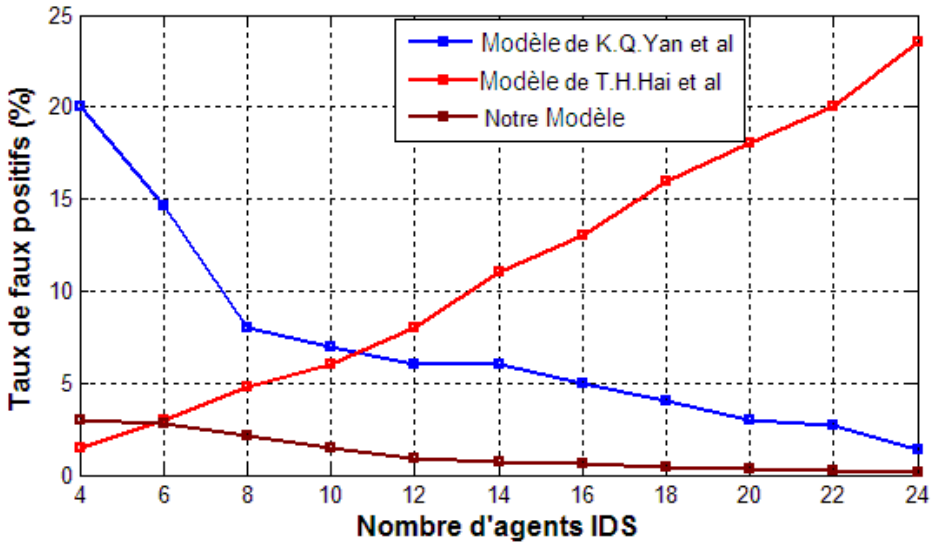


Figure 3.8. Comparaison des taux de faux positifs dans les différents modèles

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.

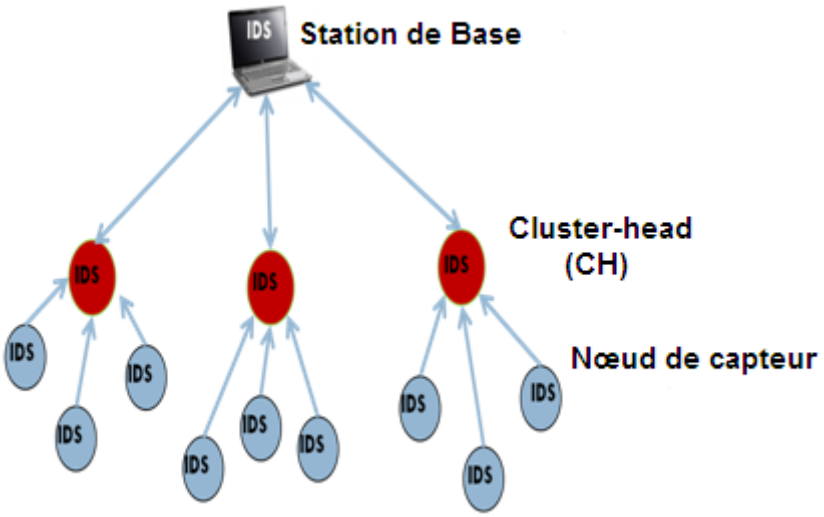


Figure 4.1. Détection hiérarchique d'intrusion

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.*

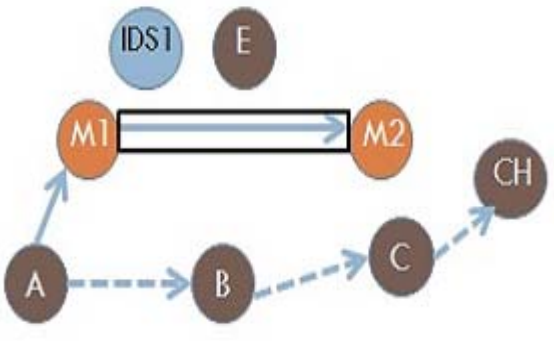


Figure 4.2. Attaque *Wormhole* active

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.

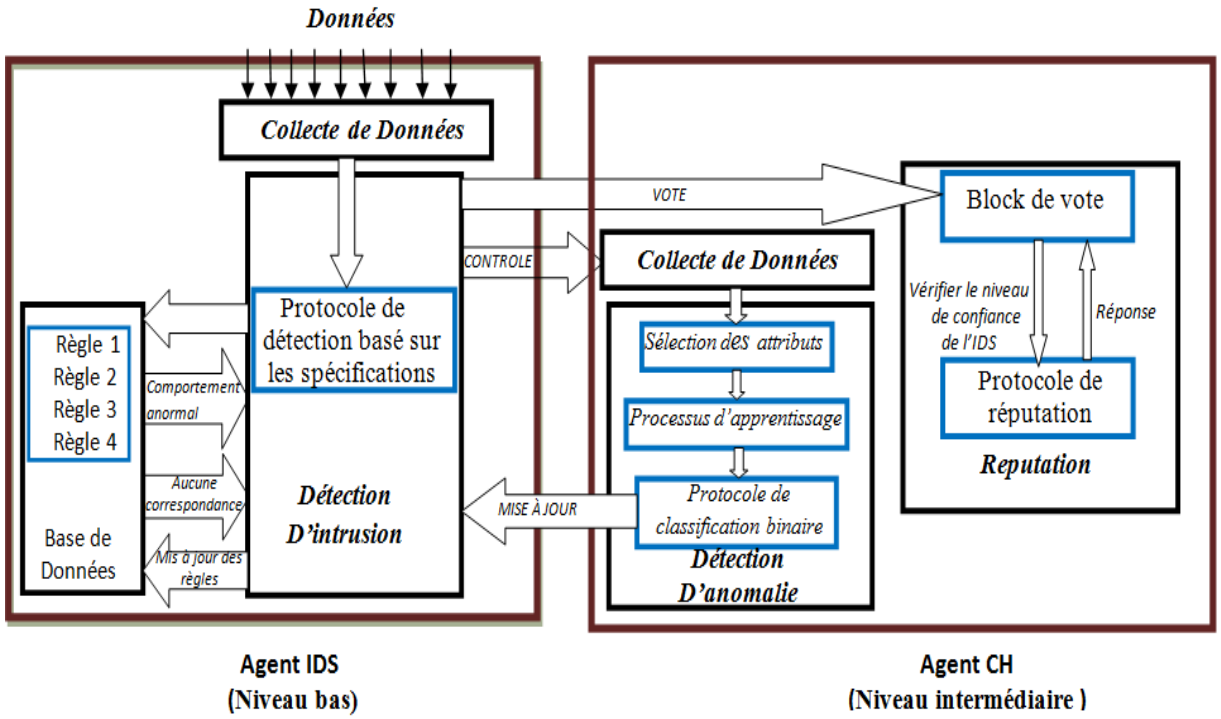


Figure 4.3. Procédé de détection entre les agents IDS et CH

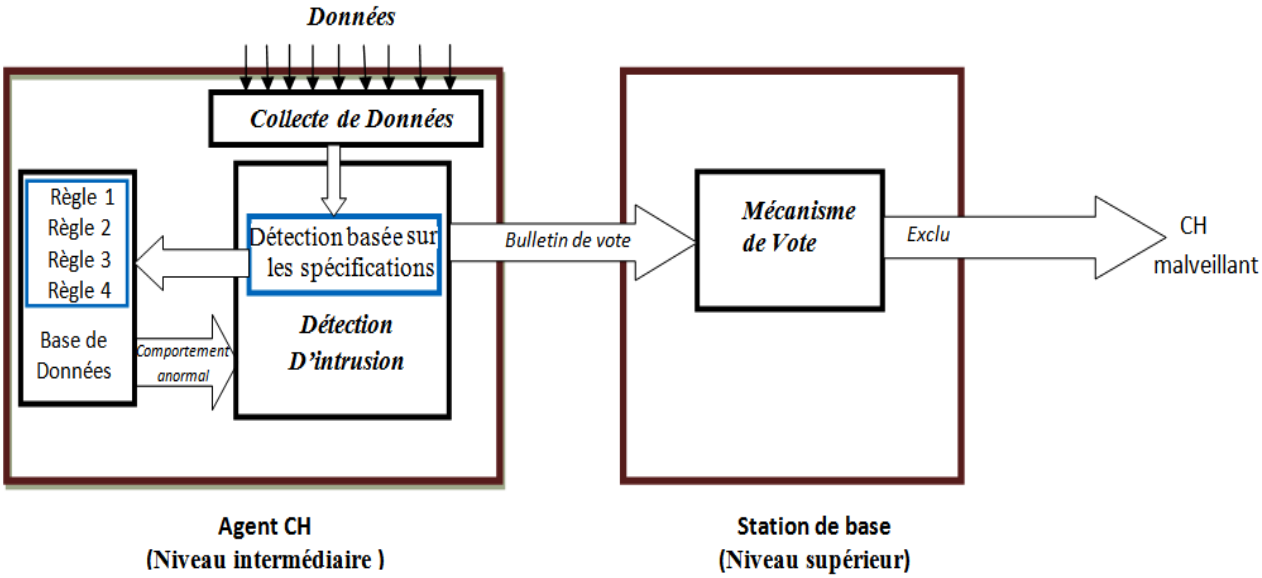


Figure 4.4. Procédé de détection entre l'agent CH et la station de base

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 (δ_{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 (δ_{rssih}). La règle pour détecter l'attaque *Black hole* est définie par le nombre de NPD (supérieur au seuilδ_{bh}) et l'excès de la puissance du signal (supérieur au seuil δ_{rssibh}). Finalement la règle pour détecter l'attaque *Wormholes* est l'excès de la puissance du signal (supérieur au seuil δ_{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 δ_{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.

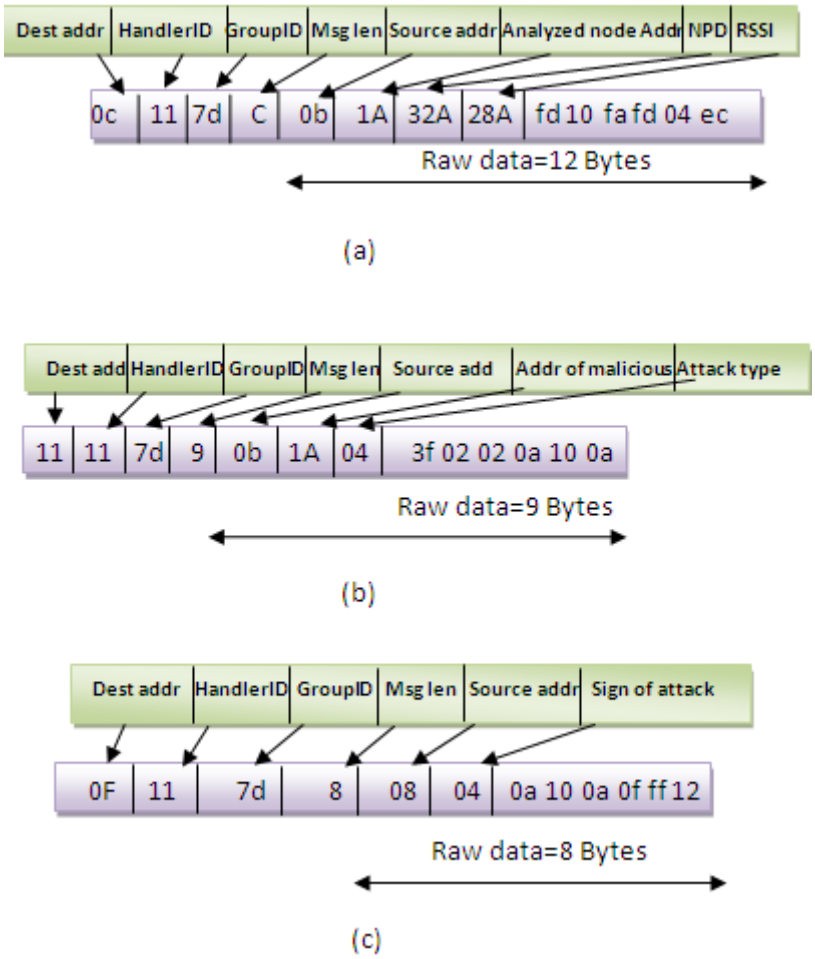


Figure 4.6. Formats de paquet des messages de:
(a) CONTROLE, (b) VOTE, (c) MISE À JOUR

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**], α_i et β_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
δ_{sf}	64 %
δ_{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.

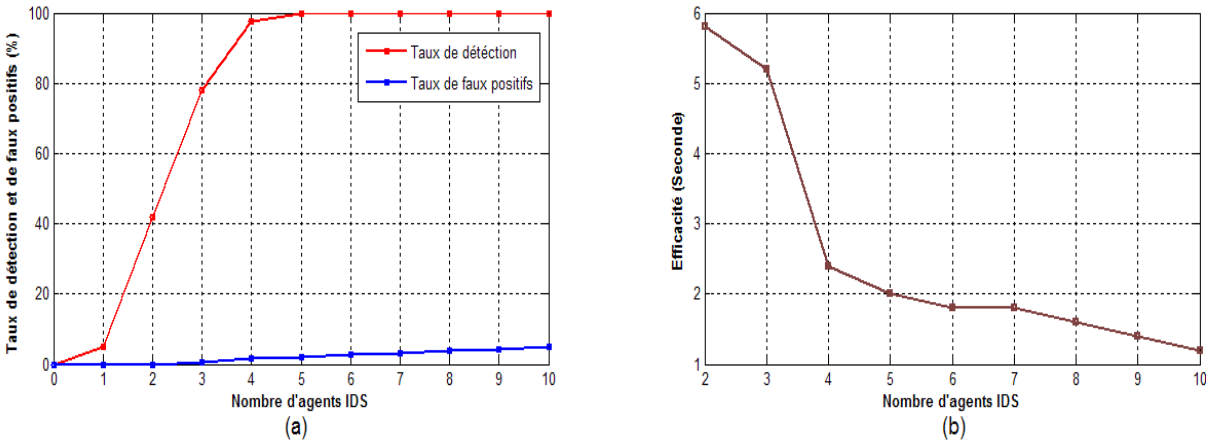


Figure 4.7. Scénario de l'attaque *Hello flood* :
(a)Taux de détection et de faux positifs, (b) Efficacité

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.

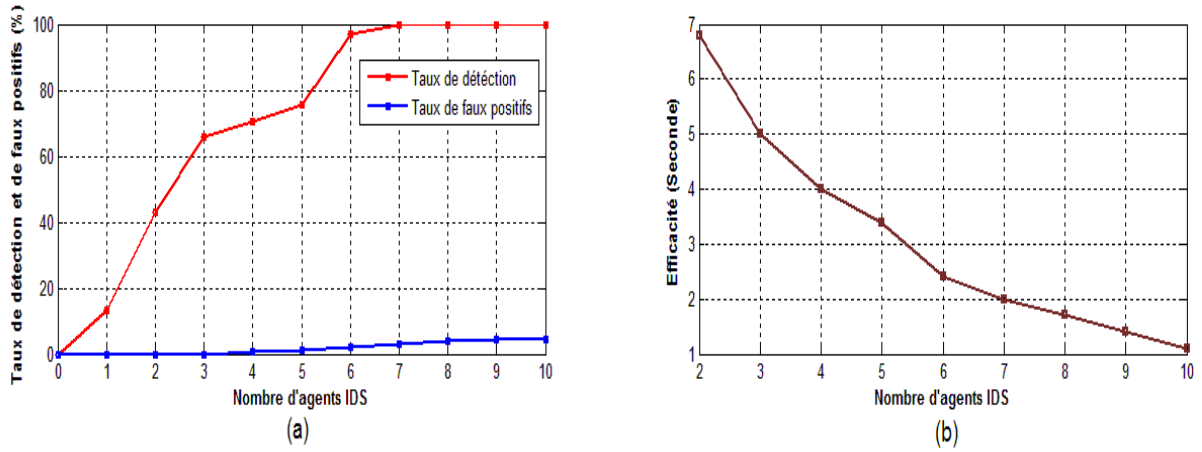


Figure 4.8. Scénario de l'attaque *Selective forwarding :*
(a)Taux de détection et de faux positifs, (b) Efficacité

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.

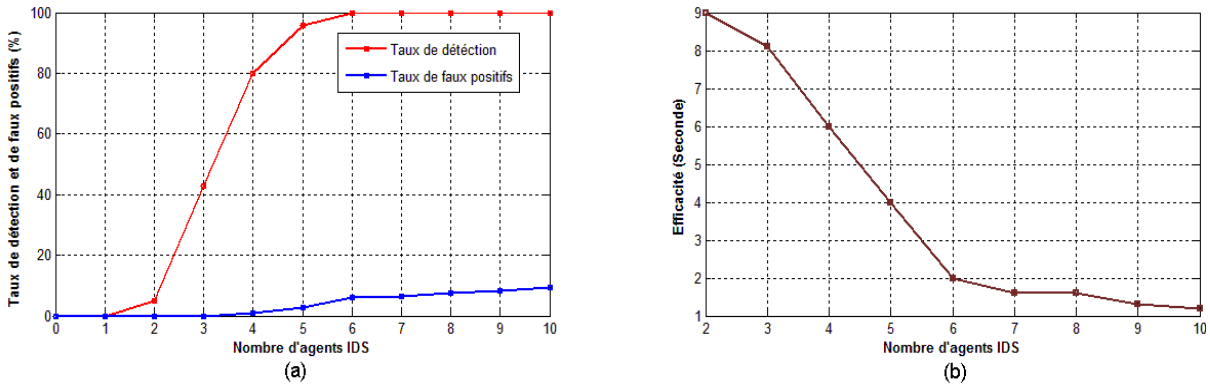


Figure 4.9. Scénario de l'attaque *Black hole :*
(a)Taux de détection et de faux positifs, (b) Efficacité

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.

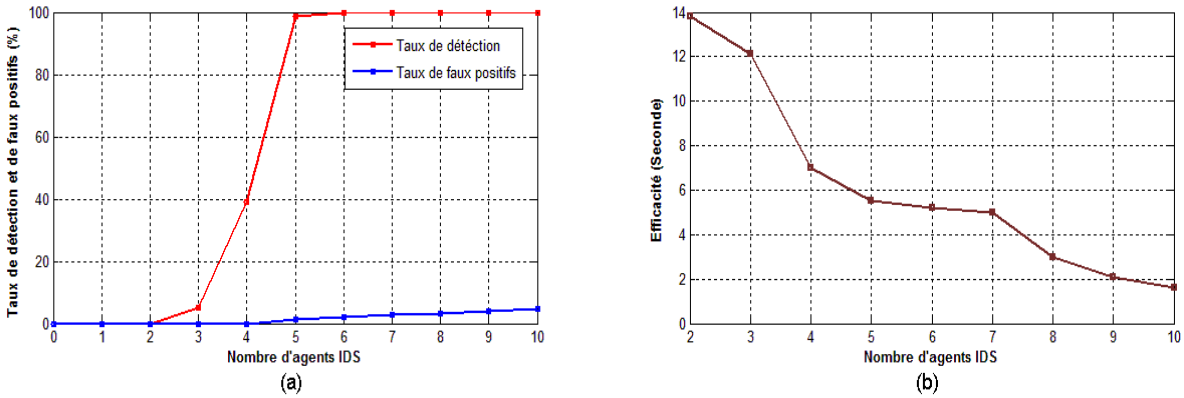


Figure 4.10. Scénario de l'attaque *Wormholes* :
(a)Taux de détection et de faux positifs, (b) Efficacité

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.

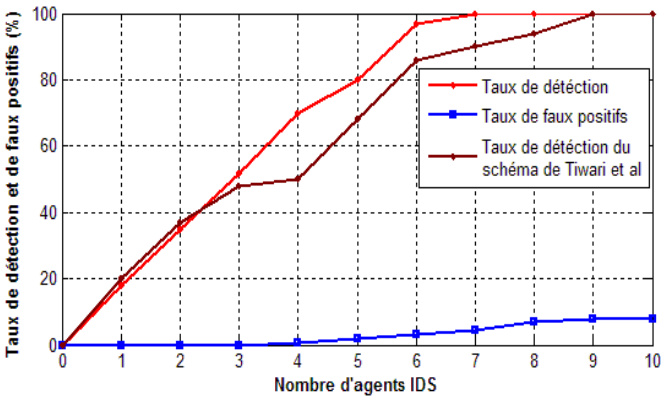


Figure 4.11.Comparaison de notre modèle sous les attaques *Black hole* et *Selective forwarding*

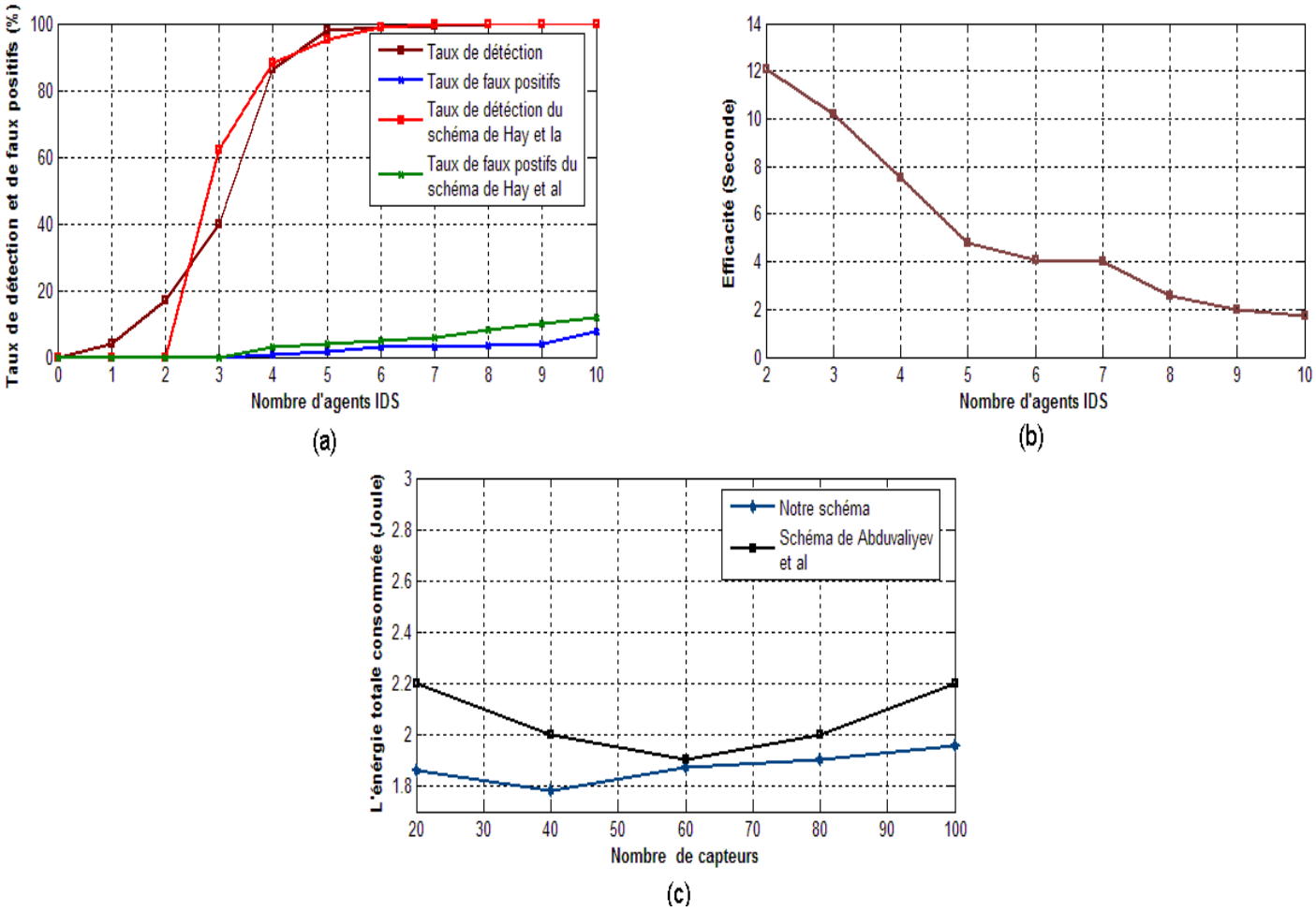


Figure 4.12. Scénario de plusieurs attaques :

(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.

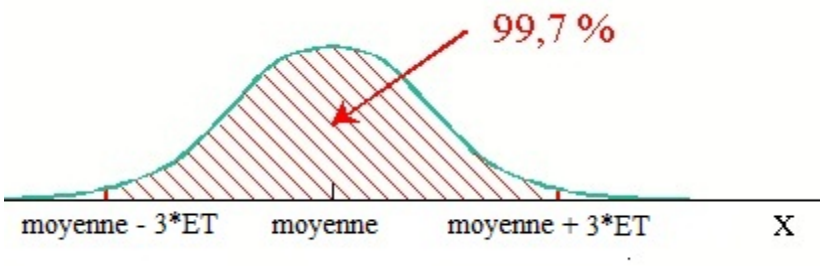


Figure 5.1. La distribution normale

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.

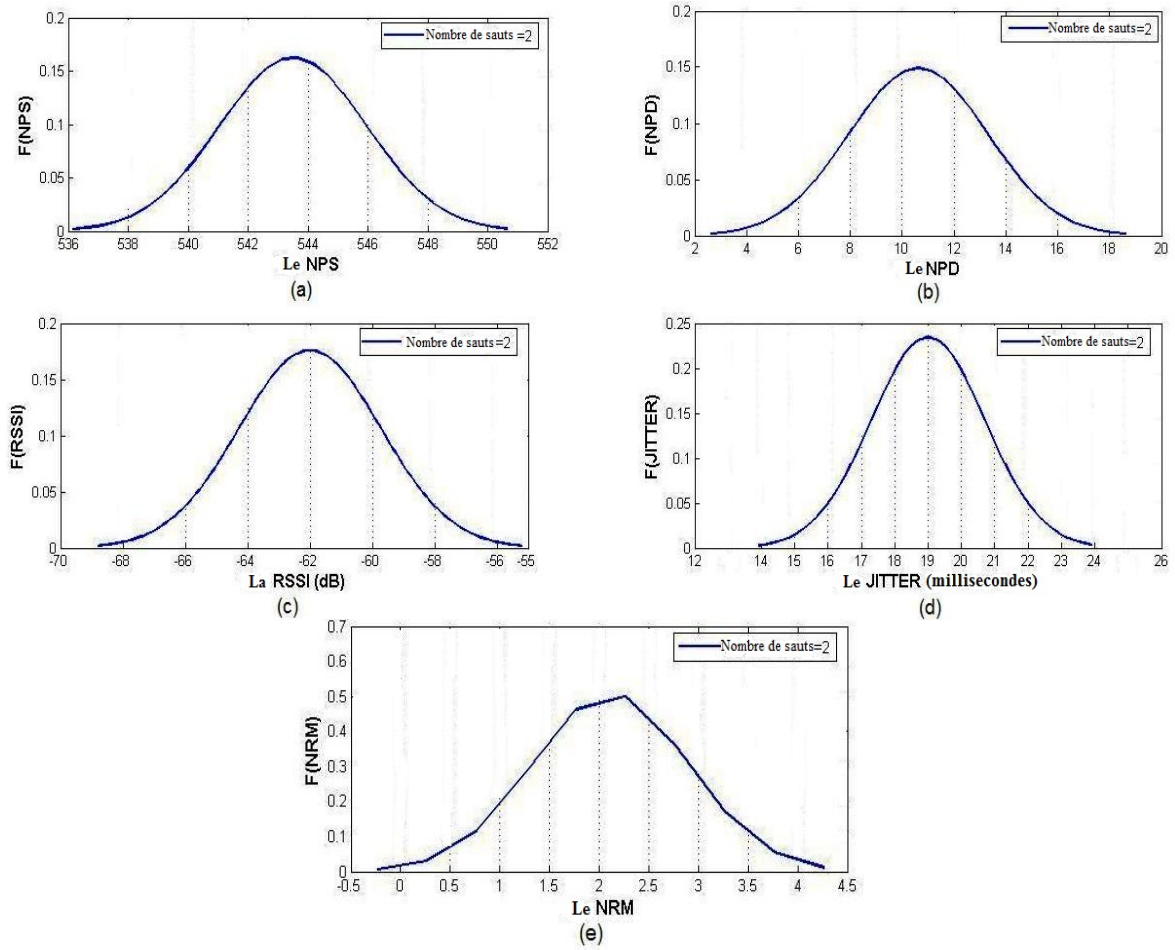


Figure 5.2. La distribution normale des comportements d'un nœud

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.

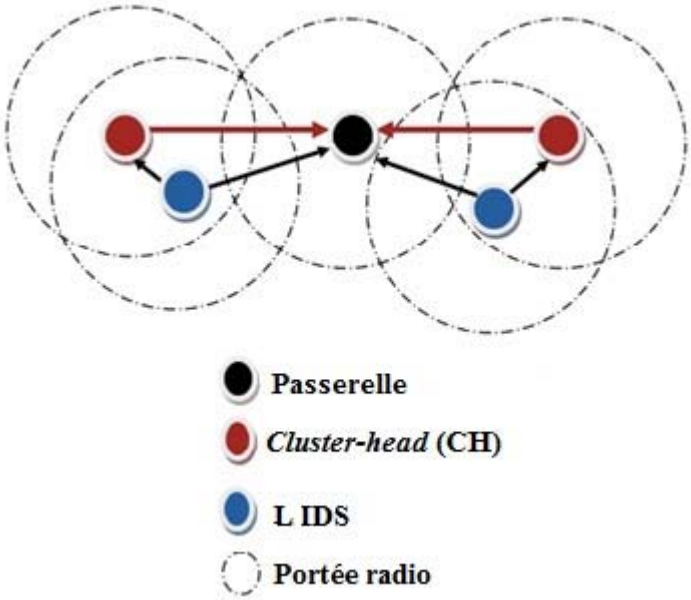


Figure 5.4. Topologie à base de cluster

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.

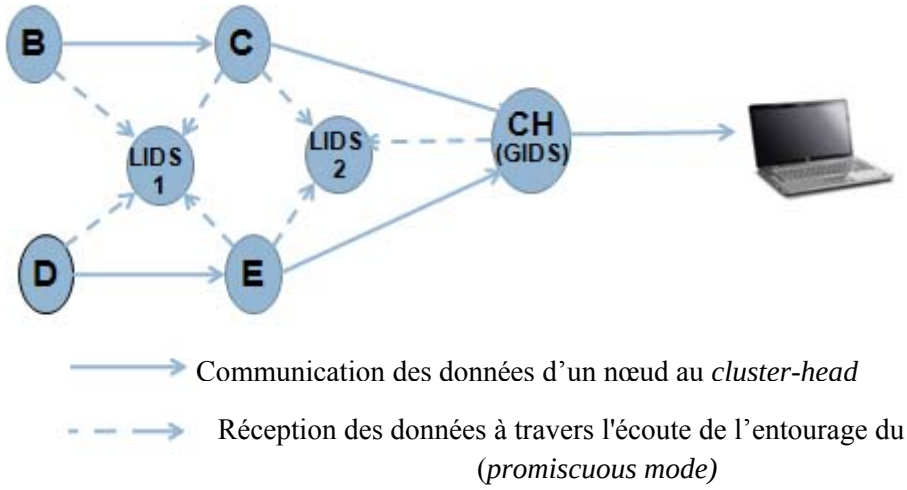


Figure 5.5. Stratégie de l'emplacement des agents LIDS

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.

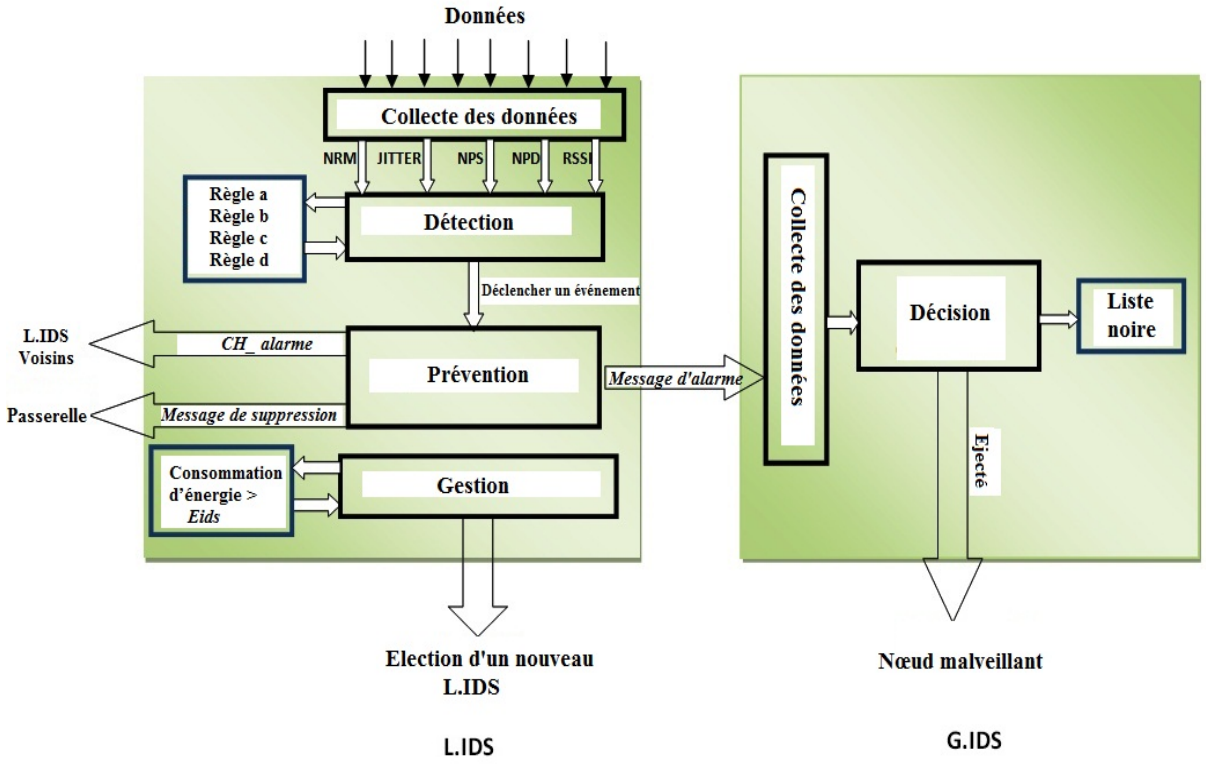


Figure 5.6. Architecture du système de détection d'intrusion par les agents IDS

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 (σ_{nps}, σ_{npd}, σ_{rssi}, σ_{jitter} et σ_{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.

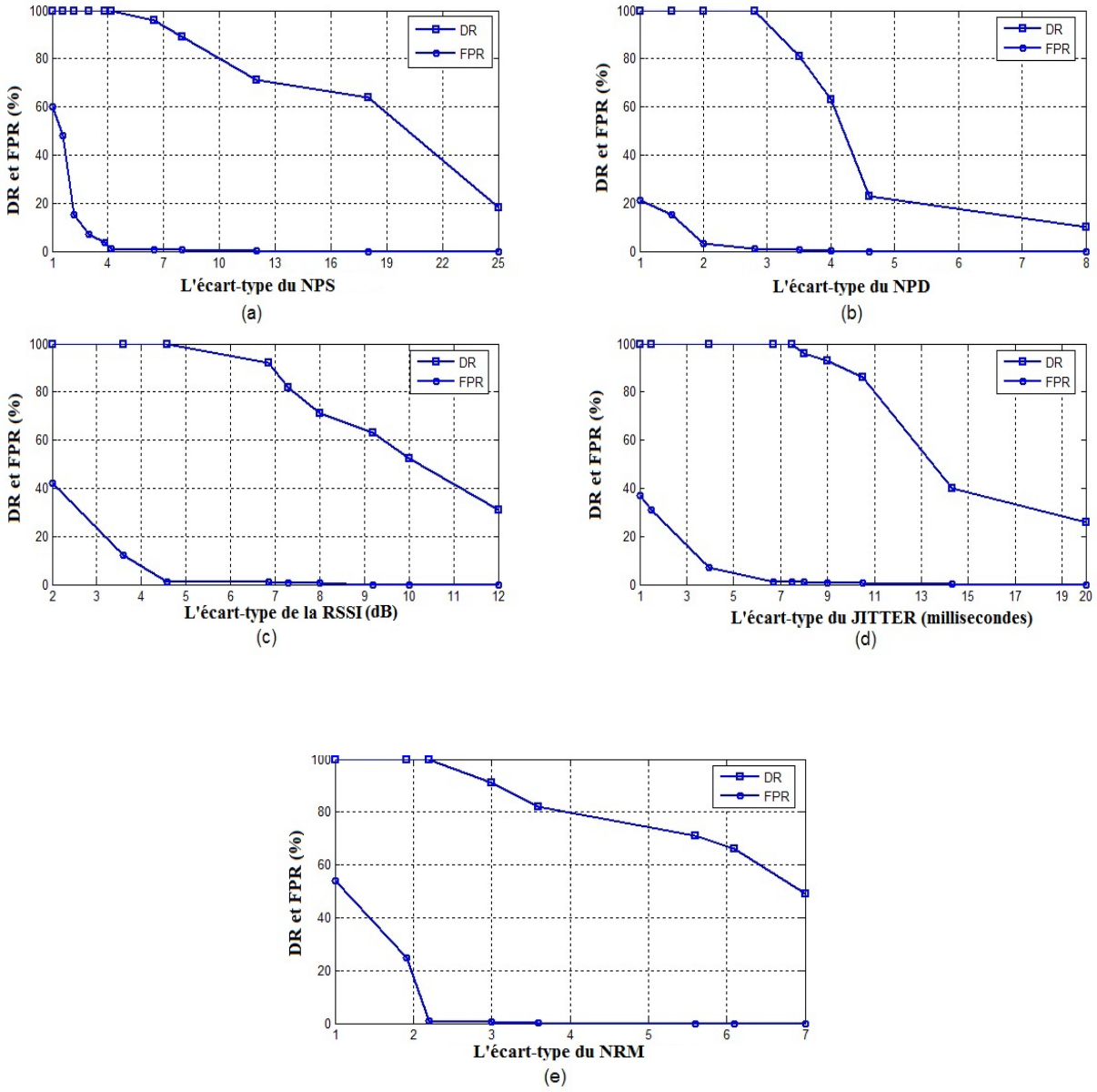


Figure 5.7. Sélection des seuils optimaux de l'écart-type pour:
(a) NPS, (b) NPD, (c) RSSI, (d) JITTER, (e) NRM

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 (σ_{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 γ_{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 γ_{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 (σ_{npd},γ_{sf} et γ_{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 γ_{sh} et γ_{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 γ_{sh}= 9 dBm et γ_{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 (γ_{rj}',γ_{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 $\gamma_{re} = 2{,}6$.

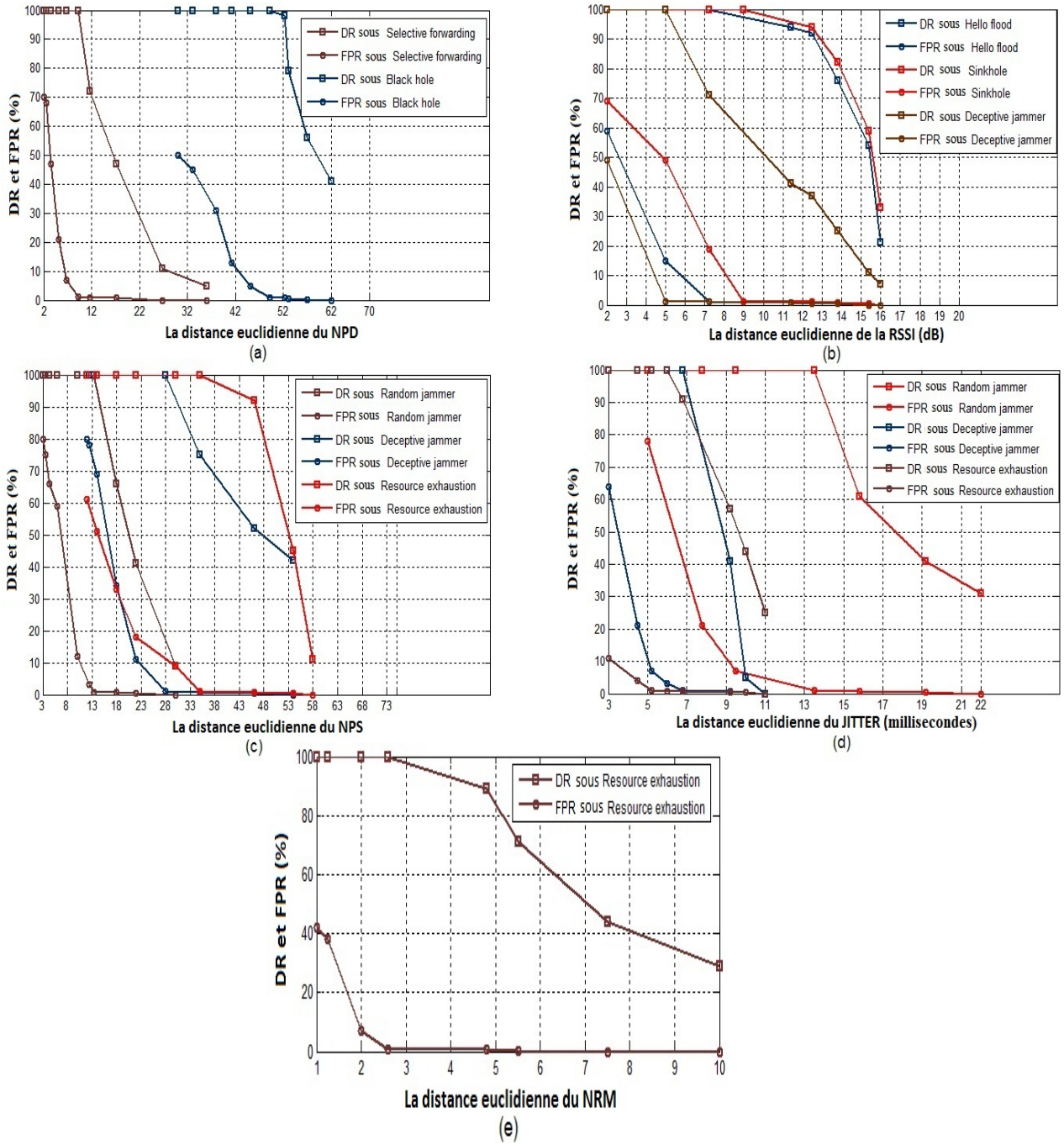


Figure 5.8. Sélection des seuils optimaux de la distance euclidienne pour:

(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.

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

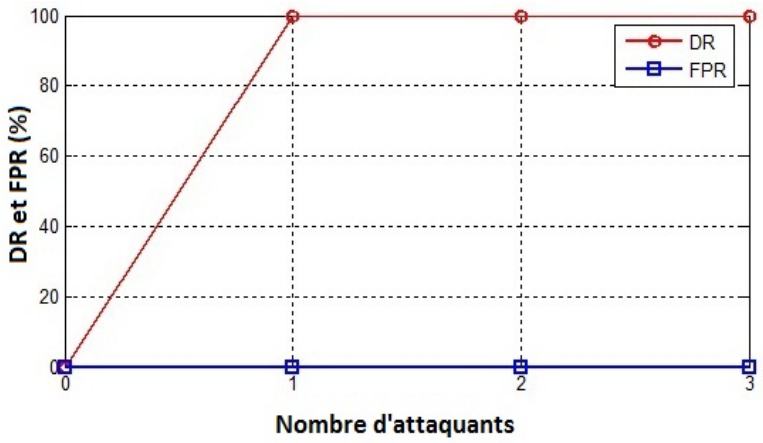
2. Selective forwarding* et *Black hole

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.

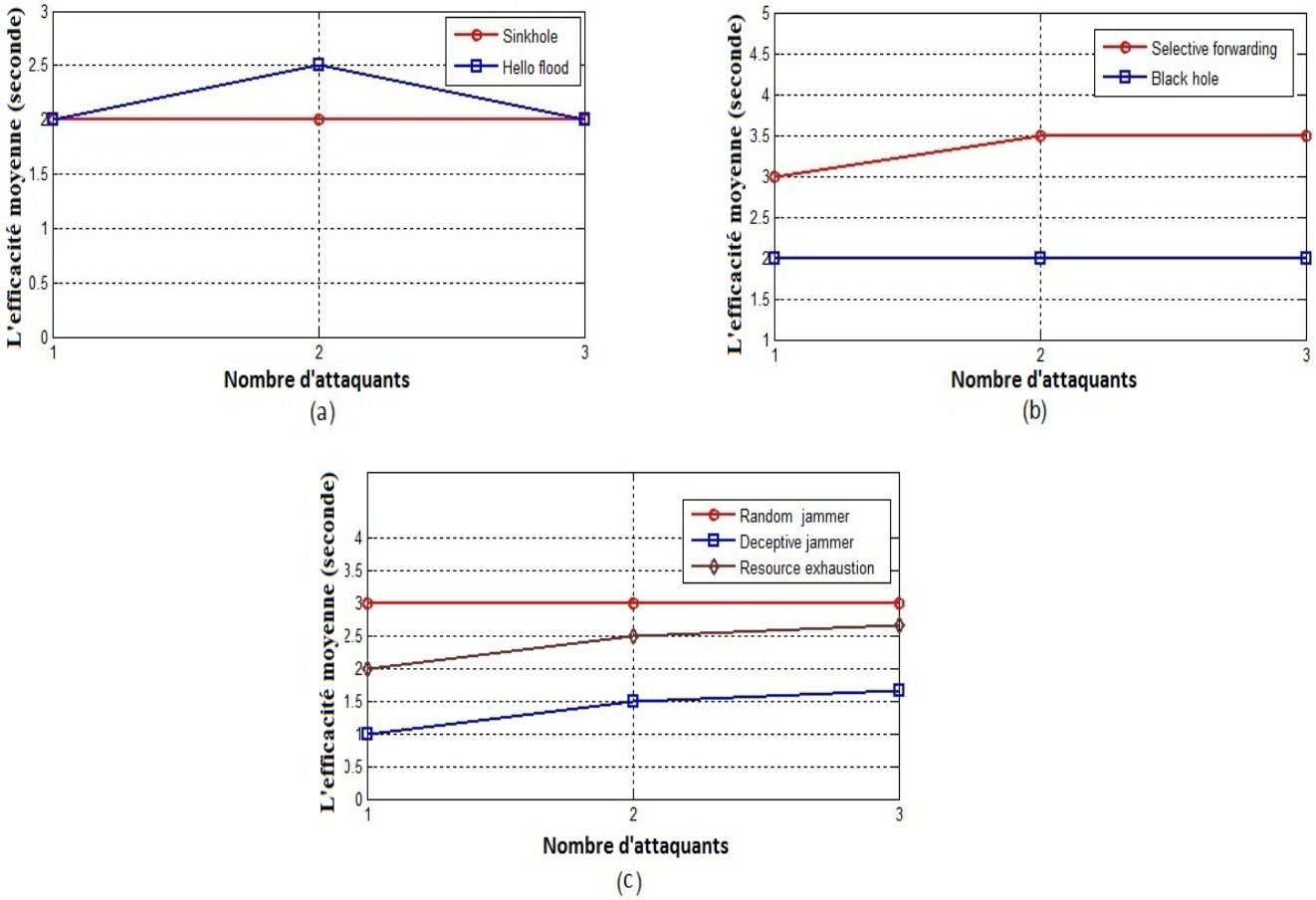
3. Random jammers, Deceptive jammers* et *Resource exhaustion

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

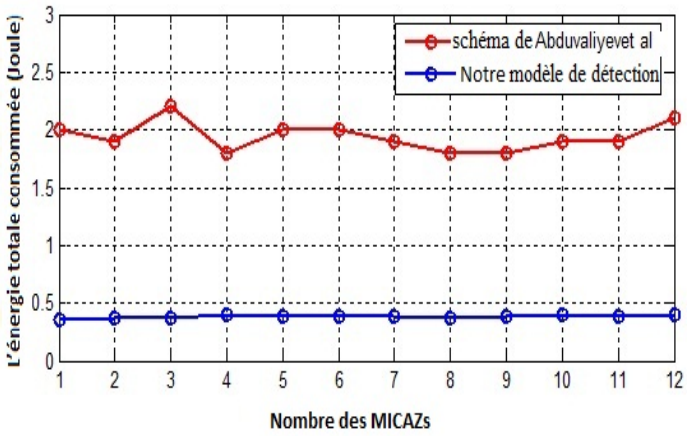


Figures 5.11. Performances expérimentales de détection d'intrusion : (a) Efficacité moyenne sous les attaques *Sinkhole* et *Hello flood*, (b) Efficacité moyenne sous les attaques *Selective forwarding* et *Black hole*, (c) Efficacité moyenne sous les attaques *Random jammer*, *Deceptive jammer* et *Resource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

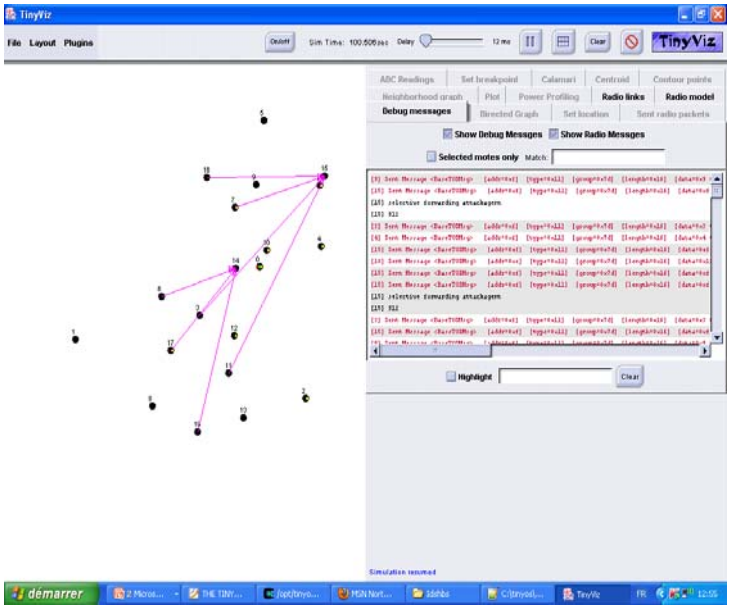


Figure B.1. Fenêtre graphique de TinyViz

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.

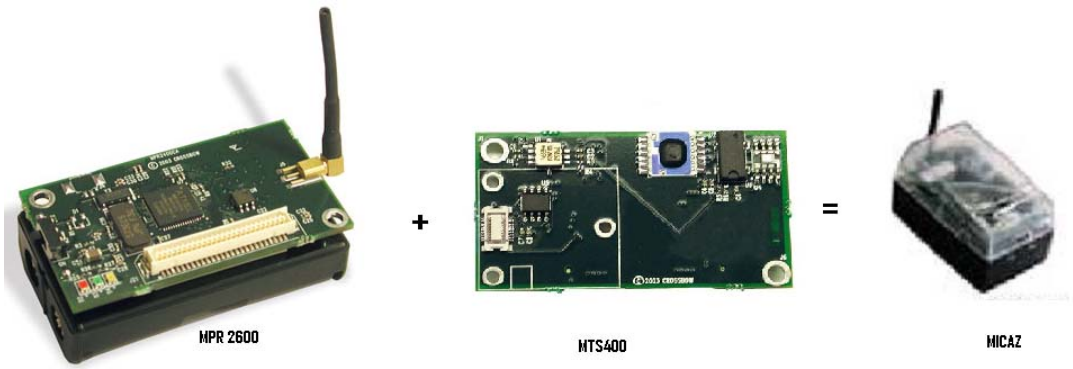


Figure B.4. Capteur MicaZ

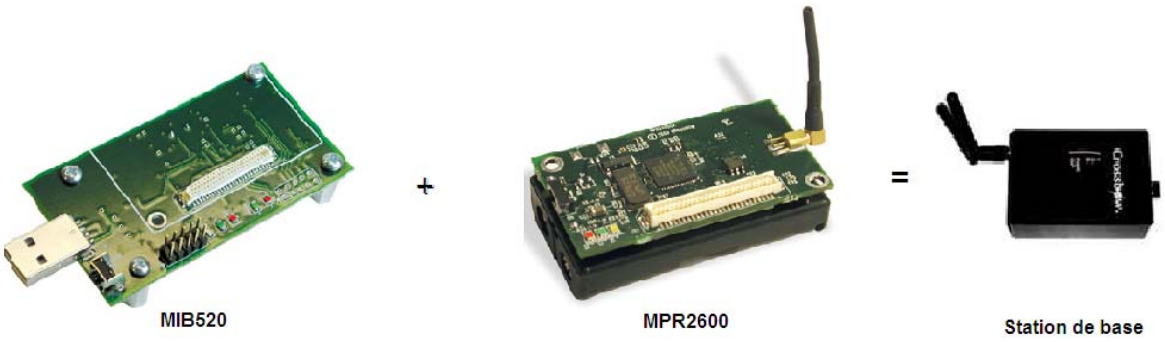


Figure B.5. Station de base

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

Intrusion Detection Framework of Cluster-based Wireless Sensor Network

Hichem Sedjelmaci[#], Sidi Mohammed Senouci[*], Mohammed Feham[#],
[#]Abou Bakr Belkaid University, STIC Lab, Tlemcen, Algeria
{hichem.sedjelmaci, m_feham }@mail.univ-tlemcen.dz
[*]University of Bourgogne, DRIVE Lab
49 Rue Mademoiselle Bourgeois, 58000, Nevers, France
Sidi-Mohammed.Senouci@u-bourgogne.fr

***Abstract*— Wireless sensor networks (WSNs) have a huge potential to be used in critical situations like military and commercial applications. However, these applications are required often to be deployed in hostile environments, where nodes and communication are attractive targets to attackers. This makes WSNs vulnerable to a variety of potential attacks. Due to their characteristics, conventional security mechanisms are not applicable. In this context, we propose an intrusion detection framework for a cluster-based WSN (CWSN) that aims to combine the advantage of anomaly and signature detection which are high detection rate and low false positive, respectively.**

Keywords-Wireless sensor network; Clustering; Intrusion detection system; Anomaly detection; Signature detection.

I. INTRODUCTION

The development of multifunctional sensors with low-cost and low-power has been behind the significant growth of Wireless Sensor Networks (WSNs). These sensors can be very useful for many military and industry applications to collect and process the corresponding information. These applications are required often to be deployed in hostile environments, where nodes and communication are attractive targets to attackers. In this context, many researchers focused on security issues for wireless sensor networks. As a result, two types of techniques have been used: cryptographic techniques and Intrusion Detection Systems (IDS). The first method is used to ensure authentication and data integrity by checking the source of the data and verifying that was not altered. However, the main weakness of this approach is its inability to detect accurately internal attacks when the attacker knows the keys and use them to encrypt and decrypt the communication messages. On the other hand, the IDS is used to protect the network against both internal and external attacks. The task of this method is to analyze a target node and trigger an alarm when suspicious activities occur.

We believe that IDS is useful and more effective to detect any malicious behavior that attempts to launch either internal or external attacks. Roman et al. [1] claim that IDS solutions for ad-hoc networks cannot be applied directly to sensor networks. This is particularly true as WSNs introduce severe resource constraints in data storage and power consumption [2]. As a consequence, the proposed IDS must meet the restrictions of WSNs. In the literature, intrusions detections are classified into two categories as shown in "Fig. 1".

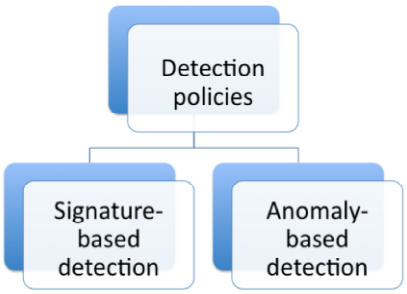


Figure 1. Detection techniques.

•*Anomaly detection:* This technique builds a normal behavior model and compares it to the captured data; any deviation from this model can be seen as an intrusion. The advantage of such technique is its ability to detect new attacks that have never occurred before. However, the main drawback is it results in higher false positive rate. Among anomaly detection techniques proposed in the literature for wireless sensor networks are: Rule Based, Neural Network and Support Vector Machine (SVM) [3].

•*Signature based detection (or misuse detection):* This technique is based on detection of the attack type by comparing a target behavior with a set of fixed rules related to each attack signature. The advantage of such technique is the low false positive rate. Nevertheless, the drawback of this technique is its inability to detect unpublished attacks [4].

Based on current hybrid IDS models used in the literature, our aim in this research is to investigate a new IDS model that combines the advantage of anomaly and signature based detections (high detection rate and low false positive respectively) in order to obtain an efficient detection system to identify malicious nodes. We use a class of machine learning algorithm called support vector machine (SVM) that classifies data into normal and anomalous types. In addition, we use a signature detection technique that is based on known attack patterns (i.e. signatures) to validate the malicious behavior of a target that is identified by the anomaly detection technique. Finally, we concentrate on clustered WSN (CWSN) where the clustering protocol is

used to reduce the energy consumption and enhance the network survivability. This is achieved by designating only one node known as the cluster head which forward a packet (i.e. aggregate data received from cluster members) to the base station instead of all nodes sending their sensed data to remote location (base station).

The outline of this paper is organized as follows: In Section 2, we highlight some background information and some of the related work that was carried out so far in the existing literature. The proposed scheme is presented in Section 3. Finally we conclude this paper by giving some perspectives that we envisage to carry out in the future.

II. BACKGROUND & RELATED WORK

In this section, we first summarize some related works about hybrid intrusion detection systems in CWSN by describing their main advantages and shortcomings. Second, we give some background information on Support Vector Machine (SVM) learning algorithm.

A. Related Work

There are some works that use a combination of a number of detection techniques (i.e. hybrid model) to make use of the best advantages of these methods in order to identify the maximum number of attacks [5], [6] and [7].

The authors in [5] proposed a hybrid approach for IDS in CWSN where the IDS node is embedded in the cluster head (CH). This node is equipped with learning based model, a rule based anomaly detection mechanism and a decision making module. First, the IDS agent gathers incoming packets and analyzes them using a rule-based method (i.e., anomaly detection). If the output of the analyzed packets is found as abnormal, it will then be forwarded to a learning process model. This later uses a supervised learning algorithm called Back Propagation Network (BPN). The abnormal packet delivered by rule based anomaly detection is used as input vectors to BPN where the algorithm trains and classifies the data into five classes (i.e. four types of attacks and one normal behavior). Finally, the decision making module combines the outputs from both, the rule and learning based models to determine if an incoming information is an intrusion or not, and subsequently determines the category of the attack. In the case of presence of an intrusion, the module reports the results to the base station. Simulation results show that the scheme exhibits a higher rate of detection and a lower false positive rate. However, the major drawback of the proposed scheme is that the static nature of the IDS node (run in cluster head) and its vulnerability to attacks by intruders.

In contrast to the previous approach, the authors in [7] propose Dynamic model of IDS (DIDS) for WSN. The IDS node is note active all the time and it moves from one node to another. The authors propose to use a clustering topology where the IDS nodes are placed at both in the cluster heads (i.e. core defense) and at the boundary of the cluster (i.e. boundary defense). In addition, each IDS agent uses the misuse detection and the anomaly detection techniques to analyze and detect any intruder. In this scheme, the dynamic process is defined as follows: if one of the IDS nodes has consumed 30 percent of the overall energy which it has before activating its IDS, the cluster reconfigures and the IDS will be activated in the new nodes and in the new clusters. This approach has a remarkable enhancement in the security and the network lifetime as compared to the static model. Nevertheless, the drawback of this scheme is the need to spend much longer time to detect all intrusions, and specifically for higher number of attackers.

B. Support vector machines

In this section we highlight some background information on SVM and the impact of features selection for detecting a malicious behavior.

SVMs are supervised learning techniques. The aim of a SVM classifier is to determine a set of vector called support vector to construct a hyper-plan in the feature spaces. The SVM algorithm is a maximal margin algorithm. It seeks to place a hyper-plan between classes of points such that the distance between the closest points is maximized [8].

In our context a distributed binary classifier for anomaly detection is performed to detect the abnormal packet. The binary classification SVM provides a decision function:

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

K(x) is the kernel function and α_i are the Lagrange multipliers, which can be been obtained by solving this optimization problem [8]:

$$\left.\begin{array}{l} maximize\ L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{i=1}^{n} \alpha_i\alpha_j y_i y_j K(x_j,x_i) \\ \text{subject to} \quad \sum_{i=1}^{n} y_i\,\alpha_i = 0, and\ 0 \le \alpha_i \le C \text{ for all } 1 \le i \le n \end{array}\right\} \qquad (2)$$

According to the condition of Kuhn-Tucker (KKT), x that corresponding to $\alpha > 0$ are called support vectors.

We use SVM algorithm for anomaly detection because it provides very good results with less training time compared to other supervised classifier (e.g. neural networks). In addition it presents a low generalized error probability [9].

The selection of most relevant features for detecting a set of attacks is an important task to increase the classification accuracy, reduce the false positive and get a fast training time.

III. PROPOSED HYBRID MODEL

The proposed approach uses a combination of the SVM based anomaly detection and the signature detection techniques. The anomaly detection uses a distributed learning algorithm for the SVM training to distinguish between normal and anomalous activities. In addition, a set of fixed rules related to each attack signature are stored at each node. We use a clustering topology that divides the sensor network into a set of clusters, each one having a CH. Finally, each node has the possibility to activate its IDS. However, simultaneous activation is not performed because

it leads to waste the network resources. At each link there must be at least one IDS node that gathers and analyzes the packets within its radio range. In addition, each IDS agent monitors its IDS neighbors based on the fact that even the IDS node could be malicious. These agents receive data through promiscuous listening or by using a multi-hop communication mode as illustrated in "Fig. 2".

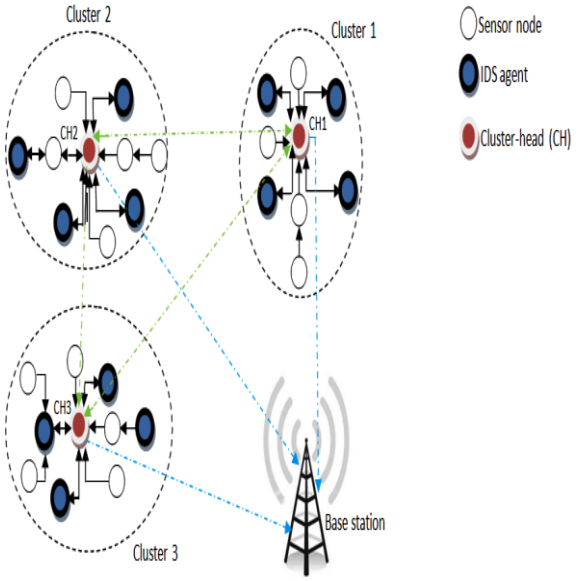


Figure 2. Clustered WSN topology.

A. *Framework of detection agents*

The IDS agent is equipped with an Audit Data System (ADS) and Intrusion Detection Framework (IDF). The cluster head agent is equipped with Collaborative Detection System (CDS). These systems are illustrated in "Fig. 3" and detailed as follow:

1) ADS: IDS nodes gather the packets within their radio range and pass it to the intrusion detection framework.

2) IDF: The intrusion detection uses anomaly and signature detection techniques.

a) Anomaly Detection: The anomaly detection involves: (i) **SVMs training process** where each IDS agent trains the SVM locally, computes a set of data vectors (called support vector), and then sends these vectors to an adjacent agent. Each agent that receives support vector from its neighbors updates its vector by combining the union of received set and its own support vectors. Subsequently, it computes the separating hyperplane (separate the anomalous points from the normal points). This process is continued until a complete pass through all agents within the same cluster. At each cluster, the IDS node that exhibits a less energy consumption is selected to send the support vector to the associated CH. All the CHs exchange then their data and communicate the computed set of support vector to their IDS nodes. As a result, the IDS agents by updating their vectors will have the same set of support vectors and they will be able to classify new captured packets as normal or anomalous. It is also important to note that this algorithm reveals less power consumption since in this case the communication takes place only with a vector support. This is particularly true as this latter is much less in number compared to the input data vector used during the training process, and (ii) **Testing process** where data is classified by the classifiers according to the training model (anomalous and normal patterns). Any deviations from normal profiles (i.e. intrusion) are delivered to the signature detection technique for further investigation.

b) Signature detection technique: A set of attack signatures, that are stored in the node database are compared with intrusion sent by the anomaly detection. If match occurs, the suspected node is removed from the cluster. Otherwise, a collaboration process is raised.

3) CDS: In the collaborative process, a vote mechanism is applied. In the case when there is no matches between the attack detected by the anomaly detection and some predefined signatures of malicious behaviors, the intrusion detection agent sends a message (i.e. report) to the CH. This report contains the suspected node and a set of features (i.e. the radio strength, the number of dropped packets, etc). When the CH receives the intrusion report, it carries out a voting mechanism. If more than half of the votes are in favor of intrusion the CH removes the suspected node from the cluster.

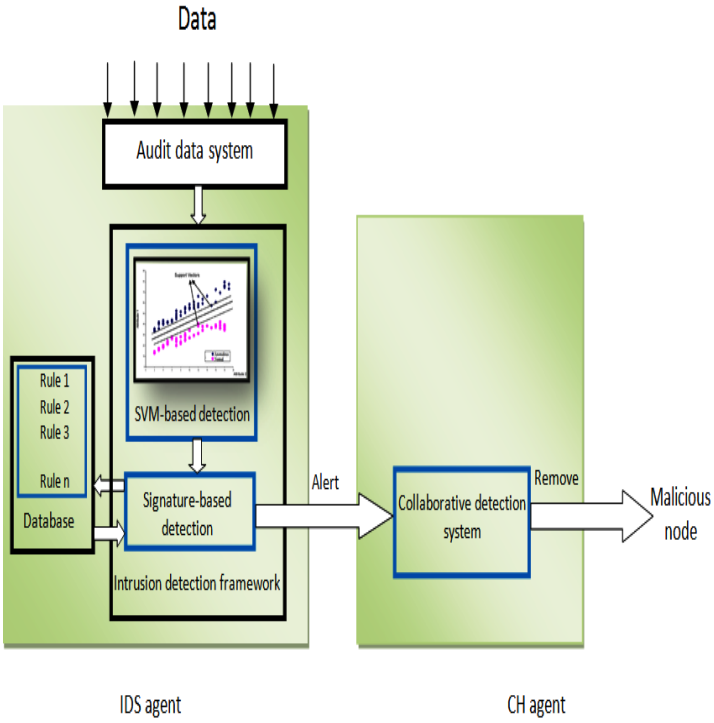


Figure 3. System architecture of intrusion detection framework.

IV. PERFORMANCES EVALUATION

We evaluate the distributed intrusion detection framework in terms of detections and false positives rates (also called false alarm). The former is the rate of correct detected malicious behavior and a later is defined as the number of misclassified connections over the number of normal connections [5]. In 1998 MIT Lincoln Lab and DARPA agency developed a real dataset of connections that represent the anomaly and normal connection's behavior that occurred in a military network environment. This real dataset is named KDD99 [10]. The output of each sample (i.e. connection) is either an anomaly or normal connection. The anomaly behavior is classified into four types of attacks: Dos, Probe, U2r, and R2l (for more detail about these attacks see reference [5]).

A. *Simulation Assumptions*

A large number of simulators have been developed for WSNs in current literature, such as NS2 [11], and TOSSIM [12]. However, due to the fact that all these tools do not use a real data during their simulation processes, we have developed a Java based simulator that uses the KDDCup'99 dataset [10]. This simulator is composed of the following components: packets, IDS agents, sensor node, and an attacker. The packets' components use the kdd99 simples, each node exchanges these simples with its neighbor nodes for routing purposes and hence it leads to simulate the real WSN. We set the key parameters of our simulation as specified in the Table 1.

TABLE1. SIMULATION PARAMETERS

Simulation time	320 Second
Simulation area	$60x50m^2$
Number of nodes	100
Number of clusters	10
Number of IDS nodes	3-24

B. *Results Analysis*

In this section, we evaluate the performances of our intrusion detection framework. First, we evaluate our IDS framework only under SVM based anomaly detection. Second, we combine both techniques (i.e. SVM and signature based detection). In both cases, we study the variations of detection and false positives rates when we increase the number of IDS agents in the network.

As illustrated in "Fig. 4", the detection rate reaches almost 100% when the number of IDS nodes is high (i.e. over 12 agents). At the same time we also notice an increase in the rate of false positives. As a consequence a balance between the number of IDS nodes and false alarms must also be considered.

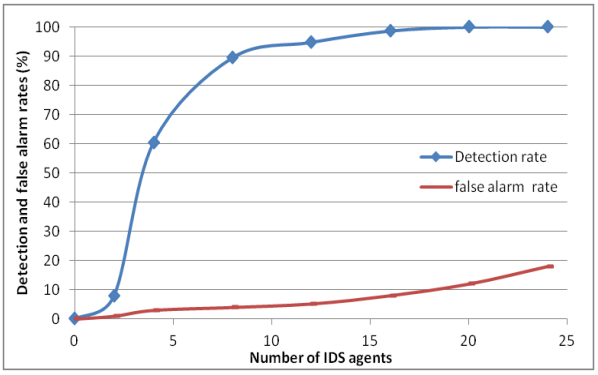


Figure 4. Detection and false positives rates with an SVM based detection.

As illustrated in "Fig. 5", the combination of both SVM and Signature based detections allows our intrusion detection framework to reach a high detection rate with a low number of false positives, specifically when the number of IDS nodes is important (i.e. exceed 12 nodes).

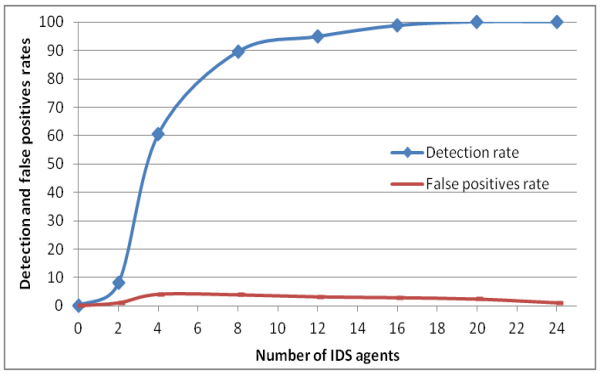


Figure 5. Detection and false positives rates with an SVM&Signature based detection.

V. CONCLUSION AND FUTURE WORK

We proposed a distributed intrusion detection framework for cluster-based WSN. Our detection framework uses a support vector machine (SVM) for anomaly detections. This technique provides a high detection rate but generates considerable number of false alarms. However, in order to mitigate this issue, a signature detection technique is added in our detection framework. In fact, this combination takes advantage of these two techniques and allows a more efficient intrusion detection system that detects any malicious behavior with a low false positive rate. Moreover, we used our framework within a CWSN since clustering aims to reduce the communication cost, which leads to an increase in the network lifetime.

In near future, we attempt to implement our approach under TINYOS sensor nodes [13].

REFERENCES

[1] R. Roman, J. Zhou, and J.Lopez, "Applying Intrusion Detection Systems to Wireless Sensor Networks", *the 3rd IEEE Consumer Communications and Networking Conference*, 2006, pp.640-644.

[2] J. P. Walters, Z.Liang, W.Shi, and V.Chaudhary, "Wireless Sensor Network Security: A Survey", *Security in Distributed Grid and Pervasive Computing*, Auerbach Publications, CRC Press, Vol.1, Issue.2, 2006, pp.1-50.

[3] S. Rajasegarar, C. Leckie, and M. Palaniswami, "Detecting Data Anomalies in Wireless Sensor Networks", *Security in Ad hoc and Sensor Network,* World Scientific Publishing Co, Vol. 3, pp.231-259, 2009.

[4] S. Kaplantzis, "Security Models for Wireless Sensor Networks", *PhD Conversion Report*, Centre of Telecommunications and Information Engineering, Monash University, Australia, 2006.

[5] K. Q. Yan, S. C. Wang, S. S. Wang, and C. W. Liu, "Hybrid Intrusion Detection System for Enhancing the Security of a Cluster-based Wireless Sensor Network", *In Proc. 3rd IEEE International Conference on Computer Science and Information Technology*, Chengdu, China, 2010, pp.114-118.

[6] A. Abduvaliyev, S. Lee, and Y.K Lee, Energy efficient hybrid intrusion detection system for wireless sensor networks, *International Conference on Electronics and Information Engineering, IEEE,* Kyoto,Japan,2010, pp.25-29.

[7] G. Huo, and X. Wang, "A Dynamic Model of Intrusion Detection System in Wireless Sensor Networks", *In Proc. International Conference on Information and Automation, IEEE*, Zhangjiajie, China, 2008, pp.374-378.

[8] J. Gama, and R.U. Pedersen, "Predictive Learning in Sensor Networks", *In learning from data streams, Springer*, 2007, pp. 143-164.

[9] 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, 2003, pp.209-216.

[10] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[11] http://www.isi.edu/nsnam/ns/

[12] http://www.cs.berkeley.edu/~pal/research/tossim.html

[13] http://www.tinyos.net/

NOVEL HYBRID INTRUSION DETECTION SYSTEM FOR CLUSTERED WIRELESS SENSOR NETWORK

Hichem Sedjelmaci[1] and Mohamed Feham[1]

STIC Lab, Department of telecommunications, Abou Bakr Belkaid University, Tlemcen, Algeria
massy2008@hotmail.fr, m_feham@mail.univ-tlemcen.dz

ABSTRACT

Wireless sensor network (WSN) is regularly deployed in unattended and hostile environments. The WSN is vulnerable to security threats and susceptible to physical capture. Thus, it is necessary to use effective mechanisms to protect the network. It is widely known, that the intrusion detection is one of the most efficient security mechanisms to protect the network against malicious attacks or unauthorized access. In this paper, we propose a hybrid intrusion detection system for clustered WSN. Our intrusion framework uses a combination between the Anomaly Detection based on support vector machine (SVM) and the Misuse Detection. Experiments results show that most of routing attacks can be detected with low false alarm.

KEYWORDS

Wireless Sensor Network, Hybrid Intrusion Detection System, Support Vector Machine (SVM), Classification Accuracy, False alarm

1. INTRODUCTION

Recent advances that deal with the technology of micro-electronics and wireless communication have enabled the development of multifunctional sensor with low-cost and low-power. These sensor nodes consist of data processing, wireless communication and capture device.

A wireless sensor network (WSN) consists of a large number of devices operating independently and communicating with each other via short-range radio transmissions. These sensors can be very useful for many military and civilian applications, as collecting and processing information from hostile environment and difficult access locations such as battlefield surveillance, environment monitoring, etc. Many researchers have focused on the security of WSN against attacks or malicious behaviors since that the characteristics of both wireless infrastructure and WSNs can cause the potential risk of attacks on the network [1]. The security mechanisms used to protect the wireless sensor network against intruders are:

Cryptographic techniques: They are used to ensure authentication and data integrity by checking the source of the data and to verify that is not altered. The cryptographic operations are based on primitives such as hash functions, symmetric encryption and public key cryptography [2] which can protect WSN against external attacks. However cryptographic techniques cannot detect internal attacks when the attacker knows the keys and uses them to perform encryption/decryption. This technique is defined as the first line of defence.

Steganography: If cryptography is the art of secrecy, steganography is the art of concealment. The main objective of steganography is hiding or embedding a message either in another one or into a multimedia data (image, sound, etc). However this technique requires significant processing resources and it is hard to implement it in WSN because of the constraints of these sensors.

DOI : 10.5121/ijnsa.2011.3401

IDS: An intrusion detection system is usually considered as the second line of defence, it can protect with high accuracy against internal attacks. This mechanism allows detecting abnormal or suspicious activities on the analyzed target and triggers an alarm when intrusion occurs. As far as cryptography is concerned it cannot provide the necessary security in WSN, that why we believe strongly that IDS is useful for both internal and external attacks. Many researches in the application of the IDS' technology in ad hoc networks were done, in comparison with wireless sensor networks where few subjects were investigated because of its limited energy and computing storage capacity.

Our aim is to introduce in the following research a novel hybrid intrusion detection system for WSN. The approach uses the clustering algorithm to reduce the amount of information and decrease the consumption of energy. In addition we have used a class of machine learning algorithm called support vector machine (SVM), that separates data into normal and anomalous (binary classification), in order to detect anomalies. We have also applied misuse detection technique to determine known attack patterns (signatures). Therefore, the combination of both techniques can achieve high detection rate with low false positive and false negative rate. Finally, we have developed a mechanism of cooperation among IDS agents that work with each other, that mechanism can make a better decision in order to verify if a node is compromised or not which might determine novel sign of intrusion.

2. RELATED WORK

Sensor networks introduce severe resource constraints due to their lack of data storage and power [3], according to Roman et al. [4] IDS solutions for ad hoc networks cannot be applied directly to sensor networks. Therefore, the proposed intrusion detection system must meet the demands and restriction of WSNs.

Kumar [5] classified intrusions detection techniques into two categories:

• Misuse detection: this technique involves the comparison between captured data and known attack signatures, any corresponding pattern can be considered as an intrusion. Updating the signature over time is necessary to keep this technique effective. However, the major drawbacks of misuse detection systems are their inability to detect unpublished attacks [6].

• Anomaly detection: is based on modeling the normal behavior of the nodes and compares the captured data with this model, any activities that deviate from this model can be seen as an anomaly. The advantage of such technique is that it can detect attacks that are unpublished [6]. On the contrary this technique requires a considerable computation time which implies high energy consumption. Therefore, the anomaly detection algorithm in WSN must take into consideration a Trade-off between detection accuracy and energy consumption. Among anomaly detection techniques proposed in literature for wireless sensor networks are: Rule Based, K-Nearest Neighbor and support vector machine (SVM), etc [7].

2.1 Anomaly detection based on SVM

There have been currently limited researches on the use of SVM classifier in WSN. Kaplantzis et al. [8] worked on centralized intrusion detection system based on support vector machine to detect selective forwarding and black hole attacks. IDS that are run in the base station use one-class SVM in training collected nodes' data for only normal network activity, by using bandwidth and hops count as feature vector (no attacks activity). Subsequently, the attacks are introduced in the network, and SVM use the training set to detect the attacks. In the simulations, the authors claim that the IDS can detect with high accuracy black hole and selective forwarding attacks. However, this scheme can detect only two kinds of attacks, and presents low detection rate for the selective forwarding attack when its number is small in the network. Besides, the base station

cannot manage the large number of packets sent by the nodes which consequently cannot be analyzed by the SVM.

Centralized SVM training method allows a better separation of the classes with the misclassification error rate close to zero [9]. However, it exhibits a high communication overhead, and it is not suitable for resource-constrained sensor networks. That is why many authors mentioned that the distributed SVM training fit the requirement of sensor nodes in terms of energy cost ([9], [10], [11], [12]) and provide closer classification as centralized approach. In [12], two distributed algorithms for training SVM in WSN are proposed. For both algorithms the SVM classifiers are run in each node to compute a set of data vector. For the first algorithm, it is called Support Vectors (that lie closet to the separating lines), but for the second one the amount of information is much larger, it represents the vectors that lie in the convex hulls boundary of each of the two classes (normal, anomalous). Each node communicates its data vector with one-hop neighbor, once this process done, the final hyperplane is computed, and all the nodes have the same discriminant plan to separate data into two classes and can classifier any new measures. In the simulations the second algorithm exhibits a better classification than the first algorithm but with additional power consumption.

2.2 Hybrid intrusion detection system integrated for clustered sensor networks

There are some researches that use a combination between anomaly detection and misuse detection (hybrid model) in order to leverage the advantages of these two techniques and try to detect a significant number of attacks. In the literature there are some hybrid intrusion detection systems for WSN such as [13], [14] and [15].

In [13], Hai et al. proposed in WSN cluster based and hybrid approach for IDS. Based on work undertaken by Roman et al. [4], they suggest that IDS agents are located in every node. The agent is divided into two modules: local IDS agent and global IDS agent. Because of energy and memory constraints of WSN, global agents are active only at a subset of nodes. For anomaly detection, the global agent IDS monitors the communication of its neighbors by using predefined rules with two-hop neighbor knowledge, then sends alarm to cluster-head (CH) when they detect malicious nodes. Each node has an internal malicious database, which contains a list of known attack patterns (signatures) computed and generated in the CH.
The authors attempt to minimize the number of nodes where the global agents IDSs are deployed by evaluating their trustworthy based on trust priority. In order to reduce the collisions and avoid the waste of energy, they propose an over-hearing mechanism that reduces the sending message alerts. When the rate of collision and the number of malicious node is not very high the proposed scheme can detect the routing attacks such as selective forwarding, sinkhole, hello flood and wormhole attacks with a better energy saving. Nevertheless, the drawback of this scheme is the high rate of false positive that is generated when using the rule based-approach of anomaly detection. In addition, this method is well defined by experts and specialists in the area of wireless security by being dependent on manual rule updating.

Yan et al. [14] have focused on using clustering approach in WSN and embedded hybrid IDS in CH. the proposed IDS have three modules: misuse detection module, anomaly detection module and decision making module.
In the anomaly detection module, the rule-based method has been used to analyze incoming packets and categorize the packet as normal or abnormal. For building misuse detection model, the supervised learning algorithm Back Propagation Network (BPN) is adopted. The abnormal packets, which are detected by anomaly detection model is used as input vectors of BPN. The algorithm trains this training dataset, then classifies the data into five classes (four types of attacks and one normal behavior), when the process of training is over, it integrates the model in the misuse detection module in order to classify the new data (testing dataset).

Finally the output of both models (anomaly detection and misuse detection models) is used as an input for the decision making module. The rule-based method is applied to determine if an incoming information is an intrusion or not, and determine the category of attack. In case of presence of an intrusion the module reports the results to the base station. The simulation results show a higher rate of detection and a lower false positive rate, but the major drawback of the proposed scheme is that IDS monitor run in a fixed cluster heads (the hot point). Therefore it's an attractive node for the intruder that uses all its capacity to attack this node. Another drawback is the number of features which is very important (twenty four features are used). Thus the cluster head consumes much more energy, which leads to minimize the life time of the node, also the names of this features are not mentioned.

3. ROUTING ATTACKS IN WSNS

A large variety of attacks against WSNs exist in the literature; therefore in this section we specify two categories of attacks: Dos attack and Probe attack.

Select Forwarding and black holes use illegitimate data forwarding to make attack, so they are classified as Dos attack [14].

Spoofed, Altered or Replayed Routing Information, Wormholes and Acknowledgment Spoofing make a probe step before they begin to attack, so they are classified as Probe attack [14].

• Selective forwarding: In a selective forwarding attack, the intruders prevent the forwarding of certain packets by dropping them. They can also forward a received packet along a wrong path, thus creating unfaithful routing information in the network [1].

• Black holes: In this attack, the intruder pretends to be as shortest path to the base station or cluster head (CH) by using a higher power transmission. The WSN are vulnerable of this kind of attacks because of their communication paradigm, where all nodes carry data to the single node, in our case, the CH.

• Wormholes: The attacker tunnels packets received at one location in the network (in our case, the cluster) to another location, where the packets are then replayed. khalil et al. propose five modes of wormhole attacks in WSN, the detailed of these modes are defined in [16].

• Spoofed, Altered or Replayed Routing Information: The attacker monitors transmissions, intercepts packets, then altering or repelling traffic, this attack can also lead to create routing loops in networks [6].

Acknowledgment Spoofing: In this attack, the intruder convincing the sender that a weak link is strong or that a dead node is alive [6]. This result in packets sent along such link or node are lost.

4. SVM FOR ANOMALY DETECTION

In this section a brief description of SVMs and feature selection are presented

4.1 Support vector machines

Support vector machines are a set of supervised learning techniques used for regression and classification .The aim of SVM classifier is to determine a set of vector called support vector to construct a hyperplan in the feature spaces.

There are several researches that use SVM based on multi-class for traditional network to classify a data into n-classes, but this approach don't meet the requirement of sensors network and remains as an open research question. In our context a distributed binary classifier (anomaly or normal) for anomaly detection is performed to detect the abnormal packet.

Given the training datasets , $(x_i, y_i)\ i = 1, \dots n\ , y_i \in \{-1, +1\}, x_i \in R^d$, we want to find the hyperplane that have a maximum margin:

$$w.x = b$$

Where w is a normal vector and the parameter b is offset.

In order to find the optimal hyperplane, we must solve the following convex optimization problem:

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

$\sum_i^n \varepsilon_i$ relax the constraints on the learning vectors, and C is a constant that controls the tradeoff between number of misclassifications and the margin maximization.

The Eq. (1) can be deal by using the Lagrange multiplier [17]:

$$\left.\begin{aligned} &maximize\ L(\alpha) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K(x_j, x_i) \\ &\text{subject to}\quad \sum_{i=1}^{n} y_i\,\alpha_i = 0, and\ 0 \leq \alpha_i \leq C \text{ for all } 1 \leq i \leq n \end{aligned}\right\} \quad (2)$$

Here $K(x_j, x_i)$ is the kernel function and α_i are the Lagrange multipliers. According to the condition of Kuhn-Tucker (KKT), the x_is that corresponding to $\alpha_i > 0$ are called support vectors (SVs).

Once the solution to Eq. (2) is found, we can get [17]:

$$w = \sum_{i=1}^{n}\alpha_i\, y_i x_i \quad (3)$$

Thus the decision function can be written as:

$$f(x\ , \alpha, b) = \{\pm 1\} = sgn\left(\sum_{i=1}^{n} y_i\,\alpha_i k(x, x_i) + b\right) \quad (4)$$

We choose SVM classifier for anomaly detection because it's provide very good results with less training time compared to neural networks. In addition, it is more suitable for intrusion detection in case where new signature is detected. Another advantage of SVM is the low expected probability of generalization errors [18].

4.2 Feature selection

Feature selection is an important factor to increase the classification accuracy, reduce the false positive and get a fast training time. In this research, the feature selection method proposed by Sung et al. [18] is adopted. Thus the most relevant features are selected.

5. PROPOSED FRAMEWORK AND ITS WORKING

The novelty of our approach is using hybridization between anomaly detection based on SVM and misuse detection, in order to achieve a more accurate intrusion detection system. The anomaly detection uses a distributed learning algorithm for the training of a SVM to solve the two-class problem (distinguish between normal and anomalous activities). In addition, we use a hierarchical topology that divide the sensor network into clusters, each one having a cluster head (CH). The objective of this architecture is to save the energy that allows the network life time prolongation. Among the Cluster-based routing protocols founded in the literature are: LEACH [19], PEGASIS [20], HEED [21]. At last, each node has the possibility to activate its IDS. Activating every node as an IDS wastes energy. So minimization of number of nodes to run intrusion detection is necessary [22]. We defined N as the average number of IDS nodes for each individual link that is expressed by the following equation [16]:

$$N=1.6r^2d$$

Where d is network density and r is the communication range.

Each IDS monitors the neighbor nodes with no trust between each pair of agents (i.e. IDS also monitor its IDS neighbor) as illustrated in Figure 1.

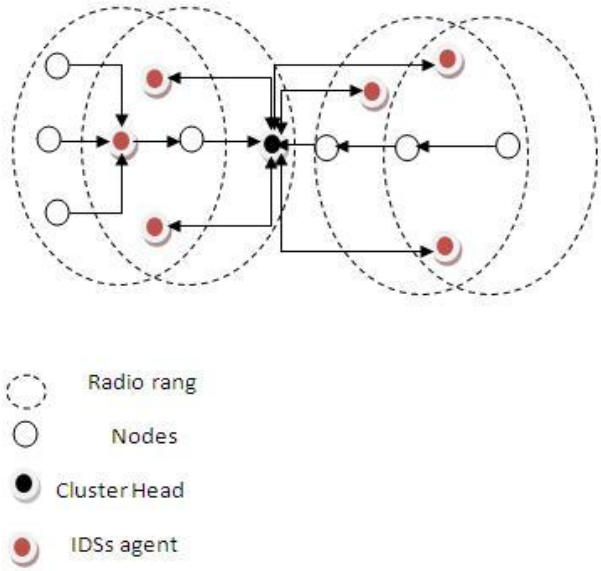


Figure 1.Placement strategy of IDSs node illustration

In our model, we assume that the sensor nodes are stationary and cluster head has more energy compared to the other ones. In the training phase, each IDS node receives the data (support vector) from the nearby IDS nodes by keeping its radio in a promiscuous mode or through multi-hop communication mode (the cluster head can act as a relay). We also make the assumption that the communication activity in this case is supposed to be secured (the algorithm is detailed bellow). In the end, we embed the selected training model into hybrid intrusion detection module (HIDMs) in order to obtain a lightweight and accurate detection system. The selected model is chosen according to the processes illustrated in Figure 2.

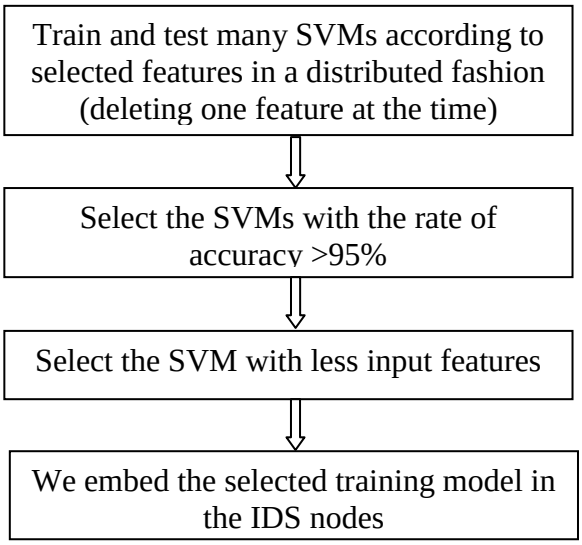


Figure 2.Optimal distributed SVM selection process

5.1 The IDS agent architecture

The proposed intrusion detection system (Figure 4) comprises three modules, which are detailed as follow:

5.1.1 Data Collection Module (DCM)

Due to broadcast nature of wireless networks, monitor nodes gather the packets within their radio range [13] and pass it to the Hybrid Intrusion Detection Module.

5.1.2 Hybrid Intrusion Detection Module (HIDM)

The hybrid intrusion detection module involves anomaly and misuse detection techniques.

A. SVM-Based Anomaly Detection Engine

The anomaly detection procedure is divided into two stages:

Stage1: The training process. Each IDS agent trains the SVM locally, then computes a set of data vectors called support vector (these set of data vectors are less in number than the input data vector used during the learning process). These later will be sent to an adjacent IDS node that is situated in the same cluster. Each monitor node that receives support vector from their IDS neighbors or cluster Head makes a combination between the union of received set and its own support vectors. These monitors update their support vector and compute the separating hyperplane. Afterward, they transmit the resulted set of support vector to the nearby IDS nodes. This process is continued until all IDS agents in the same cluster reach the same trained SVM (a complete pass through all IDSs within the same cluster). The communication activity within the cluster between IDS nodes are depicted in Figure 3. For each cluster, the selected IDS agent that

depends on its residual energy, sends its support vector to the concerned cluster head; then, all the cluster heads exchange their data and communicate the computed set of support vector to their IDS nodes. Finally, when they all compute the global vector support the result is the same, after that, they can classify new captured packets as normal or anomalous. This algorithm reveals little communication overhead and less power consumption since the communication is performed only with a vector support rather than the whole data as in the case of the centralized approach. In addition, in order to save the energy, each IDS node sends back different values of support vector from the ones sent before.

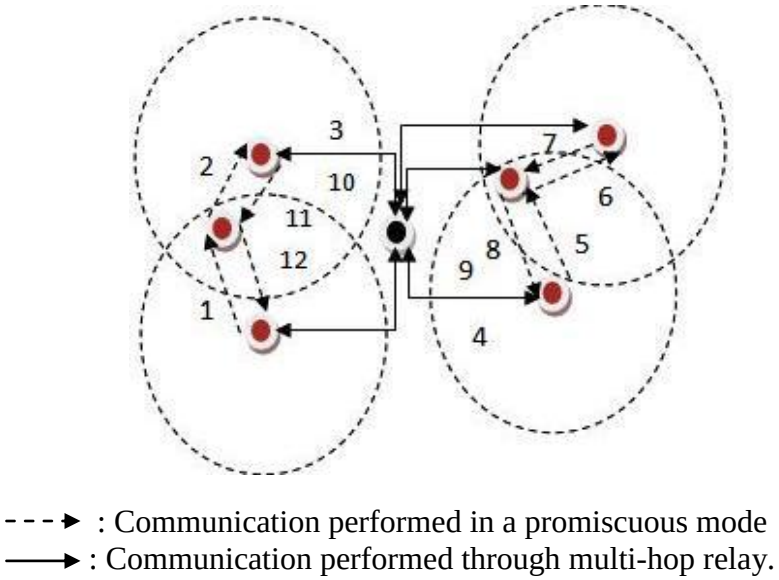


Figure 3.Communication of support vector between IDS nodes

Stage2: SVM testing process. When the process of training is over each IDS node classifies the new data according to normal and anomaly patterns.

The selected training model (described above) is used for anomaly detection engine to classify the captured data that are delivered from data collection module. Any deviations from normal pattern are considered as an intrusion and delivered to misuse detection engine for further detection.

B. Signature Based Detection Engine:

When misuse detection engine receives the intrusion report (the suspected node, a set of features) from anomaly detection engine, it uses some predefined signs of intrusion that are stored in the signature database to check the occurrence of intrusion. If match occurs, the IDS node sends an alarm to cluster head that the analyzed node is an intruder. The cluster head removes the compromised node from the cluster and inform its IDS agents and all CHs over the network about the malicious node. If no match occurs, the process of cooperation is launched. Note that we stored at all nodes in the network a predefined rule about a set of intrusion signature.

5.1.3 Cooperative Detection Module (CDM)

If there are no matches between the intrusion detected by anomaly detection engine and some predefined signatures of attacker, the IDS agent sends the intrusion report to cluster head. That node performs a voting mechanism to make a better decision about the suspect nodes. If more than half of IDS nodes within the same cluster claim that the analyzed target is an attacker, the cluster head isolates the suspect nodes from the cluster and compute a new rule regarding the novel intrusion, then sends an alert message (that include a malicious node and novel intrusion

signature) to their IDS agents and all CHs over the network. When the IDS agents receive this message they update their signature database.

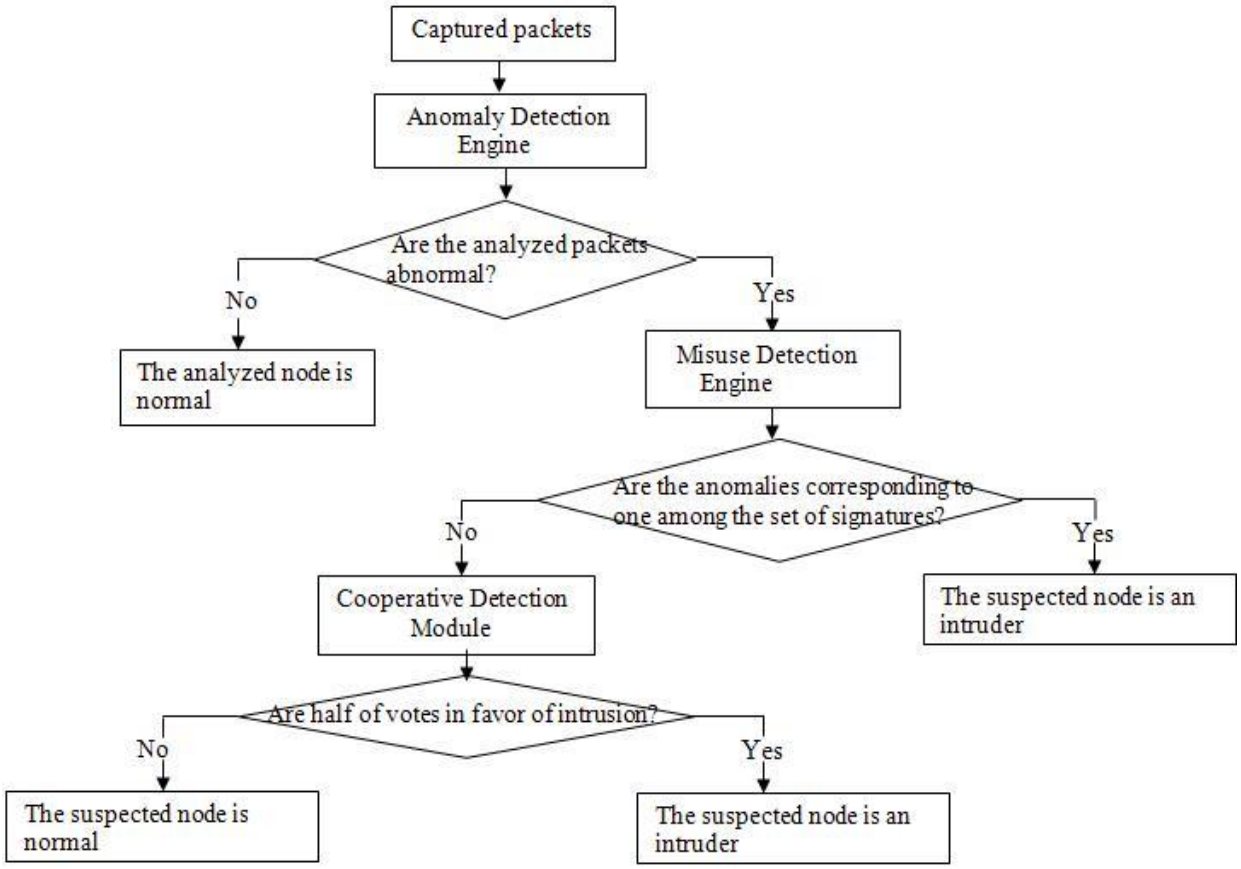


Figure 4. The flowchart of IDS framework in WSN

5.2 Dynamic process for intrusion detection system

In the suggested approach, if (3/4) of IDS nodes within the cluster have consumed more than 50% of their energy; new IDSs are elected and receive the actual set of intrusion signature from the cluster head. The older ones are designated as ordinary nodes. Note that the new IDSs election depends on the residual energy and the placement strategies suggested by Khalil et al. [16]. This mechanism helps to avoid depletion of energy nodes, thus prolonging the network lifetime. When new IDS nodes are elected, they compute locally the support vector and the distributed algorithm for training SVMs is performed as alluded above.

6. EXPERIMENTS

In this section we evaluate the performance of the proposed hybrid IDSs. In our experiments, we have used the KDDcup'99 dataset [23] as the sample to verify the efficient of the distributed anomaly detection algorithm and valid it by compare with a centralized SVM-based classifier, which achieve a high level of accuracy detection. Also, we compare the distributed hybrid IDSs with one proposed by Yan et al. [14] and Hai et al. [13] according to the false positive rate (false alarm) in order to determine the effectiveness of our scheme.

6.1 Dataset

The KDD 99 intrusion detection dataset is developed by MIT Lincoln Lab in1998, each connection in the dataset has 41 features and it's categorized into five classes: normal and four attack behaviors (Dos, Probe, U2r, R2l).

Our analysis is performed on the "10% KDD" intrusion detection benchmark by using its samples as training and testing dataset. We focus only on two categories of attacks (Dos and Probe attacks), which are defined as anomalies behavior and are classified as (-1). The normal behavior is classified as (+1).

The training data used at each IDS comprises of 50 normal and 50 anomalous samples (include both Dos and Probe attacks). In order to evaluate the proposed algorithm, the amount of the data used in test process is equal to N*60, where N is the number of IDS nodes in the network, and the amount of both anomalous and normal samples is equal respectively to 42% and 68% of all test data. The test will perform at one among the IDSs, because all IDSs have the same trained SVM classifier.

6.2 Experiments results and discussion

The radial basis function (RBF) is used as the kernel function:

$$F_{RBF} = \exp\left(-\|x_1 - x_2\|/2.\sigma^2\right), \text{where } 1/2.\sigma^2 > 0$$

The accuracy measure is used as performance metric to evaluate our algorithm. We also compute the detection rate, that represents the percentage of correctly detected intrusions, and false positive, that represents the percentage of normal connections that are incorrectly classified as anomalous.

The identification of the most relevant features is an important task, in our scenario we try to determine SVMs-based anomaly detection that achieve high classification accuracy by deleting the useless features. This task is performed by delete one feature at time according to the approach proposed by Sung et al. [18].The increased number of features led a High computational cost in the nodes, for that our aims is to obtain the SVM classifier with less number of features but able to provide high rate of accuracy, in order to save the memory storage and energy consuming in the sensor nodes. The results of the distributed SVMs binary classifier related to the most relevant features with N=18 are summarized in Table 1.

Table 1: The performance evaluation of distributed IDSs based on SVM

Number of Features	Accuracy (%)	Detection Rate (%)
9	**97.80**	**93.66**
7	**98.47**	**95.61**
5	**96.95**	**91.21**
4	**98.39**	**95.37**

From Table 1, we find out that, the binary SVM classifier with 7 features outperforms the SVMs that use (9, 5, 4) features, respectively, in terms of accuracy and detection rate. Thus these 7 features represent the most significant features. However, the difference of accuracy between both SVM with 7 and 4 features is small, and due to the resource constraints at sensor nodes, we use SVM with 4 features for anomaly detection engine. These features are:

Src_bytes: Number of bytes sent from source to destination

Dst_bytes: Number of bytes sent from destination to source
Count: Number of connection to same destination host
Srv_diff_host_rate: the percentage of connections to different host

The centralized IDS based on SVM (IDS located in the base station) exhibits high performance for solving a problem of 2-class, but this approach requires all the data to be provided by each sensor. Thus it's consuming much energy. The proposed algorithm is compared to centralized approach in term of classification accuracy by using the selected features (Src_bytes, Dst_bytes, Count, Srv_diff_host_rate). This is illustrated in figure 5, where N is the number of IDSs and sensor nodes for both distributed and centralized approaches respectively.

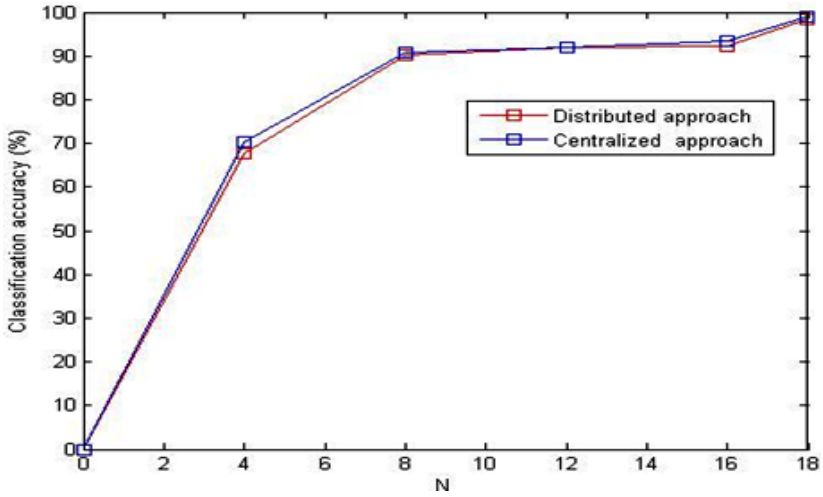


Figure 5.Classification accuracy of SVM for centralized and distributed approaches

As shown in Figure 5, the curves for both approaches coincide almost exactly, and the rate of classification accuracy for centralized and distributed approaches increases when the number of IDSs and sensor nodes increase respectively. Specifically when the number is important (in our case N=18) the rate is close to 100%, 99.07% for centralized approach and 98.39 % for distributed approach. As a result distributed IDSs based on SVM deliver highly-accurate performance, with less training data.

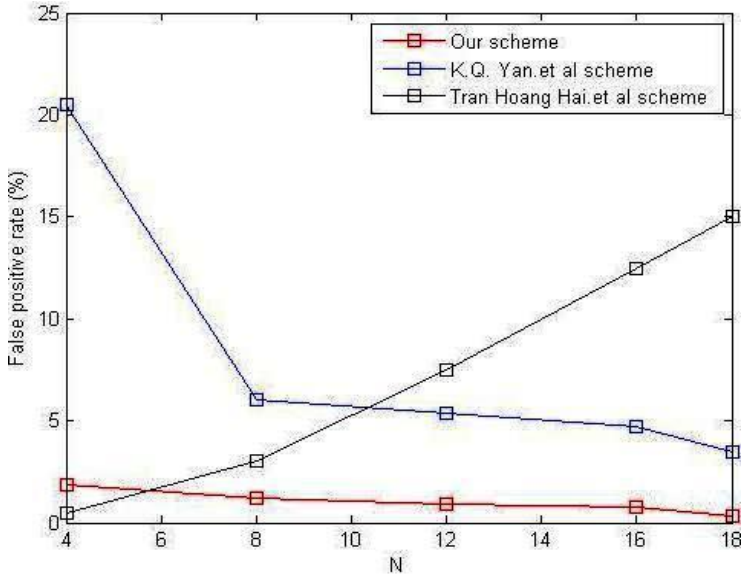


Figure 6.The comparison of the false alarm of different schemes

Increasing the number of IDS nodes for our scheme or sensor nodes (in case of Yan et al. [14] scheme) results in a decrease in the rate of false positive. However, in Hai et al. [13] scheme the number of false positive is important when the number of IDS nodes increase (due to the number of collusion).As shown in Figure 6, our approach exhibits a low false alarm compared to the other schemes, specifically when N=18 (it's equal to 0.3%). However, the distributed hybrid intrusion detection system proposed in this paper achieves a better effectiveness in terms of a low number of false alarms.

7. CONCLUSION AND FUTURE WORK

In this paper, we proposed a distributed hybrid intrusion detection system (HIDSs) for clustered wireless sensor networks. The proposed distributed learning algorithm for the training of SVM in WSN reaches high accuracy for detecting the normal and anomalous behavior (accuracy rate over 98%). Also a combination between the SVM classifier and Signature Based Detection achieve a high detection rate with low false positive rate.

Communication in WSN consumes a high energy, as an example one bit transmitted in WSNs consumes about as much power as executing 800-1000 instructions [24]. The training process is carried out with IDS nodes. These nodes need to compute and transmit only a set of data vector (support vector) between each others, instead of transmitting all captured data to a centralized point, then train a SVM classifier. Thus our approach reduces energy consumption.

In our future work, we will use PSO (particle swarm optimization) to select the relevant features, instead of delete one feature at a time and rank the important one. In the near future, we attempt to implement our approach in sunspot sensor nodes.

REFERENCES:

[1] T. H. Hai, F. khan, and E. N. Huh, "Hybrid Intrusion Detection System for Wireless Sensor Networks", In Proceeding of the ICCSA, LNCS 4706, pp. 383-396, 2007.

[2] R. Roman, C. Alcaraz, and J. Lopez, "A Survey of Cryptographic Primitives and Implementations for Hardware-Constrained Sensor Network Nodes", Mobile Networks and Applications, Springer. Vol.12, no 4, pp 231-244, 2007.

[3] J. P. Walters, Z.Liang, W.Shi, and V.Chaudhary, "Wireless Sensor Network Security: A Survey", Security in Distributed Grid and Pervasive Computing, Auerbach Publications, CRC Press, Vol.1, Issue.2, pp.1-50, 2006.

[4] R. Roman, J. Zhou, and J.Lopez, "Applying Intrusion Detection Systems to Wireless Sensor Networks", the 3rd IEEE Consumer Communications and Networking Conference, pp.640-644, 2006.

[5] S. Kumar, "Classification and detection of computer intrusions", PhD Thesis, Department of Computer Sciences, Purdue University, USA, 1995.

[6] S. Kaplantzis, "Security Models for Wireless Sensor Networks", PhD Conversion Report, Centre of Telecommunications and Information Engineering, Monash University, Australia, 2006.

[7] S. Rajasegarar, C. Leckie, and M. Palaniswami, "Detecting Data Anomalies in Wireless Sensor Networks", Security in Ad hoc and Sensor Network, Computer and Network Security , World Scientific Publishing Co, Vol. 3, pp.231-259, 2009.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] T. H. Hai, E. N. Huh and M. Jo, "A Lightweight Intrusion Detection Framework for Wireless Sensor Networks", Wireless Communications and mobile computing, Vol.10, Issue.4, pp.559-572, 2010.

[14] K. Q. Yan, S. C. Wang, S. S. Wang, and C. W. Liu, "Hybrid Intrusion Detection System for Enhancing the Security of a Cluster-based Wireless Sensor Network",In Proc. 3rd IEEE International Conference on Computer Science and Information Technology, Chengdu, China, pp.114-118, 2010.

[15] G. Huo, and X. Wang, "A Dynamic Model of Intrusion Detection System in Wireless Sensor Networks", In Proc.International Conference on Information and Automation, IEEE, Zhangjiajie, China, pp.374-378, 2008.

[16] I. Khalil, S. Bagchi, and N. B. Shroff, "LITEWORP: A Lightweight Countermeasure for the Wormhole Attack in Multihop Wireless Networks", In Proc. International Conference on Dependable Systems and Networks, IEEE, Yokohama, Japan, pp.612-621, 2005.

[17] B. Scholkopf, and A. J. Smola, "Learning with Kernels", The MIT Press, pp.204-205, 2006.

[18] 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.

[19] 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.

[20] S. Lindsey, and C. Raghavendra, "PEGASIS: Power Efficient Gathering in Sensor Information System", In Proc.IEEE Aerospace conference, vol.3, pp.1125-1130, 2002.

[21] O. Younis, and S. Fahmy, "Heed: A hybrid, Energy-Efficient Distributed Clustering Approach for Ad Hoc Sensor Networks", IEEE Transactions on Mobile Computing, vol.3, No.4, pp.366-379, 2004.

[22] 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), Vol.2, No.3, July 2010

[23] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[24] A.Stetsko, L.Folkman, and V.Matayáš, "Neighbor-Based Intrusion Detection for Wireless Sensor Network", 6th International Conference on Wireless and Mobile Communications, IEEE, Valencia, Spain, pp.420-425, 2010.

Authors

Hichem Sedjelmaci received his Master degree in Mobile Networks and Service from the University of Tlemcen (Algeria) in 2009. Member of STIC laboratory in the University of Tlemcen, his research interests include the security issue (intrusion detection in particular) of wireless sensor networks and mobile services.

Mohammed Feham received his PhD in Engineering in optical and microwave communications from the University of Limoges (France) in 1987, and his PhD in science from the university of Tlemcen (Algeria) in1996. Since 1987, he has been Assistant Professor and Professor of Microwave, Communication Engineering and Telecommunication network. His research interests cover telecommunication systems and mobile networks.

RESEARCH ARTICLE

An Efficient Intrusion Detection Framework in Cluster-Based Wireless Sensor Networks

Hichem Sedjelmaci[1]*,Sidi Mohammed Senouci [2] and Mohammed Feham [1]

[1] University of Tlemcen, STIC Lab, Tlemcen, Algeria
[2] University of Bourgogne, DRIVE Lab, 49 Rue Mademoiselle Bourgeois, 58000, Nevers, France

ABSTRACT

In the last few years, the technological evolution in the field of Wireless Sensor Networks (WSNs) was impressive, which made them extremely useful in various applications (military, commercial, etc.). In such applications, it is essential to protect the network from malicious attacks. This presents a demand for providing security mechanisms in these vulnerable networks. In this paper, we design a new framework for intrusion detection in cluster-based wireless sensor networks (CWSN). Our detection framework is composed of different protocols that run at different levels. The first protocol is a specification-based detection protocol that runs at IDS agents (low level). The second one is a binary classification detection protocol that runs at Cluster-Head (CH) node (medium level). In addition, a reputation protocol is used at each CH to evaluate the trustworthiness level of its IDSs agents. Each CH monitors its CH neighbors based on a specification detection protocol with the help of a vote mechanism applied at the base station (high level). We evaluated the performances of our framework in the presence of four well-known attacks: hello flood, selective forwarding, black hole, and wormholes attacks. We evaluated specifically the detection rate, false positive rate, energy consumption, and efficiency. Simulation results show that our detection framework exhibits a high detection rate (almost 100%), low number of false positives, a less time to detect the attack, and a less energy consumption. our intrusion detection framework outperforms other schemes proposed in the literature in terms of detection, false positive rate, and energy consumption.

KEYWORDS

Wireless sensor networks; Clustering; Intrusion detection system; Detection rate; False positive; Efficiency.

***Correspondence**

Hichem Sedjelmaci, STIC Lab, University of Tlemcen, Algeria, E-mail: hichem.sedjelmaci@mail.univ-tlemcen.dz

1. INTRODUCTION

Wireless sensor networks (WSNs) are being used in a variety of applications such as military monitoring, detection of forest fires, human vital functions monitoring etc. These sensors are autonomous and very small in sizes and they can be deployed in a random manner in a monitored field. Despite the services they provide and the advantages they bring, WSNs have several constraints related to the energy consumption, computational capability, and memory storage. These specific characteristics must be taken into account when we deploy any of these applications into the corresponding devices.

Security is one of the most important issues in WSNs as sensors are often deployed in a hostile and insecure environment such battlefield. Cryptographic technique can protect WSNs against external attackers by applying packets authentication from the source and ensuring data integrity of the ongoing communication. However cryptographic technique cannot detect an internal attacker that is aware of the cryptographic keys. In this context, Intrusion Detection System (IDS) allows a detection of a suspicious activity within the network by analyzing a target node and triggers an alarm when this node exhibits a malicious behavior. The intrusion detection system remains the best mechanism to identify and eject the intruder within the network itself.

In wireless sensor networks, IDS topology can be classified as follows[1]: (i) Distributed approach where intrusion detection load is divided among the sensor nodes, which may collaborate with each other to form a global intrusion detection mechanism. This architecture is more suitable for flat wireless sensor networks. In a flat architecture, the sender relies on a multi-hop communication to reach the remote location (base station), leading to a high communication overhead, and (ii) Hierarchical approach: this architecture has been proposed

for multilayered wireless sensor network named CWSN (Clustered WSN). In this approach, network is divided into clusters where cluster-heads aggregate data collected from the member nodes. At the same time all cluster-heads can cooperate with central base station to form a global IDS. An example of the clustering topology for WSN can be seen in Figure 1. This architecture exhibits a low communication overhead and prolongs the network lifetime. The remainder of this paper is organized as follows: In Section 2, we give some background and related work. Section 3 proposes our intrusion detection framework. In Section 4, we provide simulation results and performance analysis of our scheme. Finally, we summarize the main results and give some perspectives that we envisage to carry out in Section 5.

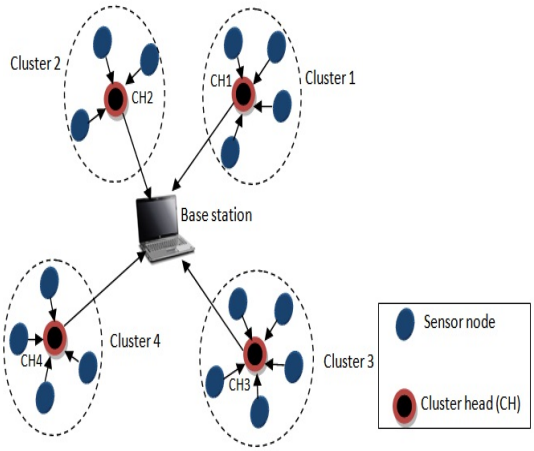


Figure 1. Clustered WSN (CWSN) topology.

2. BACKGROUND & RELATED WORK

In this section, we highlight some related works and background necessary for understanding our propositions. We organize this section in four subsections. In the first one, we summarize some intrusion detection works that we found in the literature by describing their main shortcomings. In the second sub-section, we describe some routing attacks and especially: selective forwarding, black hole, hello flood and wormholes attacks. In the third sub-section, we give some background information on the supervised learning algorithm Support Vector Machines (SVM). In particular, we describe how SVM can be used in either centralized or distributed fashion. We finally give, in the last subsection, some relevant information about clustering protocols in wireless networks. In particular, we describe the Hybrid Energy-Efficient Distributed clustering (HEED) algorithm [2], which was selected as a base of our clustering protocol used in our intrusion detection framework.

2.1. INTRUSION DETECTION IN WSN

Currently, there are limited researches that use the intrusion detection system to identify the malicious behavior within the network. The authors in [3] are among the first who use the mechanism of intrusion detection in WSN. This research work is based on naturally occurring events and the analysis of fluctuations in sensor reading [4]. In [5], the authors propose a model that relies on the number of packets being dropped to detect black hole and selective forwarding attacks. The authors in [6] analyzed the packets by using both detection policies (i.e. anomaly detection based on SVM and a set of attacks signature) to detect the routing attacks with a high accuracy. However, the major drawbacks of these schemes [5][6]are related to not taking into account that the IDS node can also be a malicious node and that the CH node is an attractive target of attackers due to their relevant data. In [7], the authors propose an intrusion prevention and detection framework in a one-hop clustering topology for WSNs. In the intrusion prevention phase, the authors propose a cryptographic technique to prevent the external threat to attack the networks. In the intrusion detection phases, each IDS monitors the nodes that are located within its radio range (one-hop). The detection framework uses only a rule-based detection to identify the malicious node. In their experiments results, the authors claim that using one-hop clustering for intrusion detection permits to all the IDS nodes within the same cluster to detect the malicious node when it occur. In this scheme, the authors don't evaluate the performance of their framework in terms of energy consumption. In [8], the authors propose a mechanism of intrusion detection and isolation of malicious nodes. In their detection mechanism, the cluster head monitors its cluster member by using a set of rules related to a specific attack behavior. When the intruder occurs, the IDS isolates the attacker nodes and records malicious nodes in its isolation table. In their hierarchical topology, all CHs within the network are managed by a node called a Primary cluster-head (PCH). This later is an attractive target of the attackers. In order to avoid this issue, the CH and the cluster members monitor this node. According to the simulation results, their scheme permits to consume a less energy compared to the schemes proposed by the authors in [9]. The major drawback of both detection frameworks proposed in [7][8] is the detection policies applied by IDS nodes, which are based only on a rule-based detection. Using only the rule based approach for the detection process leads to a low detection rate when several kinds of attacks occur. In [10], the authors propose a light-weight intrusion detection technique for clustered sensor nodes based on IDS framework developed by the authors in [11]. In this technique, the monitoring node has two detection engines identified as local agent and global agent. The former monitors only their own communication (e.g. sent and received messages, sensed data) and the latter observes the neighbors' communication. The global agent uses a rule based-approach with two-hop neighbor knowledge for

the anomaly detection, and it sends alarms to the cluster-head when the intrusion occurs. Both monitoring nodes use signature-based detection, which are computed and generated by the CH. The authors attempt to provide a cooperative mechanism between IDS agents that is based on trust priority in order to reduce the false alerts raised by the intruder. Nevertheless, the drawback of this scheme is the large increase of the size of the signature database which in turn leads to an overload of the node.

As shown in Figure 2, detection policies for the intrusion in WSN can be classified into two main techniques:

1. Signature-based detection or Misuse detection:This approach is based on comparing the observed behavior to a set of attack signatures that are stored in the node's memory. If a match occurs, the analyzed node is defined as an attacker. The technique is very accurate to detect known attacks but it cannot identify unknown attacks. As a result, it requires constant signature updates to be reliable.

2. Anomaly detection:This approach is based on first modeling the normal node behavior and then identifying anything that deviates from this model as anomalous. This technique is composed of two categories:

- **Binary classification-based detection**:This category uses a supervised learning algorithm to model the normal behavior. The advantage of this technique is to detect unknown attacks, but its high computational cost leads to a rapid decrease of the node' lifetime. As a consequence, and in order to mitigate this cost, the technique must be embedded in a node that has considerable power resources. Among detection techniques proposed in the literature for wireless sensor networks, we find neural networks and support vector machine (SVM).
- **Specification-based detection**: This approach works by simply specifying a normal behavior using a set of rules. The advantage of this technique is the ability to detect unknown malicious behaviors with a low computational cost. However, reliability of this detection approach relies on continuous updates of rules over time.

2.2. ROUTING ATTACKS

The intruder could realize one of the four following attacks: selective forwarding, black hole, hello flood, and wormholes. We describe in the following these different attacks.

1. Selective forwarding: In this case, the attacker stops forwarding certain packets and starts dropping them. This attack is therefore detected by calculating the packet-drop rate (PDR).

2. Black hole: In this attack, the intruder pretends to be in the shortest path to the cluster-head (CH) by using a high-power transmission [5]. In this case, the intruder will be able to receive the messages and subsequently swallows

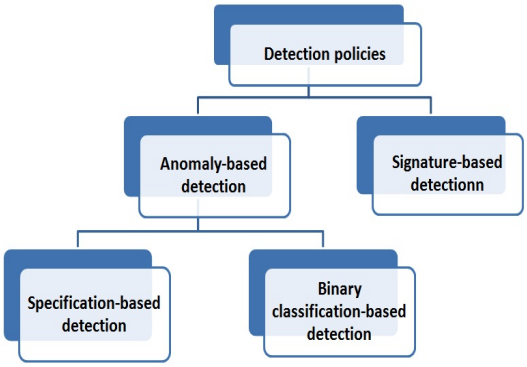


Figure 2. Detection techniques.

the corresponding packets (drops all receiving packets). This attack can be detected by computing the PDR and the received signal strength intensity(RSSI).

3. Hello flood: A malicious node broadcasts hello packets by generating a high signal strength compared to other sensor nodes. In this case, other legitimate nodes in the network will send their packets to the broadcasting node. As a result, the packets will then be dropped or altered. This attack can be detected by computing the RSSI.

4. Wormholes: According to the work undertaken by the authors in [12], wormholes attacks are classified into passive or active attacks. In our research, we focus on active wormholes. In particular, the wormholes attacks tend to pretend to be one hop away from the cluster-head by using high signal strength. As a consequence, the attacker forwards the messages received from legitimate node to another attacker as illustrated in Figure 3. In this case, both malicious nodes take part in the network routing protocol. In Figure 3, note that M1 and M2 are the endpoints of wormholes tunnel and M1 generates a high signal strength in order to convince a node that is close to the cluster-head (1 hop away from CH). Node A wants to send its packets to the CH either by following the valid route (nodes B and C) or a malicious one (nodes M1, E, and M2). In both cases, node A chooses the lower-cost route via M1-M2 wormholes (shown in solid arrows) since M1 pretends to be close to the CH. Therefore, all packets received by M1 from A are forwarded directly to M2 and are not sent to E. In this case and in order to detect this attack, we simply monitor the signal strength. In addition, nodes located in the same neighborhood of this attack don't receive the packets from this malicious node , hence the packet-dropping rate becomes high. As illustrated in Figure 3, the IDS1 agent hears the packets sent by M1 with a high RSSI. In addition, this monitoring node does not hear the message that must be forwarded by E to M2. Based on the RSSI and PDR, M1 will be detected as a malicious node that is carrying out a wormhole attack.

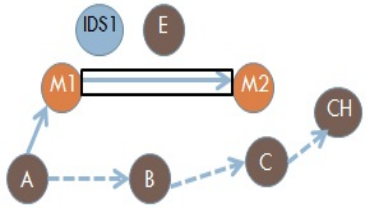


Figure 3. Active wormholes attack.

2.3. SUPPORT VECTOR MACHINES

Support vector machine (SVM) is a supervised learning method developed by Vapnik in 1995 and it is used for classification and regression analysis. The aim of the SVM classifier is to construct a hyperplane that separates data into two classes defined by the number of support vectors. These vectors define the boundary of each class. In situations where SVM cannot separate data into two classes (nonlinear separation), it solves this problem by mapping input data into high-dimensional attributes spaces using a kernel function [13]. As a result, it allows a linear separation. We note that, in our research, we focus only on two-class problems. The binary SVM classification provides a decision function [14]:

$$f(a,x,b) = sgn\big(\sum_{i=1}^{m} y_i \alpha_i k(x_i,x) + b\big) = \pm\{1\} \quad (1)$$

Here $k(x_i,x)$ is the kernel function and α are the Lagrange multipliers, which can be found by solving the nonlinear optimization equations below:

$$\begin{cases} \max \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j k(x_i,x_j) \\ \text{subject to } \sum_{i=1}^{m} y_i \alpha_i = 0,\ 0 \le \alpha_i \le C \end{cases} \quad (2)$$

SVM can be used either in a centralized or distributed fashion.

In the first case, the SVM, which is embedded at the base station, collects the packets from all nodes, and then trains an SVM classifier. This approach forces a node to send a considerable amount of data to a remote location which leads to a high communication overhead and subsequently decreases the lifetime of the sensor nodes.

In the distributed approach the cost of energy is reduced. The support vectors, that are much less than the input data, are computed at each node. These key vectors are then exchanged between nodes with the exception of the centralized approach where packets are sent to remote nodes. As a consequence, the network lifetime is increased as this approach meets the energy consumption constraint. In our anomaly detection model, a distributed SVM learning between CHs is applied to detect the anomaly behavior.

2.4. CLUSTERING

A hierarchical topology divides the sensor network into clusters, each one having a cluster-head (CH). The objective of this architecture is to help the deployment of protocols (especially routing) and save the energy that enhances the survivability of the network. This is achieved by designating a CH node to have the responsibility for forwarding a packet (which contains the aggregated data received from cluster members) to the base station rather than all nodes send their sensed data to remote location (base station).

Among the large number of Cluster-based routing protocols proposed in the literature, we cite: LEACH[15], PEGASIS[16] and HEED[2]. The aim of HEED protocol is to use a combination of the residual energy and an intra-cluster communication cost to elect the cluster-head. In our study, a modified version of this routing protocol is selected (by using only the residual energy) to embed our intrusion detection framework.

In HEED, the authors defined two kinds of nodes: 'uncovered' and 'covered'. In this case, the first node announces itself to become a cluster-head by broadcasting an announcement message to other nodes. This process occurred when the execution algorithm is completed without electing a cluster-head. The 'covered' node is a cluster member who selects a lower cost cluster-head according to the overheard message sent by the CH. To this end, a node can be elected to become a cluster-head using the probability formula below:

$$CH_{prob} = \frac{E_{residual}}{E_{max}} \quad (3)$$

Where *Eresidual* and *Emax* are the residual and maximum energy respectively in the node.

3. PROPOSED FRAMEWORK

In our framework the intrusion detection process is carried out at three levels as detailed in the following. In the low level, a set of nodes called IDS agents monitor the communication of their neighbors and report their feedbacks to their cluster-head for further detections. To identify any suspected behavior, these agents use the specification-based detection technique. This technique relies on a set of rules do detect and prevent the malicious behavior (more details in subsection **3.1**.2). Due to energy constraints, and the fact that one bit transmitted in WSNs consumes about as much power as executing 800 to 1000 instructions [17], the agent node has to limit the amount of information that is exchanged between him and the cluster-head.

In the medium level, a powerful cluster-head (CH) uses SVM training technique to detect any anomaly. This approach allows separating data into two classes (normal and anomalous). It is called a binary classification. Given that no node is assumed to be trustworthy, a reputation

mechanism is applied at the cluster-head in order to evaluate the trustworthiness of their IDSs membership. Detection process occurring between level one (IDS agents) and level two (CH) is illustrated in Figure 4.

In the high level, each CH monitors its CH neighbors based on a specification detection technique and sends a ballot form to the base station containing the suspected CH. The base station is used as the counter to collect the votes that are generated by CHs in order to take a final decision on any suspected node that may be found. Detection process occurring between levels two (CH) and three (base station) is illustrated in Figure 5.

In the following, we give more details about the different protocols of our framework.

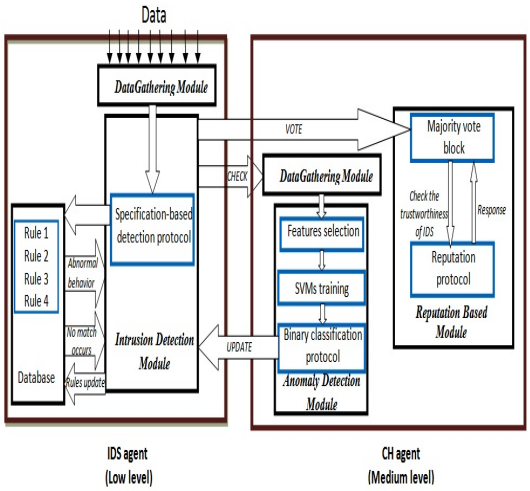


Figure 4. Detection process occurring between IDS and CH agents.

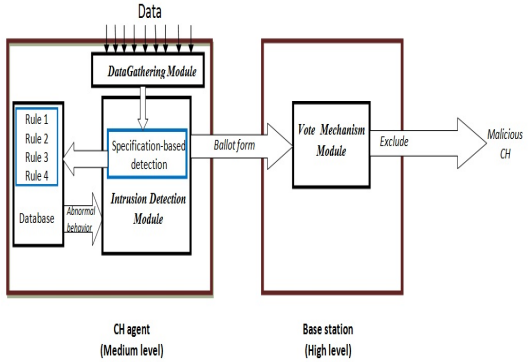


Figure 5. Detection process occurring between CH and base station.

3.1. Sensor (Low) Level Intrusion Detection

In each cluster and for each communication link, there must be at least one IDS agent for collecting and analyzing the packets according to the set of rules within the radio range. As shown in Figure 4 (see IDS agent), 'Data Gathering' and 'Intrusion Detection' modules are the most important components of this agent. These modules are detailed in the following:

1. Data Gathering Module. Due to the broadcast nature of wireless networks, IDS nodes gather the packets within their radio range [10] and pass it to the intrusion detection module for process analysis as shown in Figure 4.

2. Intrusion Detection Module. This module follows a specification based detection protocol do detect and prevent the malicious nodes. The purpose of this protocol is to categorize the target behavior as normal or abnormal according to a set of rules. In our case, there are four rules related to each attack. The rule for detecting 'selective forwarding' attack can be defined as the PDR which is greater than a certain threshold(δsf) . The rule for detecting 'hello flood' is the value of RSSI that exceeds a certain predefined threshold ($\delta rssi_h$). The rule for detecting a 'black hole' is defined as the number of PDR (which is greater than δ_{bh} threshold) and the excess in signal strength (higher than $\delta rssi_{bh}$ threshold). Finally, the rule for detecting 'wormholes' attack is the excess in signal strength (higher than $\delta rssi_{wo}$ threshold) and none of the nodes, which are located in the same neighborhood of this malicious node forwards a received packet sent by this adversary (by computing the PDR which excess δwo threshold). All these rules used for attacks detection are illustrated in Figure 6 as follows:

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

// Rule for hello flood attack
if (RSSI>δrssi_h )
  //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 (PDR>δbh  && RSSI>δrssi_bh)
  //node_ID is performing a black Hole attack
     send_VOTE_message_CH(node_ID);
else
     send_CHECK_message_CH (node_ID,PDR,RSSI);

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

Figure 6. The detection rules of the four attacks.

As illustrated in Figure 4, when abnormal behavior is detected according to the selected rule, a *VOTE* message

is submitted to the Majority Vote block (located at a CH) to make a vote process. This message includes the suspected node and the attack type. When a vote exceeds a certain threshold, the CH will not assign any time slot to this malicious node and will be removed from the cluster. However, when the detection evidence is not very conclusive (no match occurs), a *CHECK* message is forwarded by the IDS agent to Anomaly Detection module (located at the CH also) for further detections. This message includes the analyzed node with the PDR and RSSI.

3.2. Cluster (Medium) Level Intrusion Detection

Inspired by the work of authors in [2], our clustering algorithm which was implemented under TOSSIM Simulator [18], elects at each cluster a CH that has more power resources to manage and aggregate data received from the cluster members. As illustrated in Figure 4 (see CH agent), this powerful node comprises three modules: Data Gathering, Anomaly Detection and Reputation Modules. They are detailed in the following:

1. Data Gathering Module. This module is responsible to collect the *CHECK* messages sent by the IDS agent. This message includes the address of the node analyzed by IDS agent and the following features: PDR and RSSI. These features are then forwarded to the anomaly detection module for the training and classification process.

2. Anomaly Detection Module. Anomaly detection procedure is divided into three steps:

- *Step1: Features selection*. This is an important factor that increases the classification accuracy, reduces the false positive and speeds up the training time. In this research, the (PDR) and RSSI are used as input data for the training process.
- *Step2: SVMs training process*. The anomaly detection uses a distributed learning algorithm for the SVM training to classify data as normal or anomalous (a binary classification problem). Each CH trains the SVM locally, then computes a set of data vectors called support vectors that are generally less in number than the input data used during the learning process. These vectors will be sent to an adjacent CH that is located in the same radio range. Each CH that receives the support vectors from their CH neighbors updates its corresponding information by unifying the received data and its own support vectors. They will then retransmit the resulted set of support vectors to the nearby CHs.
- *Step3: Binary classification protocol*. When the training process is completed, each CH classifies new incoming data according to the attacks and the normal pattern. Any deviation from the normal behavior is considered as anomalous. In this case, an *UPDATE* message (including a new sign of attack) is send back to their IDS members to compute the new rule of this attack as illustrated in Figure 4.

The packet frame of all exchanged messages (*CHECK, VOTE and UPDATE*) are is(re) illustrated in Figure 7.

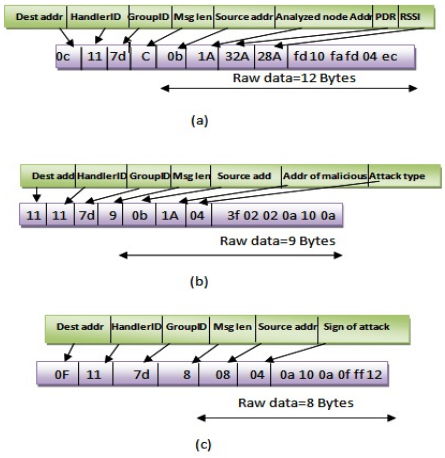


Figure 7. Packet format of (a) *CHECK, (b) VOTE, and (c) UPDATE* messages.

3. Reputation Based Module. When IDS agent detects an attack, it sends a *VOTE* message to its CH as illustrated in Figure 4. This message includes the suspected node and the type of the attack. The CH node uses a Majority Vote Block to determine if the suspected node is an intruder or not, while a beta reputation protocol has to evaluate the confidence level of IDS agents (see reference [19]). If a vote exceeds the predefined threshold, the suspected node is ejected from the network and the reputation of the IDS nodes that detect the attack will be increased. Otherwise, the reputation of IDSs will be decreased. We note that for each cluster the threshold is n/2, where n is the number of IDS agents per each cluster. Reputation-based protocol takes a step further in helping to identify compromised nodes as early as possible [20].

The reputation of the IDS_i maintained at its corresponding CH is defined as follows [19] :

$$R_i = \beta(\alpha_i + 1, \beta_i + 1) \tag{4}$$

Here α_i and β_i represent the normal and suspected behaviors respectively of IDS_i claimed by CH. The updating of this two parameters (i.e. α_i,β_i) can be found in Reference [19].

The Trust metric is defined as the level of trustworthiness of an IDS node, which can be computed as follows:

$$T_i = E[R_i] \tag{5}$$

Where $E[R]$ is the statistical expectation of the reputation function. The Trust value is classified by the following mapping function:

$$M(T_i) = \begin{cases} high & T_i \geq TH \\ low & T_i < TH \end{cases} \tag{6}$$

After computing the trust value, each CH sets this value according to the mapping function above to indicate the trust level requirement. Only IDSs having a high trust value can trigger the detection process. Otherwise they will be defined as normal node and not being able to play the IDS role. As a result, a community of trustworthy IDS nodes will be generated.

3.3. Intra Cluster-heads (High) Level Monitoring

The CH is an attractive target of an attacker since it contains relevant data. As a consequence, the intruder uses all its capacity to launch an attack against this hot point. In order to avoid this issue, each CH monitors its CH neighbors. The cluster-head is equipped with Data Gathering Module as in the Cluster Level Intrusion Detection and another module, which is Intrusion Detection. The base station is equipped with a Vote Mechanism Module. These modules are illustrated in Figure 5 and described as follows:

1. Data Gathering Module. Each cluster-head captures the packets from other CHs that are situated in the same radio range then computes the RSSI and PDR. Subsequently, this information will be forwarded to the intrusion detection module for monitoring purpose as shown in Figure 5.

2. Intrusion Detection Module. Each CH monitors its nearby CHs by adopting a specification based detection protocol as used before by IDS agents. According to the rules related to each attack (see subsection **3.1**.2 about these rules), if an abnormal behavior occurs, the monitoring CH sends a ballot form that includes the suspected CH and the attack type to the base station as shown in Figure 7(re). The base station performs a voting mechanism in order to identify suspect nodes. In particular, if more than half of votes are in favor of attack, the CH will be excluded from the network and a new CH will be elected.

4. PERFORMANCE EVALUATION

In our experiment, we used TOSSIM simulator [18]; a simulator for TINYOS application. The main advantage of this simulator compared to other tools such as NS2 [21], is the fact that we can easily embed the source code written in NESC on real sensor nodes (with TINYOS operating system). However, the TOSSIM simulator does not have the ability to model the energy dissipated during the execution of the application. To this end, an improved version of the tool was proposed by Harvard University called POWERTOSSIM [22] allowing the simulation of nodes' energy consumption and hence the determination of the network' lifetime.

4.1. Simulation Assumptions

We used in our simulations 168 nodes deployed randomly in a square area of 88*88m2. We notice that the network was composed of 8 clusters with one CH in each. All sensors are static. In order to avoid collisions, a TDMA protocol is used. We use the chipcon CC1000 [23] as a transceiver and each node transmits its packets at a frequency between 433 MHZ and 868 MHZ. All the key parameters of the simulation are summarized in Table I.

The thresholds for each detection attack were determined by carrying out several simulations. The summary of these thresholds are illustrated in Table I

The purpose of our simulations is to investigate the effect of each attack in the network in isolation and then all together. In addition, we assume that there are no attacks at the beginning of simulation. We have varied the number of IDS nodes per cluster from 1 to 10 in order to assess the performances of our detection framework for different configurations. The binary classification detection protocol used in our simulation is a simple version of SVM learning algorithm that is able to classify only the four routing attacks cited before.

In order to evaluate our framework, we used different metrics:

- **Detection Rate (DR)**: defined as the ratio between the number of correct detected intrusions and the total number of intrusions.
- **False Positives Rate (FPR)**: also called false alarms: defined as the ratio between the number of normal connections incorrectly classified as intrusions and the total number of normal connections [13].
- **Energy Consumption (EC)**: defined as the energy consumed by all sensor nodes and computed as follows:

$$E_t = \frac{\sum_{i=1}^{N} E_{node_i}}{N} \tag{7}$$

  Where E_t is the energy total of the network and N the number of nodes.
- **Efficiency (E)**: This metric determines the required time for our IDS agents to detect the occurrence of the first adversary node. It is computed as follows:

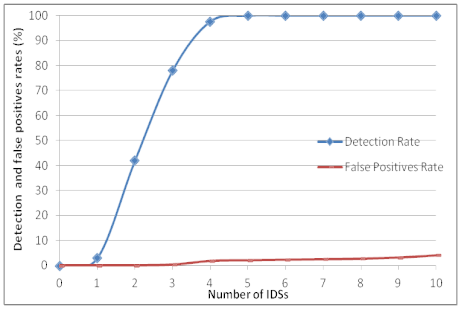
$$E = \frac{ED - ET}{Sampling frequency} \tag{8}$$

  Where ET is the time of a first malicious behavior starts and ED is the detection time of the first malicious node, respectively.

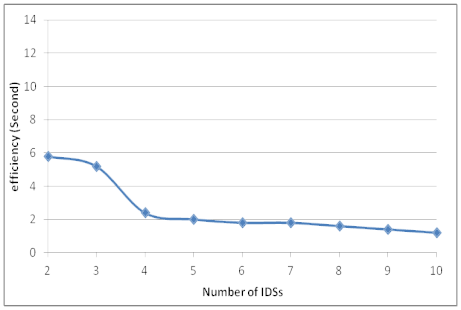
4.2. Results Analysis

1. Hello flood attack scenario. This attack was implemented as a node that has a high signal strength compared to the other nodes. As shown in Figure 8 (a), when the number of IDSs increases, the detection rate increases together with the number of false positives. When the

average number of IDSs in each cluster is 4, the detection rate and false positive rate are close to 98% and 2%, respectively. In addition, as shown in Figure 8 (b) our detection framework requires less time to detect the hello flood attack when the average number of IDSs in each cluster is 4 (the efficiency is close to 2 seconds). As a consequence, an optimal number of IDSs is a crucial characteristic that makes our scheme effective. Finally, we conclude that when an optimal number of IDS agents is determined (4 agents per clutser) our framework exhibits a high detection rate, low number of false alarms, and requires less time to detect this attack.



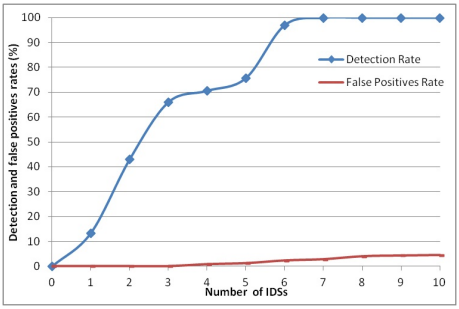
(a)



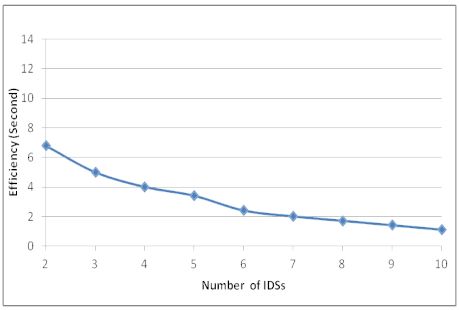
(b)

Figure 8. Hello flood attack scenario:(a) Detection and false positives rates and (b) Efficiency.

2. Selective forwarding attack scenario. The selective forwarding attack is recognized when a node drops a considerable number of packets compared to legitimate node. The detection rate and the number of false alarms are related to the number of IDS agents per each cluster. As shown in Figure 9 (a), both metrics increase when the number of agents increases. Therefore, the optimal number of IDSs for detection of Selective forwarding with less occurrence of false positive is equal to 6. In addition, according to this optimal number of agents our detection framework requires 2 seconds to detect the selective forwarding attack as shown in Figure 9 (b). Therefore, a tradeoff between the number of IDS nodes and false positives must be considered in order to suit our application requirements.



(a)



(b)

Figure 9. Selective forwarding attack scenario:(a) Detection and false positives rates and (b) Efficiency.

3. Black hole attack scenario. This attack was implemented as a node that has high signal strength and drops all receiving packets. The detection performance of our scheme under black hole attacks is illustrated in Figure 10 (a). Our detection framework yields a good detection of black hole attack; exceeding 96 % when average number of IDSs per each cluster is equal to 5. This later is an optimal number of agents under black hole attacks that meets our application requirements in terms of detection rate and low number of false positives. The required time of an IDS agent to detect this adversary reaches almost 1,5 seconds when the number of IDS agents per each cluster is equal to 10, as illustrated in Figure 10 (b). However, a high number of false alarms occurred when we select 10 agents per cluster. As a result, the optimal number of IDS nodes per each cluster that meets our application requirements in terms of fast detection time, detection rate and the number of false alarms is equal to 5.

4. Wormholes attack scenario. This attack was implemented as both the node that generates a high signal strength as well as the nodes located in the same neighborhood of the attack that do not receive the message from this adversary. The detection rate reaches almost 100% when the number of agents increases as shown in Figure 11 (a). In this case, the optimum number of IDS agents per cluster that provides a trade-off between the detection rate and the number of false alarms under wormholes attacks

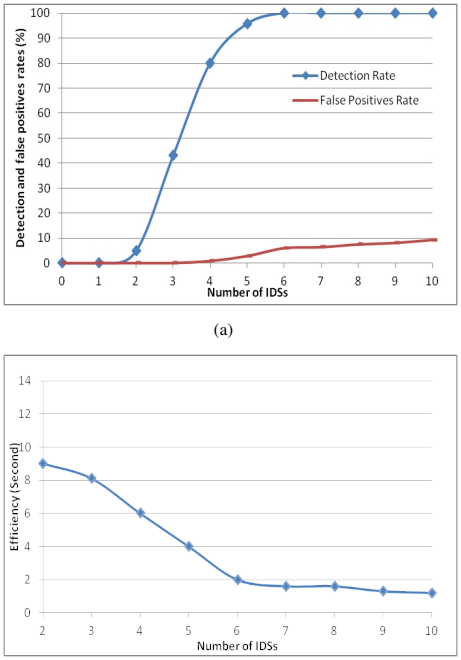


Figure 10. Black hole attack scenario:(a) Detection and false positives rates and (b) Efficiency.

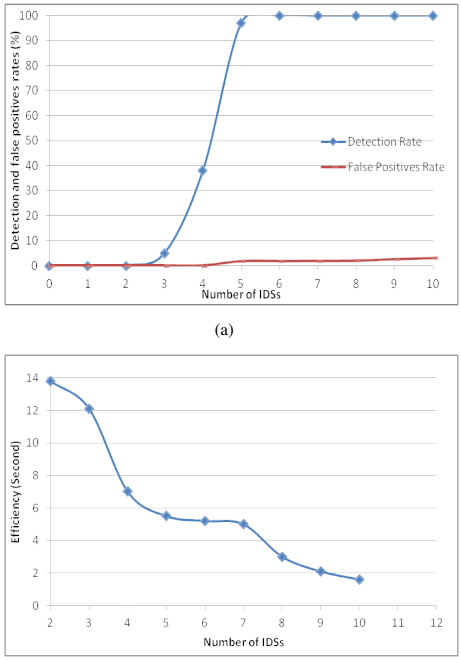


Figure 11. Wormholes attack scenario:(a) Detection and false positives rates and (b) Efficiency .

is equal to 5. The detection of wormholes attack requires a considerable amount of time compared to other detection attacks, as illustrated in Figure 11 (b). Using 6 agents per cluster yields to a detection time reaching 4.5 seconds. As a conclusion, the optimal number of IDS agents under Wormholes attacks for low number of false positives, a high detection rate, and fast detection time is equal to 6.

5. Multiple attacks scenario. In this section, we evaluate the performances of our framework when various kinds of attackers appear within the WSN. First, we evaluate our IDS framework under black hole and selective forwarding attacks with one proposed by the authors in [5] in terms of detection rate. Second, we compare our detection framework when all the attacks cited above appear (i.e. hello flood, selective forwarding, black hole and wormholes attacks). Here we compare its performances against another scheme proposed in the reference [10] in terms of detection rate, false positives rate and efficiency. In addition , in order to determine the energy efficiency of our model, we compare the results to the ones obtained in the scheme [24] .

As shown in Figure 12, our detection framework performs a better detection against black hole and selective forwarding attacks than the scheme proposed in [5], specifically when the number of IDSs is important. In this case, the number of false alarms is related to the number of IDS nodes. As a result, increasing the number of IDS agents per cluster, results in an increase in the rate of false positives. We must therefore, consider a balance between the low false positives rate and high detection rate metrics. As result, the optimal number of IDSs per cluster meeting our application requirements is equal to 6.

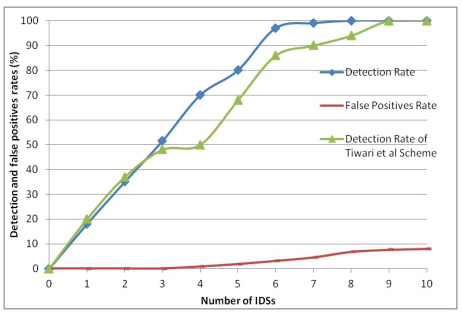
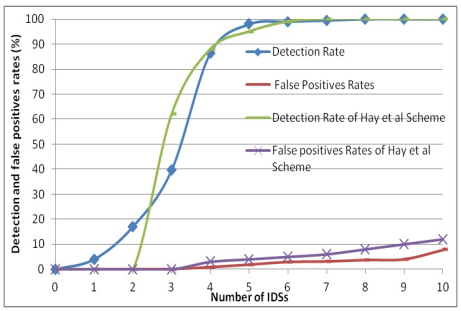


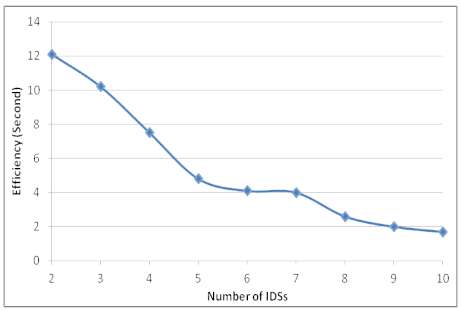
Figure 12. Comparison of our framework under selective forwarding and black hole Attacks.

As shown in Figure 13 (a), our intrusion detection framework is effective against all attacks (cited above) when the number of IDS agents increases. However the number of false positives will affect the performance of our framework when the number of IDSs is important (exceed 6 agents). Therefore, we must consider a balance between the number of IDS agents and the false positive rate. As a result, the optimal number of IDS agents per each cluster that meets our application requirements is equal to 5. The

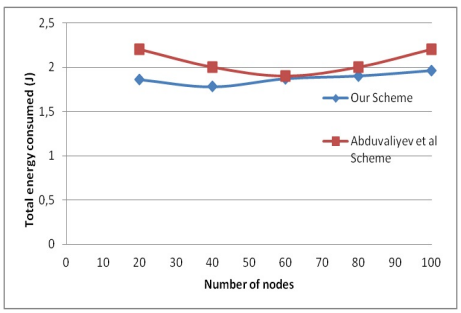
detection and false positives rates are close to 98% and 2%, respectively. As illustrated in figure (re) Figure 13 (a), both schemes exhibit a high detection and low false alarms rates. Otherwise, our scheme performs a better detection and a low number of false alarms compared to the scheme in [10] when an optimal number of IDS agents is selected (5 agents per each cluster). In other side, according to this optimal number of agents, the required time of IDS nodes to detect the first malicious node in the network is close to 4s as illustrated in Figure 13 (b), which is suitable for our application requirements. Finally, we conclude that using an optimal number of IDS agents at each cluster, our intrusion detection framework exhibits a low number of false positives, a high detection rate, and fast detection time.



(a)



(b)



(c)

Figure 13. Multiple attacks scenario:(a) Detection and false positives rates, (b) Efficiency and (c)Energy consumption.

We can observe in Figure 13 (c), that our proposed detection framework requires less energy to detect all the attacks that are given above in comparison to the approach used by the authors in [24]. This improvement has been achieved due to two main reasons: the first is that we use a clustering topology that aims to select only one node per cluster (cluster-head) that forwards the aggregated data to base station rather than all nodes sending their sensed data to remote location (base station). The second reason is the fact that each IDS agent relies on a policy that minimizes the packets transmission that in turn will save energy consumption. As a conclusion, we can state that our scheme improves the network lifetime.

5. CONCLUSION AND FUTURE WORKS

In this paper, we propose an efficient and lightweight intrusion detection framework against common routing attacks that have high severity damage in wireless sensor networks. The aim of our framework is to apply a set of intrusion detection protocols on cluster-based WSNs that run at different levels (i.e., at the sensor node level, cluster-head and base station levels) in order to identify and prevent any adversary node disturbing the network. In particular, at a sensor node level, rule based detections are implemented at the IDS agents to identify any incoming attack. At the same time, at a cluster-head level, the binary classification detection embedded at each CH aims to update the rules of the IDS agents. In addition, a reputation protocol is used at each CH to evaluate the trustworthiness level of its IDSs member. At a high level, the cluster-head agent sends an intrusion report on the suspected CH to the base station that in turn will perform a voting mechanism about the suspected node. Simulation results show that our scheme presents superior performances for detecting attacks (such as hello flood, selective forwarding, black hole and wormholes attacks) compared to other schemes. This is mainly specific for networks with an optimal number of IDS agents per cluster. In this case, the IDS agent will generate fast detection time with low number of false alarms. Simulation results confirmed the lightweight of our detection framework in term of energy used and show that our scheme uses less energy than other model proposed in current literature

In the near future, we will expand the detection range of our framework by adding a sophisticated distributed SVM training model that has the capability to detect any attack. We also intend to carry out new simulations in order to observe the performances of our scheme in the context of mobile wireless sensor networks.

REFERENCES

1. Bhattasali T, Chak R. Lightweight hierarchical model for hwsnet. *International Journal of Advanced Smart Sensor Network Systems (IJASSN)* 2011; **1**(2):17-32. DOI:10.5121/ijassn.2011.1202.

2. Younis O, Fahmy S. HEED: A hybrid energy efficient distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on Mobile Computing* 2004; **3**(4):366-379.
3. Doumit SS, Agrawal DP. Self-organized criticality and stochastic learning based intrusion detection system for wireless sensor networks. *Proceedings of IEEE Military Communications Conference (MILCOM)*, 2003; 609-614.
4. Mitrokotsa A, Karygiannis A. Intrusion detection techniques in sensor networks. *In Book :Wireless Sensor Network Security, Cryptology and Information Security Series* 2008: 251-272.
5. Tiwari M, Arya KV, Choudhari R, Choudhary KS. 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, 2009; 824-828.
6. Sedjelmaci H, Feham M. Novel hybrid intrusion detection system for clustered wireless sensor network. *International Journal of Network Security & its Applications (IJNSA)* 2011; **3**(4):1-14. DOI : 10.5121/ijnsa.2011.3401.
7. Shin S, Kwon T, Jo GY, Park Y, Rhy H. An experimental study of hierarchical intrusion detection for wireless industrial sensor networks. *IEEE Transactions on Industrial Informatics* 2010; **6**(4): 744-757.
8. Chen RC, Hsieh CF, Huang YF. A New method for intrusion detection on hierarchical wireless sensor networks. *Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication, ACM*, SKKU, Suwon, Korea, 2009; 238-245. DOI: 10.1145/1516241.1516282
9. Su WT, Chang KM., and Kuo YH. eHIP: An energy-efficient hybrid intrusion prohibition system for cluster-based wireless sensor networks. *Computer Networks* 2007; **51**(4):1151-1168.
10. Hai TH, Huh EN, Jo M. A lightweight intrusion detection framework for wireless sensor networks. *Wireless Communications and Mobile Computing* 2010; **10**(4):559-572. DOI: 10.1002/wcm.785.
11. Roman R,Zhou J,Lopez J. Applying intrusion detection systems to wireless sensor networks.*The 3rd IEEE Consumer Communications and Networking Conference*, Las vegas, USA, 2006; 640-644.
12. DeGraaf R, Hegazy I, Horton J, Safavi-Naini R. Distributed detection of wormhole attacks in wireless sensor networks. *Proceedings of 1rst International Conference on Ad oc Networks, Springer*, Niagara Falls, Canada, 2009; 208-223.
13. Haijun X, Fang P, Ling W, Hongwei L. Ad hoc-based feature selection and support vector machine classifier for intrusion detection. *Proceedings of IEEE International Conference on Grey Systems and Intelligent Services*, Nanjing, China, 2007; 1117-1121.
14. Gama J, Pedersen R. Predictive learning in sensor networks. *In Learning from Data Streams, editors Joo Gama and Mohamed Gaber, Springer* 2007; 143-164.
15. Heinzelman WR, Chandrakasan A, Balakrishnan H. Energy efficient communication protocol for wireless microsensor networks. *The 33rd IEEE International Conference on System Sciences*, Hawaii, USA, 2000, **2**; 1-10.
16. Lindsey S, Raghavendra C. PEGASIS: Power efficient gathering in sensor information system. *Proceedings of IEEE International Conference on Aerospace*, 2002, **3**; 1125-1130.
17. Stetsko A, Folkman L, Matay V. Neighbor-based intrusion detection for wireless sensor network. *6th International Conference on Wireless and Mobile Communications, IEEE*, Valencia, Spain, 2010; 420-425.
18. Simulating tinyOS networks. Available at http:// www.cs.berkeley.edu/ pal/research/tossim.html.
19. Ganeriwal S, Srivastava MB. 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, 2004; 66-77. DOI: 10.1145/1029102.1029115.
20. Alzaid H, Foo E, Nieto JG, Ahmed E. Mitigating On-Off attacks in reputation-based secure data aggregation for wireless sensor networks. *Security and Communication Networks* 2012; **5**(2):125-144. DOI: 10.1002/sec.286.
21. The network simulator ns-2. Available at http:// www.isi.edu/nsnam/ns/.
22. Efficient power simulation for tinyOS applications. Available at http://www.eecs.harvard.edu/ shnayder/ptossim/.
23. CC1000 chip, very low power RF transceiver. Available at http://www.ti.com/lit/ds/symlink/cc1000.pdf.
24. Abduvaliyev A, Lee S, Lee YK. Energy efficient hybrid intrusion detection system for wireless sensor networks. *International Conference on Electronics and Information Engineering, IEEE*, Kyoto, Japan, 2010; 25-29.

Table I. Simulation parameters.

Simulation time	875 seconds
Simulation area	88*88m^2
Number of nodes	168
Radio model	Lossy radio model
Number clusters	8
Number of IDS agents per cluster	1-10
Routing	Modified HEED
MAC	TDMA
Radio range	15m
Sensor initial energy	5 Joules
δsf	64 %
$\delta rssi_h$	-41 (dbm)
δ_{bh}, $\delta rssi_{bh}$	94 %, -47 (dbm)
$\delta rssi_{wo}$, δwo	-44 (dbm), 99%

Int J Wireless Inf Networks (2011) 18:11–23
DOI 10.1007/s10776-011-0129-1

Design and Implementation of an Algorithm for Cardiac Pathologies Detection on Mobile Phone

Rachid Merzougui · Mohammed Feham · Hichem Sedjelmaci

Received: 4 November 2009 / Accepted: 6 January 2011 / Published online: 30 January 2011

Abstract The development and the design of telemedicine services have taken a great consideration and care in the domain of wireless communication nowadays. The set of these researches is concerned with old people and lack of infrastructures of reception for those who are at risk or tend to have deterioration in their health condition. Thus, several works of research contributed to develop telemedicine services. They notably focus on the conception and the development of communication architectures between the actors of these systems, monitoring and the development of human's quality is based on the storage of the collected data at home and analytical tools, and processing of these large quantities of data. Therefore, it is useful to detect and prevent the occurrence of critical situations of a remote person, the transmission of the messages and alarms to concerned actors to be ready to intervene in a case of emergency. Many works and systems undertaken in this field carry out the complete analysis and synthesis of signals on large servers (great capacities, better resolutions…). Moreover, these systems would have required large means and a large infrastructure in their deployment (installation, configuration…), which generates the disadvantage of the excessive expenditure. In this paper, we suggest to introduce and implement this complete treatment for revealing critical situations and pathologies on a simple mobile phone by respecting theirs constraints. The principal objective is to permit a taking off for medical and social dependant people as aged ones, handicapped, in order to the adaptation with their environment domestically and make up their incapacities. In this case, it is indispensable to make a diagnostic in a real time and well manage the patient's computerized data between the various medical actors with the permanent security insurance of highly risky patients. Furthermore, the need to make a speed diagnostic of patients and to detect their health state, their parameters (medical information) of analyses with efficacy, allows us to gain time while monitoring the cardiac patient. It concerns the implementation of services on mobile terminals for transferring medical information and results of ECG analysis (calculated parameters) in a real time with ensuring the mobility, the permanent security and the reliability insurance in covered zone by the mobile network, PLMN (GSM/GPRS…). Our attention has been focused on the choice of a relevant work. It concerns an application on a mobile terminal (*MIDlet*) for detecting some cardiac pathologies and monitoring patient in a non-hospital setting. This paper recalls a complete architecture of an economic wireless transmission system with the implementation of an effective algorithm, adapted to the mobile terminal, allowing to the doctor to have the results of the ECG analysis. Thus, the stakes of setting up such systems are numerous, so much for patients, medical staff and the society in general.

Keywords Remote monitoring · Mobile · J2ME · Wireless sensors network · ECG · ECG pathologies

R. Merzougui (✉) · M. Feham · H. Sedjelmaci
Faculty of Engineering Science of Tlemcen, STIC Laboratory, Chetouane Tlemcen, Algeria
e-mail: j2me_com@hotmail.com

1 Introduction

Recent technological advances of wireless communication networks have contributed to the development of telemedicine. It appears to be a medical reality and it has already imposed the use of portable units as mobile phones. These progress applied to the medical domain (medical

imaging, transmission rate, confidentiality of data, the conviviality of systems…) and the miniaturization of devices open perspectives for medical development of remote monitoring in terms of a better care's quality and a reduction of public health cost [1, 2]. These new technologies have led to the emergence of a wide variety of new ways for users to access and use information anywhere and anytime. Then, today a simple mobile phone can contribute effectively to safeguard of the human lives.

This research orientation exploits the mobility of wireless networks to treat, monitor and detect the state of patients and remote aged ones.

The majority of work and the systems undertaken in this field implement the intelligent part (treatment, numerical calculations, analysis and synthesis of signals…) on large servers (great capacities, better resolutions…) and the mobile phone is used only for transmitting measurements values (a simple gateway) generated by sensors to the server. Moreover, these systems would have required large means and a large infrastructure in their deployment (installation, configuration…), which generates the disadvantage of the excessive expenditure.

Our suggested solution within the framework of this article is an implementation of an algorithm which transmits the data of the patient via a wireless communication in the purpose to exploit a mobile phone for medical monitoring (detection, calculation of cardiac frequency, visualization of ECG signal on the screen of the mobile phone…).

Thus the orientation of our works in this sense was dictated by mobile networks services, simplicity of management and adaptation to the context of mobility with a low cost of exploitation.

In following sections, we present different subjects concerning the solution of detecting critical situations on limited resources by an adaptation strategy to design and develop health services.

Indeed, it is necessary at first to formulate precisely the problem to identify the areas of research which effectively require to be addressed.

2 Problem

In this paper, we particularly focused on the carried out operations in the processing and analysis of the received medical signals from the installed and carried wireless sensors by persons. This step is fundamental to an effective exploitation of the potentialities for collecting a large amount of data which improves monitoring to ensure a permanent safety of remote patients and prevents a degradation of their health state (pathologies). The extracted information on the patient's situation must be relevant to diagnosis.

This approach is significant in comparison to the diverse quantities of the data, as well as the need of a personalized treatment for each patient.

The constraints of this work is related specifically to the need for an approach focused on the characterization, classification and determination of the parameters of ECG signal, specific to each person. Therefore, the physiological characteristics vary according to individuals [3]. The complexities of the problem reside in a great inter-individual variability of the recorded data, and also in broad intra-individual possible modifications are often given not very foreseeable aspect of human behaviors [4].

The problem is also posed on the implementation level of an efficient algorithm intended to solve all the preceding constraints and adapted to mobile phones. This implementation requires many constraints (low resource calculation, memory capacity, resolution…) to run properly.

In this context, the considered study leads to an inexpensive solution, efficient and comfortable for patients at anytime and anywhere, provided that they have a mobile terminal. Indeed, they could benefit of a medical monitoring security, without the inconvenience and without excessive expenditures.

3 Positioning of Work

The considered study leads to an inexpensive solution, efficient and comfortable for patients at anytime and anywhere, with the mobile terminal. Indeed, they could benefit from the medical monitoring security without obstacles and excessive expenditures.

3.1 Cost of the System

The cost of the health represents a considerable weight in the economic balance sheet on international scale. In addition, in many countries, aging or psychological shocks tend to increase the number of people who are in depth need of medical monitoring even more or less intensive care. As a result this will require more attention from the global cost of medical care.

As all the technologies, the development of mobile telephony and its possibilities are in fact more important than it were in the last decade. But indeed as usually, the majority of users use the basic functions only such as: phoning and sending messages (SMS), despite the fact that there are other multitude applications.

We are going to propose in this research the exploitation of the mobile phone in other applications apart from the vocal communications. The idea is to divert these devices of their basic function to make them useful for the detection of pathologies and the medical supervision.

3.2 Platform System and Functions

The considered system allows a patient to be in contact, at any time with the doctor for the medical monitoring and the confrontation of diagnoses from the transmitted data on mobile phone (Fig. 1).

The implemented application on a mobile phone for the detection of some pathologies, functions on all mobile terminals or PDAs equipped with a J2ME virtual machine. This algorithm allows not only of Bluetooth or ZigBee connections with the wireless sensors, but also of wide communications network (GSM/GPRS…) with other display devices installed in doctor's clinic. In this case, the transmitted data concerns the ECG signal are collected on remote sites. To implement this process on a mobile device, the following operations are executed:

- Collection of ECG data from a wireless sensors network.
- Classification and training of ECG signal.
- Detection of pathologies (ESV…).
- The remote recording of the results of training and classification 24/24 on a data base.
- Sending an alarm in the case of a dangerous pathology.
- Visualization of ECG signals on the screen of mobile terminal.
- Possibility to zoom the ECG signals.

The first part which must be realized concerns the collection of remotely medical data on the mobile terminal. These data are generated by wireless sensors placed on the body of a patient.

Thereafter, it is necessary to keep permanent interconnection of the mobile phone with the sensors network and the doctor's display devices (portable telephone, server…), in order to be easy to exchange the entered data.

Moreover, the adaptation strategy of the medical context was followed to generate analysis results of calculation algorithm implemented on the doctor's mobile terminal (detail in the following sections). Thus, the doctor is able to check the result of the diagnosis obtained by this algorithm and consequently detected pathology.

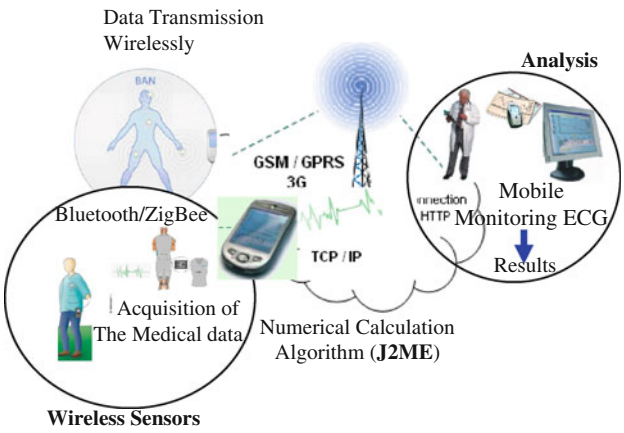


Fig. 1 Architecture of the platform system

3.3 Choice of Technology

This part is the most significant in this study. It relies on studying various technologies and protocols used in the world of wireless communications. The tackled subjects are related to the data exchange between the various parts of the platform including a mixture of networks.

The analysis carried out made it possible to a better understand of principal protocols that has to do with our development application.

After studying the various technologies, in terms of exploitation of the data sent by a sensors network on a mobile phone using a specific algorithm to treat medical data and transmit the results of diagnosis on a telecommunication network to the doctor's device (Mobile phone: result reception, ECG signal, alarm in the case of the anomaly…, or Server). It appeared out that the most adapted solution depends on the use of a simple mobile phone linked by two different systems; a wireless support WPAN (Bluetooth technology or ZigBee) and a GSM/GPRS system. Otherwise this kind of technologies is simple and rapid to implement.

This implies the exploitation of a pallet of integrated network protocols in order to establish connections that are already described between the mobile applications, wireless sensors network and doctor's tools. The choice of this pallet is justified by the following characteristics: [5]

- The protocols in question are obligatorily implemented on all terminals MIDP (J2ME).
- Technologies must be simple, effective and more easily installed.
- A reduced cost of the deployment, installation and configuration.

The transmission of remote patient's medical information is based on the following communications:

3.3.1 Transmission by Mobile Phone

Transmission between two mobile terminals. Does not exist enormously of possibilities. Indeed, it is possible to send SMS, MMS and e-mail. On these three modes of transmitted data, two are available only on the last generation phones; they are e-mail and MMS. Moreover, these two possibilities appear more sophisticated than the others. They allow sending all kind of electronic documents (text, photo, sound…).

For our application, the choice is related to the service MMS which has the following characteristics:

- MMS protocol is implemented in the optional packages of J2ME.
- A large range of use.
- An important content of multi-media which can be transmitted.

Transmission between a mobile terminal and a server. This implies the existence of an https connection (exchange of protected information) between the mobile terminal and the data base server via a WAP gateway to transfer results of final diagnosis. This choice is dictated by the following characteristics: [5]

– Https is obligatorily implemented on all terminals MIDP (J2ME).
– Https is independent of the network.
– The port of the https protocol is more easily working on the firewall.
– Https protocol is implemented by default in J2ME package. Other protocols are not necessarily available [6].

This transfer is based on a communication WSP/https. Its name suggests Wireless Session Protocol (WSP), session layer that allows the connection setting to make transactions. Thus it allows the layer application to profit from two different types of sessions:

– Connected session mode which the layer session will interact with the layer transaction.
– Non-Connected session mode where the session layer will act directly at the transport layer for sending brutes' datagram.

WSP is equivalent to the https protocol, and we find many identical implementations to the https in WSP.

3.3.2 Transmission by Sensors Network

The sensors, which are placed on the patient body, use a wireless support of WPAN technologies: Bluetooth or ZigBee. They transmit on short perimeters measured data of a patient through these kinds of technologies.

The part concerning the reception of these data on a mobile phone does not require a particular study; it's the library's research which allows such a handling of a sensor. It's on the level of the implemented application that one opens a simple tunnel (input Stream: for the reading, Output Stream: For the writing) with a Buffer to recover and to store the transmitted data.

The particularity of sensors networks is located in the routing and energy economy of the network layer. We consider in this article a context of the energy saving and the increase in lifespan of sensors networks.

The current protocols of routing use the metric (a number of jumps, stability of the bonds) which inevitably do not optimize the energy of the nodes like that of the network and this by the use of some nodes more than of other. Indeed, the protocols of routing with energy conservation must determine the optimal roads while being based on the metric related ones to the energy state of the nodes. In this context, many protocols are proposed.

As a result, we proposed an improvement of protocol DSR to include the aspect of energy economy in establishment of the roads. The original version of DSR does not take in consideration this aspect and chooses as road that has the minimum of jump which is not always effective in the sensors networks having constraints major of energy.

Our named proposal TMM-DSR (Taux Min–max Dynamic Source Routing) saves energy during the establishment of road, as in the remainder of the lifespan of the sensors because the metric used by TMM-DSR does not support the roads having the minimum of jumps but uses each time the road having the best energy rate. What balances the use of the sensors for the routing, and saves their battery and consequently gives a long lifespan for the network [7].

TMM-DSR preserves the nature of DSR as a reactive protocol, based on two operations: discovering and maintenance of road. To implement this technique each node must have information concerning energy level and the energy rate of its battery at anytime during the lifespan of network [8].

1. Technique of road selection (Max–Min). In the conventional networks, the metric used is the number of hop which separates a source node from the destination. This metric is adapted to the wire networks but for wireless networks, the number of hop as parameters to evaluate a road is insufficient, following the imposed constraints by these networks such as the mobility of the nodes, the limitation of the band-width as well as the energy constraints.

New suggested metric is based on consumption rate of nodes' batteries in order to improve the power consumption of the network. It is called "Max–Min" technique.

2. Calculate rate of energy consumption. The rate of energy consumption or battery discharge can be defined as being effective energy (the remainder of energy) divided by maximum energy (initial).

$T = (E_{ini} - e_c)/E_{ini}$
$T \cong 1$: Low consumption rate.
$T \cong 0$: Very significant rate consumption.
e_c: Power consumption.

The implemented algorithm, based on the new metric, is responsible for the choice of the roads. This algorithm will proceed in the following way:

1. Each node when it receives a new request, will insert the rate pre-calculated on the request heading of discovered road until the arrival of each node to the recipient.
2. It waits a time D after the reception of each new request. Then, it determines the minimum of the rates of each received road:

$$T_k = Min(T_i)$$

i: The number of the nodes on the road K.

After the determination of the minimum rates of each road, the recipient will choose principal road that has the maximum rates.

$$T = Max(T_k)$$

The algorithm Max–Min does not give any guarantee on the time from the beginning to the end. To solve this problem, we have introduced a factor of rate energy differences that makes it possible to switch between the ways which have performance indexes very close in such manner to choose the shortest way then the smallest time from the beginning to the end. This parameter is given by the following relation:

If $\left[Min\left(T_j\right) - Min(T_i)\right] < \varepsilon$ and $hR_j > hR_i$. So to use the road R_i

3.4 Choice of Development Environment

Java applications have been implemented under NetBeans IDE environment.

A simulation tool Sun Java™ Wireless Toolkit 2.5 for CLDC was exploited to examine all the possible wireless communications. It allows applications on devices with low calculation resources such as a mobile phone [6].

The choice of Java is justified by the different problems associated to coding in C++ on Symbian operating system:

- Management of the memory: for the majority of applications, Java system seems to be sufficient.
- Environment of execution: the proposed options on executable Java as protections for downloading or secure execution are free, whereas in C++, it is necessary to develop them, test them and maintain them.
- Perpetuity: Java seems to have been accepted for the development of applications on mobile phones. The future developments will make Java perhaps as fast as C++.

Hence, for that principal reason, Java was chosen in our project, but it is necessary to mention that both environments can be used [9].

4 Implementation of Proposed Model

As mentioned before, our implementation achieves the medical service which provides the continuity of remote monitoring at home and immediate alarms to deal with the patient in the event of need.

The schedule of conditions of our project consists of:

- The implementation of this service requires the development of two distinct applications:
  - A first to be installed on the mobile phone to detect and treat the critical situations of patient via wireless support.
  - A second function on doctor's devices in order to receive and record results relating to the patient.
- To program the application in a language which is most portable possible, the algorithm must be simple to use and install.
- To program a user interface of high quality.

The suggested model is also based on techniques of programming adapted sources and not only limited ones (devices of medical supervision) but also to the generated heterogeneous parameters. What allows in particular showing the diversity of profiles of persons and types of generated situations, including the simulation of "normal" modifications and disturbing of behavior.

The following paragraphs present (1) Development on mobile phone, (2) Development on PC, (3) the simulation of the proposed model.

4.1 Development on Mobile Phone

J2ME is a collection of technologies and specifications which are conceived for various parts of the market of the small devices. The principal part of the platform J2ME is composed of two different configurations (Fig. 2): Connected Device Configuration (CDC) and Connected Limited Device Configuration (CLDC) [5].

A configuration defines the central libraries of Java technology and virtual storage capacities of the device. CLDC is adapted to recent mobile phones. This configuration is useful for our application. To still note, that in the case of J2ME, the virtual machine is called KVM for Kilobyte Virtual Machine.

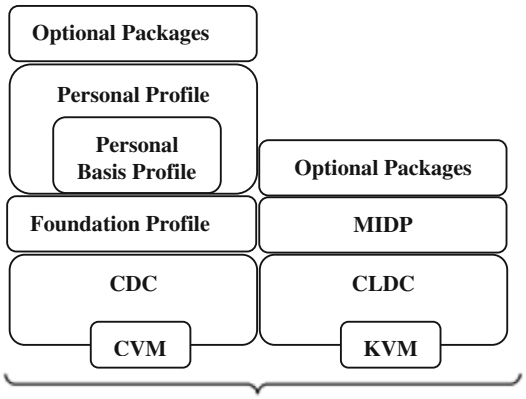


Fig. 2 Architecture of J2ME

At the top of the configurations (Fig. 2), there are the profiles which define the functionalities in each specific category of devices. The "Mobile Information Device Profile" (MIDP) is a profile for the mobile devices using configuration CLDC, like the mobile phones. Profile MIDP specifies the functionalities like the use of the interface user, the persistence of storage, the setting in network and the model of application.

On the majority of the current phones, J2ME is composed of configuration CLDC and profile MIDP.

In addition to standard MIDP (Fig. 2), additional (optional) packages can be added according to the devices, allowing the use of their specificities.

As these options are typically reserved to mobile phones, it was natural to not integrate them directly in the profile MIDP.

So, the development of our application on mobile phones is based on the use of configuration CLDC and profile MIDP. In addition to these two standard elements, we have exploited some optional packages such as WMA for the management of services SMS/MMS and Web Services API. The libraries necessary for the implementation for each component of J2ME are as follows: [6]

API MIDP: is currently that which one finds on the compatible mobiles:

- javax.microedition.Icdui: For the graphic components necessary to the creation of applications.
- javax.microedition.midlet: It provides the component application as well as the primitives managing the life of the application.
- javax.microedition.rms: A possibility of storage of information on the terminal.

API CLDC:

Javax.microedition.io: It contains the classes allowing connection via TCP/IP or UDP. The main class of this package is the class *Connector* [9].

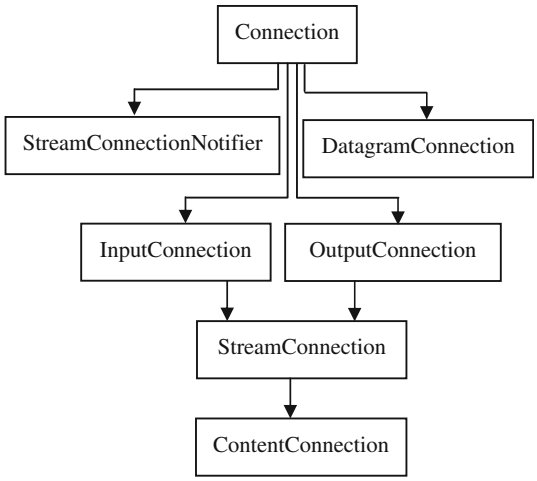


Fig. 3 Tree structure of the classes' javax.microedition.io [7–9]

This network part determines which means is used to communicate medical information (Fig. 3).

This algorithm, called *MIDlet*. It is carried out with the virtual machine J2ME (KVM) on the mobile terminal. It has the role to receive measurements of the sensors of the patient, to treat these data, to transmit or to store them if necessary.

It also allows to the doctor to send an alarm in case of a critical situation.

4.2 Simulation of Proposed Model

The networks GSM/GPRS are useful to transmit information concerning the measured data ECG. Currently, mobile phones of last generations are able to send and receive all sorts of messages (text, image, sound…). They offer in addition to the voice communication, a supply of services on a large scale, that allows the use of multitude applications for these devices.

Our investigation turns to integrate an ECG signal on a mobile phone to ensure a medical remote monitoring service of remote cardiac person.

We describe in the following the basic concept in electrophysiology of the heart in order to specify the components of the electrocardiogram (ECG), while basing on some pathologies which one can meet them on cardiac person.

4.2.1 Electrocardiography

Electrocardiography deals with the study of the electrical activity of the heart muscles. When human body is electrically conductive, the electric potentials that are produced by the activity of the heart can be collected by electrodes

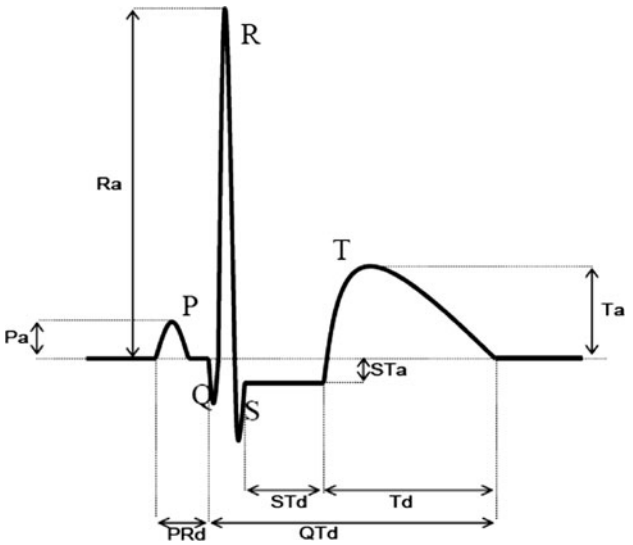


Fig. 4 The complete cardiac cycle

placed on the thorax. The model that is recorded of this electric activity of the heart, on a frontal level (derivations of the members) and on a horizontal level (derivations of precordiales) is an electrocardiogram.

For each heart beat, the electrocardiogram records following successive waves (P, Q, R, S, T) (Fig. 4).

Each of these waves is characterized by its amplitude and its duration. The Table 1 summarizes these values for a normal person.

The study of a recording ECG is founded on the analysis of some successive cardiac frequency. The study of only one frequency provides few indications for the set of a diagnosis, but the variations of characteristic parameters of each frequency during the recording constitute a source of essential information.

These characteristic parameters are: [10]

– Durations of the waves P, Q, R, S, T, and measured amplitudes from the base line (Fig. 4). The basic line is the isoelectric line of the heart rest, which is taken as a reference to measure the waves' amplitude: during the cardiac inactivity, the measured potential is thus normally null as compared to this reference. It is the case on the level:

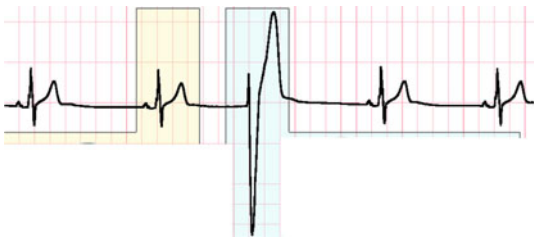


Fig. 5 Ventricular extrasystole (ESV)

- Interval between the waves T and P of two successive beats.
- Interval between the waves P and Q of the same beat.
- Interval between the waves S and T (in the absence of pathology).

– Duration of these waves.

The values of the parameters in Fig. 5, usually noted at the adult person in good health (normal person), are presented in Table 1: [11].

This diagnostic tool allows detecting the rhythmic cardiac pathologies as muscular, extra cardiac metabolic problems, medicinal and others.

4.2.2 *Pathology*

The problem of detection of a person's critical situations from the remote collected data relates particularly the development and the design of intelligent algorithms. Large quantities of temporal data and heterogeneous, are analyzed in real time for the identification of the worrying or critical situations.

This article describes the deployment and the implementation of an intelligent system on mobile terminals to detect cardiac pathologies.

In this context, one is interested more particularly in the study of the long-term evolution of a person's health to identify the installation of more or less progressive pathologies such as an ESV, ESA or ESJ. This study is carried out in three sections with defining the three parts of this document. They relate successively (I) the characteristics and types of extrasystole, (II) Diagnosis starting from the rhythm, (II) Diagnosis starting from the waves.

Table 1 Usual value of the VARIOUS PARAMETERS characterizing a cardiac beat

	Wave P	Interval PQ	Complexe QRS	Interval ST	Interval QT	Wave T
Duration (S)	(Pd) 0.08–0.1	(PQd) 0.12–0.2	0.08	(STd) 0.20	(QTd) 0.36	0.2
Amplitude (mV)	(Pa) 0.25	Isoelectric : 0	$Qa < 0$, $Ra > 0$, $Sa < 0$	Isoelectric : 0	–	$Ta > 0$

Fig. 6 Auricular extrasystole (ESA)

4.2.2.1 Characteristics and Types of the Extrasystole An extrasystole is a premature ventricular excitation compared to awaited depolarization, of auricular origin, nodal or ventricular. Sometimes physiological, it can however translate a more or less serious subjacent pathology.

Ventricular extrasystole (ESV). The ventricular extrasystole (ESV) is a sufficiently widespread abnormal beat. The ESV is almost observed on all the recordings, mainly in period of recovery after an effort (Fig. 5).

The most significant example in this context is that of the MIT base of signals, which contains too much ESV. For example the following signals:

- Signal 119: It contains 444 ESV.
- Signal 200: It contains 836 ESV.
- Signal 208: It contains 992 ESV.

For this reason, we have exploited the signals quoted above in order to validate the power and the effectiveness of this algorithm.

Auricular extrasystole (ESA). The ESA occurs when the number of samples between two successive peaks *R* is higher than the number of samples ranging between other peaks *R* (Fig. 6).

The example of the MIT base of signals where we finds the ESA, is:

- Signal 106: It contains 520 ESA.
- Signal 232: It contains 1832 ESA.

Nodal or jonctionnelles extrasystoles (ESJ). There is a presence of an ESJ if the complex QRS is very fine (except block of branch), of morphology identical to the layout in sinusal rhythm, without P wave or with a P wave called retrograde (Fig. 7).

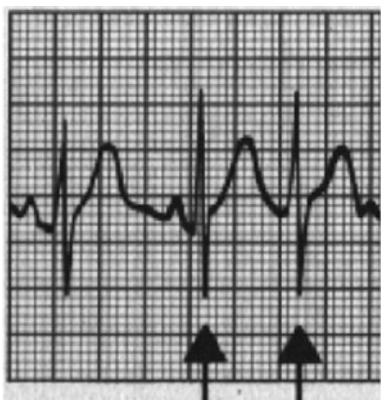


Fig. 7 Nodal or jonctionnelles extrasystole (ESJ)

4.2.2.2 Diagnosis Starting from the Rhythm Analysis of the rhythm requires only the location of the R waves. This analysis is founded on the extraction, starting from the signal of two following characteristic parameters, the frequency of the beats and their regularity.

A regular rhythm cardiac is normal when it is included in a day between 60 and 100 bpm and between 40 and 80 bpm during night.

Out of these limits, one speaks about bradycardia when it is too slow and about tachycardia when it is too fast.

Bradycardia: The bradycardia is characterized by a cardiac frequency lower than 60 bpm. It is known as of origin sinusal, jonctionnelle or ventricular, according to the initiation site of electric pulse at the origin of considered beats.

Tachycardia: Although the presence of an ESV does not indicate any particular pathology, if, in a recurrence way, their number per minute is higher than 6, they can be a precursory sign of a ventricular tachycardia which constitutes a major pathology. It is characterized by a frequency higher than 100 bpm (Fig. 8).

Rhythm: One speaks about doublet (2 successive ESV), of triplet (3 successive ESV); beyond, one also about salvo of ESV or non constant ventricular tachycardia. This one can caused sudden death.

The QRS duration normally lies between 0.06-0.10 s, and beyond 0.12 s one evokes a major disorder.

There is then a presence of ESV when the complex QRS > 0.12 s as well as the number of sample between two successive R peaks is higher than number of samples ranging between the other preceding successive R peaks.

4.2.2.3 Diagnosis Starting from the Waves This type of analysis remains at this moment primarily limited to the form of R wave, even if it starts to allow location of the repolarization disorders starting from the form of T wave.

The advantage of an individual study of each wave by including the analysis of P wave, complex QRS and T wave, is that it will make it possible to carry out a true diagnosis on the basis of expert knowledge. Thanks to the

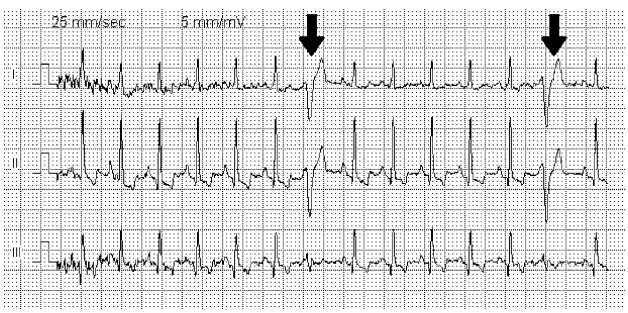


Fig. 8 Ventricular tachycardia

localization of the origin of the problem when the cardiac frequency are not normal.

Thus, the method suggested in our algorithm, allows a precise and continuous location of all the characteristic waves P, Q, R, S and T of the cardiac beat, which will have to make it possible more precisely to locate all the zones of the signal likely to carry the trace of an abnormal behavior of the heart on the 24 h of the recording.

Application. This section describes a medical application implemented on a mobile phone to treat and characterize the ECG signal.

Our application consists to develop a MIDlet to take remotely the evolution of the state of patients and to calculate parameters witch characterize the ECG signal in order to detect a critical situation.

The implementation of the proposed model for transmission simulating, storage and data processing of the electrocardiogram is realized with the J2ME environment. For reasons relating to the rapid evolution of technology, it is always preferable to avoid carrying out specific applications to a type of mobile equipment owner (Windows, Symbian, Palm/OS). J2ME allows the development of applications which can run on all compatible mobiles. It brings to the portable systems the power and the modularity of the programming JAVA and this in a way is adapted to the characteristics of the embarked terminals [12].

The following paragraphs present the principle of the implementation, the global structure of the implementation of simulation and finally the analysis result of the calculated parameters (Cardiac frequency averages, QRS duration…).

4.2.2.4 The Principle of Implementation Being given the complexity between the portable telephone's technology and number of parameters of the model which must be defined in priori, an adaptation of data to this context lies in the usage a set of files in the format text for the definition of the current exploited values for the simulation and the default values. A principal class undertakes to recover and analyze the associated flow to the transmitted files through the radio interface between wireless sensors network and diagnosis's telephone. It implements an intermediate graphic interface on mobile phone allowing the display, the transfer not only the contained parameters in the files and results of obtained diagnosis but also alarms identifiers the degree of risk of detected pathology. After execution of the application, one can then recover the values in files. At the end of the application, the generated results are remotely stored on doctor's (hospital's) data base, and possibly displayed on his screen.

Figure 9 presents the general principle of the process implementation of simulation.

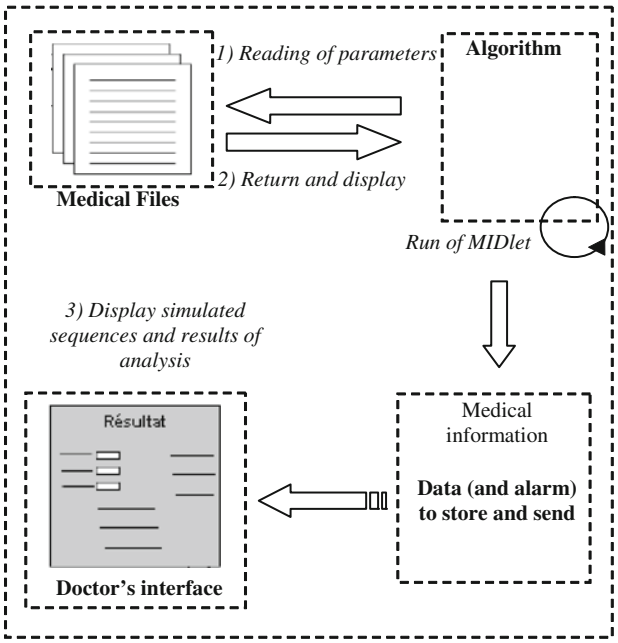


Fig. 9 Principle of the process implementation of simulation

The result interface is presented on Figs. 11 and 12. The parameters are calculated according to the simulation model sequences on which they involve. They are sent in real time to the hospital's data base via an intermediate servlet with generated alarms on doctor's telephone in the event of a critical situation (Figs. 12, 13).

4.2.2.5 Global Structure of the Implementation of Simulation The global structure of the simulation program is completely sequential. It calls one after the other, a table of 10000 samples each two seconds to fill in order to optimize the memory of storage on the telephone, the functions realizing the principle stages of the simulation and successively corresponding to the generation of the different parameters of ECG signal. These parameters include measurements taken, duration between two successive peaks R, complex duration QRS and cardiac frequency. Each called function takes as entering parameter the results' call of the previous function and provides the results of its execution to the following function [13].

The stages of the MIDlet execution are detailed in the following section.

4.2.2.6 Result of Diagnosis As we have already seen before, J2ME wireless development was exploited to implement the proposed simulation model of ECG signal on a mobile phone.

In this section, we present and discuss the various stages of execution of the algorithm. All this series of tests were made thanks to the phone emulator.

The doctor may activate the mode of the medical remote monitoring ECG (Remote Monitoring ECG) during the launching of the application.

At the beginning, the application will operate and communicate in autonomous mode with measurement sensors network. Then, the phone collects periodically the ECG samples that are generated by these wireless sensors (Fig. 10). These data are stored in a sequential way in tables of 10000 samples in order to imply them in the treatment and detection of critical situations in the event of pathological cases.

Figure 10 shows the organization in vectors (10000 samples) of ECG values transferred via wireless PAN technologies (Bluetooth, ZigBee…) to the internal memory of the mobile phone.

Such a medical application proposes a set of services to the health professionals (list of the patients, the display of the medical profile of a patient, Digital processing of the signals…). These services make treatments with variable complexities (management of data via a wireless sensors network set by the person or a data base, numeral calculation…) and exchange data with the user through a graphical interface on doctor's display device (mobile phone, PC…).

This environment type presents important and heterogeneous information, a great variability and numerous possibility of evolution. Indeed, the offered resources on the level of terminal can be extremely different according to the use of a personal assistant, a laptop or a workstation. Therefore, it is necessary to implement an adaptation strategy to conceive and develop the algorithm by respecting these required constraints.

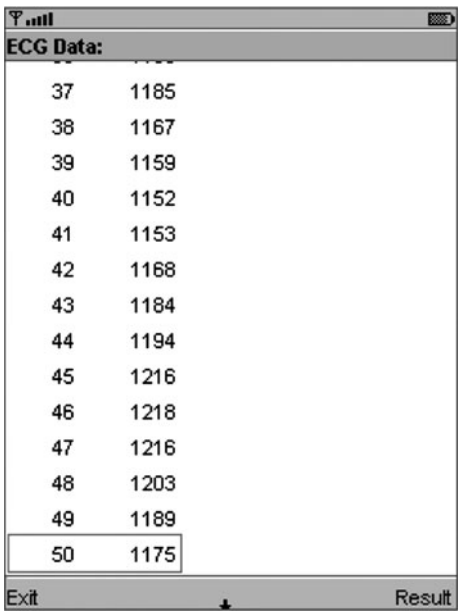


Fig. 10 Acquisition of 10000 samples of a patient (ECG signal)

The graphical interface, the mobile phone's screen, allows to collect, display, store, calculate the parameters of simulation (time between two successive peaks R, QRS duration…) and to transmit these medical information in real time to the hospital or clinic (Figs. 11, 12), after an adaptation of medical data to the context.

Our adaptation strategy consists to exploit a vector of storage of a medical file (Fig. 10), of fixed size relating to the size memory available on the mobile phone (32 KB). This contained is sequentially transmitted by sensors network what leads with time to increase the space of storage on the terminal.

For this reason it is necessary to adapt these measurements to the mobile context. This adaptation consists to recover and treat regularly each 10000 samples in a vector to be periodically erased, in order to optimize the memory for the storage of the relevant results, which makes it possible to the doctor to re-examine the files.

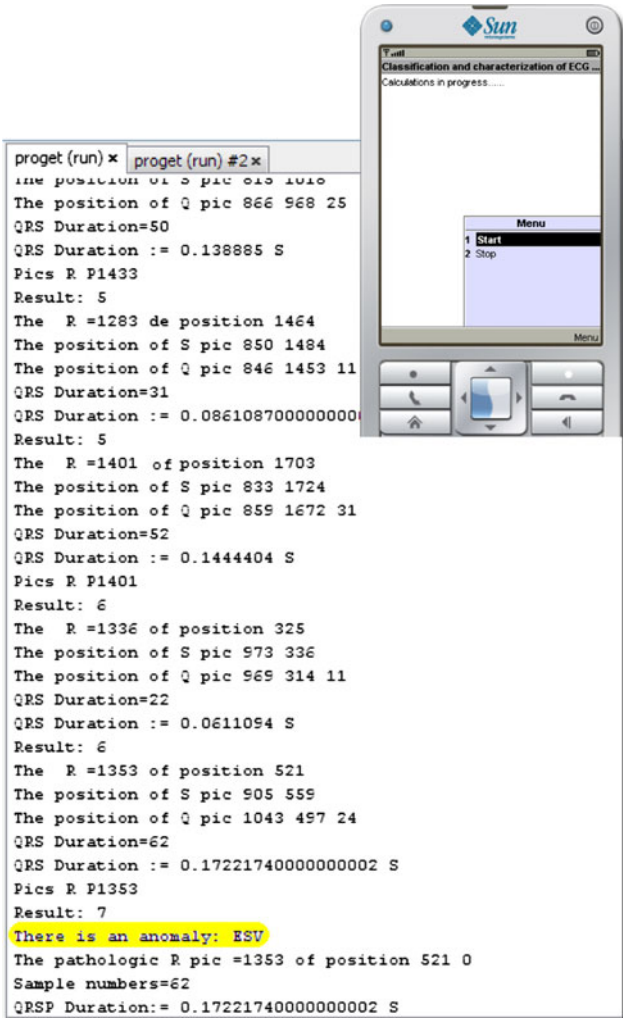


Fig. 11 The calculated parameters

Key code	R position	QRS Duration	S N	The date	Type
10	2224	0.0777756	28	Mon Jul 06 12:03:42	NORMAL
10	2740	0.0722202	26	Mon Jul 06 12:03:42	NORMAL
10	3265	0.0777756	28	Mon Jul 06 12:03:42	NORMAL
10	3777	0.0749979	27	Mon Jul 06 12:03:42	NORMAL
10	4086	0.0916641	33	Mon Jul 06 12:03:42	NORMAL
10	4281	0.1305519	47	Mon Jul 06 12:03:42	ESV
10	4754	0.0944418	34	Mon Jul 06 12:03:42	NORMAL
10	4888	0.0916641	33	Mon Jul 06 12:03:42	NORMAL
10	5082	0.13610730000000001	49	Mon Jul 06 12:03:42	ESA
10	5551	0.0944418	34	Mon Jul 06 12:03:42	NORMAL
10	5747	0.138885	50	Mon Jul 06 12:03:42	ESV
10	6218	0.08610870000000001	31	Mon Jul 06 12:03:42	NORMAL
10	6457	0.1444404	52	Mon Jul 06 12:03:42	ESJ
10	6468	0.083331	30	Mon Jul 06 12:03:42	NORMAL
10	6782	0.0611094	22	Mon Jul 06 12:03:42	NORMAL
10	6978	0.17221740000000002	62	Mon Jul 06 12:03:42	ESV
10	7180	0.1333296	48	Mon Jul 06 12:03:42	TACYCHARDIA
10	7461	0.0777756	28	Mon Jul 06 12:03:42	NORMAL
10	7462	0.13610730000000001	49	Mon Jul 06 12:03:42	ESV

Fig. 12 calculated parameters Stored on hospital's data base

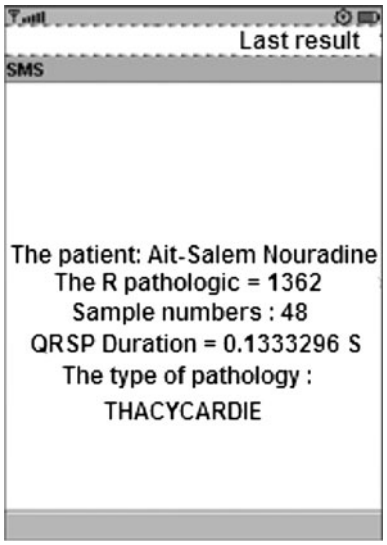


Fig. 13 Detection of Critical situation is sent to doctor's mobile phone

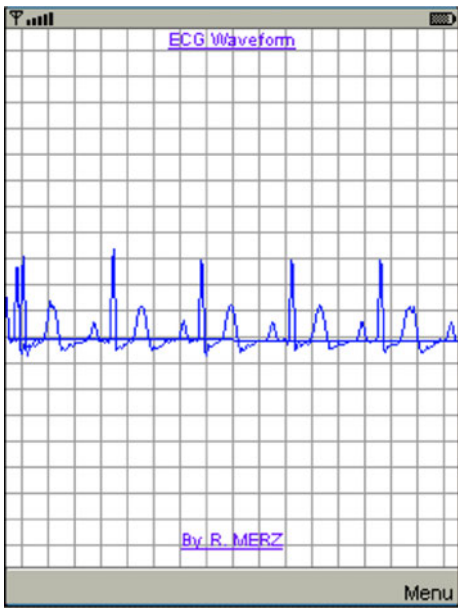


Fig. 14 Show ECG waveform of the patient

The medical data recovered on wireless diagnosis tool (Fig. 10), are exploited for the training, the characterization and the classification of ECG signal. Accordingly, the doctor is invited to consult the latest results of his patient (Fig. 12) in order to take the adapted decision.

The diagnosis and the complete treatment could be done using the implementation of the calculation algorithm on the mobile. It calculates the most significant parameters necessary to the characterization and the precise and continuous location of all the waves' characteristic of the ECG, making it possible more precisely to locate all the zones of the signal likely to cover the trace of an abnormal behavior of the heart during 24 h of the recording.

It immediately sends an SMS/MMS (message alarms) to the doctor in the event of detection of a pathology which has occurred (Fig. 13).

In this scenario, we propose the development and design of a complete analysis on an intelligent device (mobile phone). It includes a pallet of diagnoses: diagnosis starting from the rhythm and diagnosis starting from the waves (R, P…). We have implemented and associated these two types of diagnoses in our algorithm with each model containing ECG data corresponding to a patient and all parameters (time between two successive peaks, duration QRS, the rhythm cardiac and localization of the waves…) which characterize this patient.

Thus, the doctor can observe the ECG signal in real time on his screen: (Fig. 14)

A more importantly option allowing the zoom of the part which presents an anomaly, is implemented in our application. It is enough for the doctor to introduce a begin point and an end point (time interval) in order to widen the part in question (Fig. 15).

4.2.2.7 Comparison with the Simulation MATLAB The sequence data generated for the parameters of ECG in the context of medical remote monitoring (MIDlet) are validated by a comparison with simulated data under MATLAB environment (Fig. 16).

Signals 119, 200, 208…of MIT base are implied and implemented in algorithm in order to validate the effectiveness of this monitoring tool. This validation which is

Fig. 15 Zoom part of the curve

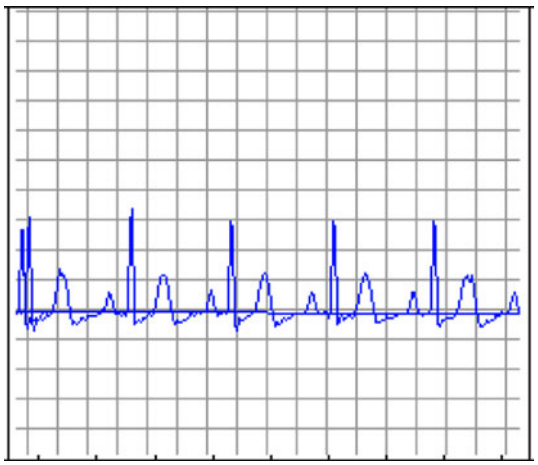


Fig. 16 ECG signal

carried out about the implemented simulation process by the tool of J2ME development makes this algorithm to be exact in terms of calculation, powerful and effective. It was developed in the context of pathology detection on mobile phone, from data sequences transmitted by a sensors network. The obtained results show the strong points and innovating qualities of our algorithm compared to those of other works related fields. At the same time our algorithm presents better perspectives of improvement of its efficiency.

4.3 Quality of Service

Within the framework of this work, the problem of the mobile networks' quality service for the second or third generation, lies on one hand in a environment with various requirements of good functioning and on the other hand in two ways of transport of the information: circuit or packet.

The doctor wants to know always about state of evolution of the patient. What requires having at least an acceptable quality of service in the mobile communication to be ready for the sanction by a correct decision-making.

We can say that the zero risk does not exist. Prevention measures in the system of remote monitoring that has chosen suitably according to needs can reduce considerably of this risk as well. For that purpose, this application requires a real-time transmission and is not tolerant in the errors. It requests for more or less raised debits.

5 Evaluation

The proposed simulation algorithm is articulated on two fundamental points: the first one relates to the simulation step in the respect of the complexity and objectives of the medical remote monitoring context at distance. The second one is a more global vision on the resolution cycle of construction problem of the behavior profile of person to ensure a critical situation. The suggested remote monitoring consists to monitor and diagnose the state of a patient using the methodology developed in this project. Thus, the doctor treating a person with risk cardiac can at any time control the state of his patient by consulting in real-time the ECG on his device (mobile terminal, personal assistant, pc…) and the classification of pathology by the developed algorithm.

6 Conclusion

This article refers to biological analysis signals of a patient transmitted by sensors and detected on a mobile phone, then transmit them remotely in a real time to the doctor. This technique allows medical remote monitoring of cardiac or hypertensive patients.

Also, the identification, by the developed algorithm, of the medical profile of a remote patient and the detection of critical situations cannot cover all medical indicators corresponding to each patient.

Thus, the improvements of this algorithm must be adapted to diagnose new pathologies.

This solution is not costly and easily realizable. It is adapted to the portable devices ensuring medical monitoring anytime and anywhere. It is in this vision that other services, associated to mobiles and intended for the telemedicine and the house automation are under development.

Acknowledgments I deeply thank my supervisor and the person in charge of the laboratory STIC, Mohamed FEHAM, Professor at the university Abou Bekr Belkaid of Tlemcen, Algeria. The

correctness of his advice, the motivation and the project financing, were very precious and brought a successful conclusion to this work. A special thank to the researchers of the laboratory who helped in this project.

References

1. A. Nemo, La télémédecine: Faire voyager les informations plutôt que le malade. *Journal du Téléphone*, Grenoble, vol. 13, p. 4, 1994.
2. C. Suarez, «La télémédecine: quelle légitimité d'une innovation radicale pour les professionnels de santé?,» *Revue de l'institut de Recherches Economiques et Sociales (IRES)*, vol. 39, pp. 1–29, 2002.
3. B. G. Celler, T. Hesketh, W. Earnshaw, and E. Ilsar, An instrumentation system for the remote monitoring of changes in functional health status of the elderly at home. In *Proceedings of the 16th Annual IEEE Engineering in Medicine and Biology Society*, Baltimore, USA, 1994, pp. 908–909.
4. L. Chwif and R. J. Paul, On simulation model complexity. In *Proceedings of the 32nd Conference on Winter Simulation*, Orlando, Florida, 2000, pp. 449–455.
5. B. Delb, Application java pour terminaux mobiles. *EYROLLES.* Paris, France, 2002.
6. A. F. Quintas, Bluetooth J2ME Java 2 micro edition. *Manual de usuario y tutorial*. Ra-Ma, Madrid, 2004.
7. M. Tommaso, P. Dario, and F. A. Ian, Optimal local topology knowledge for energy efficient geographical routing in sensor networks. *INFOCOM, Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, pp. 1705–1716, 2004.
8. B. Bouyeddou, *Implémentation d'un protocole d'économie d'énergie EMM-DSR pour les réseaux ad hoc 802.11*, These of Magister, University of TlemcenAlgeria, 2007.
9. J. Knudsen, *Wireless Java Developing with J2ME*, vol. 2nd, ApressBerkeley, USA, 2003.
10. R. Legameta, P. S. Addisson, N. Grubb, C. E. Robertson, K. Fox and J. N. Watson, Real-time classification of ECGs on a PDA, *IEEE Transactions on Information Technology in Biomedicine*, vol. 30, pp. 565–568, 2003.
11. M. Cauville, Diagnostic, soins et prevention par la telemedicine: explications de J. Demongeot,», *Siences et Technologies*, vol. 2, pp. 32–34, 1999.
12. H. Mahmoud, *Learning Wireless Java*. O'Reilly Mill Valley, Sebastopol, USA, 2002.
13. R. Merzougui and M. Feham, Algorithm of remote monitoring ECG using mobile phone: Conception and implementation. *Proceedings of the Third International Conference on Broadband Communications, Information Technology & Biomedical Applications*, 2008.

Author Biographies

Rachid Merzougui received the Master degree in Systems and Networks of Telecommunications from the University of Tlemcen (Algeria) in 2006. Since this year, he has been Assistant Professor of Mobile Networks and Services. He has served on the scientific committees of the Telecommunication Department of the University of Tlemcen. He is interested now in mobile services.

Mohammed Feham received the Dr. Eng. degree in Optical and Microwave Communications from the University of Limoges (France) in 1987, and his PhD in Science from the University of Tlemcen (Algeria) in 1996. Since 1987, he has been Assistant Professor and Professor of Microwave, Communication Engineering and Telecommunication Networks. He has served on the Scientific Council and other committees of the Electronics and Telecommunication Departments of the University of Tlemcen. His research interest now is mobile networks and services.

Hichem Sedjelmaci received the Master degree in Mobile Networks and Service from the University of Tlemcen (Algeria) in 2009.